

# Selection Check Librarys Reference Manual

## 0.7

Generated by Doxygen 1.5.1

Wed Jan 23 16:21:20 2008



# Contents

<b>1</b>	<b>Selection Check Librarys</b>	<b>1</b>
<b>2</b>	<b>Selection Check Librarys Data Structure Index</b>	<b>3</b>
2.1	Selection Check Librarys Data Structures . . . . .	3
<b>3</b>	<b>Selection Check Librarys File Index</b>	<b>5</b>
3.1	Selection Check Librarys File List . . . . .	5
<b>4</b>	<b>Selection Check Librarys Data Structure Documentation</b>	<b>7</b>
4.1	MD5Context Struct Reference . . . . .	7
4.2	SCdb_cfg Struct Reference . . . . .	8
4.3	SCr_errno Struct Reference . . . . .	9
4.4	SCs_data Struct Reference . . . . .	10
<b>5</b>	<b>Selection Check Librarys File Documentation</b>	<b>11</b>
5.1	analis/scdata_a.h File Reference . . . . .	11
5.2	manage/scdb.h File Reference . . . . .	13
5.3	manage/scmanage_common.h File Reference . . . . .	18
5.4	manage/scmanager_msg.h File Reference . . . . .	19
5.5	manage/scraport_msg.h File Reference . . . . .	20
5.6	manage/scwl2db_msg.h File Reference . . . . .	21
5.7	md5.h File Reference . . . . .	22
5.8	sccommon.h File Reference . . . . .	23
5.9	scdata.h File Reference . . . . .	24
5.10	work/schard.h File Reference . . . . .	29
5.11	work_test/schard.h File Reference . . . . .	31
5.12	work/scrunner_msg.h File Reference . . . . .	33
5.13	work_test/scrunner_msg.h File Reference . . . . .	34



# Chapter 1

## Selection Check Librarys

This are the core libraries used by programs in the Selection Check system. It is compiled in to three shared libraries:

- libscwork - compiled from sources in `./sc/work/`; is used by the `work_grp` programs,
- libscmanage - compiled from sources in `./sc/manage/`; is used by the `manage_grp` programs,
- libscanalis - compiled from sources in `./sc/analys/`; is used by programs that analyse data, mostly in the `manage_grp` and optionally in the `work_grp`.

Source files placed directly in `./sc/` are used by all this libraries.

This documentation is generated in to a HTML and PDF version. For viewing this documentation on your PC I recommend the HTML version.



# Chapter 2

## Selection Check Librarys Data Structure Index

### 2.1 Selection Check Librarys Data Structures

Here are the data structures with brief descriptions:

MD5Context	7
SCdb_cfg	8
SCr_erno	9
SCs_data	10





# Chapter 3

## Selection Check Librarys File Index

### 3.1 Selection Check Librarys File List

Here is a list of all files with brief descriptions:

<a href="#">md5.h</a>	22
<a href="#">sccommon.h</a>	23
<a href="#">scdata.h</a>	24
<a href="#">nalis/scdata_a.h</a>	11
<a href="#">manage/scdb.h</a>	13
<a href="#">manage/scmanage_common.h</a>	18
<a href="#">manage/scmanager_msg.h</a>	19
<a href="#">manage/scraport_msg.h</a>	20
<a href="#">manage/scwl2db_msg.h</a>	21
<a href="#">work/schard.h</a>	29
<a href="#">work/scranner_msg.h</a>	33
<a href="#">work_test/schard.h</a>	31
<a href="#">work_test/scranner_msg.h</a>	34



## Chapter 4

# Selection Check Librarys Data Structure Documentation

### 4.1 MD5Context Struct Reference

```
#include <md5.h>
```

#### Data Fields

- [uint32 buf](#) [4]
- [uint32 bits](#) [2]
- unsigned char [in](#) [64]

#### 4.1.1 Field Documentation

4.1.1.1 [uint32 MD5Context::buf](#)[4]

4.1.1.2 [uint32 MD5Context::bits](#)[2]

4.1.1.3 unsigned char [MD5Context::in](#)[64]

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

- [md5.h](#)

## 4.2 SCdb\_cfg Struct Reference

```
#include <scdb.h>
```

### Data Fields

- dictionary \* [d](#)
- char \* [host](#)
- char \* [user](#)
- char \* [passwd](#)
- char \* [db\\_name](#)

### 4.2.1 Detailed Description

Parameters needed to connect to the database taken from an ini file.

### 4.2.2 Field Documentation

#### 4.2.2.1 dictionary\* [SCdb\\_cfg::d](#)

dictionary from the ini file

#### 4.2.2.2 char\* [SCdb\\_cfg::host](#)

host adres or name

#### 4.2.2.3 char \* [SCdb\\_cfg::user](#)

user name

#### 4.2.2.4 char \* [SCdb\\_cfg::passwd](#)

users password

#### 4.2.2.5 char \* [SCdb\\_cfg::db\\_name](#)

database name

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

- [manage/scdb.h](#)

## 4.3 SCr\_errno Struct Reference

```
#include <scdata.h>
```

### Data Fields

- `time_t` [time](#)
- `int` [type](#)
- `int` [code](#)

### 4.3.1 Detailed Description

Error record used in errorlog file

### 4.3.2 Field Documentation

#### 4.3.2.1 `time_t` [SCr\\_errno::time](#)

date and time when error occurred

#### 4.3.2.2 `int` [SCr\\_errno::type](#)

error type

#### 4.3.2.3 `int` [SCr\\_errno::code](#)

error code

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

- [scdata.h](#)

## 4.4 SCs\_data Struct Reference

```
#include <scdata.h>
```

### Data Fields

- tms [time](#)
- unsigned int [code](#)
- unsigned int [input](#)

### 4.4.1 Detailed Description

Data record from SCslave

### 4.4.2 Field Documentation

#### 4.4.2.1 struct tms [SCs\\_data::time](#)

time in which this record was generated

#### 4.4.2.2 unsigned int [SCs\\_data::code](#)

SCslave code

#### 4.4.2.3 unsigned int [SCs\\_data::input](#)

SCslave registers state

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

- [scdata.h](#)

