

Smart Card ToolSet PRO

Card Explorer Extensions Plug-Ins Interface Specification

Document version 1.3
May 03, 2005

Card Explorer Extensions Plug-Ins Interface Specification

Copyright © 2002-2005 by SCard SOFT

All rights reserved. No parts of this work may be reproduced in any form or by any means - graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems - without the written permission of the author.

Products that are referred to in this document may be either trademarks and/or registered trademarks of the respective owners. The publisher and the author make no claim to these trademarks.

While every precaution has been taken in the preparation of this document, the publisher and the author assume no responsibility for errors or omissions, or for damages resulting from the use of information contained in this document or from the use of programs and source code that may accompany it. In no event shall the publisher and the author be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.

Printed: May 03, 2005

Publisher

SCard SOFT

<http://www.scardsoft.com>
info@scardsoft.com

Author

Igor V.Kharchenko

igor@scardsoft.com

Thank You for your interest to Smart Card ToolSet PRO!

You can send yours suggestions or any questions about the Plug-Ins' Interface to an author via e-mail.

Please visit our web site for the latest software and specifications updates.

Author.

Table of Contents

Part I Overview	1
1 Terms	1
Card Explorer Extensions Plug-In (Plug-In)	1
Main program (Program)	1
Plug-In's Interface Dispatcher (Plug-In Dispatcher)	1
Program's Interface Dispatcher (Program Dispatcher)	1
Program's COS Commands Dispatcher (Card Commands Dispatcher)	1
2 Data Types	1
3 Plug-Ins Using Fundamentals	2
Plug-In Loading Phase	2
Step 1 : Loading the Plug-In library	2
Step 2 : Loading the Plug-In Type	2
Step 3 : Loading the Plug-In Name	3
Step 4 : Loading the Plug-In Description	3
Step 5 : OnRegistered Event	3
Step 6 : Loading the Card Commands Dispatcher Interface	4
Active Plug-In Phase	4
OnActivate Event	4
OnExecute Event	4
Plug-In Unloading Phase	5
OnUnregistered Event	5
Part II Dispatchers	5
1 Interface Dispatchers	5
Fundamentals	5
Function Format	6
Commands	7
"Get Plug-In Type" command	7
"Get Plug-In Name" command	7
"Get Plug-In Description" command	8
"Get Commands Interface" command	8
"Get Plug-In's Registry Path" command	8
"Hide Program" command	9
"Show Program" command	10
Events	10
"OnRegistered" event	10
"OnError" event	11
"OnActivate" event	11
"OnExecute" event	12
"OnCardWait" event	12
"OnCardReady" event	13
"OnUnregistered" event	13
2 Commands Dispatcher	13
Fundamentals	13
Function Format	14
Index	1

1 Overview

1.1 Terms

1.1.1 Card Explorer Extensions Plug-In (Plug-In)

The Card Explorer Extensions Plug-In is a standard Dynamic Linked Library (DLL) which allows you to expand the base functionality of the Card Explorer window .

1.1.2 Main program (Program)

The main program is the Smart Card ToolSet PRO which loads this Plug-In and controls it.

1.1.3 Plug-In's Interface Dispatcher (Plug-In Dispatcher)

It's an exported by the Plug-In's function named "DispatchInterface " .

This function must has predefined parameters format because it is called by the main program each time when the command or an event occurs.

Plug-In Dispatcher is used by the program for sending the commands and events to a Plug-In.

1.1.4 Program's Interface Dispatcher (Program Dispatcher)

It's the program's function which has the same format like the Plug-In Dispatcher .

The Program Dispatcher is used by the Plug-In for sending the commands to the main program .

1.1.5 Program's COS Commands Dispatcher (Card Commands Dispatcher)

It's the program's function for sending the data to a card and for receiving the card answers.

The Card Commands Dispatcher is used by the Plug-In for sending the commands to an opened smart card.

1.2 Data Types

The program uses only the following data types as the dispatchers parameters' types:

- unsigned 32 bits integer (Integer) ;
- 32 bits pointer (Pointer) ;
- as a strings the program uses the chars arrays which are ends by the 0x00 symbol (PChar) ;

1.3 Plug-Ins Using Fundamentals

The Card Explorer Extensions Plug-In is the standard Dynamic Linked Library (DLL) and it may be created using any of programming languages and any of IDEs which allows you to create the DLL.

