

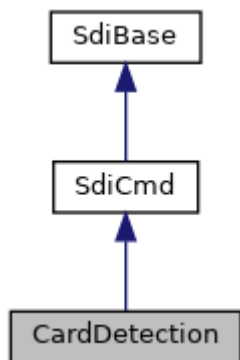
CardDetection Class Reference

Interface for SDI Card Detection Interface, command class 23.

[More...](#)

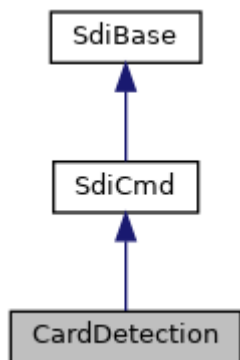
```
#include <sdi_if.h>
```

Inheritance diagram for CardDetection:



[\[legend\]](#)

Collaboration diagram for CardDetection:



[\[legend\]](#)

Public Types

```
enum DetectionMode { DETECTION\_MODE\_BLOCKING,  

DETECTION\_MODE\_POLLING,  

DETECTION\_MODE\_CALLBACK  

}
```

Public Member Functions

void [setDetectionMode](#) (const enum [DetectionMode](#) mode)
Set Card Detection mode. [More...](#)

void [setTecStartOptions](#) (const std::vector< unsigned char > &opts)
Set options for [cts_StartSelection\(\)](#) send by [startSelection\(\)](#) (DFA12A) [More...](#)

void [setTecConfig](#) (const std::vector< unsigned char > &opts)
Set TECSEL ADK configuration (DFA133) [More...](#)

void [setCancelButton](#) (bool enable)
Method for activation of cancel button. [More...](#)

void [setCardEntryValueDeactivation](#) (unsigned char b)
Configure which input is skipped at manual card data entry. [More...](#)

void [setAlternativeInputFormat](#) (const char *f)
Configure alternate input format string for CVV value. [More...](#)

void [setCardRemovalTimeout](#) (unsigned timeout_ms)
For hybrid card readers set timeout for mag. stripe data available after start of card pull-out (DFA153). [More...](#)

void [setCallback](#) (void(*cb)(unsigned char technology, void *context), void *ctx)
Set Callback function to be called in [DETECTION_MODE_CALLBACK](#) when Card Detection ends. [More...](#)

void [cardDetectedCallback](#) (unsigned char *[dataIn](#), unsigned short sizeIn)
Internal Card Detected Callback handler function. [More...](#)

int [startSelection](#) (unsigned char supportedTechnologies, unsigned seconds)
Start Technology Selection (23-01) or (23-03) according to the previously set detection mode using [setDetectionMode\(\)](#). Default mode is the blocking mode. [More...](#)

enum [libsdi::SDI_SW12](#) [stopSelection](#) ()
Abort Technology Selection via SDI Server's Sys Abort (20-02) when in blocking mode, otherwise send Stop Card Detection (23-05). [More...](#)

unsigned char [receiveTechnology](#) ()
Get the Technology Selection Result. [More...](#)

bool [cardReadAtEpp](#) ()
Check at which device the card has been read (DFA150). [More...](#)

unsigned char [pollTechnology](#) ()
Poll for the asynchronous Card Detection status (23-04). [More...](#)

enum [libsdi::SDI_SW12](#) [addTechnology](#) (unsigned char technology, const std::vector< unsigned char > &opts)