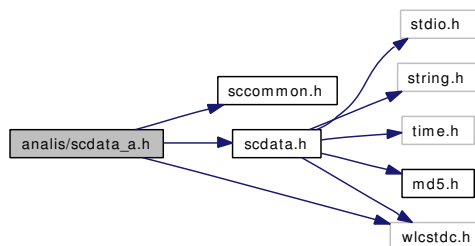
## Chapter 5

# Selection Check Librarys File Documentation

### 5.1 analis/scdata\_a.h File Reference

```
#include "sccommon.h"  
#include "scdata.h"  
#include "wlcstdc.h"
```

Include dependency graph for scdata\_a.h:



#### Functions

- int `SCs_analis_init()`
- int `SCs_data_analis` (const struct `SCs_data` \*data)
- void `SCs_analis_reset()`

#### Variables

- int `SC_min_change_time_sec`
- int `SC_score_num`

#### 5.1.1 Detailed Description

Functions and structures to analyse data from gathered from SCslave's.

## 5.1.2 Function Documentation

### 5.1.2.1 `int SCs_analis_init ()`

Initializes the analyse engine. This function does nothing for now and always returns 0.

**Returns:**

0 if successful

### 5.1.2.2 `void SCs_analis_reset ()`

Resets the analyse engine. I has to be done if you want to start an analyse another sequence of SCslave data records

### 5.1.2.3 `int SCs_data_analis (const struct SCs\_data * data)`

Analyses a data record from SCslave and returns its score. Remember to pass SCslave data records in the same order as they ware gathered by the SCrunner program.

**Parameters:**

*data* data record from SCslave

**Returns:**

analyse score which can be 0..SC\_score\_numb

## 5.1.3 Variable Documentation

### 5.1.3.1 `int SC\_min\_change\_time\_sec`

Minimum time that has to pass between SCslave data records from one SCslave. If there is a sequence of data records that don't have smaller time gaps then only the first one counts. Default valuse is 3 s.

### 5.1.3.2 `int SC\_score\_numb`

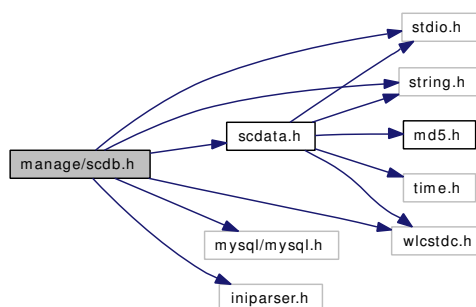
The value that SCs\_data\_analis can return is 0..SC\_score\_numb



## 5.2 manage/scdb.h File Reference

```
#include <stdio.h>
#include <mysql/mysql.h>
#include <string.h>
#include "wlcstdc.h"
#include "iniparser.h"
#include "scdata.h"
```

Include dependency graph for scdb.h:



### Data Structures

- struct [SCdb\\_cfg](#)

### Defines

- #define [SCDB\\_RES](#) MYSQL\_RES
- #define [SCDB\\_ROW](#) MYSQL\_ROW
- #define [SCdb\\_fetch\\_row](#) mysql\_fetch\_row
- #define [SCdb\\_free\\_result](#) mysql\_free\_result

### Typedefs

- typedef unsigned long [ulong](#)

### Functions

- [SCdb\\_cfg \\*](#) [make\\_SCdb\\_cfg](#) (const char \*ini\_path)
- int [SCdb\\_init](#) (const struct [SCdb\\_cfg](#) \*cfg)
- void [SCdb\\_free](#) ()
- int [SCdb\\_reset\\_connection](#) (const struct [SCdb\\_cfg](#) \*cfg)
- [SCDB\\_RES \\*](#) [SCdb\\_store\\_result](#) ()
- int [SCdb\\_query](#) (const char \*query)

### stored\_proc SCdb stored procedures

*Functions operating on SCdb stored procedures This is a set of stored procedures predefined in SCdb. See the "Selection Check Developers Manual" for a description of there parameters and about the results they give. All of these functions return 0 if the query was successful or -1 if otherwise.*

- int [SCdb\\_insert\\_wl](#) (const char \*date)
- int [SCdb\\_insert\\_SCs\\_data](#) (int over, const char \*date, const struct [SCs\\_data](#) \*data)
- int [SCdb\\_get\\_wls](#) (int year, int month)
- int [SCdb\\_get\\_planed\\_shift](#) (int year, int month)
- int [SCdb\\_get\\_workers](#) ()
- int [SCdb\\_insert\\_worker](#) (const char \*first\_name, const char \*last\_name)
- int [SCdb\\_delete\\_worker](#) (const char \*first\_name, const char \*last\_name)
- int [SCdb\\_delete\\_alldata](#) (const char \*date)
- int [SCdb\\_delete\\_shift](#) (const char \*date)
- int [SCdb\\_insert\\_shift](#) (const char \*date, const char \*first\_name, const char \*last\_name, int station, int shift)
- int [SCdb\\_get\\_shift](#) (const char \*date)
- int [SCdb\\_get\\_work](#) (const char \*first\_name, const char \*last\_name, const char \*date)
- int [SCdb\\_get\\_input](#) (const char \*date, const char \*tfrom, const char \*tto, int ws)

## 5.2.1 Detailed Description

Abstraction layer over the SCdb database.

## 5.2.2 Define Documentation

### 5.2.2.1 #define SCdb\_fetch\_row mysql\_fetch\_row

### 5.2.2.2 #define SCdb\_free\_result mysql\_free\_result

### 5.2.2.3 #define SCDB\_RES MYSQL\_RES

Stores the resoultts of a query

### 5.2.2.4 #define SCDB\_ROW MYSQL\_ROW

Stores a row from the results

## 5.2.3 Typedef Documentation

### 5.2.3.1 typedef unsigned long [ulong](#)

## 5.2.4 Function Documentation

### 5.2.4.1 struct [SCdb\\_cfg](#)\* make\_SCdb\_cfg (const char \* *ini\_path*)

Allocates a [SCdb\\_cfg](#) structure with entries from an ini file.

