

[1] [1 [Generated on Wed Aug 3 10:34:38 2005 for Programmers Guide for the Winswup-Library by Doxygen] [Generated on Wed Aug 3 10:34:38 2005 for Programmers Guide for the Winswup-Library by Doxygen

Programmers Guide for the Winswup-Library Reference Manual

Generated by Doxygen 1.3.3

Wed Aug 3 10:34:34 2005

Contents

1	The Programmers Guide for the WINSWUP-Library	1
1.1	Overview	1
1.2	Recommended Readings	1
1.3	The different DLLs and their responsibility	2
1.4	Modules and their responsibility	3
1.5	Hints about some details	5
2	Programmers Guide for the Winswup-Library Module Index	7
2.1	Programmers Guide for the Winswup-Library Modules	7
3	Programmers Guide for the Winswup-Library Data Structure Index	9
3.1	Programmers Guide for the Winswup-Library Data Structures	9
4	Programmers Guide for the Winswup-Library File Index	11
4.1	Programmers Guide for the Winswup-Library File List	11
5	Programmers Guide for the Winswup-Library Page Index	13
5.1	Programmers Guide for the Winswup-Library Related Pages	13
6	Programmers Guide for the Winswup-Library Module Documentation	15
6.1	WSWUPSER.DLL : Functions concerning serial lowest layer functionality.	15
6.2	WSWUPSER.DLL : Functions concerning block reception and sending.	20
6.3	WSWUPSER.DLL : Functionality for Debugging	26
6.4	WSWUPSER.DLL : Functionality for Error-Tracing	30
7	Programmers Guide for the Winswup-Library Data Structure Documentation	31
7.1	FileInfo Struct Reference	31
7.2	t_Block Struct Reference	34
7.3	t_ComConfig Struct Reference	35
7.4	t_enumErrorDescription Struct Reference	36

7.5	t_ErrorCodeDescription Struct Reference	37
7.6	t_ErrorField Struct Reference	38
7.7	t_ErrorInfo Struct Reference	39
7.8	t_ExtendedInfo Struct Reference	41
7.9	T_HashBinBlock Struct Reference	51
7.10	t_LangCheckInfo Struct Reference	52
7.11	t_TableEntry Struct Reference	53
7.12	tag_AllLibInfo Struct Reference	54
7.13	tag_Area Struct Reference	55
7.14	tag_BfbThreadInfo Struct Reference	56
7.15	tag_CompressInfo Struct Reference	57
7.16	tag_LocaterRecord Struct Reference	58
7.17	tag_PINstruct Struct Reference	60
7.18	tag_ThreadInfo Struct Reference	61
7.19	tag_UsbInfo Struct Reference	62
7.20	tagDebugFiles Struct Reference	63
7.21	tagMessageBlock Struct Reference	64
7.22	tagNewSplitInfo Struct Reference	65
8	Programmers Guide for the Winswup-Library File Documentation	67
8.1	boot_hex.h File Reference	67
8.2	dyna_dll.c File Reference	79
8.3	err_text.c File Reference	82
8.4	err_text.h File Reference	83
8.5	fihawrap.c File Reference	84
8.6	filehand.c File Reference	87
8.7	pc_mob.h File Reference	95
8.8	swupwork.c File Reference	105
8.9	wbfbtool.c File Reference	122
8.10	wmob_err.c File Reference	129
8.11	wseril1.c File Reference	130
8.12	wseril2.c File Reference	133
8.13	wseril3.c File Reference	139
8.14	wsw_seri.h File Reference	148
8.15	wsw_tool.c File Reference	160
8.16	wswdbgex.c File Reference	161
8.17	wswdbgin.c File Reference	162

8.18	wswerrex.c File Reference	164
8.19	wswerrin.c File Reference	166
8.20	wswupprog.dox File Reference	171
8.21	xbi_info.h File Reference	172
9	Programmers Guide for the Winswup-Library Page Documentation	177
9.1	Todo List	177

Chapter 1

The Programmers Guide for the WINSWUP-Library

1.1 Overview

WinSwup is the new program for performing software-updates for Siemens Mobilephones. It is currently developed for usage in Software-Development and Production, not for usage at traders or worst: at the enduser. To achieve a trader/enduser-proof level, there must be some modifications e.g.

- only a mini bfblibrary which exports only the necessary functions
- no dll-handling for different processors
- less information about the update-process
- less configuration possibilities
- better security :
- No usage of Bootstraploader (BOT-Files), this should be contained in the GSM-Mobile-SW, where we have better control about whats happening (no back-booting, checking of IMEI, ...). For this, the PC-Side should only connect to a living mobile via bfb-bus, and after some security-checks the mobile should be forced to reset (via bfb) and then brought into update-mode, where the Update-SW is beeing transmitted. In the moment winswup supports up to 4 parallel updates of the same software on different comports. This is especially for production purposes. To achieve a parallel processing of the updates each update has it's own thread.

1.2 Recommended Readings

Since Windows-programming is not as easy as normal 'c'-programming, I recommend the following books for understanding the program:

- Charles Petzold : Programming Windows (95/98) : the standard
- Jeffrey Richer : Windows Programming for Experts : some usefull details about memory/dll-programming

- Nancy Cluts : Programming the new user interface (or so) : some hints about new common controls etc.

For understanding the program itself you should read the following Siemens-internal papers:

- The SWUPAPI - Documentation (swupapi2.pdf)
- The HEX-BIN-Converter - Documentation (hex-bin-doc.pdf)
- The also via Doxygen generated Dokumentation for the user of this Library
- evt. the via Doxygen generated Dokumentation for the GUI of WINSWUP

1.3 The different DLLs and their responsibility

Currently the WSWUPLIB.DLL needs several other DLLs to work. The whole functionality is split into several DLLs to allow the dynamically loading of a Mobile-Processor-dependent DLL which contains :

- the Bootstrap-Loader for this Processor
- the Update-SW that runs on the Mobile and is responsible for the flash-processing Since version 0.96 of the whole winswup-program the differentiation is done via the project-name (not the product-name) found in the XBI-Header.

latex swupdlls.jpg

The following DLLs exist in the moment:

1.3.1 WSWUPSER.DLL

Contains all serial low-level stuff, the debugging-support and some error-routines. This DLL is linked via export-lib to the different processor-dependent DLLs and via export-lib to the Main-DLL: WSWUPLIB.DLL. All functions that are exported from WSWUPSER.DLL are described in the Interface-Header: [wsu_seri.h](#) To get a DLL all responsible modules must be compiled with the following define set: #define __WSWUPSERIDLL__ (Done in the IDE)

1.3.2 WSW_????.DLL

These are the "entwproj"-dependent DLLs and contain the Boot-Strap-Loader and the Update-SW. In the moment there are 4 of them:

- WSW_K45.DLL: for K45-Family-Projects
- WSW_P35.DLL: for P35-Family-Projects
- WSW_U35.DLL: for SL45-Projects
- WSW_SU35.DLL : for SL45-Plus-Projects, updated via Synch-Station (other BSL) All of these DLLs are loaded dynamically from WSWUPLIB.DLL, so there exists no export-library

1.3.3 WSWPLIBx.DLL

The DLL is available in different variants, see [WSWPLIBx.DLL](#). This is the main-DLL that contains all the "intelligence" necessary for a software-update. This DLL forms the interface for the WINSWUP-Programm or for the PV-Program that runs in the factory. Since in the factory no "real" windows-programming is possible, the interface of this DLL is done via callback-functions to the caller. The interface is described in detail in the Interface-Header : wswuplib.h. The DLL performs things like :

- reading the XBI-File
- loading dependent of the XBI-Header one of the WSW????-Update-DLLs
- loading dependent of the XBI-Header one of the BFB-DLLs
- performing a voltage-check (in normal-mode) and switching off the mobile
- performing the whole SW-Update (BSL/ Update-SW / Erasing / Transmitting new SW)
- performing a update-success via BFB

The DLL brings its export-library, so it can be "linked" to the Main-Program.

1.4 Modules and their responsibility

1.4.1 WSWUPSER.DLL

- [wseril1.c](#) : serial communication lowest layer (opening and configuring ports, reading and writing bytes or streams)
- [wseril2.c](#) : serial communication layer2, reading and writing bin-messages, writing bot-blocks , reading bytes with check for message-blocks
- [wsbdbgin.c](#) : debugging purposes for the software-update. Internal, working functions
- [wsverrin.c](#) : error-handling routines for the software-update. Internal, working functions

1.4.2 WSW_????.DLL

- [swup_eg.c](#) : the bootstrap-loader and update-sw for EGOLD and EGOLD+, forms a DLL. Since the difference between EGOLD/ EGOLD+ and EGOLD+ via Synch-Station is very low, all 3 different update-means are contained in this one module. But on compilation there must be set different defines, so the result are 3 different DLLs. Responsible defines : #define EGOLD_PLUS => result is a DLL for EGOLD+ #define SYNC.STATION => result is a DLL for EGOLD+ for update with a synch-station. This #define must be set together with EGOLD_PLUS ! All of the defines are already set in the relevant settings in the IDE !
- [*.jcj](#) : All the mobile-SW-parts, the Bootstrap-Loader (BOT-Files) and the Update-SW as C-Arrays, so it can be compiled into the PC-Project.

1.4.3 WSWPLIBx.DLL

Because of the effort to outsource the programming of a WINSWUP-GUI, we need a new interface which provides a data access of the bin data only from heap. So the programmers of the external firm have to open and read the complete XBI-file as it is to the heap, and the rest of the new interface is able to manage the update with the information of the heap location and the amount of data in heap (== file length).

The sources for wswuplib were splitted in a shared pool in the directory "shardsrc" and wswuplib specific directories wswuplibf(ile) and wswplibh(eap). Because the differences are only two functions, in a first step the same export header "fuko-header" in terms of mach-tools is used.

The old wswuplib is therefor renamed to wswplibf, the new library is named wswplibh. For a graphical overview please refer to "wswplib.jpg" in the doc-directory in Continuous.

Please be carefull: Never build an executable with dependencies to both DLLs!

The sources were moved to an own continuous project, in our next steps we should move some other sources to own projects or directories.

latex wswuplib.jpg

1. Shared sources between the File-Read and the Heap-Read-Library:

- **erase.c** : Contains all functions necessary for erasing the flash-parts in the mobile. This means it doesn't contain any mobile-SW, instead this means it contains the necessary control-functions.
- **wbfbtool.c** : Contains all relevant functions for communication with the mobile in normal-mode (voltage-check ...)
- **wmob_err.c** : functions for error-handling on mobile-side.
- **wseril3.c** : "High Layer" functions for serial-communications (all Message-Block-Sending)
- **wsw_tool.c** : usefull tools for winswup (conversion routines etc ...)
- **wswdbgex.c** : interface for external calling to the internal debug-functions contained in the wswupser.dll
- **wswdbgex.c** : interface for external calling to the internal error functions contained in the wswupser.dll
- **swupwork.c** : this is the "main" of the DLL, here is all real work done.

2. Sources for the File-Read-Library:

- **filehand.c** : Contains all functions for reading the XB-file and its header.
- **fiwrap.c** : Contains some wrapper-functions for **filehand.c** for the external interface that's used by factory/ WSWUP32 ...

3. Sources for the Heap-Read-Library:

- **heaphand.c** : Contains all functions for reading the XB-file and its header from heap.
- **heapwrap.c** : Contains some wrapper-functions for **heaphand.c** for the external interface that's used by factory/ WSWUP32 ...

1.4.4 WSWUP32.EXE

- **wswup32.c** : the main module, creating all the necessary windows
- **registry.c** : contains all accesses to registry for saving information over the runtime of winswup

- `whatsnew.c` : some news, presented on first start of new version or via menu (only usefull during beta-test, could be removed after first public release)
- `fileopen.c` : contains handling for the file-open-dialog-box
- `about.c` : contains the Dialog-Box procedure for the About-Box
- `config.c` : all Configure-Dialog-Box-Procedures (in the moment there are two configuration Dialogs, one concerning the whole winswup and one concerning the update over the comport (since there are up to four parallel updates)
- `statbar.c` : contains the statusbar-functions of WSWUP32
- `toolbar.c` : contains the toolbar-functions of WSWUP32

1.5 Hints about some details

1.5.1 Parallel updating

The Main-Window creates all the child-windows, each responsible for a software-update on a dedicated comport. The information about all updates are stored in a (shame) global array. The Main-Window stores on creation of each childwindow the number of the childwindow in the childwindows extra bytes (DLGWINDOWEXTRA). So the ChildWindow gains access to it's specific information via `GetWindowWord()`. The ChildWindows create on demand (via prssure of the start-button) a thread that is responsible for this update. Remember : All functions called from this thread must be either reentrant (no static variables etc) or must store their specific information in an array, so each instance of the function has it's own set of static variables !