enum libsdi::SDI_SW12	Add a technology to a running asynchronous Card Detection (23-06) More... removeTechnology (unsigned char technology)
std::string	Remove a technology from a running asynchronous Card Detection (23-07) More... getPan () Access PAN from mag. stripe or manual entry (DF1A) More...
std::string	getTrack2 () Access track 2 from mag. stripe (DF49) More...
int	getTrack2Bin (std::vector< unsigned char > &data) Access obfuscated track 2 data (DF1B) Note: Data is appended to existing content. More...
std::string	getCardholderName () Access card holder name from track 1 (5F20) More...
std::string	getServiceCode () Access service code from track 3 (5F30) More...
std::string	getTrack1 () Access mag. stripe track 1 (DFA040) More...
int	getPluginResponseData (std::vector< unsigned char > &data) Access plugin response data (DFA101) in case one plugin has been triggered. More...
int	getPluginResponseData (unsigned index, int32_t &pluginId, int32_t &pluginResponseCode, std::vector< unsigned char > &data) Access plugin response data (FFA107, DFA101, DFA10C, DFA10D) More...
bool	getPluginResponseData (std::vector< PluginResult > &results) Access plugin response data (FFA107, DFA101, DFA10C, DFA10D) More...
std::string	getString (unsigned CTS_DATA_TAG) Access cts_WaitSelection result data (FFA102) CTS_DATA_TAG_VAS_DATA: JSON string returned by NFC_VAS_Activate or NFC_VAS_Decrypt (DFDB24) More...
unsigned char	getValue (unsigned CTS_DATA_TAG, unsigned char defaultValue) Access cts_WaitSelection result data tags of 1 byte length (FFA102) CTS_DATA_TAG_NFC_RESULT CTS_DATA_TAG_CARDS_TOTAL_COUNT CTS_DATA_TAG_CARDS_A CTS_DATA_TAG_CARDS_B CTS_DATA_TAG_CARDS_F CTS_DATA_TAG_CARD_TYPE CTS_DATA_TAG_VAS_RESULT CTS_DATA_TAG_EMV_RESULT CTS_DATA_TAG_VAS_DECRYPT_DATA_RESULT. More...
unsigned char	getValue (unsigned CTS_DATA_TAG, int n, unsigned char defaultValue) Access cts_WaitSelection result data tags of 1 byte length. More...
int	getData (std::vector< unsigned char > &data) Access cts_WaitSelection result data (complete FFA102 content) More...
int	getData (unsigned CTS_DATA_TAG, std::vector< unsigned char > &data) Access cts_WaitSelection result data CTS_DATA_TAG_CARD_INFO. More...
int	getData (unsigned CTS_DATA_TAG, int n, std::vector< unsigned char > &data) Access cts_WaitSelection result data CTS_DATA_TAG_CARD_INFO. More...

int [getTecselData](#) (std::vector< unsigned char > &data)
 Access cts_WaitSelection result data in case CTS_DATA_TLV is not set ([SDI](#) tag DFA154) This can include either the EMV-ADK CTLS framework response code (result of continue offline) or the contactless card type (result of smart reset). [More...](#)

int [startMsrRead](#) (unsigned timeout_sec)
 Send MSR read request (21-01) [More...](#)

int [msrSetOptions](#) (const std::vector< unsigned char > &opts)
 Set MSR options (21-03) [More...](#)

int [msrGetTrackStatus](#) (int track)

int [msrGetCardSpecificToken](#) (std::vector< unsigned char > &token)
[CardDetection](#) ()
[~CardDetection](#) ()

virtual void [clear](#) ()

► 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](#) (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 Attributes

const static unsigned char [SDI_TEC_CHIP](#) = 1

const static unsigned char [SDI_TEC_MAGN](#) = 2

const static unsigned char [SDI_TEC_CTLs](#) = 4

const static unsigned char [SDI_TEC_MANU](#) = 8

Additional Inherited Members

► Protected Member Functions inherited from [SdiBase](#)

void [setSdiSw12](#) (enum [SDI_SW12](#) s)

void	setClientError (int libsdiprotocol_result)
▶ Protected Attributes inherited from	SdiCmd
void *	dataIn
	Command input buffer collecting TLV data items. More...
void *	dataOut
	Command response TLV data items. More...
▶ Protected Attributes inherited from	SdiBase
unsigned short	sw12
int	additionalResultValue
SDICLIENT_ERROR	clientErr

Detailed Description

Interface for SDI Card Detection Interface, command class 23.

SDI server offers three forms of processing: Blocking, callback and polling mode. For all forms be aware that contactless card acception might generate so called EMV query call backs (Candidate List and Application Selected) that give PPSE and Final Select response discretionary data to an application to control the candidate list or contactless L2 kernel used. These call backs invoke the application's call back function given by [EMV_CTLS_Init_Framework\(\)](#). On processing these call backs the application is allowed to gather transaction information from the SDI system or change parameters e.g. by [SDI_CTLS_GetCandidateData\(\)](#) or [SDI_fetchTxnTags\(\)](#). Apart from these blocking call backs there are some others like light and sound control that do not need an answer (one-way call back). See [startSelection\(\)](#).