#### Parameters:

*ini\_path* path of the ini file

**Returns:**

a pointer to the allocated [SCdb\\_cfg](#) structure if successful; NULL if couldn't open the file or if the file didn't have the required entries

**5.2.4.2 int SCdb\_delete\_alldata (const char \* *date*)**

Deletes worker. See "Selection Check Developer Manual for more information.

**5.2.4.3 int SCdb\_delete\_shift (const char \* *date*)**

Deletes a shift. See "Selection Check Developer Manual for more information.

**5.2.4.4 int SCdb\_delete\_worker (const char \* *first\_name*, const char \* *last\_name*)**

Deletes worker. See "Selection Check Developer Manual for more information.

**5.2.4.5 void SCdb\_free ()**

Frees the database connection.

**5.2.4.6 int SCdb\_get\_input (const char \* *date*, const char \* *tfrom*, const char \* *tto*, int *ws*)**

Gets inputs from a SCslave generated in the specified time. See "Selection Check Developer Manual for more information.

**5.2.4.7 int SCdb\_get\_planed\_shift (int *year*, int *month*)**

Gets planed shift dates. See "Selection Check Developer Manual for more information.

**5.2.4.8 int SCdb\_get\_shift (const char \* *date*)**

Gets a shift. See "Selection Check Developer Manual for more information.

**5.2.4.9 int SCdb\_get\_wls (int *year*, int *month*)**

Gets worklog dates. See "Selection Check Developer Manual for more information.

**5.2.4.10 int SCdb\_get\_work (const char \* *first\_name*, const char \* *last\_name*, const char \* *date*)**

Gets work done by one worker in day 'date'. See "Selection Check Developer Manual for more information.

**5.2.4.11 int SCdb\_get\_workers ()**

Gets worker names. See "Selection Check Developer Manual for more information.

**5.2.4.12 int SCdb\_init (const struct [SCdb\\_cfg](#) \* *cfg*)**

Initializes a database connection with parameters from *cfg*.

**Parameters:**

*cfg* database parameters

**Returns:**

0 if successful else -1

**5.2.4.13 int SCdb\_insert\_SCs\_data (int *over*, const char \* *date*, const struct [SCs\\_data](#) \* *data*)**

Inserts SCslaves data record. See "Selection Check Developer Manual for more information.

**5.2.4.14 int SCdb\_insert\_shift (const char \* *date*, const char \* *first\_name*, const char \* *last\_name*, int *station*, int *shift*)**

Inserts a shift. See "Selection Check Developer Manual for more information.

**5.2.4.15 int SCdb\_insert\_wl (const char \* *date*)**

Inserts worklog date. See "Selection Check Developer Manual for more information.

**5.2.4.16 int SCdb\_insert\_worker (const char \* *first\_name*, const char \* *last\_name*)**

Inserts worker. See "Selection Check Developer Manual for more information.

**5.2.4.17 int SCdb\_query (const char \* *query*)**

Execute query from *query*.

**Parameters:**

*query* a string containing the query

**Returns:**

0 if successful else -1

**5.2.4.18 int SCdb\_reset\_connection (const struct [SCdb\\_cfg](#) \* *cfg*)**

Closes the connection and establishes a new one with the parameters from *cfg*.

**Parameters:**

*cfg* new connection parameters

**Returns:**

0 if successful else -1

**5.2.4.19 SCDB\_RES\* SCdb\_store\_result ()**

Returns the results from last query

## 5.3 manage/scmanage\_common.h File Reference

### Defines

- `#define SCM_SEPARATOR "—"`

### Enumerations

- `enum manage_common_exit_codes {  
    exit_err_ini = 1, exit_err_db = 2, exit_err_argc = 3, exit_err_cmd = 4,  
    exit_err_data = 5 }`

#### 5.3.1 Detailed Description

Common data in the manage\_grp.

#### 5.3.2 Define Documentation

##### 5.3.2.1 `#define SCM_SEPARATOR "—"`

Used in output do separate diffrent sequences of data

#### 5.3.3 Enumeration Type Documentation

##### 5.3.3.1 `enum manage_common_exit_codes`

Common exit codes in programs from the manage\_grp

##### Enumerator:

*exit\_err\_ini*   ini file error  
*exit\_err\_db*   database error  
*exit\_err\_argc*   wrong number of arguments  
*exit\_err\_cmd*   wrong command  
*exit\_err\_data*   wrong or corrupted data

## 5.4 manage/scmanager\_msg.h File Reference

### Defines

- #define [SCM\\_CMD\\_GET\\_WLS](#) "get\_wls"
- #define [SCM\\_CMD\\_INSERT\\_WORKER](#) "insert\_worker"
- #define [SCM\\_CMD\\_GET\\_WORKERS](#) "get\_workers"
- #define [SCM\\_CMD\\_DELETE\\_WORKER](#) "delete\_worker"
- #define [SCM\\_CMD\\_INSERT\\_SHIFT](#) "insert\_shift"
- #define [SCM\\_CMD\\_GET\\_SHIFT](#) "get\_shift"
- #define [SCM\\_CMD\\_DELETE\\_ALLDATA](#) "delete\_alldata"

### 5.4.1 Detailed Description

Commands useful for writing a fornt-end for SCmanager.

All of the commands are documented in the "Selection Check Developers Documentation".

