

Smart Card ToolSet PRO

# **Card Explorer Extensions Plug-Ins Interface Specification**

Document version 1.3  
May 03, 2005

СОДЕРЖАНИЕ		СТРАНИЦА
I	Термины и обозначения	3
II	Базовые принципы	4
III	Типы данных	5
IV	Диспетчеры интерфейсов	6
V	Команды диспетчеров интерфейсов	8
VI	Диспетчер команд	10
VII	Подключение	12
VIII	События	14
IX	Заключение	17

## I. ТЕРМИНЫ И ОБОЗНАЧЕНИЯ

### Функциональный плагин (далее - Плагин):

- программный модуль, подключаемый динамически к программе Smart Card ToolSet в момент готовности смарт карты к работе и выполняющий определенный набор действий со смарт картой и ее данными;

### Главная программа (далее - Программа):

- программа Smart Card ToolSet, управляющая данным плагином;

### Диспетчер интерфейса плагина (далее - Диспетчер плагина):

- экспортируемая плагином функция, имеющая определенный заданный формат и предопределенное имя "DispatchInterface", предназначенная для передачи команд и данных из программы в плагин;

### Диспетчер интерфейса программы (далее - Диспетчер программы):

- функция в программе, имеющая тот же формат что и Диспетчер плагина, предназначенная для передачи команд и данных из плагина в программу;

### Диспетчер передачи команд в смарт карту (далее - Диспетчер команд):

- функция в программе, предназначенная для передачи команд и данных из плагина в текущую открытую смарт карту;

## II. БАЗОВЫЕ ПРИНЦИПЫ

Функциональные плагины для программы Smart Card ToolSet представляют собой обычные dll - т.е. динамически подключаемые библиотеки MS Windows.

Выбор среды разработки плагинов зависит только от профессиональных навыков или от личных предпочтений разработчика.

Каждый плагин обязан экспортировать одну функцию "DispatchInterface", имеющую заданный формат (см. ниже) и представляющую собой Диспетчер интерфейса плагина.

В процессе динамического подключения плагина программа, используя Диспетчер плагина, производит проверку соответствия и передает через параметры необходимые данные, в том числе указатели на Диспетчер интерфейса программы и на Диспетчер команд, а так же событие готовности к работе.

Все взаимодействие между плагином и программой организовано в виде прямых вызовов функций диспетчеров и передаче необходимых данных через их параметры.

### III. ТИПЫ ДАННЫХ

При работе с диспетчерами интерфейсов и команд в качестве передаваемых параметров и возвращаемых значений используются следующие типы данных:

1. беззнаковое 32-битное целое (далее - Integer)
2. 32-битный указатель (далее - Pointer)
3. В качестве текстовых строк используются массивы символов, всегда оканчивающиеся символом 0x00;

## IV. ДИСПЕТЧЕРЫ ИНТЕРФЕЙСОВ

Диспетчер интерфейса плагина - это единственная экспортируемая библиотекой плагина функция.

Обязательным требованием является экспорт данной функции под именем "DispatchInterface". Если в момент подключения библиотеки такая функция не будет в ней найдена - данная библиотека не считается плагином.

Для каждого подключаемого плагина программа динамически создает его собственный Диспетчер интерфейса программы, указатель на который передается в плагин на этапе его подключения.

Параметры функции:

№	Параметр	Тип	Описание
1	Command Code	Integer	код команды
2	Command Parameter	Integer	код параметра команды
3	Command Data	Pointer	адрес буфера данных

Все параметры функции передаются по ссылке, т.е. реально передаются указатели на локальные переменные типа Integer и Pointer.