1.5.2 Resources(To be Done ...)

All the resources (Dialog-Boxes, Strings etc) reside in a DLL. So it's very easy to add some new languages. For this the Resource-DLL have to follow this convention:

- the DLL containing the resources is called `wswl????.dll`
- the DLL exports an additional function `GiveResourceLanguageString` where the name of the language is given . With this convention WinSwup can scan it's directory for DLLs following the convention and build up a dynamical menu with all the names of the found resource-dlls To achieve this the DLLs are loaded dynamically.

Chapter 2

Programmers Guide for the Winswup-Library Module Index

2.1 Programmers Guide for the Winswup-Library Modules

Here is a list of all modules:

WSWUPSER.DLL : Functions concerning serial lowest layer functionality.	15
WSWUPSER.DLL : Functions concerning block reception and sending.	20
WSWUPSER.DLL : Functionality for Debugging	26
WSWUPSER.DLL : Functionality for Error-Tracing	30

Chapter 3

Programmers Guide for the Winswup-Library Data Structure Index

3.1 Programmers Guide for the Winswup-Library Data Structures

Here are the data structures with brief descriptions:

FileInfo (A structure which holds all informations of the WSWUP input File)	31
t_Block (A structure which holds information of one XBI Header block)	34
t_ComConfig (This describes one serial connection)	35
t_enumErrorDescription	36
t_ErrorCodeDescription (Structure containing the error-code and a description)	37
t_ErrorField (A struct containing all errors from the indicated update, bIndex is a pointer to the next free t.ErrorStruct)	38
t_ErrorInfo (A struct containing all info about an error)	39
t_ExtendedInfo (The "Master"-stuct, containing all info in the XBI-Header All arrays containing text (especially dates) are 1 Byte longer to allow the storage of the)	41
T_HashBinBlock (A structure used to save the hash entries or the signature like bin blocks) . . .	51
t_LangCheckInfo (Information about Adresses on Language-Group-Bootng)	52
t_TableEntry (For each Message-Identifier it is possible to define a default-Handler)	53
tag_AllLibInfo (A linked list containing the library-infos)	54
tag_Area (Definition of a listenelement of the Erase/Not-Erase/Generate/And-so-on - Table) . .	55
tag_BfbThreadInfo	56
tag_CompressInfo (A (complete useles) information about parameters on compressing)	57
tag_LocaterRecord (A new structure for "locating" Up to now only implemented in the HEX-BIN-Converter)	58
tag_PINstruct (A structure which holds PIN information)	60
tag_ThreadInfo (A structure which holds the information of Threads (COM, speed, Update Nr))	61
tag_UsbInfo (A struct containing all information needed for UsbUpdate)	62
tagDebugFiles	63
tagMessageBlock (A structure containing a message-block with Message-Identifier, Len and additional Data)	64
tagNewSplitInfo (Definition of an element of the New SPlit information (can be Tegic, Language, FileSystem,...) Implemented as a linked list)	65

Chapter 4

Programmers Guide for the Winswup-Library File Index

4.1 Programmers Guide for the Winswup-Library File List

Here is a list of all files with brief descriptions:

boot_hex.h (One of two Interface-Headers between HEX-Converter and SWUP-Program) . . .	67
dyna.dll.c (Responsible for the dynamical loading of the project-dependant DLLs, Part of the Main-DLL)	79
err_text.c	82
err_text.h	83
fihawrap.c (Contains a wrapper for the former file-handling-library, part of the Main-DLL) . . .	84
filehand.c (Contains all necessary routines to deal with XBI-Files and complete EXE-Files for traders, part of the MAIN-DLL)	87
pc_mob.h (Interface between PC-Part and Mobile-Part of SWUP, Part of the MAIN-DLL) . . .	95
swupwork.c (Contains the "working" routine for swup, here is done "everything", Part of the MAIN-DLL)	105
wbfbtool.c (Contains the bfb-access-functions , part of the MAIN-DLL)	122
wmob_err.c (Responsible for decoding errors sent from mobile, contained in the MAIN-DLL) .	129
wseril1.c (Serial Communication : Layer1 -functionality, contained in WSWUPSER-DLL) . . .	130
wseril2.c (Serial Communication : Layer2 -functionality, contained in WSWUPSER-DLL) . . .	133
wseril3.c (Layer-3-functionality for the serial communication, Part of the MAIN-DLL)	139
wsw_seri.h	148
wsw_tool.c (Some tools like GetWord and the stuff, part of the MAIN-DLL)	160
wsbdbgex.c (The external-debug-interfaces to the internal functions, part of the MAIN-DLL) .	161
wsbdbgin.c (Debugging-Functionality contained in WSWUPSER-DLL)	162
wswerrex.c (The external-error-interfaces to the internal functions, part of the MAIN-DLL) . .	164
wswerrin.c (Error-Logging-Functionality contained in WSWUPSER-DLL)	166
wswupprog.dox	171
xbi_info.h (Interface between the HEX-BIN-Converter and SWUP, part of the MAIN-DLL) . .	172

Chapter 5

Programmers Guide for the Winswup-Library Page Index

5.1 Programmers Guide for the Winswup-Library Related Pages

Here is a list of all related documentation pages:

 Todo List [177](#)

Chapter 6

Programmers Guide for the Winswup-Library Module Documentation

6.1 WSWUPSER.DLL : Functions concerning serial lowest layer functionality.

These functions deal with the physical layer of the transmission.

Data Structures

- struct [t.ComConfig](#)
This describes one serial connection.

Functions

- unsigned long [ConvertBaudRatesForAddiData](#) (t.SwupCom WhichCom, unsigned long ulBaudRate)
Converts Baudrates depending on the (via INI-File) selected card.
- BOOL SWUPSERILOWEXIMPORT [WSwupComOpen](#) (t.SwupCom ComPort, DWORD [dwBaudrate](#), [t.SwupParity](#) WhichParity)
Opens the given Com-Port with the given Baud-Rate and Parity.
- BOOL SWUPSERILOWEXIMPORT [WSwupComConfigure](#) (t.SwupCom ComPort, DWORD [dwBaudrate](#))
Changes the Baudrate of the Given ComPort.
- BOOL SWUPSERILOWEXIMPORT [WSwupComClose](#) (t.SwupCom ComPort)
Closes the given Com-Port.

- **BOOL SWUPSERILOWEXIMPORT WSwupComWriteByte** (t_SwupCom ComPort, BYTE b-Value)
Writes a single Byte to the given ComPort.
- **BOOL SWUPSERILOWEXIMPORT WSwupComWriteDataBlock** (t_SwupCom ComPort, BYTE *pbValue, DWORD len)
Writes a Data-Block to the given comports.
- **BOOL SWUPSERILOWEXIMPORT WSwupComReadByteTimeOut** (t_SwupCom ComPort, LPBYTE lpbByte, DWORD dwTimeOut)
Reads a Byte from the given serial port with timeout.
- **BOOL SWUPSERILOWEXIMPORT WSwupComReadByte** (t_SwupCom ComPort, LPBYTE lpbByte)
Reads a Byte from the given serial port.
- **BOOL SWUPSERILOWEXIMPORT WSwupSetIgnition** (t_SwupCom ComPort)
Sets the Ignition-Port on the serial port (DTR-Line).
- **BOOL SWUPSERILOWEXIMPORT WSwupClearIgnition** (t_SwupCom ComPort)
Clears the Ignition-Port on the serial port (DTR-Line).

6.1.1 Detailed Description

These functions deal with the physical layer of the transmission.

All functions described here are part of the WSWUPSER-DLL, the DLL containing the lower layer serial functions and some other stuff. The functions are responsible for opening comports, setting the speed, writing single bytes and this stuff.

6.1.2 Function Documentation

6.1.2.1 unsigned long ConvertBaudRatesForAddiData (t_SwupCom WhichCom, unsigned long ulBaudRate)

Converts Baudrates depending on the (via INI-File) selected card.

The driver for the ADDIDATA-Card and for the old FASTBOOT-Card is the same. From drivers point of view the differentiation between the two cards is done via different baudrates. (ex. 203000 Bd vs 199219 Bd) To hide this from the enduser, this function is provided, it contains a compatibility-layer, so that from users point of view the baudrates are independent of the serial card

Parameters:

WhichCom a t_SwupCom-enum, for which comport shall the conversion be done

ulBaudRate a long value containing the baudrate that shall be converted

Returns:

the converted baudrate for this ComPort

Read the information from the INI-File

And determine the card-type

Is the given string "fastboot" ?

Now, depending on the Baudrate and the Card-Type do the conversion

6.1.2.2 **BOOL SWUPSERILOWEXIMPORT WSwupClearIgnition (t_SwupCom ComPort)**

Clears the Ignition-Port on the serial port (DTR-Line).

Should be done after switch on the mobile

Parameters:

ComPort : the opened ComPort

Return values:

TRUE -> ok

FALSE -> Error

6.1.2.3 **BOOL SWUPSERILOWEXIMPORT WSwupComClose (t_SwupCom ComPort)**

Closes the given Com-Port.

Parameters:

ComPort : the opened ComPort

Return values:

TRUE -> ok

FALSE -> Error

6.1.2.4 **BOOL SWUPSERILOWEXIMPORT WSwupComConfigure (t_SwupCom ComPort, DWORD dwBaudrate)**

Changes the Baudrate of the Given ComPort.

Parameters:

ComPort : the opened ComPort

dwBaudrate : the new Baudrate

Return values:

TRUE -> ok

FALSE -> Error

6.1.2.5 **BOOL SWUPSERILOWEXIMPORT WSwupComOpen (t_SwupCom ComPort, DWORD dwBaudrate, t_SwupParity WhichParity)**

Opens the given Com-Port with the given Baud-Rate and Parity.

All other Parameters are fix-coded according to the swup-requirements. All further needed values are stored in this module in a global array of structs, so its very easy to retrieve needed settings via a field-access. Otherwise (HANDLE is passed from the user) sometimes a HANDLE-compare would have been necessary ...

Parameters:

ComPort : the Com to be opened

dwBaudrate : the Baudrate (a double-value)

WhichParity : which parity do we use

Return values:

TRUE -> ok

FALSE -> Error

6.1.2.6 **BOOL SWUPSERILOWEXIMPORT WSwupComReadByte (t_SwupCom ComPort, LPBYTE lpbByte)**

Reads a Byte from the given serial port.

Does not wait if no byte is available

Parameters:

ComPort : the opened ComPort

lpbByte : pointer to the expected byte

Return values:

TRUE -> ok

FALSE -> Error

6.1.2.7 **BOOL SWUPSERILOWEXIMPORT WSwupComReadByteTimeOut (t_SwupCom ComPort, LPBYTE lpbByte, DWORD dwTimeOut)**

Reads a Byte from the given serial port with timeout.

Waits the given time in milliseconds for a byte

Parameters:

ComPort : the opened ComPort

dwTimeOut : amount of time to wait max for the byte

lpbByte : pointer to the expected byte

Return values:

TRUE -> ok

FALSE -> Error

6.1.2.8 BOOL SWUPSERILOWEXIMPORT WSwupComWriteByte (t_SwupCom ComPort, BYTE bValue)

Writes a single Byte to the given ComPort.

Dont use this function for a stream, use WSwupComWriteDataBlock instead because of faster timing on stream-write

Parameters:

ComPort : the opened ComPort

bValue : the Byte to write to the serial port

Return values:

TRUE -> ok

FALSE -> Error

6.1.2.9 BOOL SWUPSERILOWEXIMPORT WSwupComWriteDataBlock (t_SwupCom ComPort, BYTE * pbValue, DWORD len)

Writes a Data-Block to the given comport.