### 5.4.2 Define Documentation

**5.4.2.1** #define [SCM\\_CMD\\_DELETE\\_ALLDATA](#) "delete\_alldata"

**5.4.2.2** #define [SCM\\_CMD\\_DELETE\\_WORKER](#) "delete\_worker"

**5.4.2.3** #define [SCM\\_CMD\\_GET\\_SHIFT](#) "get\_shift"

**5.4.2.4** #define [SCM\\_CMD\\_GET\\_WLS](#) "get\_wls"

**5.4.2.5** #define [SCM\\_CMD\\_GET\\_WORKERS](#) "get\_workers"

**5.4.2.6** #define [SCM\\_CMD\\_INSERT\\_SHIFT](#) "insert\_shift"

**5.4.2.7** #define [SCM\\_CMD\\_INSERT\\_WORKER](#) "insert\_worker"

## 5.5 manage/scraport\_msg.h File Reference

### Defines

- #define [SCM\\_CMD\\_GEN\\_DAY\\_RAPORT](#) "gen\_day\_raport"
- #define [SCM\\_CMD\\_GEN\\_MONTH\\_RAPORT](#) "gen\_month\_raport"
- #define [SCM\\_CMD\\_GEN\\_SHORT\\_RAPORT](#) "gen\_short\_raport"
- #define [SCM\\_FORMAT\\_TXT](#) "txt"
- #define [SCM\\_FORMAT\\_HTML](#) "html"

### 5.5.1 Detailed Description

Commands and arguments useful for writing a front-end for SCraport.

All of them are documented in the "Selection Check Developers Documentation".

### 5.5.2 Define Documentation

**5.5.2.1** #define [SCM\\_CMD\\_GEN\\_DAY\\_RAPORT](#) "gen\_day\_raport"

**5.5.2.2** #define [SCM\\_CMD\\_GEN\\_MONTH\\_RAPORT](#) "gen\_month\_raport"

**5.5.2.3** #define [SCM\\_CMD\\_GEN\\_SHORT\\_RAPORT](#) "gen\_short\_raport"

**5.5.2.4** #define [SCM\\_FORMAT\\_HTML](#) "html"

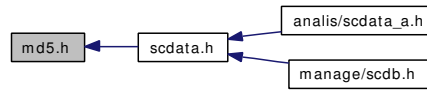
**5.5.2.5** #define [SCM\\_FORMAT\\_TXT](#) "txt"



## 5.6 manage/scwl2db\_msg.h File Reference

## 5.7 md5.h File Reference

This graph shows which files directly or indirectly include this file:



### Data Structures

- struct [MD5Context](#)

### Typedefs

- typedef unsigned long [uint32](#)
- typedef [MD5Context](#) [MD5\\_CTX](#)

### Functions

- void [MD5Init](#) (struct [MD5Context](#) \*context)
- void [MD5Update](#) (struct [MD5Context](#) \*context, unsigned char const \*buf, unsigned len)
- void [MD5Final](#) (unsigned char digest[16], struct [MD5Context](#) \*context)
- void [MD5Transform](#) ([uint32](#) buf[4], [uint32](#) const in[16])

#### 5.7.1 Typedef Documentation

5.7.1.1 typedef struct [MD5Context](#) [MD5\\_CTX](#)

5.7.1.2 typedef unsigned long [uint32](#)

#### 5.7.2 Function Documentation

5.7.2.1 void [MD5Final](#) (unsigned char *digest*[16], struct [MD5Context](#) \* *context*)

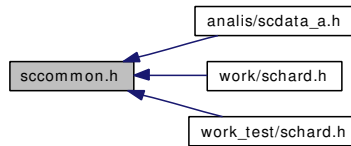
5.7.2.2 void [MD5Init](#) (struct [MD5Context](#) \* *context*)

5.7.2.3 void [MD5Transform](#) ([uint32](#) *buf*[4], [uint32](#) const *in*[16])

5.7.2.4 void [MD5Update](#) (struct [MD5Context](#) \* *context*, unsigned char const \* *buf*, unsigned *len*)

## 5.8 sccommon.h File Reference

This graph shows which files directly or indirectly include this file:



### Defines

- `#define` [SCS\\_NUMB\\_MAX](#) 16
- `#define` [INBITS](#) 8

#### 5.8.1 Detailed Description

General definitions used in the whole SC system.

#### 5.8.2 Define Documentation

##### 5.8.2.1 `#define` INBITS 8

Number of bits of data stored in SCslave modules

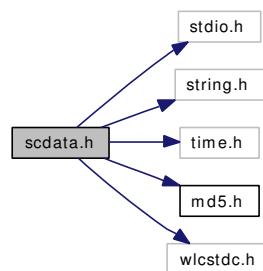
##### 5.8.2.2 `#define` SCS\_NUMB\_MAX 16

Maximal number of SCslave modules.

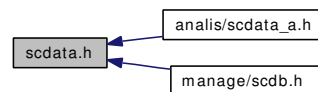
## 5.9 scdata.h File Reference

```
#include <stdio.h>
#include <string.h>
#include <time.h>
#include "md5.h"
#include "wlcstdc.h"
```

Include dependency graph for scdata.h:



This graph shows which files directly or indirectly include this file:



### Data Structures

- struct [SCs\\_data](#)
- struct [SCr\\_erno](#)

### Defines

- #define [SCR\\_MAX\\_NOCHECK](#) 5
- #define [TIMELEN](#) 9
- #define [DATELEN](#) 11
- #define [MD5LEN](#) 16

### Enumerations

- enum [SCr\\_error\\_type](#) {  
[SCr\\_err\\_startup](#) = 0, [SCr\\_err\\_noccheck](#) = 1, [SCr\\_err\\_file\\_format](#) = 2, [SCr\\_err\\_sys\\_disk](#) = 3,  
[SCr\\_err\\_transfer](#) = 4, [SCr\\_err\\_unlogged](#) = 5, [SCr\\_err\\_types](#) = 6 }
- enum [SCt\\_error\\_type](#) {  
[SCt\\_err\\_argc](#) = 1, [SCt\\_err\\_no\\_twisset](#) = 3, [SCt\\_err\\_no\\_owlset](#) = 5, [SCt\\_err\\_no\\_wlset](#) = 7,  
[SCt\\_err\\_no\\_terr](#) = 11, [SCt\\_err\\_pre\\_sh](#) = 13, [SCt\\_err\\_post\\_sh](#) = 15, [SCt\\_err\\_over](#) = 17 }

## Functions

- const char \* [SCr\\_err\\_to\\_str](#) (int type)
- void [scode\\_time](#) (char \*give, time\_t timer)
- void [scode\\_date](#) (char \*give, time\_t timer)
- void [SCr\\_err\\_fwrite](#) (FILE \*dst, const struct [SCr\\_erno](#) \*err)
- void [SCr\\_err\\_fread](#) (FILE \*src, struct [SCr\\_erno](#) \*err)
- int [SCs\\_data\\_fwrite](#) (FILE \*dst, const struct [SCs\\_data](#) \*data)
- int [SCs\\_data\\_fread](#) (FILE \*src, struct [SCs\\_data](#) \*data)
- int [md5file](#) (FILE \*fp, unsigned char \*digest)

## Variables

- const char \* [SC\\_wlset\\_name](#)
- const char \* [SC\\_worklogs\\_dir](#)
- const char \* [SC\\_errorlog\\_name](#)

### 5.9.1 Detailed Description

Functions and structures used to manipulate data in the whole SC system.

