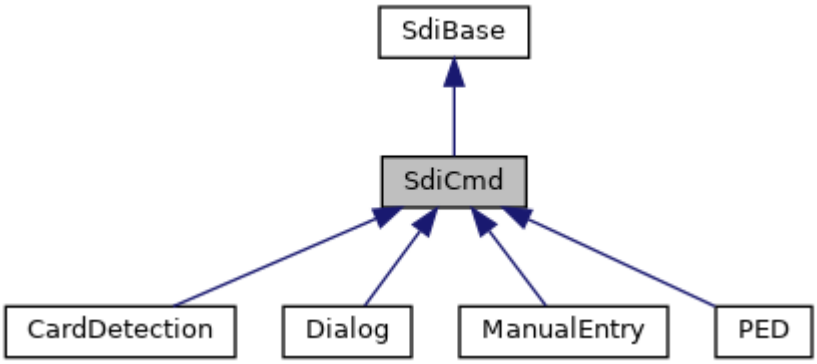
SdiCmd Class Reference

Composition for TLV based SDI commands.
More...

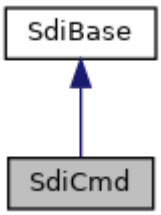
```
#include <sdi_if.h>
```

Inheritance diagram for SdiCmd:



[legend]

Collaboration diagram for SdiCmd:



[legend]

Public Member Functions

	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...

Protected Attributes

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

Additional Inherited Members

▸ Protected Member Functions inherited from SdiBase

void	setSdiSw12 (enum SDI_SW12 s)
void	setClientError (int libsdiprotocol_result)

Detailed Description

Composition for TLV based SDI commands.

General instruction:

- Create instance of SdiCmd
- Populate input data by calling the set functions

- call sendReceive() to execute the SDI command
- use the getter functions to access response data

Path Format

The path specification is a terminated ASCII string composed of the hexadecimal encoded tag names with '/' separating sub tag levels. **Note: The leading "F0/" is implicitly added and must not be used.**To support TLV data not enclosed in 'F0' template the path has to start with '/'. Additionally a counter is possible separated by ',' E.g "E0,3/90,2" gives the second primitive '90' in the third 'E0' container. Note: Count starts at 1: "90,1" is the same as "90,0" and "90".

Example showing use in subclass PED:

```
void PED::setAmount(const unsigned char* amount){    set("9F02", amount, 6
);}
```

Constructor & Destructor Documentation

? SdiCmd()

SdiCmd ( )

? ~SdiCmd()

virtual ~SdiCmd ( ) virtual

Member Function Documentation

? clear() [1/2]

virtual void clear ( ) virtual

Remove all SDI command command input data.

Reimplemented in CardDetection.

? clear() [2/2]

virtual void clear (const char * *path*) virtual

Remove single input data stored in dataIn.

Parameters

[in] path TLV tag name path (without enclosing F0), see Path Format

? clearResults()

virtual void clearResults ( ) virtual

Remove all SDI response data.

? get() [1/6]

bool get (const char * *path*,
int & *value*
)

Get integer value from SDI response

Parameters

[in] path TLV path (without enclosing F0), see Path Format
[out] value target buffer

Returns

true if value was found in SDI response

? get() [2/6]

bool get (const char * *path*,
std::string & *value*
)

Read text from SDI response

Parameters

[in] path TLV path (without enclosing F0), see Path Format
[out] value string reference for the result data

Returns

true for success

? get() [3/6]

```
bool get ( const char *                    path,
           std::vector< unsigned char > &  buffer
         )
```

Read data from SDI response

Parameters

[in] path TLV path (without enclosing F0), see Path Format

[out] buffer target vector

Returns

true if data was found in SDI response

? get() [4/6]

```
bool get ( const char *  path,
           uint32_t &    value
         )
```

Get unsigned integer value from SDI response

Parameters

[in] path TLV path (without enclosing F0), see Path Format

[out] value target buffer

Returns

true if value was found in SDI response

? get() [5/6]

```
bool get ( const char *         path,
           unsigned char &  value   inline
         )
```

Read single byte from SDI response

Parameters

[in] path TLV path (without enclosing F0), see Path Format

[out] value target buffer, remains unchanged if tag not found

Returns

true if SDI contains requested data with length 1

? get() [6/6]

```
int get ( const char *        path,
          unsigned char *  buffer,
          unsigned         bufferSize
        )
```

Read data from SDI response

Parameters

[in] path TLV path (without enclosing F0), see Path Format

[out] buffer target buffer

[in] bufferSize size of target buffer

Returns

length of data found in SDI response, -1 if absent

? getString()

```
std::string getString ( const char *  path )
```

Read text from SDI response

Parameters

[in] path TLV path (without enclosing F0), see Path Format

Returns

Text from SDI response, empty string if absent

? importResults() [1/2]

```
void importResults ( const SdiCmd &  intermediate )
```

Transfer status word, client error and response data from an intermediate command while keeping command input data.