No assumptions about data-block-format

Parameters:

ComPort : the opened ComPort

pbValue : pointer to the bytes to be written

len : amount of bytes to write

Return values:

TRUE -> ok

FALSE -> Error

6.1.2.10 BOOL SWUPSERILOWEXIMPORT WSwupSetIgnition (t_SwupCom ComPort)

Sets the Ignition-Port on the serial port (DTR-Line).

Used to switch on the mobile

Parameters:

ComPort : the opened ComPort

Return values:

TRUE -> ok

FALSE -> Error

6.2 WSWUPSER.DLL : Functions concerning block reception and sending.

These functions deal with Blocks, Bin-Blocks, CMD-Blocks and this stuff.

Data Structures

- struct [t_TableEntry](#)

For each Message-Identifier it is possible to define a default-Handler.

Defines

- #define [ACK_LEN](#) 1

Sends ACK/NAK in SIPC format.

Functions

- BOOL SWUPSERILOWEXIMPORT [WSwupSeriLowGiveVersion](#) (t_VersionInformation *pInfo)
Returns information about the Serial-Low-DLL.
- BOOL [InstallDefaultHandlerFunction](#) (unsigned char ucThisMi, [pfnDefaultHandler](#) ThisDefault-Handler)
Installs a defaulthandler-function for a specific Message-Identifier.
- void SWUPSERILOWEXIMPORT [WSwupSendBotBlock](#) (t_SwupCom ComPort, unsigned char *pucSendBuf, unsigned int unDatLen)
Sends a so called BOT-Block to the Mobile.
- void SWUPSERILOWEXIMPORT [WSwupSendSGoldBotBlock](#) (t_SwupCom ComPort, unsigned char *pucSendBuf, unsigned int unDatLen)
Sends a so called BOT-Block to the SGOLD-Mobile.
- [t_RecResult](#) SWUPSERILOWEXIMPORT [WSwupReceiveValueFromSerial](#) (t_SwupCom Com-Port, unsigned char *pucRecBuf, DWORD dwTimeOutMilli)
Read a single byte from the serial Port.
- BOOL SWUPSERILOWEXIMPORT [WSwupReceiveCmdBlock](#) (t_SwupCom ComPort, [t_MessageBlock](#) *pMessageBlock, unsigned char ucExpectedMi, DWORD dwTimeOutMilli)
Waites for a CMD-Block with the given Message-Identifier on the serial port.
- [t_RecResult](#) [ReceiveCmdBlockInBinMode](#) (t_SwupCom ComPort, [t_MessageBlock](#) *pMessage-Block, BOOL fReadFirstChar, DWORD dwTimeOut)
Receives a CMD-Block in BIN-Mode.
- BOOL SWUPSERILOWEXIMPORT [WSwupSendMessageBlock](#) (t_SwupCom ComPort, unsigned char ucMi, unsigned short unLen, unsigned char *pabBuffer)
Sends a CMD-Block (currently only in BIN-Mode) .

- void SWUPSERILOWEXIMPORT [WSwupSetCurrentReceiveMode](#) (t_SwupCom ComPort, [t_KnownReceiveModes](#) WhichMode)
Sets the current receive-mode .
- [t_KnownReceiveModes](#) SWUPSERILOWEXIMPORT [WSwupWSwupGetCurrentReceiveMode](#) (t_SwupCom ComPort)
Retrieves the current receive-mode .
- void SWUPSERILOWEXIMPORT [SendSimpleMessageInSipcMode](#) (WORD wUsbPort, unsigned char ucMi, unsigned char ucPayload)
Sends message in SIPC format without waiting for response.
- [t_RecResult](#) [ReceiveBlockInSipcMode](#) (WORD wUpdateNr, [t_MessageBlock](#) *pMessageBlock)
Receives a CMD-Block in SipcMode.

6.2.1 Detailed Description

These functions deal with Blocks, Bin-Blocks, CMD-Blocks and this stuff.

All functions described here are part of the WSWUPSER-DLL, the DLL containing the lower layer serial functions and some other stuff, like serial layer2-functions.

6.2.2 Define Documentation

6.2.2.1 #define ACK_LEN 1

Sends ACK/NAK in SIPC format.

Parameters:

wUsbPort : the UsbPort to receive the message-block, same as UpdateNr
ucValue : ACK or NAK

6.2.3 Function Documentation

6.2.3.1 BOOL InstallDefaultHandlerFunction (unsigned char ucThisMi, [pfnDefaultHandler](#) ThisDefaultHandler)

Installs a defaulthandler-function for a specific Message-Identifier.

For each Message-Identifier it is possible to define a default-Handler. This default-handler will be called if a Message-Block with this identifier is received and this message-block is not expected. Used for example for Mobile-Errors which are normally "unexpected"

Parameters:

ucThisMi : the Message-Identifier that will be handled
ThisDefaultHandler : functionpointer to the handlerfunction

Return values:

TRUE -> everything is fine, handler is installed
FALSE -> some error occurred

6.2.3.2 **t_RecResult** ReceiveBlockInSipcMode (WORD wUsbPort, t_MessageBlock * pMessageBlock)

Receives a CMD-Block in SipcMode.

The received messageblock will be returned in the via pointer given t_MessageBlock structure.

Parameters:

wUsbPort : the UsbPort to receive the message-block, same as UpdateNr

pMessageBlock : a pointer to a t_MessageBlock - struct, filled on success of the function

Return values:

TRUE -> Message-Block was received, struct is filled with received data

FALSE -> an error occurred

6.2.3.3 **t_RecResult** ReceiveCmdBlockInBinMode (t_SwupCom ComPort, t_MessageBlock * pMessageBlock, BOOL fExpectFirstFF, DWORD dwTimeoutMilli)

Receives a CMD-Block in BIN-Mode.

Depending on where it comes from it receives the first FF or not. Thats because a Message-Block can be received:

- by request of the user, the protocol expects a cmd-block
- on error, the protocol expects an ACK or NAK, but the Mobile sends the error

The received messageblock will be returned in the via pointer given t_MessageBlock structure.

Parameters:

ComPort : the comport to receive the message-block

pMessageBlock : a pointer to a t_MessageBlock - struct, filled on success of the function

fExpectFirstFF : does the routine have to wait for the first FF or not

dwTimeOutMilli : timeout to wait for the first byte of the message-block

Return values:

TRUE -> Message-Block was received, struct is filled with received data

FALSE -> an error occurred

6.2.3.4 **void** SWUPSERILOWEXIMPORT SendSimpleMessageInSipcMode (WORD wUsbPort, unsigned char ucMi, unsigned char ucPayload)

Sends message in SIPC format without waiting for response.

Parameters:

wUsbPort : the UsbPort to receive the message-block, same as UpdateNr

ucValue : ACK or NAK

6.2.3.5 **BOOL** SWUPSERILOWEXIMPORT WSwupReceiveCmdBlock (t_SwupCom *ComPort*, t_MessageBlock * *pMessageBlock*, unsigned char *ucExpectedMi*, DWORD *dwTimeOutMilli*)

Waites for a CMD-Block with the given Message-Identfier on the serial port.

If a Message-Block is received and it is the expected Message, the function fills the given t_MessageBlock - struct. If it is not the expected message the default-handler is executed. If a single char or nothing is received, the function will return FALSE.

Parameters:

ComPort : the comport to receive the message-block

pMessageBlock : a pointer to a t_MessageBlock - struct, filled on success of the function

ucExpectedMi : the Message-Identifier that is expected by the caller of the function

dwTimeOutMilli : timeout to wait for the first byte of the message-block

Return values:

TRUE -> Message-Block was received, struct is filled with received data

FALSE -> an error occurred

6.2.3.6 **t_RecResult** SWUPSERILOWEXIMPORT WSwupReceiveValueFromSerial (t_SwupCom *ComPort*, unsigned char * *pucRecBuf*, DWORD *dwTimeOutMilli*)

Read a single byte from the serial Port.

This function returns if the received value is in the array of allowed single chars, if it is not in the allowed chars it will continue to wait. If the character received is the start of a message-block it will process the message-block.

Parameters:

ComPort : the Comport for waiting for chars

pucRecBuf : pointer to a buffer for the received character

dwTimeOutMilli : timeout in milliseconds to wait for the byte

Returns:

t_RecResult -> one of the t_RecResult - values

6.2.3.7 **void** SWUPSERILOWEXIMPORT WSwupSendBotBlock (t_SwupCom *ComPort*, unsigned char * *pucSendBuf*, unsigned int *unDatLen*)

Sends a so called BOT-Block to the Mobile.

First len is send, then the data and at last a xor-checksum over all the data is send.

Parameters:

ComPort : the Comport for sending the block

pucSendBuf : pointer to the buffer containing the data

unDatLen : amount of data to be send

Returns:

void

6.2.3.8 **BOOL SWUPSERILOWEXIMPORT WSwupSendMessageBlock (t_SwupCom ComPort, unsigned char ucMi, unsigned short unLen, unsigned char * pabBuffer)**

Sends a CMD-Block (currently only in BIN-Mode) .

This routine forms a BIN-Block with the given Message-Identifier and the given date, calculates the checksum and sends it out via function WSwInternalSendDataBlockInBinFormat()

Parameters:

ComPort : the comport to send the message-block

ucMi : the Message-Identifier

unLen : the amount of data following in

pabBuffer : the buffer to the data to send

Return values:

TRUE -> Message-Block was sent

FALSE -> an error occurred

6.2.3.9 **void SWUPSERILOWEXIMPORT WSwupSendSGoldBotBlock (t_SwupCom ComPort, unsigned char * pucSendBuf, unsigned int unDatLen)**

Sends a so called BOT-Block to the SGOLD-Mobile.

Transmission starts with len-info: UINT16 and LSB first, followed by data and at last a xor-checksum over all the data sent.

Parameters:

ComPort : the Comport for sending the block

pucSendBuf : pointer to the buffer containing the data

unDatLen : amount of data to be send

Returns:

void

6.2.3.10 **BOOL SWUPSERILOWEXIMPORT WSwupSeriLowGiveVersion (t_VersionInformation * pInfo)**

Returns information about the Serial-Low-DLL.

Every DLL that is part of the "WINSWUP-Library" returns information about itself with a function like this, so that the user of the library can retrieve information about all concerned parts.

Parameters:

pInfo : a pointer to a t_VersionInformation-struct, that is filled from this function

Return values:

TRUE -> everything is fine, struct is filled with information

FALSE -> error, struct-size is not the expected size

6.2.3.11 void SWUPSERILOWEXIMPORT WSwupSetCurrentReceiveMode (t_SwupCom ComPort, [t_KnownReceiveModes WhichMode](#))

Sets the current receive-mode .

Part of this module/dll are prepared to switch between different send/reception-modes The fully supported mode is the so-called BIN-FORMAT with the (hopefully) wellknown Format : 3 Byte Address, 1 Byte Len, up to 64 Byte Data and 1 Byte Checksum. The other format will be (hopefully) the format of the future , the LEN-CHECK-FORMAT: 1 Byte Len, up to 0x7F Bytes Data and 1 Byte Checksum. Address-Information will be no longer in this future-format, this information will be transmitted in a CMD-Block (CMD-Blocks can be recognised in the MSB of the Len-Byte)

Parameters:

ComPort : the comport to set the receive/transmit-format

WhichMode : a [t_KnownReceiveModes](#) -enum-value which sets the format

Returns:

void

6.2.3.12 [t_KnownReceiveModes](#) SWUPSERILOWEXIMPORT WSwupWSwupGetCurrentReceiveMode (t_SwupCom ComPort)

Retrieves the current receive-mode .

For a more detailed description of the receivemodes see [WSwupSetCurrentReceiveMode\(\)](#)

Parameters:

ComPort : the comport to set the receive/transmit-format

WhichMode :

Returns:

a [t_KnownReceiveModes](#) -enum-value with the current receive-mode

6.3 WSWUPSER.DLL : Functionality for Debugging

Enables and Disables (internally) debugging and contains the debug-print.