Since returned data is pretty the same this class is also used for the MSR Read command (21-01). See [startMsrRead\(\)](#).

Member Enumeration Documentation

? [DetectionMode](#)

enum [DetectionMode](#)

The card detection mode given to [setDetectionMode\(\)](#) controls which SDI card detection approach is used when [startSelection\(\)](#) is invoked

Enumerator

DETECTION_MODE_BLOCKING	using 23-01 command with SDI_Send. Before use SDI_SetDataAvailableCallback() to install notification function. After notification, call receiveTechnology() .
-------------------------	---

DETECTION_MODE_POLLING using 23-03 command in polling mode: Poll for technology selection result using [pollTechnology\(\)](#). This mode allows [addTechnology\(\)](#) and [removeTechnology\(\)](#) during wait for card.

DETECTION_MODE_CALLBACK using 23-03 command in callback mode: This mode allows [addTechnology\(\)](#) and [removeTechnology\(\)](#) during wait for card. Before use [setCallback\(\)](#) to install notification function().

Constructor & Destructor Documentation

[?](#) CardDetection()

[CardDetection](#) ()

[?](#) ~CardDetection()

[~CardDetection](#) ()

Member Function Documentation

[?](#) addTechnology()

```
enum libsdi::SDI\_SW12 addTechnology ( unsigned char technology,  
                                     const std::vector< unsigned char > & opts  
                                     )
```

Add a technology to a running asynchronous Card Detection (23-06)

Parameters

[in] *technology* technology to be added

[in] *opts* options for `cts_StartSelection` as described in TECSEL ADK

Returns

SW12 returned by the SDI Server

[?](#) cardDetectedCallback()

```
void cardDetectedCallback ( unsigned char * dataIn,
```

```
        unsigned short  sizeIn
    )
```

Internal Card Detected Callback handler function.

Parameters

[in] *dataIn* callback data received from SDI Server

[in] *sizeIn* callback data length

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

```
bool cardReadAtEpp ( )
```

Check at which device the card has been read (DFA150).

This function will deliver valid results once the card detection response has been received.

Returns

true for card has been read at EPP, false otherwise

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

```
virtual void clear ( ) virtual
```

Clean up object status

Buffers, status variables and callback settings are brought back to initial state. Object is afterwards in the same state as if just constructed.

Reimplemented from [SdiCmd](#).

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

```
std::string getCardholderName ( )
```

Access card holder name from track 1 (5F20)

Returns

Name/Christian name according ISO 7810 (ans ..26)

[?_getData\(\)](#) [1/3]

```
int getData ( std::vector< unsigned char > & data )
```

Access cts_WaitSelection result data (complete FFA102 content)

Parameters

[out] data destination buffer

existing data is erased

Returns

length of output, -1 if not present

[?_getData\(\)](#) [2/3]

```
int getData ( unsigned          CTS_DATA_TAG,  
              int               n,  
              std::vector< unsigned char > & data  
              )
```

Access cts_WaitSelection result data CTS_DATA_TAG_CARD_INFO.

Parameters

[in] CTS_DATA_TAG tag to be extracted

[in] *n* number of card in case multiple cards detected, count starts with 1

[out] data destination buffer

Data is appended to existing content

Returns

length of appended data, -1 if not present

[?_getData\(\)](#) [3/3]

```
int getData ( unsigned          CTS_DATA_TAG,  
              std::vector< unsigned char > & data  
              )
```

Access cts_WaitSelection result data CTS_DATA_TAG_CARD_INFO.

Parameters

[in] CTS_DATA_TAG tag to be extracted

[out] data destination buffer

Data is appended to existing content

Returns

length of appended data, -1 if not present

? getPan()

std::string getPan ()

Access PAN from mag. stripe or manual entry (DF1A)

Returns

an ..19

? getPluginResponseData() [1/3]

bool getPluginResponseData (std::vector< [PluginResult](#) > & *results*)

Access plugin response data (FFA107, DFA101, DFA10C, DFA10D)

Parameters

[out] results plugin ids, response codes and response data

Returns

true if any plugin result has been found

? getPluginResponseData() [2/3]

int getPluginResponseData (std::vector< unsigned char > & *data*)

Access plugin response data (DFA101) in case one plugin has been triggered.

