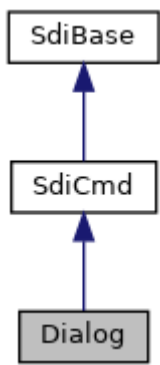


Dialog Class Reference

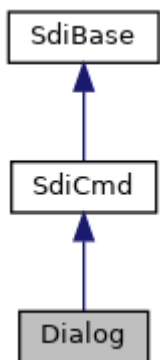
```
#include <sdi_if.h>
```

Inheritance diagram for Dialog:



[\[legend\]](#)

Collaboration diagram for Dialog:



[\[legend\]](#)

Public Types

```
enum REQUEST\_CARD\_MODE { REQUEST\_CARD\_STANDARD = 0,  
  REQUEST\_CARD\_FALLBACK = 1,  
  REQUEST\_CARD\_RETRY = 2  
}
```

Public Member Functions

bool	<u>clearScreen</u> (bool epp=true) Clear display (24 03) <u>More...</u>
int	<u>display</u> (const std::string &text, bool epp=true) Call <i>Handle Display</i> (24 03) <u>More...</u>
int	<u>secureInput</u> (bool epp=true) Start SDI command <i>Handle Secure Input</i> (24 04) <u>More...</u>
int	<u>menu</u> (bool epp=true) Start SDI command <i>Handle Menu</i> (24 05) <u>More...</u>
int	<u>requestCard</u> (unsigned char technology, enum <u>REQUEST_CARD_MODE</u> mode= <u>REQUEST_CARD_STANDARD</u> , bool epp=true) Start SDI command <i>Handle Card Request Display</i> (24 06) <u>More...</u>
bool	<u>idleScreen</u> (bool epp=true) Send SDI command <i>Display Idle Screen</i> (24 07) <u>More...</u>
int	<u>captureSignature</u> (std::vector< unsigned char > &signature, std::string &format, bool epp=true) Start SDI command <i>Handle Signature Capture</i> (24 08) <u>More...</u>
int	<u>htmlDialog</u> (const std::string &fname, bool epp=true) Call SDI command <i>Handle HTML Dialog</i> (24 0A) <u>More...</u>
int	<u>getAsyncResult</u> (bool epp=true) Get result of asynchronous dialog invocation (24-0B) <u>More...</u>
void	<u>setTemplate</u> (unsigned char id) Set HTML template identifier (DFA010) <u>More...</u>
void	<u>setInputTemplate</u> (unsigned char id) Set input template identifier (DFA007) <u>More...</u>
void	<u>setLanguage</u> (unsigned char language) Set language (DFA008) <u>More...</u>
void	<u>addData</u> (const std::string &name, const std::string &value) Add display data name-value pair (FFA011, DFA012, DFA013) <u>More...</u>
void	<u>setTimeout</u> (int seconds) Set dialog timeout (DFA005) <u>More...</u>
void	<u>setOptions</u> (unsigned options) Set dialog options (DFA13D) <u>More...</u>
void	<u>setEnterKeyLabel</u> (const std::string &label) Set label for the <i>Enter</i> soft key or the key helper in the dialog control bar (DFA13E) <u>More...</u>
void	<u>setClearKeyLabel</u> (const std::string &label) Set label for the <i>Clear</i> soft key or the key helper in the dialog control bar (DFA13F) <u>More...</u>
void	<u>setCancelKeyLabel</u> (const std::string &label)

Set label for the *Cancel* soft key or the key helper in the dialog control bar (DFA140) [More...](#)

void [setUpKeyLabel](#) (const std::string &label)
Set label for the *up* soft key or the key helper in the dialog control bar (DFA141) [More...](#)

void [setDownKeyLabel](#) (const std::string &label)
Set label for the *down* soft key or the key helper in the dialog control bar (DFA142) [More...](#)

void [setHeader](#) (const std::string &text)
Set Text shown in the header area (DFA143) [More...](#)

void [setBeep](#) (bool active=true)
Enable/Disable a beep played when the dialog appears (DFA00D) [More...](#)

void [addAction](#) (const std::string &key, const std::string &action)
Set additional key action (FFA106, DFA13B, DFA13C) [More...](#)