Each Plug-In must export one function named "DispatchInterface" . This function must has predefined parameter's format because it is called by program each time when the command or an event occurs.

The communication between the main program and the Plug-In is based on a dispatchers calls.

The main program calls:

1. Plug-In Dispatcher;

The Plug-In calls:

1. Program Dispatcher;
2. Card Commands Dispatcher;

The program controls the Plug-In by calling of the Plug-In Dispatcher function and sending the commands and events.

The Plug-In sends the control commands into the program using the Program Dispatcher .

The Plug-In can exchange the data with an opened smart card using the program's function Card Commands Dispatcher .

1.3.1 Plug-In Loading Phase

The Card Explorer window of the main program loads the Plug-In every time when the card is opened in the case if this Plug-In is in the "Type Plug-Ins"s list of the Card Type of an opened card.

1.3.1.1 Step 1 : Loading the Plug-In library

The program attempts to connect the plug-in's file as a standard DLL library.

In the case of success the program attempts to register the Plug-In's Dispatcher function named "**DispatchInterface** ".

If this function is found it will be used as the Plug-In Dispatcher.

1.3.1.2 Step 2 : Loading the Plug-In Type

Using the registered Plug-In's Dispatcher the program sends to the Plug-In the "Get Plug-In Type" command:

Command Code	Command Parameter	Command Data
1	2	-

The Plug-In must return 1 as a value of the Command Data parameter.

Warning! Only in the case of this command (Get Plug-In Type) the Command Data is a variable of an Integer type! It's not a pointer! And the Plug-In must set the value of this variable to **10**.

1.3.1.3 Step 3 : Loading the Plug-In Name

Using the Plug-In's Dispatcher the program requests the Plug-In's Name:

Command Code	Command Parameter	Command Data
30	1	-

The Plug-In must returns the pointer on the its local chars array as a value of the Command Data parameter .

This chars array must contains the name of this Plug-In and it must ends by the 0x00 symbol.

1.3.1.4 Step 4 : Loading the Plug-In Description

Using the Plug-In's Dispatcher the program requests the Plug-In's Description:

Command Code	Command Parameter	Command Data
30	2	-

The Plug-In must returns the pointer on the its local chars array as a value of the Command Data parameter .

This chars array must contains the description of this Plug-In and it must ends by the 0x00 symbol.

1.3.1.5 Step 5 : OnRegistered Event

The program sends to the Plug-In the "OnRegistered" event:

Command Code	Command Parameter	Command Data
1	1	the pointer on the Program Dispatcher function

On this event the Plug-In must store the received pointer on the Program Dispatcher into its local variable.

This event is the best time for executing the command "Get Commands Dispatcher" by the Plug-In.

1.3.1.6 Step 6 : Loading the Card Commands Dispatcher Interface

The Plug-In sends to the program the command "Get Commands Dispatcher":

Command Code	Command Parameter	Command Data
1	1000	the pointer on the Card Commands Dispatcher function

On this command the program returns the pointer on the Card Commands Dispatcher function.

The Plug-In may store this address into its local variable.

1.3.2 Active Plug-In Phase

After loading the Plug-In may be activated by the following two ways:

1. Automatically by the program:
 - If this Plug-In is a default Plug-In for the opened card and if the check box "Allow activate the default Plug-In ..." of the Card Type Editor is checked;
2. Manually by user:
 - if the user has clicked on the Plug-In name menu item of the "Plug-Ins" menu of the Card Explorer window or in the main menu of the program;

In the case of a Plug-In activating the program sends into a Plug-In the "OnActivate" event using its Dispatcher.

1.3.2.1 OnActivate Event

Important! Only one event binds the user and the Plug-In - it is "OnActivate" event!
The Plug-In must always process this event correctly!

Command Code	Command Parameter	Command Data
10	1	-

After receiving of this event the Plug-In must run its activating function which makes its visible to user. It may be the function of showing of the Plug-In's main window or something else.

1.3.2.2 OnExecute Event

This event occurs only after the OnActivate event if the check box "Allow autoexecute on activate" of the Card Type Editor is checked and if the OnActivate event was created automatically.

This event do not occurs if the Plug-In was activated by user manually by clicking on the menu item.

Command Code	Command Parameter	Command Data
10	2	-

After receiving of this event the Plug-In must run its main function.