Parameters

[out] data plugin response data (Note: Data is appended to existing content)

Returns

length of appended data, -1 if not present

? getPluginResponseData() [3/3]

```
int getPluginResponseData ( unsigned          index,  
                           int32_t &         pluginId,  
                           int32_t &         pluginResponseCode,  
                           std::vector< unsigned char > & data  
                           )
```

Access plugin response data (FFA107, DFA101, DFA10C, DFA10D)

Parameters

[in] index	for iterating the result data, starting with 0
[out] pluginId	return value of a plugin's <code>moduleID</code> function also used as INS byte of SDI 26-xx command to invoke a plugin, 0 if not available
[out] pluginResponseCode	plugin's <code>processTrigger</code> function return value, SDI_SW12_NONE if not available
[out] data	plugin response data (Note: Data is appended to existing content)

Returns

length of appended data, -1 if not present = end of iteration

? getServiceCode()

```
std::string getServiceCode ( )
```

Access service code from track 3 (5F30)

Returns

n 3

? getString()

```
std::string getString ( unsigned CTS_DATA_TAG )
```

Access cts_WaitSelection result data (FFA102) CTS_DATA_TAG_VAS_DATA: JSON string returned by NFC_VAS_Activate or NFC_VAS_Decrypt (DFDB24)

Parameters

[in] CTS_DATA_TAG item to be read

Returns

JSON string from NFC ADK

[?](#) **getTecselData()**

int getTecselData (std::vector< unsigned char > & *data*)

Access cts_WaitSelection result data in case CTS_DATA_TLV is not set ([SDI](#) tag DFA154) This can include either the EMV-ADK CTLS framework response code (result of continue offline) or the contactless card type (result of smart reset).

Parameters

[out] data destination buffer

Data is appended to existing content

Returns

length of appended data, -1 if not present

[?](#) **getTrack1()**

std::string getTrack1 ()

Access mag. stripe track 1 (DFA040)

Returns

ans ..79

[?](#) **getTrack2()**

std::string getTrack2 ()

Access track 2 from mag. stripe (DF49)

Returns

ans ..40

[? getTrack2Bin\(\)](#)

```
int getTrack2Bin ( std::vector< unsigned char > & data )
```

Access obfuscated track 2 data (DF1B) Note: Data is appended to existing content.

Returns

length of appended data ..20, -1 if not present

[? getValue\(\) \[1/2\]](#)

```
unsigned char getValue ( unsigned      CTS_DATA_TAG,  
                        int           n,  
                        unsigned char defaultValue  
                        )
```

Access cts_WaitSelection result data tags of 1 byte length.

Parameters

[in] CTS_DATA_TAG tag to be extracted

[in] n number of card in case multiple cards detected, count starts with 1

[in] defaultValue default value to be returned for the case that requested data is not found

Returns

requested value or defaultValue if tag not present

[? getValue\(\) \[2/2\]](#)

```
unsigned char getValue ( unsigned      CTS_DATA_TAG,  
                        unsigned char defaultValue  
                        )
```

Access cts_WaitSelection result data tags of 1 byte length (FFA102) CTS_DATA_TAG_NFC_RESULT
CTS_DATA_TAG_CARDS_TOTAL_COUNT CTS_DATA_TAG_CARDS_A CTS_DATA_TAG_CARDS_B

CTS_DATA_TAG_CARDS_F CTS_DATA_TAG_CARD_TYPE CTS_DATA_TAG_VAS_RESULT
CTS_DATA_TAG_EMV_RESULT CTS_DATA_TAG_VAS_DECRYPT_DATA_RESULT.

Parameters

[in] CTS_DATA_TAG tag to be extracted

[in] defaultValue default value to be returned for the case that requested data is not found

Returns

requested value or defaultValue if tag not present

? **msrGetCardSpecificToken()**

int msrGetCardSpecificToken (std::vector< unsigned char > & *token*)

Read card specific token (DFA014) optionally returned by MSR Read command

Parameters

[out] token buffer for token data **Note** data is appended

Returns

data length, -1 if not present

? **msrGetTrackStatus()**

int msrGetTrackStatus (int *track*)

Access MSR-ADK track status code (MSR_STATUS_xxx in msr/msr_common.h) (DFA034)

Only available if technology selection result is magnetic stripe read.