### 5.9.2 Define Documentation

#### 5.9.2.1 #define DATELEN 11

The length of a string with a date in ISO format

#### 5.9.2.2 #define MD5LEN 16

The lenght of a string with the MD5 checksum

#### 5.9.2.3 #define SCR\_MAX\_NOCHECK 5

The number of maximum unsuccessful communication tray's with SCslave module

#### 5.9.2.4 #define TIMELEN 9

The length of a string with time in format HH:MM:SS

### 5.9.3 Enumeration Type Documentation

#### 5.9.3.1 enum [SCr\\_error\\_type](#)

Error types codes returned and logged in errorlog file by the SCrunner program. See the "Selection Check Developers Manual" for more detail.

#### Enumerator:

*SCr\_err\_startup* initialization error

*SCr\_err\_nocheck* raised when the maximum number of unsuccessful communication tray's with SCslave module have occurred

*SCr\_err\_file\_format* wrong file format

*SCr\_err\_sys\_disk*

*SCr\_err\_transfer* transfer error

*SCr\_err\_unlogged* multiple unsuccessful data logs

*SCr\_err\_types* number of error types

### 5.9.3.2 enum [SCt\\_error\\_type](#)

Transfer type error codes

#### Enumerator:

*SCt\_err\_argc* wrong number of arguments

*SCt\_err\_no\_twlset* couldn't create a wlset file in the transfer directory

*SCt\_err\_no\_owlset* couldn't open the old wlset file

*SCt\_err\_no\_wlset* couldn't open the new wlset file

*SCt\_err\_no\_terr* couldn't create a terror file in the transfer directory

*SCt\_err\_pre\_sh* pre-trans.sh error

*SCt\_err\_post\_sh* post-trans.sh error

*SCt\_err\_over* tried to override an old transfer

## 5.9.4 Function Documentation

### 5.9.4.1 int md5file (FILE \* *fp*, unsigned char \* *digest*)

Generates a MD5 checksum for a file. This is a modification of a function from md5sum source.

#### Parameters:

*fp* file for witch the checksum will be generated

*digest* byte array in witch the checksum will be stored, must be at least MD5LEN big

#### Returns:

0 if successful else -1

### 5.9.4.2 void scode\_date (char \* *give*, time\_t *timer*)

Generates a string with the date from 'timer' in ISO format.

#### Parameters:

*give* generated string

*timer* source date

**5.9.4.3 void scode\_time (char \* *give*, time\_t *timer*)**

Generates a string with time 'timer' in format HH:MM:SS.

**Parameters:**

*give* generated string  
*timer* source time

**5.9.4.4 void SCr\_err\_fread (FILE \* *src*, struct SCr\_errno \* *err*)**

Reads an error from the error log file

**Parameters:**

*src* source file  
*err* the error

**5.9.4.5 void SCr\_err\_fwrite (FILE \* *dst*, const struct SCr\_errno \* *err*)**

Writes an error to the errorlog file

**Parameters:**

*dst* destination file  
*err* the error

**5.9.4.6 const char\* SCr\_err\_to\_str (int *type*)**

Converts code of an error type to its name.

**Parameters:**

*type* error type from SCr\_error\_type

**Returns:**

a pointer to a string with the name of the error type

**5.9.4.7 int SCs\_data\_fread (FILE \* *src*, struct SCs\_data \* *data*)**

Reads a SCslave data record from worklog file.

**Parameters:**

*src* source file  
*data* SCslave data record

**Returns:**

0 if successful else -1

**5.9.4.8 int SCs\_data\_fwrite (FILE \* *dst*, const struct SCs\_data \* *data*)**

Writes a data record from SCslave to a worklog file.

**Parameters:**

*dst* destination file

*data* data record from SCslave

**Returns:**

0 if successful else -1

**5.9.5 Variable Documentation****5.9.5.1 const char \* SC\_errorlog\_name**

the name of the error log file

**5.9.5.2 const char\* SC\_wlset\_name**

the name of a wlset type file

**5.9.5.3 const char \* SC\_worklogs\_dir**

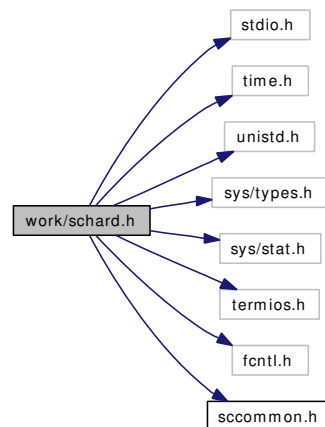
the name of the directory that holds worklog files



## 5.10 work/schard.h File Reference

```
#include <stdio.h>
#include <time.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <termios.h>
#include <fcntl.h>
#include "sccommon.h"
```

Include dependency graph for schard.h:



### Enumerations

- enum [SCs\\_returns](#) {  
[no\\_new\\_data](#) = -1, [com\\_error](#) = -2, [not\\_init](#) = -3, [no\\_new\\_data](#) = -1,  
[com\\_error](#) = -2, [not\\_init](#) = -3 }

### Functions

- int [SC\\_hard\\_init](#) ()
- int [SCs\\_init](#) (const char \*path, int mach\_code)
- int [SCs\\_data\\_cycle](#) (int mach\_code)
- void [SCs\\_off](#) (int mach\_code)

#### 5.10.1 Enumeration Type Documentation

##### 5.10.1.1 enum [SCs\\_returns](#)

Return values of function SCslave

**Enumerator:**

*no\_new\_data* no new data in module  
*com\_error* communication error  
*not\_init* module not initialized  
*no\_new\_data* no new data in module  
*com\_error* communication error  
*not\_init* module not initialized

## 5.10.2 Function Documentation

### 5.10.2.1 int SC\_hard\_init ()

Initializes global hardware mechanisms.