void [setMaskingCharacter](#) (char c)
Set masking character (DFA019) [More...](#)

void [setAlternativeInputFormat](#) (const char *f)
Configure alternative input format string (DFA01A) [More...](#)

void [setAmount](#) (const unsigned char *amount)
Configure amount (9F02) [More...](#)

void [setCurrency](#) (const unsigned char *currency)
Configure currency (5F2A) [More...](#)

void [setTransactionType](#) (unsigned char txnType)
Configure transaction type (9C) for *Handle Card Request Display* [More...](#)

void [setMenuText](#) (const std::string &text)
Set menu text prompt (DFA00A) for SDI command *Handle Menu* [More...](#)

void [addItem](#) (const std::string &item)
add menu item (DFA00B) for SDI command *Handle Menu* [More...](#)

void [setPreSelected](#) (unsigned char itemNumber)
Set a preselected item other than the first or disable pre-selection (DFA00C) [More...](#)

void [setAdminMenu](#) (unsigned char adminMenu)
Enable showing of Info/Menu softkeys/keyhelpers or control admin menu access for idle screen to display (DFA155). This tag is ignored, if administration mode is disabled (default). [More...](#)

unsigned char [getSelected](#) ()
Get selected item number (DFA00C) from *Handle Menu* invocation. [More...](#)

std::string [get](#) (const std::string &name)
Get output data of dialog invocation (FFA014, DFA015, DFA016) [More...](#)

std::vector< std::string
> [getInputFieldNames](#) ()
List up data input field names found in the dialog invocation result (FFA014, DFA015) [More...](#)

► Public Member Functions inherited from [SdiCmd](#)

	SdiCmd ()
virtual	~SdiCmd ()
enum SDI_SW12	sendReceive (unsigned char cla, unsigned char ins, unsigned char p1=0, unsigned char p2=0, unsigned maxResponseSize=2048) Generic SDI command exchange function combining sending of request, waiting and receiving the response. More...
int	send (unsigned char cla, unsigned char ins, unsigned char p1=0, unsigned char p2=0) Like sendReceive() but without waiting for response. More...
enum SDI_SW12	receive (unsigned maxResponseSize=2048) Receive SDI response after request has been send with send() and data availability has been signaled. More...
void	set (const char *path, int value , unsigned fixedLength=0) Set integer data object SDI command. More...
void	set (const char *path, uint32_t value , unsigned fixedLength=0) Set unsigned integer data object SDI command. More...
void	set (const char *path, const unsigned char *data, unsigned dataLen) Set byte array data object in SDI command. More...
void	set (const char *path, unsigned char byteValue) Set single byte value in SDI command. More...
void	set (const char *path, const std::vector< unsigned char > &data) Set byte array data object in SDI command. More...
void	set (const char *path, const std::string & value) Set character data object in SDI command. More...
void	setCommandDestination (bool epp, bool force=false) Set command destination for systems with EPP Unless force the command destination is set only if parameter epp is true. More...
virtual void	clear () Remove all SDI command command input data. More...
virtual void	clear (const char *path) Remove single input data stored in dataIn . More...
virtual void	clearResults () Remove all SDI response data. More...
void	importResults (const SdiCmd &intermediate) Transfer status word, client error and response data from an intermediate command while keeping command input data. More...
void	importResults (const unsigned char * sw12 , const unsigned char *tlvData, unsigned tlvSize) Inject result data from a call back. More...
bool	get (const char *path, int & value)
bool	get (const char *path, uint32_t & value)
int	get (const char *path, unsigned char *buffer, unsigned bufferSize)

bool [get](#) (const char *path, unsigned char &[value](#))
 bool [get](#) (const char *path, std::vector< unsigned char > &buffer)
 bool [get](#) (const char *path, std::string &[value](#))
 std::string [getString](#) (const char *path)
 ▶ Public Member Functions inherited from [SdiBase](#)
 [SdiBase](#) ()
 enum [SDI_SW12](#) [getSdiSw12](#) ()
 int [getAdditionalResultValue](#) ()
 Access Additional Result Value if returned in SDI response. [More...](#)
[SDICLIENT_ERROR](#) [getClientError](#) ()
 Access client side error codes. [More...](#)
 enum [SDI_SW12](#) [receiveSW12](#) ()
 Receive SDI server response with no data. [More...](#)
 void [clear](#) ()
 clear result data obtained from SDI communication [More...](#)
 void [importResults](#) (const [SdiBase](#) &intermediate)
 set result data obtained from intermediate SDI communication [More...](#)