1.3.3 Plug-In Unloading Phase

The program unloads all opened Plug-Ins every time when the card is closed.

1.3.3.1 OnUnregistered Event

On the card closing the program sends to the Plug-In the "OnUnregistered" event:

Command Code	Command Parameter	Command Data
1	100	-

After receiving this event the Plug-In must stop all its operations and prepare all its data for closing.

2 Dispatchers

2.1 Interface Dispatchers

2.1.1 Fundamentals

The Plug-In's Interface Dispatcher (Plug-In Dispatcher) is exported by the Plug-In's function named "DispatchInterface". It is used by the program for sending commands and events from the program to a Plug-In.

The Program's Interface Dispatcher (Program Dispatcher) is the program's function with the same parameters like the Plug-In Dispatcher. It is used by the Plug-In for sending its commands to a program.

Any command is a combination of values of the three parameters of the dispatcher function.

The first two parameters "Command Code" and "Command Parameter" of the dispatcher must be defined for any command. The third parameter "Command Data" may be either defined or not depending on the command.

Calling the Program Dispatcher

Prepare the data before the dispatcher calling:

1. define the local variable of an integer type and assign to it the Command Code value of the selected command;
2. define the local variable of an integer type and assign to it the Command Parameter value of the selected command;

3. define the local variable of a pointer type and assign to it the address of the data buffer according to current command if need;
4. define the three pointer variables and assign to the addresses of the previous three variables; you must use only these pointers as the dispatcher's parameters;

Call the dispatcher using the prepared parameters. The function will returns the result code. In the some cases the parameter Command Data may points to the data buffer after calling the dispatcher.

Receiving the program's calls by the Plug-In Dispatcher

You need to classify the received parameters according to the [Commands](#) using the logical operator like if, switch, case e.t.c.

2.1.2 Function Format

The Plug-In's Interface Dispatcher is a function of an integer type with the three parameters :

N	Parameter Name	Parameter Type	Variable Value Description
1	Command Code	Pointer on Integer	The Code of the command (Integer)
2	Command Parameter	Pointer on Integer	The Parameter of the command (Integer)
3	Command Data	Pointer on Pointer	The pointer on the data buffer

Warning! All parameters are the pointers on the your local variables of an Integer or a Pointer types!

The function returns the Result Code of an Integer type:

Value	Result
1	Success
2	The Command Code is invalid or has an unsupported value
3	The Command Parameter is invalid or has an unsupported value

2.1.3 Commands

Both the Program's and Plug-In's Dispatchers use the one common set of the commands which are based on the values of the dispatchers functions' parameters.

2.1.3.1 "Get Plug-In Type" command

Caller : Program

Receiver : Plug-In

Parameter Name	Parameter Type	Variable	Description
Command Code	pointer on the variable of an Integer type	1 Integer	
Command Parameter	pointer on the variable of an Integer type	2 Integer	
Command Data	pointer on the variable of an Integer type	10 Integer	The address of the Card Commands Dispatcher function of the program;

The Plug-In receives this command by its Dispatcher and it must set the variable of the Command Data parameter to 1.

2.1.3.2 "Get Plug-In Name" command

Caller : Program

Receiver : Plug-In

Parameter Name	Parameter Type	Variable	Description
Command Code	pointer on the variable of an Integer type	30 Integer	
Command Parameter	pointer on the variable of an Integer type	1 Integer	
Command Data	pointer on the variable of a pointer type	the address of the Plug-In's local text buffer; Pointer	The Plug-In must assign to this variable the address of the local chars array which contains the Plug-In's Name string;

The Plug-In receives this command by its Dispatcher and it must assign to the Command Data parameter the address of the Plug-In's local chars array which contains the Plug-In's Name string.

2.1.3.3 "Get Plug-In Description" command

Caller : Program
Receiver : Plug-In

Parameter Name	Parameter Type	Variable	Description
Command Code	pointer on the variable of an Integer type	30 Integer	
Command Parameter	pointer on the variable of an Integer type	2 Integer	
Command Data	pointer on the variable of a pointer type	the address of the Plug-In's local text buffer; Pointer	The Plug-In must assign to this variable the address of the local chars array which contains the Plug-In's Description string;

The Plug-In receives this command by its Dispatcher and it must assign to the Command Data parameter the address of the Plug-In's local chars array which contains the Plug-In's Description string.