**Returns:**

0 if successful

### 5.10.2.2 int SCs\_data\_cycle (int *mach\_code*)

Gets data from a SCslave module.

**Parameters:**

*mach\_code* modules id

**Returns:**

if the value is below 0 then its one from SCs\_returns, else its the data from the SCslave module

### 5.10.2.3 int SCs\_init (const char \* *path*, int *mach\_code*)

Initializes a SCslave module.

**Parameters:**

*path* the path o the serial port  
*mach\_code* modules id

**Returns:**

0 if successful else -1

### 5.10.2.4 void SCs\_off (int *mach\_code*)

Releases the SCslave module.

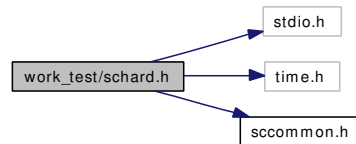
**Parameters:**

*mach\_code* modules id

## 5.11 work\_test/schard.h File Reference

```
#include <stdio.h>
#include <time.h>
#include "sccommon.h"
```

Include dependency graph for schard.h:



### Enumerations

- enum [SCs\\_returns](#) {  
[no\\_new\\_data](#) = -1, [com\\_error](#) = -2, [not\\_init](#) = -3, [no\\_new\\_data](#) = -1,  
[com\\_error](#) = -2, [not\\_init](#) = -3 }

### Functions

- int [SC\\_hard\\_init](#) ()
- int [SCs\\_init](#) (const char \*path, int mach\_code)
- int [SCs\\_data\\_cycle](#) (int mach\_code)
- void [SCs\\_off](#) (int mach\_code)

### 5.11.1 Enumeration Type Documentation

#### 5.11.1.1 enum [SCs\\_returns](#)

Return values of function SCslave

**Enumerator:**

*no\_new\_data* no new data in module  
*com\_error* communication error  
*not\_init* module not initialized  
*no\_new\_data* no new data in module  
*com\_error* communication error  
*not\_init* module not initialized

### 5.11.2 Function Documentation

#### 5.11.2.1 int [SC\\_hard\\_init](#) ()

Initializes global hardware mechanizems.

**Returns:**

0 if succesfull

**5.11.2.2 int SCs\_data\_cycle (int *mach\_code*)**

Gets data from a SCslave module.

**Parameters:**

*mach\_code* modules id

**Returns:**

if the value is below 0 then its one from SCs\_returns, else its the data from the SCslave module

**5.11.2.3 int SCs\_init (const char \* *path*, int *mach\_code*)**

Initializes a SCslave module.

**Parameters:**

*path* the path o the serial port

*mach\_code* modules id

**Returns:**

0 if succesfull else -1

**5.11.2.4 void SCs\_off (int *mach\_code*)**

Relases the SCslave module.

**Parameters:**

*mach\_code* modules id

## 5.12 work/scrunner\_msg.h File Reference

### Defines

- #define [SCR\\_MAX\\_LINE](#) 100
- #define [SCR\\_INIT\\_MSG](#) "SCrunner vr:%s pid:%u"
- #define [SCR\\_PROCLOG\\_START\\_MSG](#) "Proclog output:\n"
- #define [SCR\\_END\\_MSG](#) "SCrunner was ended\n"

### 5.12.1 Define Documentation

#### 5.12.1.1 #define SCR\_END\_MSG "SCrunner was ended\n"

Last line of output

#### 5.12.1.2 #define SCR\_INIT\_MSG "SCrunner vr:%s pid:%u"

First line of output

#### 5.12.1.3 #define SCR\_MAX\_LINE 100

Maximum output line

#### 5.12.1.4 #define SCR\_PROCLOG\_START\_MSG "Proclog output:\n"

First line of proclog output

## 5.13 work\_test/srunner\_msg.h File Reference

### Defines

- `#define SCR_MAX_LINE 100`
- `#define SCR_INIT_MSG "SCrunner vr:%s pid:%u"`
- `#define SCR_PROCLOG_START_MSG "Proclog output:\n"`
- `#define SCR_END_MSG "SCrunner zakonczyl prace\n"`

### 5.13.1 Define Documentation

**5.13.1.1** `#define SCR_END_MSG "SCrunner zakonczyl prace\n"`

**5.13.1.2** `#define SCR_INIT_MSG "SCrunner vr:%s pid:%u"`

**5.13.1.3** `#define SCR_MAX_LINE 100`

**5.13.1.4** `#define SCR_PROCLOG_START_MSG "Proclog output:\n"`

# Index

analisis/scdata\_a.h, [11](#)

bits  
    MD5Context, [7](#)

buf  
    MD5Context, [7](#)

code  
    SCr\_errno, [9](#)  
    SCs\_data, [10](#)

com\_error  
    schard.h, [30](#)  
    test/schard.h, [31](#)

d  
    SCdb\_cfg, [8](#)

DATELEN  
    scdata.h, [25](#)

db\_name  
    SCdb\_cfg, [8](#)

exit\_err\_argc  
    scmanage\_common.h, [18](#)

exit\_err\_cmd  
    scmanage\_common.h, [18](#)

exit\_err\_data  
    scmanage\_common.h, [18](#)

exit\_err\_db  
    scmanage\_common.h, [18](#)

exit\_err\_ini  
    scmanage\_common.h, [18](#)

host  
    SCdb\_cfg, [8](#)

in  
    MD5Context, [7](#)