Static Public Member Functions

static void [showLedArea](#) (bool show)
 Show or hide area with status indicators for contactless transactions by SDI command *Activate LEDs* (24-09) [More...](#)
 static void [abort](#) ()
 Abort a dialog by sending *SDI SysAbort* (20 02) [More...](#)

Static Public Attributes

const static int [DIALOG_SUCCESS](#) = 0
 Dialog invocation return value on success [More...](#)
 const static int [DIALOG_CANCEL_PRESSED](#) = -1
 Dialog invocation return value in case of cancel key pressed [More...](#)
 const static int [DIALOG_TIMEOUT](#) = -3
 Dialog invocation return value in case dialog timeout [More...](#)
 const static int [DIALOG_SYS_ABORT](#) = -8
 Dialog invocation return value in case of abort by *SDI Sys Abort* (20-02), e.g. [abort\(\)](#) [More...](#)
 const static int [DIALOG_CLIENT_ERROR](#) = -14
 Dialog invocation return value in case of client side error [More...](#)
 const static int [DIALOG_SDI_SW12](#) = -15
 Dialog invocation return value in case of other SDI status word [More...](#)

const static int	<u>DIALOG_NO_ASYNC_DIALOG</u> = 1 <u>getAsyncResult()</u> return value when there is no asynchronous dialog running More...
const static int	<u>DIALOG_IN_PROGRESS</u> = 2 <u>getAsyncResult()</u> return value when the asynchronous dialog is in progress More...
const static unsigned	<u>DLG_DisplayOnly</u> = 0x00000000
const static unsigned	<u>DLG_CancelKey</u> = 0x00000002
const static unsigned	<u>DLG_ClearKey</u> = 0x00000004
const static unsigned	<u>DLG_EnterKey</u> = 0x00000008
const static unsigned	<u>DLG_NoLEDs</u> = 0x00000010
const static unsigned	<u>DLG_CtlsLogo</u> = 0x00000020
const static unsigned	<u>DLG_QuestionLogo</u> = 0x00000040
const static unsigned	<u>DLG_WarningLogo</u> = 0x00000080
const static unsigned	<u>DLG_ErrorLogo</u> = 0x00000100
const static unsigned	<u>DLG_SuccessLogo</u> = 0x00000200
const static unsigned	<u>DLG_WaitLogo</u> = 0x00000400
const static unsigned	<u>DLG_Async</u> = 0x00000800
const static unsigned	<u>DLG_StoreAsyncResult</u> = 0x00001000
const static unsigned	<u>DLG_HideSoftKeys</u> = 0x00002000
const static unsigned	<u>DLG_TextAlignLeft</u> = 0x00004000
const static unsigned	<u>DLG_TextAlignRight</u> = 0x00008000
const static unsigned	<u>DLG_TextAlignTop</u> = 0x00010000
const static unsigned	<u>DLG_TextAlignButton</u> = 0x00020000
const static unsigned	<u>DLG_NoHeader</u> = 0x00040000
const static unsigned	<u>DLG_ClearOnReturn</u> = 0x00100000

const static unsigned	<u>DLG_AbortOnCardRemove</u> = 0x00200000
const static unsigned	<u>DLG_IgnoreExtAbort</u> = 0x00400000
const static unsigned	<u>MENU_NoOpts</u> = 0x00000000
const static unsigned	<u>MENU_NoLEDs</u> = 0x00000002
const static unsigned	<u>MENU_ClearOnReturn</u> = 0x00000004
const static unsigned	<u>MENU_AbortOnCardRemove</u> = 0x00000008
const static unsigned	<u>MENU_IgnoreExtAbort</u> = 0x00000010
const static unsigned	<u>MENU_NoHeader</u> = 0x00000020