Функция возвращает значение кода выполнения операции типа Integer:

Возвращаемое значение	Код выполнения
0	Завершение без ошибки
1	Передан неправильный код команды
2	Передан неправильный параметр для использованного кода команды

Принципы использования:

- в системе задано некое пространство значений команд (см.таблицу далее), задающее соответствие между необходимой командой и значением параметра Command Code функции Диспетчера интерфейса;
- каждому значению Command Code в зависимости от соответствующей ему команды может принадлежать одно или несколько значений Command Parameter, позволяющие более конкретно детализировать специфику выполнения команды;
- в зависимости от команды значение параметра Command Data может либо не использоваться, либо указывать на передаваемые данные;

Каждый вызов диспетчера подразумевает следующие действия:

1. в локальную переменную типа Integer заносится значение кода команды; она будет передана по ссылке в качестве параметра Command Code;
2. в локальную переменную типа Integer заносится значение параметра команды; она будет передана по ссылке в качестве параметра Command Parameter;

3. в локальную переменную типа Pointer при необходимости заносится адрес области памяти содержащей требуемые для этой команды данные; она будет передана по ссылке в качестве параметра Command Data;
4. Вызывается функция Диспетчер интерфейса с перечисленными выше локальными переменными переданными по ссылке в качестве ее параметров (атрибутов).
5. Функция возвращает значение кода выполнения, при этом в локальной переменной, соответствующей параметру Command Data, может при некоторых командах возвращаться адрес на область памяти, содержащую соответствующие этой команде данные;

## V. КОМАНДЫ ДИСПЕТЧЕРОВ ИНТЕРФЕЙСОВ

### Команды как параметры функций

КОМАНДА	COMMAND CODE	COMMAND PARAMETER	COMMAND DATA	ОПИСАНИЕ	ИСПОЛЬЗОВАНИЕ
КОМАНДЫ, ИСПОЛЬЗУЕМЫЕ НА ЭТАПАХ ПОДКЛЮЧЕНИЯ/ОТКЛЮЧЕНИЯ Command Code = 1					
OnRegistered	1	1	Указатель на Диспетчер программы	Программа сообщает о подключении Передается указатель на Диспетчер программы Плагин может выполнить необходимые ему подготовительные операции	Программа -> Плагин
Get Plugin Type	1	2	10 (целое, приведенное к типу Pointer)	Программа запрашивает тип плагина Для функциональных плагинов это значение всегда равно 1	Программа -> Плагин
OnUnregistered	1	100	---	Программа сообщает об отключении Плагин обязан прекратить работу с картой и освободить память	Программа -> Плагин
Get Command Interface	1	1000	Указатель на Диспетчер команд	Плагин запрашивает у программы интерфейс Диспетчера программы	Плагин -> Программа
СОБЫТИЯ Command Code = 10					
onError	10	0	Указатель на строку с сообщением об ошибке	Возникает при любой ошибке API в программе	Программа -> Плагин
onActivate	10	1	---	Событие активации плагина	Программа -> Плагин
onExecute	10	2	---	Событие необходимости выполнения плагином операции "по-умолчанию"	Программа -> Плагин
onCardWait	10	10	---	Возникает при извлечении карты из ридера	Программа -> Плагин
onCardReady	10	11	---	Возникает при помещении карты в ридер и ее успешном	Программа -> Плагин



				открытии	
КОМАНДЫ, ИСПОЛЬЗУЕМЫЕ ДЛЯ ВЫПОЛНЕНИЯ ОДИНОЧНЫХ КОМАНД Command Code = 20					
Show	20	1001	---	Команда плагина программе перейти в видимый режим	Плагин -> Программа
Hide	20	1002	---	Команда плагина программе перейти в невидимый режим	Плагин -> Программа
КОМАНДЫ, ИСПОЛЬЗУЕМЫЕ ДЛЯ ПОЛУЧЕНИЯ ДАННЫХ Command Code = 30					
Get PlugIn Name	30	1	Указатель на строку	Плагин должен вернуть в Command Data указатель на строку со своим именем	Программа -> Плагин
Get PlugIn Description	30	2	Указатель на строку	Плагин должен вернуть в Command Data указатель на строку со своим описанием	Программа -> Плагин
Get PlugIns Base Registry Path	30	1003	Указатель на строку	Программа возвращает в Command Data указатель на строку, содержащую путь к ключу реестра, содержащего настройки данного плагина  Плагин может работать со своими настройками в реестре	Плагин -> Программа

## VI. ДИСПЕТЧЕР КОМАНД

Для каждого подключенного плагина программа динамически создает его собственный Диспетчер команд, который предназначен для посылки плагином команд в карту и получения ее ответа.