INBITS  
    scommon.h, [23](#)

input  
    SCs\_data, [10](#)

make\_SCdb\_cfg  
    scdb.h, [14](#)

manage/scdb.h, [13](#)

manage/scmanager\_common.h, [18](#)

manage/scmanager\_msg.h, [19](#)

manage/scraport\_msg.h, [20](#)

manage/scw12db\_msg.h, [21](#)

manage\_common\_exit\_codes  
    scmanage\_common.h, [18](#)

md5.h, [22](#)  
    MD5\_CTX, [22](#)  
    MD5Final, [22](#)  
    MD5Init, [22](#)  
    MD5Transform, [22](#)  
    MD5Update, [22](#)  
    uint32, [22](#)

MD5\_CTX  
    md5.h, [22](#)

MD5Context, [7](#)  
    bits, [7](#)  
    buf, [7](#)  
    in, [7](#)

md5file  
    scdata.h, [26](#)

MD5Final  
    md5.h, [22](#)

MD5Init  
    md5.h, [22](#)

MD5LEN  
    scdata.h, [25](#)

MD5Transform  
    md5.h, [22](#)

MD5Update  
    md5.h, [22](#)

no\_new\_data  
    schard.h, [30](#)  
    test/schard.h, [31](#)

not\_init  
    schard.h, [30](#)  
    test/schard.h, [31](#)

passwd  
    SCdb\_cfg, [8](#)

SC\_errorlog\_name  
    scdata.h, [28](#)

SC\_hard\_init  
    schard.h, [30](#)

- test/schard.h, 31
- SC\_min\_change\_time\_sec
  - sdata\_a.h, 12
- SC\_score\_num
  - sdata\_a.h, 12
- SC\_wlset\_name
  - sdata.h, 28
- SC\_worklogs\_dir
  - sdata.h, 28
- scommon.h, 23
  - INBITS, 23
  - SCS\_NUMB\_MAX, 23
- sdata.h, 24
  - DATELEN, 25
  - md5file, 26
  - MD5LEN, 25
  - SC\_errorlog\_name, 28
  - SC\_wlset\_name, 28
  - SC\_worklogs\_dir, 28
  - scode\_date, 26
  - scode\_time, 26
  - SCr\_err\_file\_format, 26
  - SCr\_err\_fread, 27
  - SCr\_err\_fwrite, 27
  - SCr\_err\_nocheck, 25
  - SCr\_err\_startup, 25
  - SCr\_err\_sys\_disk, 26
  - SCr\_err\_to\_str, 27
  - SCr\_err\_transfer, 26
  - SCr\_err\_types, 26
  - SCr\_err\_unlogged, 26
  - SCr\_error\_type, 25
  - SCR\_MAX\_NOCHECK, 25
  - SCs\_data\_fread, 27
  - SCs\_data\_fwrite, 27
  - SCt\_err\_argc, 26
  - SCt\_err\_no\_owlset, 26
  - SCt\_err\_no\_terr, 26
  - SCt\_err\_no\_twiset, 26
  - SCt\_err\_no\_wlset, 26
  - SCt\_err\_over, 26
  - SCt\_err\_post\_sh, 26
  - SCt\_err\_pre\_sh, 26
  - SCt\_error\_type, 26
  - TIMELEN, 25
- sdata\_a.h
  - SC\_min\_change\_time\_sec, 12
  - SC\_score\_num, 12
  - SCs\_analis\_init, 12
  - SCs\_analis\_reset, 12
  - SCs\_data\_analis, 12
- scdb.h
  - make\_SCdb\_cfg, 14
  - SCdb\_delete\_alldata, 15
  - SCdb\_delete\_shift, 15
  - SCdb\_delete\_worker, 15
  - SCdb\_fetch\_row, 14
  - SCdb\_free, 15
  - SCdb\_free\_result, 14
  - SCdb\_get\_input, 15
  - SCdb\_get\_planed\_shift, 15
  - SCdb\_get\_shift, 15
  - SCdb\_get\_wls, 15
  - SCdb\_get\_work, 15
  - SCdb\_get\_workers, 15
  - SCdb\_init, 15
  - SCdb\_insert\_SCs\_data, 16
  - SCdb\_insert\_shift, 16
  - SCdb\_insert\_wl, 16
  - SCdb\_insert\_worker, 16
  - SCdb\_query, 16
  - SCDB\_RES, 14
  - SCdb\_reset\_connection, 16
  - SCDB\_ROW, 14
  - SCdb\_store\_result, 16
  - ulong, 14
- SCdb\_cfg, 8
  - d, 8
  - db\_name, 8
  - host, 8
  - passwd, 8
  - user, 8
- SCdb\_delete\_alldata
  - scdb.h, 15
- SCdb\_delete\_shift
  - scdb.h, 15
- SCdb\_delete\_worker
  - scdb.h, 15
- SCdb\_fetch\_row
  - scdb.h, 14
- SCdb\_free
  - scdb.h, 15
- SCdb\_free\_result
  - scdb.h, 14
- SCdb\_get\_input
  - scdb.h, 15
- SCdb\_get\_planed\_shift
  - scdb.h, 15
- SCdb\_get\_shift
  - scdb.h, 15
- SCdb\_get\_wls
  - scdb.h, 15
- SCdb\_get\_work
  - scdb.h, 15
- SCdb\_get\_workers
  - scdb.h, 15
- SCdb\_init
  - scdb.h, 15