Additional Inherited Members

► Protected Member Functions inherited from <u>SdiBase</u>	
void	<u>setSdiSw12</u> (enum <u>SDI_SW12</u> s)
void	<u>setClientError</u> (int libsdiprotocol_result)
► Protected Attributes inherited from <u>SdiCmd</u>	
void *	<u>dataIn</u> Command input buffer collecting TLV data items. <u>More...</u>
void *	<u>dataOut</u> Command response TLV data items. <u>More...</u>
► Protected Attributes inherited from <u>SdiBase</u>	
unsigned short	<u>sw12</u>
int	<u>additionalResultValue</u>
<u>SDICLIENT_ERROR</u>	<u>clientErr</u>

Detailed Description

Interface for SDI Display Interface, command class 24. This is basically required for External PIN pad control.

This table lists up which getters and setters are relevant for a given SDI command out of

- *Handle Display* [display\(\)](#)
- *Handle Secure Input* [secureInput\(\)](#)

- Handle Menu [menu\(\)](#)
- Handle Card Request Display [requestCard\(\)](#)
- Handle Signature Capture [captureSignature\(\)](#)
- Handle HTML Dialog [htmlDialog\(\)](#)

Tag	Setter/Getter	Function List
DFA005	setTimeout()	display() , secureInput() , htmlDialog() , captureSignature()
DFA007	setInputTemplate()	secureInput()
DFA008	setLanguage()	secureInput() , htmlDialog() , captureSignature()
DFA00A	setMenuText()	menu()
DFA00B	addItemMenu()	menu()
DFA00C	setPreSelected()	menu()
DFA00D	setBeep()	display() , secureInput() , htmlDialog() , menu()
DFA010	setTemplate()	display()
DFA019	setMaskingCharacter()	secureInput()
DFA01A	setAlternativeInputFormat()	secureInput()
DFA13D	setOptions()	display() , menu() , htmlDialog()
DFA13E	setEnterKeyLabel()	display() , menu()
DFA13F	setClearKeyLabel()	display()
DFA140	setCancelKeyLabel()	display() , menu()
DFA141	setUpKeyLabel()	menu()
DFA142	setDownKeyLabel()	menu()
DFA143	setHeader()	display() , menu()
FFA011	addData()	display() , htmlDialog()
FFA106	addAction()	display() , menu()
9F02	setAmount()	secureInput() , requestCard()
5F2A	setCurrency()	secureInput() , requestCard()
9C	setTransactionType()	requestCard()

Other functions lack that any getters and setters are

- Handle Display without parameters [clearScreen\(\)](#)
- Display Idle Screen [idleScreen\(\)](#)
- Activate LEDs [showLedArea\(\)](#)
- [abort\(\)](#)
- [getAsyncResult\(\)](#)

Member Enumeration Documentation

? REQUEST_CARD_MODE

enum [REQUEST_CARD_MODE](#)

Dialog use cases for [requestCard\(\)](#)

Enumerator

REQUEST_CARD_STANDARD Initial request card dialog, e.g. "please tab".

REQUEST_CARD_FALLBACK Modified dialog for a fall back use case.

REQUEST_CARD_RETRY Modified dialog, e.g. "please tab again".

Member Function Documentation

[?](#) abort()

static void abort () static

Abort a dialog by sending *SDI SysAbort* (20 02)

[?](#) addAction()

```
void addAction ( const std::string & key,  
                const std::string & action  
                )
```

Set additional key action (FFA106, DFA13B, DFA13C)

Setter for

[display\(\)](#)

[menu\(\)](#)

Parameters

[in] key ADKGUI key code, utf-8

[in] action associated dialog action, utf-8

[?](#) addData()

```
void addData ( const std::string & name,
```

```
        const std::string & value
    )
```

Add display data name-value pair (FFA011, DFA012, DFA013)

Adds another name-value pair for the dialog

Setter for

[display\(\)](#)

[htmlDialog\(\)](#)

Parameters

[in] name variable name DFA012, ans

[in] value variable value DFA013, utf-8

[?](#) **addMenuItem()**

```
void addMenuItem ( const std::string & item )
```

add menu item (DFA00B) for SDI command *Handle Menu*

Setter for

[menu\(\)](#)

Parameters

[in] item menu item

[?](#) **captureSignature()**

```
int captureSignature ( std::vector< unsigned char > & signature,
                      std::string & format,
                      bool epp = true
                      )
```

Start SDI command *Handle Signature Capture* (24 08)

Send SDI command *Handle Menu* with parameters configured with this object's setters.