Параметры функции:

№	Параметр	Тип	Описание
1	Send Buffer	Pointer	указатель на буффер посылаемого APDU
2	Send Buffer Length	Integer	длина буффера
3	Received Buffer	Pointer	указатель на буффер ответа карты
4	Received Buffer Length	Integer	длина буффера

Все параметры функции передаются по ссылке, т.е. реально передаются указатели на локальные переменные типа Integer и Pointer.

Функция возвращает значение кода выполнения операции типа Integer:

Возвращаемое значение	Код выполнения
0	Команда успешно послана в карту
-1	Команда в карту не послана

Каждая посылка команды в карту подразумевает следующие действия:

- подготовка буффера посылаемой команды: - значение каждого его байта должно быть установлено в соответствии со структурой посылаемого APDU согласно ISO 7816-4:
  - q Cla Ins P1 P2 P3 [DataIn] - для протокола T0
  - q Cla Ins P1 P2 Lc [DataIn] Le - для протокола T1
- в локальную переменную типа Pointer заносится адрес этого буффера; она будет передана по ссылке в качестве параметра Send Buffer;
- в локальную переменную типа Integer заносится значение используемой длины буффера передаваемой команды - т.е. длина APDU; она будет передана по ссылке в качестве параметра Send Buffer Length;
- локальная переменная типа Pointer, используемая для получения указателя на буффер Received Buffer, может иметь любое значение; она будет передана по ссылке в качестве параметра Received Buffer; при возврате из Диспетчера команд программой в нее будет помещен адрес буффера с полученными от карты данными;
- локальная переменная типа Integer, используемая для определения длины полученных данных в буфере Received Buffer, может иметь любое значение; она будет передана по ссылке в качестве параметра Received Buffer Length; при возврате из Диспетчера команд программой в нее будет вписана длина буффера с полученными от карты данными;
- Вызывается функция Диспетчер команд с перечисленными выше локальными переменными переданными по ссылке в качестве ее параметров (атрибутов).

7. В случае успешного диалога с картой функция возвращает значение 0, при этом в локальных переменных, соответствующих параметрам Received Buffer и Received Buffer Length, возвращаются соответственно адрес буфера с полученными от карты данными и его длина;
8. Формат возвращенных данных соответствует ISO 7816-4:  
  - q [DataOut] SW1SW2

## VII. ПОДКЛЮЧЕНИЕ

Ниже приводится алгоритм подключения плагина вызывающей программой.

1. Программа ищет файл плагина и, предполагая что это dll, пытается его подключить.
2. Производится поиск экспортируемой функции с предопределенным именем "DispatchInterface"; в случае успеха данная функция регистрируется как Диспетчер интерфейса данного плагина.
3. Программа проверяет тип плагина

Используя Диспетчер интерфейса плагина программой посылается команда (далее - "посылается команда"):

Command Code	Command Parameter	Command Data Pointer
1 Register Disp Interface	2 Get Plugin Type	---

Плагин обязан вернуть значение типа. Для функциональных плагинов это значение всегда должно быть равно 10.

Локальная переменная, которая будет возвращена как Command Data Pointer, здесь используется не как указатель - вместо этого она приводится к типу Integer и ей присваивается значение 10.

4. Программа запрашивает имя плагина

Посылается команда :

Command Code	Command Parameter	Command Data Pointer
30 Get Data	1 Get Plugin Name	---

Плагин обязан вернуть в Command Data Pointer указатель на локальный массив символов, содержащий строку имени плагина.

Внимание! Строка всегда должна заканчиваться символом 0x00 !

5. Программа запрашивает строку описания плагина

Посылается команда :

Command Code	Command Parameter	Command Data Pointer
30 Get Data	1 Get Plugin Description	---

Плагин обязан вернуть в Command Data Pointer указатель на локальный массив символов, содержащий строку краткого описания плагина.