Functions

- void [WSwupDebugString](#) (WORD wUpdateNr, t_DebugLevel ShowOnDebugLevel, WORD w-DebugGroup, const char *pszFormat,...)
Debugging to files and to standard-debug-port.
- void SWUPSERILOWEXIMPORT [WSwupInt_EnableOnlineDebugging](#) (t_DebugLevel Requested-DebugLevel, WORD wRequestedDebugGroups)
Enables Debugging.
- void SWUPSERILOWEXIMPORT [WSwupInt_DisableOnlineDebugging](#) (void)
Disables Debugging.
- void SWUPSERILOWEXIMPORT [WSwupInt_EnableDebuggingToFile](#) (t_DebugLevel Requested-DebugLevel, WORD wRequestedDebugGroups, char *pszFileNamePrefix)
Enables Debugging to File.
- void SWUPSERILOWEXIMPORT [WSwupInt_DisableDebuggingToFile](#) (void)
Disables Debugging to File.
- WORD SWUPSERILOWEXIMPORT [WSwupInt_GiveUpdateNrFromComPort](#) (t_SwupCom Com-Port)
Gives UpdateNumber in dependency with ComPort.
- void SWUPSERILOWEXIMPORT [WSwupInt_SetUpdateNrToComPort](#) (t_SwupCom ComPort, WORD wUpdateNr)
Sets UpdateNumber in dependency from ComPort.

Variables

- t_DebugLevel [CurrentOnlineDebugLevel](#) = enNoDebug
CurrentDebug-Level for Online-Debugging, initailaised to not debug.
- WORD [wEnabledOnlineDebugGroups](#)
the different groups that are enabled for online debugging
- t_DebugLevel [CurrentFileDebugLevel](#) = enNoDebug
CurrentDebug-Level for File-Debugging, initailaised to not debug.
- WORD [wEnabledFileDebugGroups](#)
the different groups that are enabled for file-debugging
- BOOL [fDebugToFile](#) = FALSE
Bool, No debug to file as default.

- **BOOL fDebugToWindow** = FALSE
Bool, No debug to window as default.

6.3.1 Detailed Description

Enables and Disables (internally) debugging and contains the debug-print.

All functions described here are part of the WSWUPSER-DLL, the DLL containing the lower layer serial functions and some other stuff, like debugging. There was need to split this functionality into the external and the internal part because the debugging-function is dependent on variables set by the external user. And debugging itself must be possible from all three parts of the library, the serial DLL, the project-dependent DLL and the main-dll.

6.3.2 Function Documentation

6.3.2.1 void WSwupDebugString (WORD wUpdateNr, t_DebugLevel ShowOnDebugLevel, WORD wDebugGroup, const char * pszFormat, ...)

Debugging to files and to standard-debug-port.

Depending on the internal state of the current debuglevel and the switched on debug-groups, this function acts like printf. The output is written into File (not yet implemented) as well as to the debug-output-”port” which can be watched in the debugging-version of the library with DebugWatch from www.sysinternal.com or which is printed in the debug-window of the IDE

Parameters:

wUpdateNr : Achtung !!! neu

ShowOnDebugLevel : If the current set Debuglevel is >= this value, it is output

wDebugGroup : the group this ”printf” belongs to. Output is done, if this group is enabled.

pszFormat : like printf : the format-string

... : like printf : the values to output

Returns:

void

6.3.2.2 void SWUPSERILOWEXIMPORT WSwupInt_DisableDebuggingToFile (void)

Disables Debugging to File.

Returns:

void

6.3.2.3 void SWUPSERILOWEXIMPORT WSwupInt_DisableOnlineDebugging (void)

Disables Debugging.

Returns:

void

6.3.2.4 void SWUPSERILOWEXIMPORT WSwupInt_EnableDebuggingToFile (t_DebugLevel RequestedDebugLevel, WORD wRequestedDebugGroups, char * pszFileNamePrefix)

Enables Debugging to File.

Sets the debuglevel and the groups for which output shall be done.

Parameters:

RequestedDebugLevel : the new debuglevel

wRequestedDebugGroups : the groups for which output shall be done.

pszFileNamePrefix : the filename of the file to be debugged to, the extension is generated automatically

Returns:

void

wer macht fclose ??? am Schluss automatisch ???

6.3.2.5 void SWUPSERILOWEXIMPORT WSwupInt_EnableOnlineDebugging (t_DebugLevel RequestedDebugLevel, WORD wRequestedDebugGroups)

Enables Debugging.

Sets the debuglevel and the groups for which output shall be done.

Parameters:

RequestedDebugLevel : the new debuglevel

wRequestedDebugGroups : the groups for which output shall be done.

Returns:

void

6.3.2.6 WORD SWUPSERILOWEXIMPORT WSwupInt_GiveUpdateNrFromComPort (t_SwupCom ComPort)

Gives UpdateNumber in dependency with ComPort.

Parameters:

ComPort

Returns:

WORD wUpdateNr

return Fehler ??? Default = UpdateNr 0 ... ;

6.3.2.7 void SWUPSERILOWEXIMPORT WSwupInt_SetUpdateNrToComPort (t_SwupCom ComPort, WORD wUpdateNr)

Sets UpdateNumber in dependency from ComPort.

Parameters:

ComPort The Swup-Comport

wUpdateNr The Update-Number

Returns:

void

6.3.3 Variable Documentation**6.3.3.1 t_DebugLevel CurrentFileDebugLevel = enNoDebug**

CurrentDebug-Level for File-Debugging, initailaised to not debug.

6.3.3.2 t_DebugLevel CurrentOnlineDebugLevel = enNoDebug

CurrentDebug-Level for Online-Debugging, initailaised to not debug.

6.3.3.3 BOOL fDebugToFile = FALSE

Bool, No debug to file as default.

6.3.3.4 BOOL fDebugToWindow = FALSE

Bool, No debug to window as default.

6.3.3.5 WORD wEnabledFileDebugGroups

the different groups that are enabled for file-debugging

6.3.3.6 WORD wEnabledOnlineDebugGroups

the different groups that are enabled for online debugging

6.4 WSWUPSER.DLL : Functionality for Error-Tracing

Functions to store error-information from anywhere and to retrieve it.

Data Structures

- struct [t_ErrorCodeDescription](#)
Structure containing the error-code and a description.
- struct [t_ErrorField](#)
A struct containing all errors from the indicated update, bIndex is a pointer to the next free t_ErrorStruct.
- struct [t_ErrorInfo](#)
A struct containing all info about an error.

Enumerations

- enum [t_ErrTextInfo](#) { [enStandardErrText](#), [enAdditionalErrText](#) }
An enumeration containing information about the error-text, some errors have predefined texts, some errors have dynamical error-texts.

6.4.1 Detailed Description

Functions to store error-information from anywhere and to retrieve it.

All functions described here are part of the WSWUPSER-DLL, the DLL containing the lower layer serial functions and some other stuff, like error-tracing, not handling. There was need to split this functionality into the external and the internal part because some of the error-functions get accessed from the external user. And error-tracing itself must be possible from all three parts of the library, the serial DLL, the project-dependent DLL and the main-dll.

6.4.2 Enumeration Type Documentation

6.4.2.1 enum [t_ErrTextInfo](#)

An enumeration containing information about the error-text, some errors have predefined texts, some errors have dynamical error-texts.

Enumeration values:

enStandardErrText this errortext is the predefined error-text

enAdditionalErrText the errortext is an additional dynamic error-text

Chapter 7

Programmers Guide for the Winswup-Library Data Structure Documentation

7.1 FileInfo Struct Reference

a structure which holds all informations of the WSWUP input File

Data Fields

- unsigned long [ulFileSize](#)
length of the input File
- unsigned long [ulSwupExeLen](#)
length of the SWUP part of an EXE File (other part=XBI)
- unsigned long [ulXbiLen](#)
length of the XBI Part (=File Size if XBI File)
- unsigned long [ulXbiHeaderLen](#)
length of the XBI Header
- unsigned long [ulHashClusterLen](#)
length of Hash Clusters in File
- unsigned long [ulSignatureLen](#)
length of Signature in File
- unsigned long [ulXbiDatenStart](#)
start address of XBI Data
- unsigned long [ulXbiDatenEnde](#)
end address of XBI Data

- unsigned int [unExeEndungLen](#)
length of EXE ending in File
- [t_XbiOrExe](#) [KindofFile](#)
type of File (XBI, EXE)
- [t_InitInfo](#) [enInitStatus](#)
type of initialisation (not, normal or for heap)

7.1.1 Detailed Description

a structure which holds all informations of the WSWUP input File

7.1.2 Field Documentation

7.1.2.1 [t_InitInfo](#) [FileInfo::enInitStatus](#)

type of initialisation (not, normal or for heap)

7.1.2.2 [t_XbiOrExe](#) [FileInfo::KindofFile](#)

type of File (XBI, EXE)

7.1.2.3 unsigned long [FileInfo::ulFileSize](#)

length of the input File

7.1.2.4 unsigned long [FileInfo::ulHashClusterLen](#)

length of Hash Clusters in File

7.1.2.5 unsigned long [FileInfo::ulSignatureLen](#)

length of Signature in File

7.1.2.6 unsigned long [FileInfo::ulSwupExeLen](#)

length of the SWUP part of an EXE File (other part=XBI)

7.1.2.7 unsigned long [FileInfo::ulXbiDatenEnde](#)

end address of XBI Data

7.1.2.8 unsigned long [FileInfo::ulXbiDatenStart](#)

start address of XBI Data

7.1.2.9 unsigned long [FileInfo::ulXbiHeaderLen](#)

length of the XBI Header

7.1.2.10 unsigned long [FileInfo::ulXbiLen](#)

length of the XBI Part (=File Size if XBI File)

7.1.2.11 unsigned int [FileInfo::unExeEndungLen](#)

length of EXE ending in File

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

- [filehand.c](#)

7.2 t_Block Struct Reference

a structure which holds information of one XBI Header block

Data Fields

- byte [ucLen](#)
- byte [ucId](#)
- byte [aucDataBuf](#) [64]

7.2.1 Detailed Description

a structure which holds information of one XBI Header block

7.2.2 Field Documentation

7.2.2.1 byte [t_Block::aucDataBuf](#)[64]

7.2.2.2 byte [t_Block::ucId](#)

7.2.2.3 byte [t_Block::ucLen](#)

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

- [filehand.c](#)

7.3 **t_ComConfig Struct Reference**

This describes one serial connection.

Data Fields

- DCB [dcb](#)
device control block structure for serial device
- COMMTIMEOUTS [ctmo](#)
timeout parameters for serial device
- HANDLE [hCom](#)
the Handle to the ComPort
- BOOL [fInit](#)
is it initialised ?

7.3.1 Detailed Description

This describes one serial connection.

For each possible connection (see AMOUNT_OF_SWUP_PORTS)we need such a struct

7.3.2 Field Documentation

7.3.2.1 COMMTIMEOUTS [t_ComConfig::ctmo](#)

timeout parameters for serial device

7.3.2.2 DCB [t_ComConfig::dcb](#)

device control block structure for serial device

7.3.2.3 BOOL [t_ComConfig::fInit](#)

is it initialised ?

7.3.2.4 HANDLE [t_ComConfig::hCom](#)

the Handle to the ComPort

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

- [wseril1.c](#)

7.4 t_enumErrorDescription Struct Reference

```
#include <err_text.h>
```

Data Fields

- unsigned int [ui_enError](#)
- unsigned int [ui_enAltError](#)
- char * [pszDescription](#)

7.4.1 Field Documentation

7.4.1.1 char* [t_enumErrorDescription::pszDescription](#)

7.4.1.2 unsigned int [t_enumErrorDescription::ui_enAltError](#)

7.4.1.3 unsigned int [t_enumErrorDescription::ui_enError](#)

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

- [err_text.h](#)

7.5 `t_ErrorCodeDescription` Struct Reference

Structure containing the error-code and a description.

Data Fields

- `DWORD dwErrCode`
the error-code, more or less useless for external usage
- `char * pszDescription`
pointer to a descriptive text of this error

7.5.1 Detailed Description

Structure containing the error-code and a description.

7.5.2 Field Documentation

7.5.2.1 `DWORD t_ErrorCodeDescription::dwErrCode`

the error-code, more or less useless for external usage

7.5.2.2 `char* t_ErrorCodeDescription::pszDescription`

pointer to a descriptive text of this error

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

- `wswerrin.c`

7.6 t_ErrorField Struct Reference

A struct containing all errors from the indicated update, bIndex is a pointer to the next free t_ErrorStruct.

```
#include <wsw_seri.h>
```

Data Fields

- BYTE [bIndex](#)
the index for enum field next to filled
- t_ErrorStruct [Error](#) [MAX_ENTRIES]

7.6.1 Detailed Description

A struct containing all errors from the indicated update, bIndex is a pointer to the next free t_ErrorStruct.