Parameters

[in] intermediate reference to intermediate command that received SDI response

? importResults() [2/2]

```
void importResults ( const unsigned char *  sw12,
                     const unsigned char *  tlvData,
                     unsigned               tlvSize
                   )
```

Inject result data from a call back.

Parameters

[in] sw12 pointer to SDI status word
[in] tlvData pointer to SDI TLV data (should start with F0)
[in] tlvSize length of tlvData in bytes

? receive()

```
enum SDI_SW12 receive ( unsigned  maxResponseSize = 2048 )
```

Receive SDI response after request has been send with send() and data availability has been signaled.

Parameters

[in] maxResponseSize optional required buffer size for entire SDI response

Returns

SDI status word (the two leading bytes of the SDI response as word)

? send()

```
int send ( unsigned char  cla,
           unsigned char  ins,
           unsigned char  p1 = 0,
           unsigned char  p2 = 0
         )
```

Like sendReceive() but without waiting for response.

If send() succeeded (return code 0) wait for signal by data available callback. Thereafter call receive() to get SDI response.

Parameters

[in] cla SDI command class

[in] ins SDI command instruction for given class

[in] p1 optional SDI command parameter 1

[in] p2 optional SDI command parameter 2

Returns

0 for success, SDI_PROTOCOL_ERR_xxx on failure

? sendReceive()

```
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.

The SDI command TLV data part is created from the public dataIn attribute, which has been populated with this classes setter functions before function invocation. The SDI response TLV data part will be imported to public dataOut attribute.

Parameters

[in] cla	SDI command class
[in] ins	SDI command instruction for given class
[in] p1	optional SDI command parameter 1
[in] p2	optional SDI command parameter 2
[in] maxResponseSize	optional required buffer size for entire SDI response

Returns

SDI status word (the two leading bytes of the SDI response as word)

? set() [1/6]

```
void set ( const char *          path,
           const std::string &   value
         )
```

Set character data object in SDI command.

Set text for TLV path (without enclosing F0). If a tags already exists it's data will be overwritten.

Parameters

path TLV tag name path, see Path Format

value alpha-numeric data

? set() [2/6]

```
void set ( const char *                          path,
           const std::vector< unsigned char > &  data
         )
```

Set byte array data object in SDI command.

Set binary data for TLV path (without enclosing F0). If a tags already exists it's data will be overwritten.

Parameters

path TLV tag name path, see Path Format

data byte vector

? set() [3/6]

```
void set ( const char *           path,
           const unsigned char *  data,
           unsigned               dataLen
         )
```

Set byte array data object in SDI command.

Set binary data for TLV path (without enclosing F0). If a tags already exists it's data will be overwritten.

Parameters

path TLV tag name path, see Path Format

data byte array

dataLen byte count of data length

? set() [4/6]

```
void set ( const char *  path,
           int           value,
           unsigned      fixedLength = 0
         )
```

Set integer data object SDI command.

Set integer for TLV path (without enclosing F0). If a tags already exists it's data will be overwritten. The data length after serialisation is 1-4 bytes as needed to encode the value unless otherwise specified by parameter fixedLength.

Parameters

path	TLV tag name path, see Path Format
value	signed integer value
fixedLength	optional, might be required for some parameters

? set() [5/6]

```
void set ( const char *  path,
           uint32_t      value,
           unsigned      fixedLength = 0
         )
```

Set unsigned integer data object SDI command.

Set integer for TLV path (without enclosing F0). If a tags already exists it's data will be overwritten. The data length after serialisation is 1-4 bytes as needed to encode the value unless otherwise specified by parameter fixedLength.

Parameters

path	TLV tag name path, see Path Format
value	signed unsigned integer value
fixedLength	optional, might be required for some parameters

? set() [6/6]

void set (const char * *path*,
unsigned char *byteValue* inline
)

Set single byte value in SDI command.

Set binary data for TLV path (without enclosing F0). If a tags already exists it's data will be overwritten.

Parameters
path TLV tag name path, see Path Format
byteValue single byte

? setCommandDestination()

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.

Parameters
[in] epp true if command shall be set to EPP
[in] force if true do not omit command destination in case of countertop

Field Documentation

? dataIn

void* dataIn protected

Command input buffer collecting TLV data items.

? dataOut

void* dataOut protected

Command response TLV data items.

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

- sdiclient/client/sdi_if.h