Внимание! Строка всегда должна заканчиваться символом 0x00 !

6. Программа посылает плагину событие успешной регистрации

Посылается команда :

Command Code	Command Parameter	Command Data Pointer
1 Register Disp Interface	1 Registering	адрес функции Диспетчера интерфейса программы

## VIII. СОБЫТИЯ

События играют важнейшую роль в рабочем цикле плагина. Ниже приводятся обязательные к обработке плагином события.

### 1. Подключение

Принимаемая команда :

Command Code	Command Parameter	Command Data Pointer
1 Register Disp Interface	1 Registering	адрес функции Диспетчера интерфейса программы

Событие возникает :

- ✓ однократно на этапе подключения плагина;

Реакция плагина на событие :

- ✓ плагин должен зарегистрировать у себя адрес Диспетчера программы;
- ✓ плагин при необходимости может выполнить инициализацию локальных данных;
- ✓ плагин может запросить через Диспетчер программы необходимые дополнительные данные, в том числе указатель на функцию Диспетчер команд;

### 2. Отключение

Принимаемая команда :

Command Code	Command Parameter	Command Data Pointer
1 Register Disp Interface	100 Unregistering	---

Событие возникает :

- ✓ однократно перед отключением плагина;

Реакция плагина на событие :

- ✓ плагин ОБЯЗАН прекратить все операции с картой;
- ✓ плагин ОБЯЗАН закрыть все его видимые окна;
- ✓ плагин ОБЯЗАН очистить занятую им память;
- ✓ плагин должен подготовиться к отключению библиотеки;

### 3. Ошибка

Принимаемая команда :

Command Code	Command Parameter	Command Data Pointer
10 Event	0 OnError	Указатель на строку сообщения

Событие возникает :

- ✓ при любой ошибке API;

Реакция плагина на событие :

- ✓ плагин может при необходимости обработать сообщение ошибки;

#### 4. Активация

Принимаемая команда :

Command Code	Command Parameter	Command Data Pointer
10 Event	1 OnActivate	...

Событие возникает :

- ✓ при выборе пользователем данного плагина из списка в Окне Ридера;
- ✓ при назначении данного плагина "по умолчанию" и разрешении его автоактивации в окне настроек типа карты: - в этом случае событие наступает при помещении карты в ридер;

Реакция плагина на событие :

- ✓ плагин должен активизироваться: сделать видимым главное окно и подготовиться к выполнению главной операции (если таковая имеется);

#### 5. Выполнение

Принимаемая команда :

Command Code	Command Parameter	Command Data Pointer
10 Event	2 OnExecute	...

Событие возникает :

- ✓ при выборе пользователем соответствующей команды в главном окне плагина;
- ✓ при назначении данного плагина "по умолчанию" и разрешении его автовыполнения в окне настроек типа карты: - в этом случае событие наступает при помещении карты в ридер после события активации;

Реакция плагина на событие :

- ✓ плагин должен выполнить его главную операцию (если таковая имеется);

#### 6. Ожидание карты

Принимаемая команда :

Command Code	Command Parameter	Command Data Pointer
10 Event	10 OnCardWait	...

Событие возникает :

- ✓ при извлечении карты из ридера;

Реакция плагина на событие :

- ✓ плагин обязан прекратить все выполняемые в данный момент операции с картой и перейти в режим ожидания карты;

#### 7. Карта готова к работе

Принимаемая команда :

Command Code	Command Parameter	Command Data Pointer
10 Event	11 OnCardReady	...

Событие возникает :

- ✓ при помещении карты в ридер и ее успешном открытии;

Реакция плагина на событие :

- ✓ плагин должен перейти в режим готовности к работе с новой картой;



## IX. ЗАКЛЮЧЕНИЕ

Интерфейс плагинов программы Smart Card ToolSet, как и сама программа, претерпевает изменения и постоянно развивается.

Мы приветствуем все Ваши пожелания по расширению функциональных возможностей интерфейса плагинов.

Пишите нам по адресу : <mailto:partners@scardsoft.com>