2.1.3.4 "Get Commands Interface" command

Caller : Plug-In
Receiver : Program

Parameter Name	Parameter Type	Variable	Description
Command Code	pointer on the variable of an Integer type	1 Integer	
Command Parameter	pointer on the variable of an Integer type	1000 Integer	
Command Data	pointer on the variable of a pointer type	the address of the function; Pointer	the address of the Card Commands Dispatcher function;

The Plug-In sends this command for receiving the address of the Card Commands Dispatcher function.

The program returns the pointer to the Card Commands Dispatcher and the Plug-In must store it into its local variable;

2.1.3.5 "Get Plug-In's Registry Path" command

Caller : Plug-In
Receiver : Program

Parameter Name	Parameter Type	Variable	Description
Command Code	pointer on the variable of an Integer type	30 Integer	
Command Parameter	pointer on the variable of an Integer type	1003 Integer	
Command Data	pointer on the variable of a pointer type	the address of the program's local text buffer; Pointer	The address of the program's local chars array which contains the full registry path to the key which contains the Plug-In's preferences;

2.1.3.6 "Hide Program" command

Caller : Plug-In
Receiver : Program

Parameter Name	Parameter Type	Variable	Description
Command Code	pointer on the variable of an Integer type	20 Integer	
Command Parameter	pointer on the variable of an Integer type	1002 Integer	
Command Data	pointer not used	-	

Hide the program's main window from the desktop.

2.1.3.7 "Show Program" command

Caller : Plug-In
Receiver : Program

Parameter Name	Parameter Type	Variable	Description
Command Code	pointer on the variable of an Integer type	20 Integer	
Command Parameter	pointer on the variable of an Integer type	1001 Integer	
Command Data	pointer not used	-	

Restore the main program's window on the desktop.

2.1.4 Events

The program notifies the Plug-In by sending an events using the Plug-In Dispatcher interface.

2.1.4.1 "OnRegistered" event

Caller : Program
Receiver : Plug-In

Parameter Name	Parameter Type	Variable	Description
Command Code	pointer on the variable of an Integer type	1 Integer	
Command Parameter	pointer on the variable of an Integer type	1 Integer	
Command Data	pointer on the variable of a pointer type	the address of the function; Pointer	The address of the Program Dispatcher

This event occurs when the Plug-In was successfully loaded by the program.

After receiving of this event the Plug-In must store the received pointer on the Program Dispatcher into its local variable.

2.1.4.2 "OnError" event

Caller : Program

Receiver : Plug-In

Parameter Name	Parameter Type	Variable	Description
Command Code	pointer on the variable of an Integer type	10 Integer	
Command Parameter	pointer on the variable of an Integer type	0 Integer	
Command Data	pointer on the variable of a pointer type	the address of the program's local text buffer; Pointer	The address of the program's local chars array which contains an error string;

This event occurs on the any API error.

2.1.4.3 "OnActivate" event

Caller : Program

Receiver : Plug-In

Parameter Name	Parameter Type	Variable	Description
Command Code	pointer on the variable of an Integer type	10 Integer	
Command Parameter	pointer on the variable of an Integer type	1 Integer	
Command Data	pointer not used	- Pointer	

This event occurs when:

- if the user has clicked on the Plug-In name menu item of the "Plug-Ins" menu of the Card Explorer window or in the main menu of the program;
- If this Plug-In is a default Plug-In for the opened card and if the check box "Allow activate the default Plug-In ..." of the Card Type Editor is checked;

After receiving of this event the Plug-In must run its activating function. It may be the function of showing of its main window or something else.

Note : - the activating function of the demo plug-in "GSM Demo :: IMSI Code Loader" shows its main window on the desktop.

2.1.4.4 "OnExecute" event

Caller : Program

Receiver : Plug-In

Parameter Name	Parameter Type	Variable	Description
Command Code	pointer on the variable of an Integer type	10 Integer	
Command Parameter	pointer on the variable of an Integer type	2 Integer	
Command Data	pointer not used	- Pointer	

This event always occurs after OnActivate event if the check box "Allow autoexecute on activate" of the Card Type Editor is checked.

After receiving of this event the Plug-In must run its main function.

Note: - the main function of the demo plug-in "GSM Demo :: IMSI Code Loader" starts the loading of the IMSI code from an opened SIM card.