Parameters

[in] track track number 1..3

Returns

MSR_STATUS_xxx or -1 if not available

? **msrSetOptions()**

int msrSetOptions (const std::vector< unsigned char > & *opts*)

Set MSR options (21-03)

These options configure the behavior of magnetic card reader library, set by [MSR_SetOptions\(\)](#).

Serialization information can be found in [SDI](#) docu [MSR Set Options \(21-03\)](#)

Returns

0 if successful, -1 invalid parameter, SDI command was not sent -2 SDI server error, check sw12 for more details

[?](#) pollTechnology()

unsigned char pollTechnology ()

Poll for the asynchronous Card Detection status (23-04).

In case of problem, when '0' is returned use [getSW12\(\)](#) to retrieve the SDI Server SW1 SW2.

Returns

selected technology SDI_TEC_CHIP, SDI_TEC_MAGN, SDI_TEC_CTLs, SDI_TEC_MANU or 0 in case of in progress or failure

[?](#) receiveTechnology()

unsigned char receiveTechnology ()

Get the Technology Selection Result.

In case of problem, when '0' is returned use [getSdiSw12\(\)](#) to retrieve the SDI Server SW1 SW2.

Returns

selected technology SDI_TEC_CHIP, SDI_TEC_MAGN, SDI_TEC_CTLs, SDI_TEC_MANU or 0 in case of failure or MSR Read

Getters - Card Detection

[getPan\(\)](#) (DF1A)

[getTrack2\(\)](#) (DF49)

[getTrack2Bin\(\)](#) (DF1B)

[getCardholderName\(\)](#) (5F20)

[getServiceCode\(\)](#) (5F30)

[msrGetTrackStatus\(\)](#) (DFA034)

[msrGetTrack1\(\)](#) (DFA040)

[getPluginResponseData\(\)](#) (DFA101)

[cardReadAtEpp\(\)](#) (DFA150)

[getValue\(\)](#) (FFA102)

[getData\(\)](#) (FFA102)

[getString\(\)](#) (FF102)

[getSdiSw12\(\)](#) inherited from [SdiBase](#)

[getClientError\(\)](#) inherited from [SdiBase](#)

Getters - MSR Read

[getPan\(\)](#) (DF1A)

[getTrack2\(\)](#) (DF49)

[getTrack2Bin\(\)](#) (DF1B)

[getCardholderName\(\)](#) (5F20)

[getServiceCode\(\)](#) (5F30)

[msrGetTrackStatus\(\)](#) (DFA034)

[msrGetCardSpecificToken\(\)](#) (DFA014)

[msrGetTrack1\(\)](#) (DFA040)

[getPluginResponseData\(\)](#) (DFA101)

[getSdiSw12\(\)](#) inherited from [SdiBase](#)

[getClientError\(\)](#) inherited from [SdiBase](#)

? removeTechnology()

enum [libsdi::SDI_SW12](#) removeTechnology (unsigned char *technology*)

Remove a technology from a running asynchronous Card Detection (23-07)

Parameters

[in] technology technology to be removed

Returns

SW12 returned by the SDI Server

? setAlternativeInputFormat()

void setAlternativeInputFormat (const char * *f*)

Configure alternate input format string for CVV value.

Parameters

[in] *f* format string

? setCallback()

```
void setCallback ( void(*) (unsigned char technology, void *context) cb,  
                  void * ctx  
                  )
```

Set Callback function to be called in [DETECTION_MODE_CALLBACK](#) when Card Detection ends.

Parameters

[in] *cb* callback function pointer

[in] *ctx* callback context to be provided back

? setCancelButton()

void setCancelButton (bool *enable*)

Method for activation of cancel button.

Parameters

[in] enable flag

? setCardEntryValueDeactivation()

void setCardEntryValueDeactivation (unsigned char *b*)

Configure which input is skipped at manual card data entry.

Parameters

[in] *b* bit field

? setCardRemovalTimeout()

void setCardRemovalTimeout (unsigned *timeout_ms*)

For hybrid card readers set timeout for mag. stripe data available after start of card pull-out (DFA153).