- SCdb\_insert\_SCs\_data
  - scdb.h, 16
- SCdb\_insert\_shift
  - scdb.h, 16
- SCdb\_insert\_wl
  - scdb.h, 16
- SCdb\_insert\_worker
  - scdb.h, 16
- SCdb\_query
  - scdb.h, 16
- SCDB\_RES
  - scdb.h, 14
- SCdb\_reset\_connection
  - scdb.h, 16
- SCDB\_ROW
  - scdb.h, 14
- SCdb\_store\_result
  - scdb.h, 16
- schard.h
  - com\_error, 30
  - no\_new\_data, 30
  - not\_init, 30
  - SC\_hard\_init, 30
  - SCs\_data\_cycle, 30
  - SCs\_init, 30
  - SCs\_off, 30
  - SCs\_returns, 29
- SCM\_CMD\_DELETE\_ALLDATA
  - scmanager\_msg.h, 19
- SCM\_CMD\_DELETE\_WORKER
  - scmanager\_msg.h, 19
- SCM\_CMD\_GEN\_DAY\_RAPORT
  - scraport\_msg.h, 20
- SCM\_CMD\_GEN\_MONTH\_RAPORT
  - scraport\_msg.h, 20
- SCM\_CMD\_GEN\_SHORT\_RAPORT
  - scraport\_msg.h, 20
- SCM\_CMD\_GET\_SHIFT
  - scmanager\_msg.h, 19
- SCM\_CMD\_GET\_WLS
  - scmanager\_msg.h, 19
- SCM\_CMD\_GET\_WORKERS
  - scmanager\_msg.h, 19
- SCM\_CMD\_INSERT\_SHIFT
  - scmanager\_msg.h, 19
- SCM\_CMD\_INSERT\_WORKER
  - scmanager\_msg.h, 19
- SCM\_FORMAT\_HTML
  - scraport\_msg.h, 20
- SCM\_FORMAT\_TXT
  - scraport\_msg.h, 20
- SCM\_SEPARATOR
  - scmanage\_common.h, 18
- scmanage\_common.h
  - exit\_err\_argc, 18
  - exit\_err\_cmd, 18
  - exit\_err\_data, 18
  - exit\_err\_db, 18
  - exit\_err\_ini, 18
  - manage\_common\_exit\_codes, 18
  - SCM\_SEPARATOR, 18
- scmanager\_msg.h
  - SCM\_CMD\_DELETE\_ALLDATA, 19
  - SCM\_CMD\_DELETE\_WORKER, 19
  - SCM\_CMD\_GET\_SHIFT, 19
  - SCM\_CMD\_GET\_WLS, 19
  - SCM\_CMD\_GET\_WORKERS, 19
  - SCM\_CMD\_INSERT\_SHIFT, 19
  - SCM\_CMD\_INSERT\_WORKER, 19
- scode\_date
  - sdata.h, 26
- scode\_time
  - sdata.h, 26
- SCR\_END\_MSG
  - scranner\_msg.h, 33
  - test/scranner\_msg.h, 34
- SCr\_err\_file\_format
  - sdata.h, 26
- SCr\_err\_fread
  - sdata.h, 27
- SCr\_err\_fwrite
  - sdata.h, 27
- SCr\_err\_nocheck
  - sdata.h, 25
- SCr\_err\_startup
  - sdata.h, 25
- SCr\_err\_sys\_disk
  - sdata.h, 26
- SCr\_err\_to\_str
  - sdata.h, 27
- SCr\_err\_transfer
  - sdata.h, 26
- SCr\_err\_types
  - sdata.h, 26
- SCr\_err\_unlogged
  - sdata.h, 26
- SCr\_errno, 9
  - code, 9
  - time, 9
  - type, 9
- SCr\_error\_type
  - sdata.h, 25
- SCR\_INIT\_MSG
  - scranner\_msg.h, 33
  - test/scranner\_msg.h, 34
- SCR\_MAX\_LINE
  - scranner\_msg.h, 33

- test/scranner\_msg.h, 34
- SCR\_MAX\_NOCHECK
  - scdata.h, 25
- SCR\_PROCLOG\_START\_MSG
  - scranner\_msg.h, 33
  - test/scranner\_msg.h, 34
- scraport\_msg.h
  - SCM\_CMD\_GEN\_DAY\_RAPORT, 20
  - SCM\_CMD\_GEN\_MONTH\_RAPORT, 20
  - SCM\_CMD\_GEN\_SHORT\_RAPORT, 20
  - SCM\_FORMAT\_HTML, 20
  - SCM\_FORMAT\_TXT, 20
- scranner\_msg.h
  - SCR\_END\_MSG, 33
  - SCR\_INIT\_MSG, 33
  - SCR\_MAX\_LINE, 33
  - SCR\_PROCLOG\_START\_MSG, 33
- SCs\_analis\_init
  - scdata\_a.h, 12
- SCs\_analis\_reset
  - scdata\_a.h, 12
- SCs\_data, 10
  - code, 10
  - input, 10
  - time, 10
- SCs\_data\_analis
  - scdata\_a.h, 12
- SCs\_data\_cycle
  - schard.h, 30
  - test/schard.h, 32
- SCs\_data\_fread
  - scdata.h, 27
- SCs\_data\_fwrite
  - scdata.h, 27
- SCs\_init
  - schard.h, 30
  - test/schard.h, 32
- SCS\_NUMB\_MAX
  - sccommon.h, 23
- SCs\_off
  - schard.h, 30
  - test/schard.h, 32
- SCs\_returns
  - schard.h, 29
  - test/schard.h, 31
- SCt\_err\_argc
  - scdata.h, 26
- SCt\_err\_no\_owlset
  - scdata.h, 26
- SCt\_err\_no\_terr
  - scdata.h, 26
- SCt\_err\_no\_twlset
  - scdata.h, 26
- SCt\_err\_no\_wlset
  - scdata.h, 26
- SCt\_err\_over
  - scdata.h, 26
- SCt\_err\_post\_sh
  - scdata.h, 26
- SCt\_err\_pre\_sh
  - scdata.h, 26
- SCt\_error\_type
  - scdata.h, 26
- test/schard.h
  - com\_error, 31
  - no\_new\_data, 31
  - not\_init, 31
  - SC\_hard\_init, 31
  - SCs\_data\_cycle, 32
  - SCs\_init, 32
  - SCs\_off, 32
  - SCs\_returns, 31
- test/scranner\_msg.h
  - SCR\_END\_MSG, 34
  - SCR\_INIT\_MSG, 34
  - SCR\_MAX\_LINE, 34
  - SCR\_PROCLOG\_START\_MSG, 34
- time
  - SCr\_errno, 9
  - SCs\_data, 10
- TIMELEN
  - scdata.h, 25
- type
  - SCr\_errno, 9
- uint32
  - md5.h, 22
- ulong
  - scdb.h, 14
- user
  - SCdb\_cfg, 8
- work/schard.h, 29
- work/scranner\_msg.h, 33
- work\_test/schard.h, 31
- work\_test/scranner\_msg.h, 34