Setters

[setLanguage\(\)](#)

[setTimeout\(\)](#)

Parameters

[out] signature signature data (DFA01D)
[out] format signature format (DFA01C)
[in] epp command destination DFA150 is External PIN pad

Returns

[Dialog invocation return values](#)

[? clearScreen\(\)](#)

bool clearScreen (bool epp = true)

Clear display (24 03)

Issue *Handle Display* without parameters except destination (DFA150)

Parameters

[in] epp command destination (DFA150) is External PIN pad

Returns

true for success

[? display\(\)](#)

```
int display ( const std::string & text,  
             bool epp = true  
            )
```

Call *Handle Display* (24 03)

Send SDI command *Handle Display* with parameters configured with this object's setters.

Setters

[setTemplate\(\)](#)

[addData\(\)](#)

[setTimeout\(\)](#)

[setOptions\(\)](#)

[setEnterKeyLabel\(\)](#)

[setClearKeyLabel\(\)](#)

[setCancelKeyLabel\(\)](#)

[setHeader\(\)](#)

[setBeep\(\)](#)

[addAction\(\)](#)

Parameters

[in] text value for parameter *text* (FFA011/DFA013), utf-8

[in] epp command destination (DFA150) is External PIN pad

Returns

[Dialog invocation return values](#)

[? get\(\)](#)

std::string get (const std::string & *name*)

Get output data of dialog invocation (FFA014, DFA015, DFA016)

Getter for

[secureInput\(\)](#)

[htmlDialog\(\)](#)

Parameters

[in] name variable name (DFA015), ans

Returns

variable value, empty string if not found

[? getAsyncResult\(\)](#)

int getAsyncResult (bool *epp* = true)

Get result of asynchronous dialog invocation (24-0B)

Parameters

[in] epp command destination DFA150 is External PIN pad

Returns

dialog status

0, [DIALOG_SUCCESS](#) Dialog finished with *UI_ERR_OK*

1, [DIALOG_NO_ASYNC_DIALOG](#) No asynchronous dialog started

2, [DIALOG_IN_PROGRESS](#)

or any other of [Dialog invocation return values](#)

[?](#) **getInputFieldNames()**

std::vector<std::string> getInputFieldNames ()

List up data input field names found in the dialog invocation result (FFA014, DFA015)

Can be used to iterate through all entered data in conjunction with the [get\(\)](#) function.

Getter for

[secureInput\(\)](#)

[htmlDialog\(\)](#)

Returns

list of input field names found in the dialog result

[?](#) **getSelected()**

unsigned char getSelected ()

Get selected item number (DFA00C) from *Handle Menu* invocation.

Getter for

[menu\(\)](#)

Returns

number of selected item, count starts with 1

[?](#) **htmlDialog()**

```
int htmlDialog ( const std::string & fname,  
                bool                epp = true  
                )
```

Call SDI command *Handle HTML Dialog* (24 0A)

Send SDI command *Handle HTML Dialog* with parameters configured with this object's setters.

Setters

[addData\(\)](#)

[setTimeout\(\)](#)

[setOptions\(\)](#)

[setBeep\(\)](#)

[setLanguage\(\)](#)

Parameters

[in] fname HTML file name without path (DFA020)

[in] epp command destination (DFA150) is External PIN pad

Returns

[Dialog invocation return values](#)

Getters

[get\(\)](#)

[?](#) **idleScreen()**

```
bool idleScreen ( bool epp = true )
```

Send SDI command *Display Idle Screen* (24 07)

Parameters

[in] epp command destination DFA150 is External PIN pad

Returns

true for success

[?](#) **menu()**

```
int menu ( bool epp = true )
```

Start SDI command *Handle Menu* (24 05)

Send SDI command *Handle Menu* with parameters configured with this object's setters.