This setter modifies the [startSelection\(\)](#) (23-01, 23-03) and [startMsrRead\(\)](#) (21-01) commands. It is forwarded to technology selection [CTS_OPTION_TAG_UX_MSR_TIMEOUT](#).

Parameters

[in] *timeout_ms* timeout in milliseconds ..65535

? setDetectionMode()

void setDetectionMode (const enum [DetectionMode](#) *mode*)

Set Card Detection mode.

Parameters

[in] *mode* Card Detection mode

? setTecConfig()

void setTecConfig (const std::vector< unsigned char > & *opts*)

Set TECSEL ADK configuration (DFA133)

Parameters

[in] *opts* [cts_SetOptions\(\)](#) parameter as described by TECSEL ADK

? setTecStartOptions()

void setTecStartOptions (const std::vector< unsigned char > & *opts*)

Set options for [cts_StartSelection\(\)](#) send by [startSelection\(\)](#) (DFA12A)

Parameters

[in] opts cts_StartSelection options of up to 16 bytes length

? startMsrRead()

int startMsrRead (unsigned *timeout_sec*)

Send MSR read request (21-01)

This command is like [startSelection\(\)](#) in blocking mode, use it with [SDI_SetDataAvailableCallback\(\)](#). Once response is available use [receiveTechnology\(\)](#) or [receive\(\)](#) to receive the response and then call the appropriate getters for reading the result data.

Setters

[setCardRemovalTimeout\(\)](#) (DFA153)

Parameters

[in] timeout_sec timeout in seconds (P2 & DFA01B)

Returns

result of [SDI_Send\(\)](#)

? startSelection()

```
int startSelection ( unsigned char supportedTechnologies,  
                    unsigned      seconds  
                    )
```

Start Technology Selection (23-01) or (23-03) according to the previously set detection mode using [setDetectionMode\(\)](#). Default mode is the blocking mode.

- In Blocking mode, this function is to be used with [SDI_SetDataAvailableCallback\(\)](#). Once response is available use [receiveTechnology\(\)](#) to get the output.
- In Callback mode this function is to be used with the callback set using [setCallback\(\)](#). Once response is available the [CardDetection](#) object will be automatically filled with the data received from SDI Server and the callback function will be called.

- In Polling mode this function is to be used with the [pollTechnology\(\)](#) to get the status and the result of the card detection.

Other functions controlling technology selection apart from EMV CTLS configuration are [setTecConfig\(\)](#) and [msrSetOptions\(\)](#).

Setters

[setCancelButton\(\)](#) (DFA109)

[setTecStartOptions\(\)](#) (DFA12A)

[setCardEntryValueDeactivation\(\)](#) (DFA017)

[setAlternativeInputFormat\(\)](#) (DFA01A)

[setTecConfig\(\)](#) (DFA133)

[setCardRemovalTimeout\(\)](#) (DFA153)

Parameters

[in] supportedTechnologies technologies to be detected (DFA108)

[in] seconds timeout in seconds [1 .. 1280, 65536] values above one hour trigger the maximal value (P2 & DFA01B)

Returns

return value from [SDI_Send\(\)](#)

? stopSelection()

enum [libsdi::SDI_SW12](#) stopSelection ()

Abort Technology Selection via SDI Server's Sys Abort (20-02) when in blocking mode, otherwise send Stop Card Detection (23-05).

Returns

SW12 returned by the SDI Server

You need to call [receiveTechnology\(\)](#) hereafter to eat the abort result from the IP stack and hereby stick to the rules described in the documentation of the [startSelection\(\)](#) function, i.e. if the Data Available callback has been configured then you need to wait for it before calling [receiveTechnology\(\)](#). If the technology selection has been successfully stopped then [receiveTechnology\(\)](#) will return 0 and SW1SW2 is 6500. If a transaction is already in progress or a card has been detected prior to the [stopSelection\(\)](#) call, the [receiveTechnology\(\)](#) will return the detected card technology.

Field Documentation

? SDI_TEC_CHIP

const static unsigned char SDI_TEC_CHIP = 1 static

? SDI_TEC_CTL5

const static unsigned char SDI_TEC_CTL5 = 4 static

? SDI_TEC_MAGN

const static unsigned char SDI_TEC_MAGN = 2 static

? SDI_TEC_MANU

const static unsigned char SDI_TEC_MANU = 8 static

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

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