7.6.2 Field Documentation

7.6.2.1 BYTE [t_ErrorField::bIndex](#)

the index for enum field next to filled

7.6.2.2 t_ErrorStruct [t_ErrorField::Error](#)[MAX_ENTRIES]

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

- [wsw_seri.h](#)

7.7 `t_ErrorInfo` Struct Reference

A struct containing all info about an error.

Data Fields

- `DWORD dwError`
the error-code
- `char * pszFile`
the file which throw the error
- `int nLineNo`
the line in which the error occurred
- `char szErrMsg [1000]`
the text of the error
- `t_ErrTextInfo ErrTextInfo`
what kind of error-text is it

7.7.1 Detailed Description

A struct containing all info about an error.

7.7.2 Field Documentation

7.7.2.1 `DWORD t_ErrorInfo::dwError`

the error-code

7.7.2.2 `t_ErrTextInfo t_ErrorInfo::ErrTextInfo`

what kind of error-text is it

7.7.2.3 `int t_ErrorInfo::nLineNo`

the line in which the error occurred

7.7.2.4 `char* t_ErrorInfo::pszFile`

the file which throw the error

7.7.2.5 char t.ErrorInfo::szErrBuff[1000]

the text of the error

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

- [wswerrin.c](#)

7.8 *t_ExtendedInfo Struct Reference*

The "Master"-stuct, containing all info in the XBI-Header All arrays containg text (especially dates) are 1 Byte longer to allow the storage of the .

```
#include <xbi_info.h>
```

Data Fields

- char [szLocDate](#) [9]
Date when SW was located (r b c - Date).
- char [szLocTime](#) [9]
Time when SW was located (r b c - Time).
- char [szRbmDate](#) [9]
Date when Make-File was generated (r b m - Date).
- char [szRbmTime](#) [9]
Time when Make-File was generated (r b m - Time).
- char [szProjName](#) [9]
Name of the project eg.
- char [szDevlName](#) [9]
Name of the developper of this Software.
- char [szLib_1_Date](#) [9]
Date of Lib 1 , obsolete, but dont drop !
- char [szLib_1_Time](#) [9]
Time of Lib 1 , obsolete, but dont drop !
- char [szLib_2_Date](#) [9]
Date of Lib 2 , obsolete, but dont drop !
- char [szLib_2_Time](#) [9]
Time of Lib 2 , obsolete, but dont drop !
- char [szLangGroup](#) [17]
string containin the Language-Group eg
- char [szProductName](#) [17]
official name of the product, eg SL45
- char [szVendorName](#) [17]
Name of the vendor of this Software , eg Siemens or Sony.
- char [szSwGeneration](#) [17]
SW-Directory where this Software was built eg.

- [t_SwType SwType](#)
a [t_SwType](#) -enum, describing the contents of the Data-Part
- unsigned short [nReleaseMajor](#)
one of these is used for the official SVN
- unsigned short [nReleaseMinor](#)
one of these is used for the official SVN
- unsigned long [ulRbmPutCount](#)
Net-Putcount when Make-Files were generated.
- unsigned long [ulRbcPutCount](#)
Net-Putcount when SW was located.
- unsigned long [ulConstAdress](#)
where is swup allowed to store its data
- unsigned short [nErtecSum](#)
Checksum, only for production !
- unsigned long [ulEpromSizeInBytes](#)
Size of this Software (in terms of Flash-Size).
- unsigned int [nTableEntries](#)
Amount of locater-entries of the linked list below.
- [t_LocaterRecord](#) * [ptrLocTable](#)
linked list for locater-entries of type [t_LocaterRecord](#)
- unsigned short [nSwupVersion](#)
at least needed SWUP-Version to update this Software
- char [szSwupDllName](#) [17]
SWUP DLL name used for each project.
- unsigned char [fForceDllNameFromNhk](#)
force the DLL to be used from the NHK Configuration File (not from SWUP)
- unsigned short [nHexVersion](#)
generated with HEX-BIN-Converter-Version
- char * [pucText](#)
a text the developper wants to show during swup
- unsigned long [ulTextLen](#)
len of the text the developper wants to show during swup
- [t_Area](#) * [ptrEraseList](#)

what do we want to erase, linked list of type [t_Area](#)

- [t_AllLibInfo * pLibInfo](#)
a linked list of type [t_AllLibInfo](#) containing all the infos about the LNO-Projects
- [t_AsicTyp WhichAsic](#)
a [t_AsicTyp](#) - Value, used for adjusting the watchdog during swup
- [t_WriteTyp HowToWrite](#)
a [t_WriteTyp](#) - Value, describing how we allow writing to flash in this mobile
- unsigned short [unRamSize](#)
Ram-Size (in kb ???).
- [t_ProcType ProcessorInfo](#)
this software was compiled for a [t_ProcType](#) - Processor-Type
- [t_IgnType IgnitionInfo](#)
how can we detect, if ignition-line is okay ?
- [t_LangCheckInfo LanguageCheckInfo](#)
infos about the language-booting a [t_LangCheckInfo](#) - structure
- unsigned long [ulSplitID](#)
what is the Split-ID of this Software
- unsigned long [ulSplitIdAdress](#)
where is it located in the Mobile ???
- unsigned int [unAlignement](#)
what alignement does the data have in the data-part
- [t_FileFormat FormatInfo](#)
what format has the data, a [t_FileFormat](#) - Value
- unsigned int [unCompAlg](#)
which compression-algorithm was used to compress the data
- unsigned int [unCompressRatio](#)
what compression-ratio did we get
- [t_CompressInfo CompressionInfo](#)
which parameters were used for comporession ? Useless !
- [t_FileFormat CompFormatBefore](#)
which format had the data before we put it in the compressor
- [t_FileFormat CompFormatAfter](#)
which format has the data after the compressor, must be bin in the moment (Mar 2001)

- char [szOldLocDate](#) [9]
for a diff-sw : against which SW was the diff done
- char [szOldLocTime](#) [9]
for a diff-sw : against which SW was the diff done
- char [szOldRbmDate](#) [9]
for a diff-sw : against which SW was the diff done
- char [szOldRbmTime](#) [9]
for a diff-sw : against which SW was the diff done
- char [szOldProjName](#) [9]
for a diff-sw : against which SW was the diff done
- char [szOldDevlName](#) [9]
for a diff-sw : against which SW was the diff done
- unsigned int [unAdditionalMapLen](#)
new additional-map-info : how long is it
- unsigned char * [ucAdditionalMapInfo](#)
new additional-map-info : what is it
- unsigned int [fAdditionalInfoAvail](#)
new additional-map-info : is it present
- [t_NewSplitInfo](#) * [ptrNewSplitInfoList](#)
newsplit info where, what, ...
- unsigned long [HashInfo](#)
*Length of hash information, (Hashclusterentry size*hashclusterNumber)+30.*

7.8.1 Detailed Description

The "Master"-stuct, containing all info in the XBI-Header All arrays containg text (especially dates) are 1 Byte longer to allow the storage of the .

7.8.2 Field Documentation

7.8.2.1 [t_FileFormat](#) [t_ExtendedInfo::CompFormatAfter](#)

which format has the data after the compressor, must be bin in the moment (Mar 2001)

7.8.2.2 [t_FileFormat](#) [t_ExtendedInfo::CompFormatBefore](#)

which format had the data before we put it in the compressor

7.8.2.3 [t_CompressInfo](#) [t_ExtendedInfo::CompressionInfo](#)

which parameters were used for comporession ? Useless !

7.8.2.4 unsigned int [t_ExtendedInfo::fAdditionalInfoAvail](#)

new additional-map-info : is it present

7.8.2.5 unsigned char [t_ExtendedInfo::fForceDllNameFromNhk](#)

force the DLL to be used from the NHK Configuration File (not from SWUP)

7.8.2.6 [t_FileFormat](#) [t_ExtendedInfo::FormatInfo](#)

what format has the data, a [t_FileFormat](#) - Value

7.8.2.7 unsigned long [t_ExtendedInfo::HashInfo](#)

Length of hash information, (Hashclusterentry size*hashclusterNumber)+30.

7.8.2.8 [t_WriteTyp](#) [t_ExtendedInfo::HowToWrite](#)

a [t_WriteTyp](#) - Value, describing how we allow writing to flash in this mobile

7.8.2.9 [t_IgnType](#) [t_ExtendedInfo::IgnitionInfo](#)

how can we detect, if ignition-line is okay ?

7.8.2.10 [t_LangCheckInfo](#) [t_ExtendedInfo::LanguageCheckInfo](#)

infos about the language-booting a [t_LangCheckInfo](#) - structure

7.8.2.11 unsigned short [t_ExtendedInfo::nErtecSum](#)

Checksum, only for production !

7.8.2.12 unsigned short [t_ExtendedInfo::nHexVersion](#)

generated with HEX-BIN-Converter-Version

7.8.2.13 unsigned short [t_ExtendedInfo::nReleaseMajor](#)

one of these is used for the official SVN

7.8.2.14 unsigned short [t_ExtendedInfo::nReleaseMinor](#)

one of these is used for the official SVN

7.8.2.15 unsigned short [t_ExtendedInfo::nSwupVersion](#)

at least needed SWUP-Version to update this Software

7.8.2.16 unsigned int [t_ExtendedInfo::nTableEntries](#)

Amount of locator-entries of the linked list below.

7.8.2.17 [t_AllLibInfo*](#) [t_ExtendedInfo::pLibInfo](#)

a linked list of type [t_AllLibInfo](#) containing all the infos about the LNO-Projects

7.8.2.18 [t_ProcType](#) [t_ExtendedInfo::ProcessorInfo](#)

this software was compiled for a [t_ProcType](#) - Processor-Type

7.8.2.19 [t_Area*](#) [t_ExtendedInfo::ptrEraseList](#)

what do we want to erase, linked list of type [t_Area](#)

7.8.2.20 [t_LocatorRecord*](#) [t_ExtendedInfo::ptrLocTable](#)

linked list for locator-entries of type [t_LocatorRecord](#)

7.8.2.21 [t_NewSplitInfo*](#) [t_ExtendedInfo::ptrNewSplitInfoList](#)

newsplit info where, what, ...

7.8.2.22 [char*](#) [t_ExtendedInfo::pucText](#)

a text the developer wants to show during swup

7.8.2.23 [t_SwType](#) [t_ExtendedInfo::SwType](#)

a [t_SwType](#) -enum, describing the contents of the Data-Part

7.8.2.24 [char](#) [t_ExtendedInfo::szDevName](#)[9]

Name of the developer of this Software.

7.8.2.25 char [t_ExtendedInfo::szLangGroup](#)[17]

string containin the Language-Group eg
lg1

7.8.2.26 char [t_ExtendedInfo::szLib_1_Date](#)[9]

Date of Lib 1 , obsolete, but dont drop !

7.8.2.27 char [t_ExtendedInfo::szLib_1_Time](#)[9]

Time of Lib 1 , obsolete, but dont drop !

7.8.2.28 char [t_ExtendedInfo::szLib_2_Date](#)[9]

Date of Lib 2 , obsolete, but dont drop !

7.8.2.29 char [t_ExtendedInfo::szLib_2_Time](#)[9]

Time of Lib 2 , obsolete, but dont drop !

7.8.2.30 char [t_ExtendedInfo::szLocDate](#)[9]

Date when SW was located (rb c - Date).

7.8.2.31 char [t_ExtendedInfo::szLocTime](#)[9]

Time when SW was located (rb c - Time).