Setters

[setMenuText\(\)](#)

[addMenuItem\(\)](#)

[setTimeout\(\)](#)

[setOptions\(\)](#)

[setEnterKeyLabel\(\)](#)

[setCancelKeyLabel\(\)](#)

[setUpKeyLabel\(\)](#)

[setDownKeyLabel\(\)](#)

[setHeader\(\)](#)

[setBeep\(\)](#)

[setPreSelected\(\)](#)

[addAction\(\)](#)

Parameters

[in] *epp* command destination DFA150 is External PIN pad

Returns

selected item index or `UIError` as defined in http://html/gui_error.h, `UI_ERR_CONNECTION_LOST` for client side error - inspect with [getClientError\(\)](#), `UI_ERR_PROTOCOL` for other SDI SW12 - inspect with [getSdiSw12\(\)](#)

Getters

[getSelected\(\)](#) (DFA00C)

[getAdditionalResultValue\(\)](#) (DFA000)

? requestCard()

```
int requestCard ( unsigned char
```

technology,

```

        enum REQUEST\_CARD\_MODE mode = REQUEST_CARD_STANDARD,
        bool epp = true
    )

```

Start SDI command *Handle Card Request Display* (24 06)

Send SDI command *Handle Menu* with parameters configured with this object's setters.

Setters

[setAmount\(\)](#)

[setCurrency\(\)](#)

[setTransactionType\(\)](#)

Parameters

- [in] technology (DFA108, see [Card Detection \(23-01\)](#))
- [in] mode use case variant out of [REQUEST_CARD_MODE](#) (DFA10A)
- [in] epp command destination DFA150 is External PIN pad

Returns

[Dialog invocation return values](#)

? secureInput()

```
int secureInput ( bool epp = true )
```

Start SDI command *Handle Secure Input* (24 04)

Send SDI command *Handle Secure Input* with parameters configured with this object's setters.

Setters

[setInputTemplate\(\)](#)

[setLanguage\(\)](#)

[setTimeout\(\)](#)

[setBeep\(\)](#)

[setMaskingCharacter\(\)](#)

[setAlternativeInputFormat\(\)](#)

[setAmount\(\)](#)

[setCurrency\(\)](#)

Parameters

[in] epp command destination DFA150 is External PIN pad

Returns

[Dialog invocation return values](#)

Getters

[get\(\)](#)

[?](#) setAdminMenu()

void setAdminMenu (unsigned char *adminMenu*)

Enable showing of Info/Menu softkeys/keyhelpers or control admin menu access for idle screen to display (DFA155). This tag is ignored, if administration mode is disabled (default).

Setter for

[idleScreen\(\)](#)

Parameters

admin menu access mode in idle screen to display. Possible values are: 00 = Don't show admin menu softkeys/keyhelpers and allow to enable admin menu by key combination
[in] adminMenu 2-5-8 or clockwise touch presses on four display corners (default). 01 = Don't show admin menu softkeys/keyhelpers, but disallow to enable admin menu. 02 = Show admin menu softkeys/keyhelpers.

[?](#) setAlternativeInputFormat()

void setAlternativeInputFormat (const char * *f*)

Configure alternative input format string (DFA01A)

Setter for

[secureInput\(\)](#)

Parameters

[in] f format string

[?](#) setAmount()

void setAmount (const unsigned char * *amount*)

Configure amount (9F02)

Setter for

[secureInput\(\)](#)

[requestCard\(\)](#)

Parameters

[in] amount amount in numeric format n12 (6 bytes BCD)

[**? setBeep\(\)**](#)

void setBeep (bool *active* = **true**)

Enable/Disable a beep played when the dialog appears (DFA00D)

Setter for

[display\(\)](#)

[secureInput\(\)](#)

[menu\(\)](#)

[htmlDialog\(\)](#)

Parameters

[in] active flag: True for beep, false for no beep

[**? setCancelKeyLabel\(\)**](#)

void setCancelKeyLabel (const std::string & *label*)

Set label for the *Cancel* soft key or the key helper in the dialog control bar (DFA140)

Setter for

[display\(\)](#)

Parameters

[in] label label, utf-8

[**? setClearKeyLabel\(\)**](#)

void setClearKeyLabel (const std::string & *label*)

Set label for the *Clear* soft key or the key helper in the dialog control bar (DFA13F)