2.1.4.5 "OnCardWait" event

Caller : Program

Receiver : Plug-In

Parameter Name	Parameter Type	Variable	Description
Command Code	pointer on the variable of an Integer type	10 Integer	
Command Parameter	pointer on the variable of an Integer type	10 Integer	
Command Data	pointer not used	-	

This event occurs when the reader becomes empty.

The Plug-In must stop the card operations.

2.1.4.6 "OnCardReady" event

Caller : Program
Receiver : Plug-In

Parameter Name	Parameter Type	Variable	Description
Command Code	pointer on the variable of an Integer type	10 Integer	
Command Parameter	pointer on the variable of an Integer type	11 Integer	
Command Data	pointer not used	-	

This event occurs when the card is successfully opened in the reader.

2.1.4.7 "OnUnregistered" event

Caller : Program
Receiver : Plug-In

Parameter Name	Parameter Type	Variable	Description
Command Code	pointer on the variable of an Integer type	1 Integer	
Command Parameter	pointer on the variable of an Integer type	100 Integer	
Command Data	pointer not used	-	

This event occurs before the Plug-In will be unloaded.

After receiving of this event the Plug-In must stop all its operations and prepare all its data for closing.

2.2 Commands Dispatcher

2.2.1 Fundamentals

The address of the Card Commands Dispatcher may be received by Plug-In using the command "Get Commands Interface".

How to send the command into a card:

1. you need to define the Plug-In's local buffer (send buffer) for preparing of the sending data;
2. you need to prepare the data in the send buffer according to ISO 7816-4:
 - for T0 protocol :
 - Cla Ins P1 P2 P3 [DataIn]
 - for T1 protocol :
 - Cla Ins P1 P2 Lc [DataIn] Le
3. you need to define the local variable of an Integer type and assign to it the length of the data into a send buffer;
4. you need to define the local variable of the Pointer type; the program will assign to it an address of the received data buffer automatically;
5. you need to define the local variable of an Integer type; the program will assign to it the length value of the received data automatically;
6. you need to define the four variables of the pointer type and assign to its the addresses of the four previous prepared variables; call the Card Commands Dispatcher using these four pointers as a parameters;
7. in the case of successful data sending the function will return the Result Code 0; the program automatically assigns the address of the buffer of the received data to your variable of the pointer type and the received data length to your variable of an integer type;
8. the received data is formatted by program according to the ISO 7816-4:
 - [DataOut] SW1SW2

2.2.2 Function Format

The Card Commands Dispatcher is the main program's function . It used by the Plug-In for sending data into a smart card and receiving the card's answers:

N	Parameter Name	Parameter Type	Variable Description
1	Send Buffer	Pointer on Pointer	Pointer on the Plug-In's local data buffer which contains the command data for sending to card
2	Send Buffer Length	Pointer on Integer	Length of the Send Buffer data in the bytes (Integer)
3	Received Buffer	Pointer on Pointer	Pointer on the program's local data buffer which will contains the received data
4	Received Buffer Length	Pointer on Integer	Length of the received data in the Received Buffer in bytes (Integer)

Warning! All parameters are the pointers on the your local variables of an Integer or Pointer type!

The function returns the Result Code of an Integer type:

Value	Result
0	The command data was successfully sent into a card and the card's answer was received
-1	The command data was not sent

Index

- A -

Active Plug-In Phase 4

- C -

Card Commands Dispatcher 1
Card Explorer Extensions Plug-In 1
Command "Get Commands Interface" 8
Command "Get Plug-In Description" 8
Command "Get Plug-In Name" 7
Command "Get Plug-In Type" 7
Command "Get Plug-In's Registry Path" 8
Command "Hide Program" 9
Command "Show Program" 10
Commands Parameters 7

- D -

Data Types 1

- E -

Event "OnActivate" 11
Event "OnCardReady" 13
Event "OnCardWait" 12
Event "OnError" 11
Event "OnExecute" 12
Event "OnRegistered" 10
Event "OnUnregistered" 13

- F -

Function Format : Commands Dispatcher 14
Function Format : Interface Dispatcher 6
Fundamentals : Commands Dispatcher 13
Fundamentals : Interface Dispatchers 5
Fundamentals : Plug-Ins Using 2

- M -

Main Program 1

- P -

Plug-In Dispatcher 1
Plug-In Loading Phase 2
Plug-In Unloading Phase 5
Program Dispatcher 1