7.8.2.32 char [t_ExtendedInfo::szOldDevlName](#)[9]

for a diff-sw : against which SW was the diff done

7.8.2.33 char [t_ExtendedInfo::szOldLocDate](#)[9]

for a diff-sw : against which SW was the diff done

7.8.2.34 char [t_ExtendedInfo::szOldLocTime](#)[9]

for a diff-sw : against which SW was the diff done

7.8.2.35 char [t_ExtendedInfo::szOldProjName](#)[9]

for a diff-sw : against which SW was the diff done

7.8.2.36 char [t_ExtendedInfo::szOldRbmDate](#)[9]

for a diff-sw : against which SW was the diff done

7.8.2.37 char [t_ExtendedInfo::szOldRbmTime](#)[9]

for a diff-sw : against which SW was the diff done

7.8.2.38 char [t_ExtendedInfo::szProductName](#)[17]

official name of the product, eg SL45

7.8.2.39 char [t_ExtendedInfo::szProjName](#)[9]

Name of the project eg.

epu35

7.8.2.40 char [t_ExtendedInfo::szRbmDate](#)[9]

Date when Make-File was generated (rb m - Date).

7.8.2.41 char [t_ExtendedInfo::szRbmTime](#)[9]

Time when Make-File was generated (rb m - Time).

7.8.2.42 char [t_ExtendedInfo::szSwGeneration](#)[17]

SW-Directory where this Software was built eg.

gen7u.z1

7.8.2.43 char [t_ExtendedInfo::szSwupDllName](#)[17]

SWUP DLL name used for each project.

7.8.2.44 char [t_ExtendedInfo::szVendorName](#)[17]

Name of the vendor of this Software , eg Siemens or Sony.

7.8.2.45 unsigned char* [t_ExtendedInfo::ucAdditionalMapInfo](#)

new additional-map-info : what is it

7.8.2.46 unsigned long [t_ExtendedInfo::ulConstAddress](#)

where is swup allowed to store its data

7.8.2.47 unsigned long [t_ExtendedInfo::ulEpromSizeInBytes](#)

Size of this Software (in terms of Flash-Size).

7.8.2.48 unsigned long [t_ExtendedInfo::ulRbcPutCount](#)

Net-Putcount when SW was located.

7.8.2.49 unsigned long [t_ExtendedInfo::ulRbmPutCount](#)

Net-Putcount when Make-Files were generated.

7.8.2.50 unsigned long [t_ExtendedInfo::ulSplitID](#)

what is the Split-ID of this Software

7.8.2.51 unsigned long [t_ExtendedInfo::ulSplitIdAdress](#)

where is it located in the Mobile ???

7.8.2.52 unsigned long [t_ExtendedInfo::ulTextLen](#)

len of the text the developper wants to show during swup

7.8.2.53 unsigned int [t_ExtendedInfo::unAdditionalMapLen](#)

new additional-map-info : how long is it

7.8.2.54 unsigned int [t_ExtendedInfo::unAlignement](#)

what alignement does the data have in the data-part

7.8.2.55 unsigned int [t_ExtendedInfo::unCompAlg](#)

which compression-algorithm was used to compress the data

7.8.2.56 unsigned int [t_ExtendedInfo::unCompressRatio](#)

what compression-ratio did we get

7.8.2.57 unsigned short [t_ExtendedInfo::unRamSize](#)

Ram-Size (in kb ???).

7.8.2.58 [t_AsicTyp](#) [t_ExtendedInfo::WhichAsic](#)

a [t_AsicTyp](#) - Value, used for adjusting the watchdog during swup

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

- [xbi_info.h](#)

7.9 T_HashBinBlock Struct Reference

a structure used to save the hash entries or the signature like bin blocks.

Data Fields

- byte [Address](#) [3]
- byte [Length](#)
- byte [MI](#)
- byte [PhaseID](#)
- unsigned short [offset](#)
- byte [Data](#) [60]
- byte [Checksum](#)

7.9.1 Detailed Description

a structure used to save the hash entries or the signature like bin blocks.

7.9.2 Field Documentation

7.9.2.1 byte [T_HashBinBlock::Address](#)[3]

7.9.2.2 byte [T_HashBinBlock::Checksum](#)

7.9.2.3 byte [T_HashBinBlock::Data](#)[60]

7.9.2.4 byte [T_HashBinBlock::Length](#)

7.9.2.5 byte [T_HashBinBlock::MI](#)

7.9.2.6 unsigned short [T_HashBinBlock::offset](#)

7.9.2.7 byte [T_HashBinBlock::PhaseID](#)

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

- [filehand.c](#)

7.10 t_LangCheckInfo Struct Reference

Information about Addresses on Language-Group-Booting.

```
#include <xbi_info.h>
```

Data Fields

- unsigned long [ulStartAdr](#)
Start-Adress of language-group.
- unsigned long [ulEndAdr](#)
End-Adress of language-group.
- unsigned long [ulPatchAdr](#)
where do we Patch the Boot-Bad-recognition
- unsigned long [ulChecksumAdr](#)
where is the checksum located

7.10.1 Detailed Description

Information about Addresses on Language-Group-Booting.

7.10.2 Field Documentation

7.10.2.1 unsigned long [t_LangCheckInfo::ulChecksumAdr](#)

where is the checksum located

7.10.2.2 unsigned long [t_LangCheckInfo::ulEndAdr](#)

End-Adress of language-group.

7.10.2.3 unsigned long [t_LangCheckInfo::ulPatchAdr](#)

where do we Patch the Boot-Bad-recognition

7.10.2.4 unsigned long [t_LangCheckInfo::ulStartAdr](#)

Start-Adress of language-group.

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

- [xbi_info.h](#)

7.11 `t_TableEntry` Struct Reference

For each Message-Identifier it is possible to define a default-Handler.

Data Fields

- unsigned char `ucMi`
Message-Identifier that the default-handler handles.
- `pfnDefaultHandler` `DefaultHandler`
pointer to the handler-function

7.11.1 Detailed Description

For each Message-Identifier it is possible to define a default-Handler.

This default-handler will be called if a Message-Block with this identifier is received and this message-block is not expected. Used for example for Mobile-Errors which are normally "unexpected"

7.11.2 Field Documentation

7.11.2.1 `pfnDefaultHandler` `t_TableEntry::DefaultHandler`

pointer to the handler-function

7.11.2.2 unsigned char `t_TableEntry::ucMi`

Message-Identifier that the default-handler handles.

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

- `wseril2.c`

7.12 tag_AllLibInfo Struct Reference

A linked list containing the library-infos.

```
#include <xbi_info.h>
```

Data Fields

- char [abInfo](#) [0x50]
50 Bytes containing all the Library-Information
- [tag_AllLibInfo](#) * [ptrNext](#)
ptr to the next Library -Information

7.12.1 Detailed Description

A linked list containing the library-infos.

7.12.2 Field Documentation

7.12.2.1 char [tag_AllLibInfo::abInfo](#)[0x50]

50 Bytes containing all the Library-Information

7.12.2.2 struct [tag_AllLibInfo](#)* [tag_AllLibInfo::ptrNext](#)

ptr to the next Library -Information

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

- [xbi_info.h](#)

7.13 tag_Area Struct Reference

Definition of a listenelement of the Erase/Not-Erase/Generate/And-so-on - Table.

```
#include <xbi_info.h>
```

Data Fields

- unsigned long [ulStartAddress](#)
Start of Erasure.
- unsigned long [ulEndAddress](#)
End of Erasure.
- [tag_Area](#) * [pNext](#)
pointer to the next list-element

7.13.1 Detailed Description

Definition of a listenelement of the Erase/Not-Erase/Generate/And-so-on - Table.

Implemented as a linked list.

7.13.2 Field Documentation

7.13.2.1 struct [tag_Area](#)* [tag_Area::pNext](#)

pointer to the next list-element

7.13.2.2 unsigned long [tag_Area::ulEndAddress](#)

End of Erasure.

7.13.2.3 unsigned long [tag_Area::ulStartAddress](#)

Start of Erasure.

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

- [xbi_info.h](#)

7.14 tag_BfbThreadInfo Struct Reference

Data Fields

- WORD [wThreadUpdateNr](#)
- t_SwupCom [ThreadWhichCom](#)
- unsigned short * [punThreadVoltage](#)

7.14.1 Field Documentation

7.14.1.1 unsigned short* [tag_BfbThreadInfo::punThreadVoltage](#)

7.14.1.2 t_SwupCom [tag_BfbThreadInfo::ThreadWhichCom](#)

7.14.1.3 WORD [tag_BfbThreadInfo::wThreadUpdateNr](#)

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

- [wbfbtool.c](#)

7.15 tag_CompressInfo Struct Reference

A (complete useless) information about parameters on compressing.

```
#include <xbi_info.h>
```

Data Fields

- unsigned short [unInfoOne](#)
three infos, content depending on compression-algorithm
- unsigned short [unInfoTwo](#)
- unsigned short [unInfoThree](#)

7.15.1 Detailed Description

A (complete useless) information about parameters on compressing.

Was never used, is intended to allow different parameters on compression in HEX-BIN-Converter and then to transmit to mobile, so that mobile can adapt it's decompressor.

7.15.2 Field Documentation

7.15.2.1 unsigned short [tag_CompressInfo::unInfoOne](#)

three infos, content depending on compression-algorithm

7.15.2.2 unsigned short [tag_CompressInfo::unInfoThree](#)

7.15.2.3 unsigned short [tag_CompressInfo::unInfoTwo](#)

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

- [xbi_info.h](#)

7.16 tag_LocaterRecord Struct Reference

A new structure for "locating" Up to now only implemented in the HEX-BIN-Converter.

```
#include <xbi_info.h>
```

Data Fields

- unsigned int [nSegInRom](#)
where is the segment in the master-rom (or XBI-File)
- unsigned int [nSegInMobile](#)
where is it located in the mobile
- unsigned int [nAltSeg1](#)
where else should it be seen in the mobile
- unsigned int [nAltSeg2](#)
and where else should it be seen in the mobile
- [tag_LocaterRecord](#) * [ptrNext](#)

7.16.1 Detailed Description

A new structure for "locating" Up to now only implemented in the HEX-BIN-Converter.

switched on via Flag in the NHK-File.

7.16.2 Field Documentation

7.16.2.1 unsigned int [tag_LocaterRecord::nAltSeg1](#)

where else should it be seen in the mobile

7.16.2.2 unsigned int [tag_LocaterRecord::nAltSeg2](#)

and where else should it be seen in the mobile

7.16.2.3 unsigned int [tag_LocaterRecord::nSegInMobile](#)

where is it located in the mobile

7.16.2.4 unsigned int [tag_LocaterRecord::nSegInRom](#)

where is the segment in the master-rom (or XBI-File)

7.16.2.5 **struct [tag_LocaterRecord](#)* [tag_LocaterRecord::ptrNext](#)**

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

- [xbl_info.h](#)

7.17 tag_PINstruct Struct Reference

a structure which holds PIN information

Data Fields

- unsigned int [unPINSize](#)
Number of data bytes of Boot PIN (16 Byte(128Bit) in the moment).
- unsigned char [aucBootPIN](#) [MAX_SIZE_OF_PIN]
PIN Data.

7.17.1 Detailed Description

a structure which holds PIN information

7.17.2 Field Documentation

7.17.2.1 unsigned char [tag_PINstruct::aucBootPIN](#)[MAX_SIZE_OF_PIN]

PIN Data.

7.17.2.2 unsigned int [tag_PINstruct::unPINSize](#)

Number of data bytes of Boot PIN (16 Byte(128Bit) in the moment).

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

- [swupwork.c](#)

7.18 tag_ThreadInfo Struct Reference

a structure which holds the information of Threads (COM, speed, Update Nr)

Data Fields

- WORD [wThreadUpdateNr](#)
Update number (for each swup).
- t_SwupCom [ThreadWhichCom](#)
which COM from the PC is used for Thread
- unsigned long [ulThreadSpeed](#)
the speed of Thread

7.18.1 Detailed Description

a structure which holds the information of Threads (COM, speed, Update Nr)

7.18.2 Field Documentation

7.18.2.1 t_SwupCom [tag_ThreadInfo::ThreadWhichCom](#)

which COM from the PC is used for Thread

7.18.2.2 unsigned long [tag_ThreadInfo::ulThreadSpeed](#)

the speed of Thread

7.18.2.3 WORD [tag_ThreadInfo::wThreadUpdateNr](#)

Update number (for each swup).

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

- [swupwork.c](#)

7.19 tag_UsbInfo Struct Reference

A struct containing all information needed for UsbUpdate.

```
#include <wsw_seri.h>
```

Data Fields

- **BOOL** [UsbWanted](#)
set if this thread has to be updated via USB & Sipc protocol
- **BOOL** [ForceUsb](#)
set if thread is forced to defined UsbPort
- **t_SwupCom** [UsbPort](#)
which Usb from the PC is used for thread

7.19.1 Detailed Description

A struct containing all information needed for UsbUpdate.