Setter for

[display\(\)](#)

[menu\(\)](#)

Parameters

[in] label label, utf-8

[?](#) **setCurrency()**

void setCurrency (const unsigned char * *currency*)

Configure currency (5F2A)

Setter for

[secureInput\(\)](#)

[requestCard\(\)](#)

See [Supported Currencies](#)

Parameters

[in] currency currency in numeric format n3 (2 bytes BCD)

[?](#) **setDownKeyLabel()**

void setDownKeyLabel (const std::string & *label*)

Set label for the *down* soft key or the key helper in the dialog control bar (DFA142)

Setter for

[menu\(\)](#)

Parameters

[in] label label, utf-8

[?](#) **setEnterKeyLabel()**

void setEnterKeyLabel (const std::string & *label*)

Set label for the *Enter* soft key or the key helper in the dialog control bar (DFA13E)

Setter for

[display\(\)](#)

[menu\(\)](#)

Parameters

[in] label label, utf-8

[?](#) **setHeader()**

void setHeader (const std::string & *text*)

Set Text shown in the header area (DFA143)

Setter for

[display\(\)](#)

[menu\(\)](#)

Parameters

[in] text header text, utf-8

[?](#) **setInputTemplate()**

void setInputTemplate (unsigned char *id*)

Set input template identifier (DFA007)

Setter for

[secureInput\(\)](#)

Parameters

[in] id input template id

[?](#) **setLanguage()**

void setLanguage (unsigned char *language*)

Set language (DFA008)

Setter for

[secureInput\(\)](#)

[captureSignature\(\)](#)

[htmlDialog\(\)](#)

Parameters

[in] language [%SDI language code](#)

[?](#) **setMaskingCharacter()**

void setMaskingCharacter (char c)

Set masking character (DFA019)

Setter for

[secureInput\(\)](#)

Parameters

[in] c character used to illustrate the input length

[?](#) **setMenuText()**

void setMenuText (const std::string & text)

Set menu text prompt (DFA00A) for SDI command *Handle Menu*

Setter for

[menu\(\)](#)

Parameters

[in] text menu text prompt

[?](#) **setOptions()**

void setOptions (unsigned options)

Set dialog options (DFA13D)

Setter for

[display\(\)](#) with [Dialog options](#)

[menu\(\)](#) with [Menu options](#)

[htmlDialog\(\)](#) with [Dialog options](#)

Parameters

[in] options dialog options, see [Dialog options](#) or [Menu options](#)

[?](#) **setPreSelected()**

void setPreSelected (unsigned char *itemNumber*)

Set a preselected item other than the first or disable pre-selection (DFA00C)

Setter for

[menu\(\)](#)

Parameters

[in] itemNumber number of item to select, count starts with 1, invalid value disables pre-selection

[?](#) **setTemplate()**

void setTemplate (unsigned char *id*)

Set HTML template identifier (DFA010)

Setter for

[display\(\)](#)

Parameters

[in] id identifier >= '02'

[?](#) **setTimeout()**

void setTimeout (int *seconds*)

Set dialog timeout (DFA005)

Setter for

[display\(\)](#)

[secureInput\(\)](#)

[captureSignature\(\)](#)

[htmlDialog\(\)](#)

Parameters

[in] seconds timeout in seconds, 0 or -1 for infinite depending on template id

[?](#) setTransactionType()

void setTransactionType (unsigned char *txnType*)

Configure transaction type (9C) for *Handle Card Request Display*

See [Supported Currencies](#)

Setter for

[requestCard\(\)](#)

Parameters

[in] txnType transaction type

[?](#) setUpKeyLabel()

void setUpKeyLabel (const std::string & *label*)

Set label for the *up* soft key or the key helper in the dialog control bar (DFA141)

Setter for

[menu\(\)](#)

Parameters

[in] label label, utf-8

[?](#) showLedArea()

```
static void showLedArea ( bool show ) static
```

Show or hide area with status indicators for contactless transactions by SDI command *Activate LEDs* (24-09)

Parameters

[in] show *true* for show, *false* for hide

The documentation for this class was generated from the following file:

- sdiClient/client/[sdi_if.h](#)