7.19.2 Field Documentation

7.19.2.1 **BOOL** [tag_UsbInfo::ForceUsb](#)

set if thread is forced to defined UsbPort

7.19.2.2 **t_SwupCom** [tag_UsbInfo::UsbPort](#)

which Usb from the PC is used for thread

7.19.2.3 **BOOL** [tag_UsbInfo::UsbWanted](#)

set if this thread has to be updated via USB & Sipc protocol

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

- [wsw_seri.h](#)

7.20 tagDebugFiles Struct Reference

Data Fields

- t.SwupCom [ComPort](#)
ComPort für Umwandlung.
- FILE * [fpDebug](#)

7.20.1 Field Documentation

7.20.1.1 t.SwupCom [tagDebugFiles::ComPort](#)

ComPort für Umwandlung.

7.20.1.2 FILE* [tagDebugFiles::fpDebug](#)

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

- [wsbdbgin.c](#)

7.21 tagMessageBlock Struct Reference

A structure containing a message-block with Message-Identifier, Len and additional Data.

```
#include <wsw_ser_i.h>
```

Data Fields

- unsigned char [ucMi](#)
the Message-Identifier of a Message-Block
- unsigned char [ucLen](#)
the len of the additional data of the Message-Block
- unsigned char [ucMessageData](#) [MAX_DATA_IN_BIN_BLOCK]
the additional data of the Message-Block

7.21.1 Detailed Description

A structure containing a message-block with Message-Identifier, Len and additional Data.

7.21.2 Field Documentation

7.21.2.1 unsigned char [tagMessageBlock::ucLen](#)

the len of the additional data of the Message-Block

7.21.2.2 unsigned char [tagMessageBlock::ucMessageData](#)[MAX_DATA_IN_BIN_BLOCK]

the additional data of the Message-Block

7.21.2.3 unsigned char [tagMessageBlock::ucMi](#)

the Message-Identifier of a Message-Block

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

- [wsw_ser_i.h](#)

7.22 tagNewSplitInfo Struct Reference

Definition of an element of the New Split information (can be Tegic, Language, FileSystem,...) Implemented as a linked list.

```
#include <xbi_info.h>
```

Data Fields

- unsigned long [ulStartAdr](#)
Start address of checksum area.
- unsigned long [ulEndAdr](#)
End-address of checksum area.
- unsigned long [ulChecksumAdr](#)
Checksum address.
- unsigned short [SplitCountID](#)
Kind of info Tegic, Language, Filesytem ..
- [t_NewSplitIdentifier](#) [NewSplitId](#)
Kind of split , info , info+code.
- char [szCommentAboutSplitEntity](#) [MAX_SPLIT_COMMENT_LENGTH]
a human-readable name
- [tagNewSplitInfo](#) * [pNext](#)
pointer to the next list-element

7.22.1 Detailed Description

Definition of an element of the New Split information (can be Tegic, Language, FileSystem,...) Implemented as a linked list.

7.22.2 Field Documentation

7.22.2.1 [t_NewSplitIdentifier](#) [tagNewSplitInfo::NewSplitId](#)

Kind of split , info , info+code.

7.22.2.2 **struct** [tagNewSplitInfo](#)* [tagNewSplitInfo::pNext](#)

pointer to the next list-element

7.22.2.3 unsigned short [tagNewSplitInfo::SplitCountID](#)

Kind of info Tegic, Language, Filesytem ..

7.22.2.4 char [tagNewSplitInfo::szCommentAboutSplitEntity](#)[MAX_SPLIT_COMMENT_LENGTH]

a human-readable name

7.22.2.5 unsigned long [tagNewSplitInfo::ulChecksumAdr](#)

Checksum address.

7.22.2.6 unsigned long [tagNewSplitInfo::ulEndAdr](#)

End-address of checksum area.

7.22.2.7 unsigned long [tagNewSplitInfo::ulStartAdr](#)

Start address of checksum area.

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

- [xbi_info.h](#)

Chapter 8

Programmers Guide for the Winswup-Library File Documentation

8.1 boot_hex.h File Reference

One of two Interface-Headers between HEX-Converter and SWUP-Program.

Defines

- #define [MEMO2XBL_VERSION_MAJOR](#) 0
(VERY OLD !!!) Version of HEX-BIN-Converter for Voice-Memos, was last used for the S10 !
- #define [MEMO2XBL_VERSION_MINOR](#) 70
(VERY OLD !!!) Version of HEX-BIN-Converter for Voice-Memos, was last used for the S10 !
- #define [HEXUNIX_VERSION_MAJOR](#) 1
Standard : current version of Hex-Bin-Converter, Major-Version.
- #define [HEXUNIX_VERSION_MINOR](#) 19
Standard : current version of Hex-Bin-Converter, Minor part of the version.
- #define [CONVERTER_VERSION_MAJOR](#) [HEXUNIX_VERSION_MAJOR](#)
- #define [CONVERTER_VERSION_MINOR](#) [HEXUNIX_VERSION_MINOR](#)
- #define [LEAST_SWUP_VERSION_MAJOR](#) 1
- #define [LEAST_SWUP_VERSION_MINOR](#) 12
- #define [LEAST_LANGSWUP_VERSION_MAJOR](#) 1
the SWUP-Version which is needed at least to update a language-file, major-part of the version
- #define [LEAST_LANGSWUP_VERSION_MINOR](#) 61
the SWUP-Version which is needed at least to update a language-file, minor-part of the version
- #define [LEAST_ZIPSWUP_VERSION_MAJOR](#) 1
the SWUP-Version which is needed at least to update a compressed file, major-part of the version
- #define [LEAST_ZIPSWUP_VERSION_MINOR](#) 72

the SWUP-Version which is needed at least to update a compressed file, minor-part of the version

- #define [ID_END_OF_HEADER](#) 0x04
End of Header, so that later versions also can recognize the end of the header without problems.
- #define [ID_HEXER_VERSION](#) 0x10
Version of HEX-converter with which this file was generated.
- #define [ID_LEAST_SWUP_VERSION](#) 0x11
needed version of SWUP to perform an update of this FILE
- #define [ID_MOB_SW_LOC_DATE](#) 0x12
Date and Time when this Mobile-SW was "located".
- #define [ID_MOB_SW_RBM_DATE](#) 0x13
Date and Time when the makefiles for this SW were generated.
- #define [ID_LOCATER_ENTRY](#) 0x14
how was the software "located" from processor-view to flash-view (segment-oriented)
- #define [ID_END_LOCATER_ENTRY](#) 0x15
End of the locater-table.
- #define [ID_DEVELOPER](#) 0x16
Name of the developer which generated this mobile-SW.
- #define [ID_PROJECT_NAME](#) 0x17
Name of the project, for ex.
- #define [ID_PROJECT_ID](#) 0x18
I have no idea if this ever was used ...????????????????????
- #define [ID_SPEECHGROUP_ID](#) 0x19
(OLD SPLIT !!!) Language group Identifier, was this ever used ???
- #define [ID_SPEECHGROUP_NAME](#) 0x1A
(OLD SPLIT !!!) Language group Name, for ex
- #define [ID_RBM_PUT_COUNT](#) 0x1B
put-count on generating the makefiles for this mobile SW
- #define [ID_SW_STATE](#) 0x1C
SW State, was in my opinion never used (local/ official / ..
- #define [ID_SW_VERSION_NUMBER](#) 0x1D
offical GSM-SW-Version-Number <=> SVN
- #define [ID_INTERNAL_VERSION](#) 0x1E
internal Version-Number, was never used in my opinion ??????????????????????

- #define [ID_NO_CHECK](#) 0x1F
Entries where we dont want to calc.
- #define [ID_TEXT](#) 0x20
Text that the developer added during the hex-bin-conversion.
- #define [ID_TEXT_END](#) 0x21
End of the text.
- #define [ID_TEXT_LENGTH](#) 0x22
Length of the text.
- #define [ID_EPROM_SIZE](#) 0x23
Size of Flash, this mobile-SW was generated for.
- #define [ID_ERTEC_SUM](#) 0x24
only for information : the ertec-checksum
- #define [ID_CONSTANTS](#) 0x25
where are the SWUP-constants located
- #define [ID_OLD_PROJECT](#) 0x26
(VERY VERY VERY OLD !!!) Project-ID
- #define [ID_RBC_PUT_COUNT](#) 0x27
put-count on compiling (locating) the Software
- #define [ID_SW_PRODUCT_NAME](#) 0x28
the official Name of the product, for ex
- #define [ID_SW_VENDOR_NAME](#) 0x29
the official Vendor of the SW, for ex
- #define [ID_SW_GENERATION](#) 0x2A
on which SUN-directory this SW was generated, for ex
- #define [ID_EXT_LIB_1_DATE](#) 0x2B
(VERY OLD !!!) from the time when we distributed only two lno-projects as obj to other sites
- #define [ID_EXT_LIB_2_DATE](#) 0x2C
(VERY OLD !!!) from the time when we distributed only two lno-projects as obj to other sites
- #define [ID_ALL_LIB_INFO_P1](#) 0x2D
(NEW !!!) informations about the lno-project, each lno = 0x50 bytes, here are the first 0x30 bytes
- #define [ID_ALL_LIB_INFO_P2](#) 0x2E
(NEW !!!) the rest 0x20 Bytes information about lno-project
- #define [ID_ERASE_INFO](#) 0x30
Parts to be erased.

- #define [ID_ASIC_TYPE](#) 0x31
ASIC-Type only used in terms of watchdog-timing.
- #define [ID_FLASH_WRITE_TYPE](#) 0x32
how is the flash-writing enabled for this SW
- #define [ID_RAM_SIZE](#) 0x33
which RAM Size do we have in the mobile
- #define [ID_PROC_TYPE](#) 0x34
for which Processor Type is this SW compiled
- #define [ID_IGNITION_INFO](#) 0x35
how is the ignition detected (old ..
- #define [ID_LANG_CHECK_INFO](#) 0x36
(OLD SPLIT !!!) informations about the Language Group checksums
- #define [ID_LANG_SPLIT_ID](#) 0x37
(OLD SPLIT !!!) the Split ID (unique number do determine if SW and Language Group fit together
- #define [ID_ALIGNEMENT](#) 0x38
1/2/4-Byte-Alignement in the appended bin-data
- #define [ID_FORMAT_INFO](#) 0x39
Information about the format (bin lck raw ..
- #define [ID_COMPRESS_INFO](#) 0x3A
Information about compression and parameters.
- #define [ID_OLD_LOC_DATE](#) 0x3B
on Incremental-XBI: derived from, compiled Date and Time
- #define [ID_OLD_RBM_DATE](#) 0x3C
on Incremental-XBI: derived from, rb m Date and Time
- #define [ID_OLD_PROJ_NAME](#) 0x3D
on Incremental-XBI: derived from, Project Name
- #define [ID_OLD_DEVELOPER](#) 0x3E
on Incremental-XBI: derived from, Developer Name
- #define [ID_SW_TYPE](#) 0x40
What kind of SW is the appended SW : Code, Language, etc..
- #define [ID_ADDITIONAL_INFO_LEN](#) 0x50
Additional Information, Length.
- #define [ID_ADDITIONAL_INFO_ENTRY](#) 0x51

Additional Information, for ex.

- #define `ID_ADDITIONAL_INFO_END` 0x52
Additional Information-End.
- #define `ID_NEW_SPLIT_INFO` 0x53
New Split Information for Generic splitting (Tegic, Language, ...
- #define `ID_NEW_SPLIT_INFO_TEXT` 0x54
New Split Information-Text (for ex.
- #define `ID_NEW_SPLIT_ADD_INFO_TEXT` 0x55
New Split Information, Additional Info-Text (example : lg1, lg2 , lg3).
- #define `ID_HASH_TABLE_INFO` 0x56
Info about Hash table length for signed sw update.
- #define `ID_TEGIC_GROUP` 0x60
holds which Tegic-Group is in the file
- #define `ID_SWUP_DLL` 0x70
holds which DLL the SWUP has to use for the project
- #define `XBI_KENNUNGS_STRING` "Siemens Mobile Phones Software"
when you are going to change this, you must also change SWUP !
- #define `XBI_KENNUNGS_STRING_SGOLD` "Siemens Mobile Phones:SOFTWARE:"
beginning of SGOLD header without version
- #define `HASH_ID_STRING` "EXT.SIGNATURE.HASH.AREA"
Begin of hash area info.
- #define `SIG_ID_STRING` "Siemens Mobile Phones Signature File"
Signature File ID.
- #define `SIG_ID_STRING_SGOLD` "Siemens Mobile Phones:SIGNATURE:"
beginning of SGOLD Signature File ID without version
- #define `EXE_KENNUNG` "SAG_JK_WH"
recognition of a Trader executable

8.1.1 Detailed Description

One of two Interface-Headers between HEX-Converter and SWUP-Program.

This header describes all possible IDs that can occur as XBI-Header-Block in an XBI-Header. Since the XBI-Header represents a "struct written to a file", all members of the struct are written down to file in special BIN-Blocks. A normal BIN-Block consists of

ADR | ADR | ADR | LEN | Data Data Data | CHK

This "special" BIN-Block has Address FFFFFFFE and the first Byte of the Data is the so called Block-Identifier (BI). So a XBI-Header-Block looks like this

FF | FF | FE | LEN | BI | Additional Data | CHK

In most cases a Block-Identifier corresponds to one element of the structure.

8.1.2 Define Documentation

8.1.2.1 #define CONVERTER_VERSION_MAJOR HEXUNIX_VERSION_MAJOR

8.1.2.2 #define CONVERTER_VERSION_MINOR HEXUNIX_VERSION_MINOR

8.1.2.3 #define EXE_KENNUNG "SAG_JK_WH"

recognition of a Trader executable

8.1.2.4 #define HASH_ID_STRING "EXT_SIGNATURE_HASH_AREA"

Begin of hash area info.

8.1.2.5 #define HEXUNIX_VERSION_MAJOR 1

Standard : current version of Hex-Bin-Converter, Major-Version.

8.1.2.6 #define HEXUNIX_VERSION_MINOR 19

Standard : current version of Hex-Bin-Converter, Minor part of the version.

8.1.2.7 #define ID_ADDITIONAL_INFO_END 0x52

Additional Information-End.

8.1.2.8 #define ID_ADDITIONAL_INFO_ENTRY 0x51

Additional Information, for ex.

mapping

8.1.2.9 #define ID_ADDITIONAL_INFO_LEN 0x50

Additional Information, Length.

8.1.2.10 #define ID_ALIGNEMENT 0x38

1/2/4-Byte-Alignement in the appended bin-data

8.1.2.11 #define ID_ALL_LIB_INFO_P1 0x2D

(NEW !!!) informations about the lno-project, each lno = 0x50 bytes, here are the first 0x30 bytes

8.1.2.12 #define ID_ALL_LIB_INFO_P2 0x2E

(NEW !!!) the rest 0x20 Bytes information about lno-project

8.1.2.13 #define ID_ASIC_TYPE 0x31

ASIC-Type only used in terms of watchdog-timing.

8.1.2.14 #define ID_COMPRESS_INFO 0x3A

Information about compression and parameters.

8.1.2.15 #define ID_CONSTANTS 0x25

where are the SWUP-constants located

8.1.2.16 #define ID_DEVELOPER 0x16

Name of the developer which generated this mobile-SW.

8.1.2.17 #define ID_END_LOCATER_ENTRY 0x15

End of the locater-table.

8.1.2.18 #define ID_END_OF_HEADER 0x04

End of Header, so that later versions also can recognize the end of the header without problems.

8.1.2.19 #define ID_EPROM_SIZE 0x23

Size of Flash, this mobile-SW was generated for.

8.1.2.20 #define ID_ERASE_INFO 0x30

Parts to be erased.

8.1.2.21 #define ID_ERTEC_SUM 0x24

only for information : the ertec-checksum

8.1.2.22 #define ID_EXT_LIB_1_DATE 0x2B

(VERY OLD !!!) from the time when we distributed only two lno-projects as obj to other sites

8.1.2.23 #define ID_EXT_LIB_2_DATE 0x2C

(VERY OLD !!!) from the time when we distributed only two lno-projects as obj to other sites

8.1.2.24 #define ID_FLASH_WRITE_TYPE 0x32

how is the flash-writing enabled for this SW

8.1.2.25 #define ID_FORMAT_INFO 0x39

Information about the format (bin lck raw ..
)

8.1.2.26 #define ID_HASH_TABLE_INFO 0x56

Info about Hash table length for signed sw update.

8.1.2.27 #define ID_HEXER_VERSION 0x10

Version of HEX-converter with which this file was generated.

8.1.2.28 #define ID_IGNITION_INFO 0x35

how is the ignition detected (old ..
form c25-times)

8.1.2.29 #define ID_INTERNAL_VERSION 0x1E

internal Version-Number, was never used in my opinion ????????????????????

8.1.2.30 #define ID_LANG_CHECK_INFO 0x36

(OLD SPLIT !!!) informations about the Language Group checksums

8.1.2.31 #define ID_LANG_SPLIT_ID 0x37

(OLD SPLIT !!!) the Split ID (unique number do determine if SW and Language Group fit together

8.1.2.32 #define ID_LEAST_SWUP_VERSION 0x11

needed version of SWUP to perform an update of this FILE

8.1.2.33 #define ID_LOCATER_ENTRY 0x14

how was the software "located" from processor-view to flash-view (segment-oriented)

8.1.2.34 #define ID_MOB_SW_LOC_DATE 0x12

Date and Time when this Mobile-SW was "located".

8.1.2.35 #define ID_MOB_SW_RBM_DATE 0x13

Date and Time when the makefiles for this SW were generated.

8.1.2.36 #define ID_NEW_SPLIT_ADD_INFO_TEXT 0x55

New Split Information, Additional Info-Text (example : lg1, lg2 , lg3).

8.1.2.37 #define ID_NEW_SPLIT_INFO 0x53

New Split Information for Generic splitting (Tegic, Language, ...

8.1.2.38 #define ID_NEW_SPLIT_INFO_TEXT 0x54

New Split Information-Text (for ex.

: language/tegic)

8.1.2.39 #define ID_NO_CHECK 0x1F

Entries where we dont want to calc.

the check-sums (was never used, replaced by fixed located values in the flash) ????????????????????

8.1.2.40 #define ID_OLD_DEVELOPER 0x3E

on Incremental-XBI: derived from, Developer Name

8.1.2.41 #define ID_OLD_LOC_DATE 0x3B

on Incremental-XBI: derived from, compiled Date and Time

8.1.2.42 #define ID_OLD_PROJ_NAME 0x3D

on Incremental-XBI: derived from, Project Name

8.1.2.43 #define ID_OLD_PROJECT 0x26

(VERY VERY VERY OLD !!!) Project-ID

8.1.2.44 #define ID_OLD_RBM_DATE 0x3C

on Incremental-XBI: derived from, rb m Date and Time

8.1.2.45 #define ID_PROC_TYPE 0x34

for which Processor Type is this SW compiled

8.1.2.46 #define ID_PROJECT_ID 0x18

I have no idea if this ever was used ...????????????????????

8.1.2.47 #define ID_PROJECT_NAME 0x17

Name of the project, for ex.

epu35 (which is sold as SL45 product name)

8.1.2.48 #define ID_RAM_SIZE 0x33

which RAM Size do we have in the mobile

8.1.2.49 #define ID_RBC_PUT_COUNT 0x27

put-count on compiling (locating) the Software

8.1.2.50 #define ID_RBM_PUT_COUNT 0x1B

put-count on generating the makefiles for this mobile SW

8.1.2.51 #define ID_SPEECHGROUP_ID 0x19

(OLD SPLIT !!!) Language group Identifier, was this ever used ???

8.1.2.52 #define ID_SPEECHGROUP_NAME 0x1A

(OLD SPLIT !!!) Language group Name, for ex

”lg1”

8.1.2.53 #define ID_SW_GENERATION 0x2A

on which SUN-directory this SW was generated, for ex

gen7.z1

8.1.2.54 #define ID_SW_PRODUCT_NAME 0x28

the official Name of the product, for ex
SL45

8.1.2.55 #define ID_SW_STATE 0x1C

SW State, was in my opinion never used (local/ official / ..
) ????????????????????

8.1.2.56 #define ID_SW_TYPE 0x40

What kind of SW is the appended SW : Code, Language, etc..

8.1.2.57 #define ID_SW_VENDOR_NAME 0x29

the official Vendor of the SW, for ex
"Siemens"

8.1.2.58 #define ID_SW_VERSION_NUMBER 0x1D

official GSM-SW-Version-Number <=> SVN

8.1.2.59 #define ID_SWUP_DLL 0x70

holds which DLL the SWUP has to use for the project

8.1.2.60 #define ID_TEGIC_GROUP 0x60

holds which Tegic-Group is in the file

8.1.2.61 #define ID_TEXT 0x20

Text that the developer added during the hex-bin-conversion.

8.1.2.62 #define ID_TEXT_END 0x21

End of the text.

8.1.2.63 #define ID_TEXT_LENGTH 0x22

Length of the text.

8.1.2.64 #define LEAST_LANGSWUP_VERSION_MAJOR 1

the SWUP-Version which is needed at least to update a language-file, major-part of the version

8.1.2.65 #define LEAST_LANGSWUP_VERSION_MINOR 61

the SWUP-Version which is needed at least to update a language-file, minor-part of the version

8.1.2.66 #define LEAST_SWUP_VERSION_MAJOR 1**8.1.2.67 #define LEAST_SWUP_VERSION_MINOR 12****8.1.2.68 #define LEAST_ZIPSWUP_VERSION_MAJOR 1**

the SWUP-Version which is needed at least to update a compressed file, major-part of the version

8.1.2.69 #define LEAST_ZIPSWUP_VERSION_MINOR 72

the SWUP-Version which is needed at least to update a compressed file, minor-part of the version

8.1.2.70 #define MEMO2XBI_VERSION_MAJOR 0

(VERY OLD !!!) Version of HEX-BIN-Converter for Voice-Memos, was last used for the S10 !

8.1.2.71 #define MEMO2XBI_VERSION_MINOR 70

(VERY OLD !!!) Version of HEX-BIN-Converter for Voice-Memos, was last used for the S10 !

8.1.2.72 #define SIG_ID_STRING "Siemens Mobile Phones Signature File"

Signature File ID.

8.1.2.73 #define SIG_ID_STRING_SGOLD "Siemens Mobile Phones:SIGNATURE:"

beginning of SGOLD Signature File ID without version

8.1.2.74 #define XBI_KENNUNGS_STRING "Siemens Mobile Phones Software"

when you are going to change this, you must also change SWUP !

8.1.2.75 #define XBI_KENNUNGS_STRING_SGOLD "Siemens Mobile Phones:SOFTWARE:"

beginning of SGOLD header without version

8.2 dyna_dll.c File Reference

Responsible for the dynamical loading of the project-dependant DLLs, Part of the Main-DLL.

```
#include <windows.h>
#include <io.h>
#include "wswuplib.h"
#include "wsw_proj.h"
#include "wswlibv1.h"
#include "werrenum.h"
#include "wsw_serli.h"
```

Defines

- #define [DLL_FILE_PATTERN](#) "wsw*.dll"

Functions

- char * [InternGiveDllNameToLoad](#) (char *szProject, unsigned char bSvn, BOOL fSynchStation-Necessary)
Looks for the project name into all DLLs given.
- char * [GiveDllNameToLoad](#) (char *szProject, unsigned char bSvn, BOOL fSynchStation-Necessary)
returns the first DLL name among all DLLS stored where the right project is found in one DLL in the working direcorey, and fills all conditions otherwise try to simulate the project from a new unknown project (from registrys)
- BOOL [ScanDirectoryForAvailableProjects](#) (void)
stores all dlls (in pInfoAboutAllDlls) which contain projects in the working directory => the same dll is stored x times the x projects are in this dll
- char * [GetDllPath](#) (char *pszDllName)
get the DLL path from the DLL name (with working directory) (used for new configurable DLL name from Hexer feature)
- BOOL [IsDllInWorkingDirectory](#) (char *pszDllPath)
Verifies if the DLL Name given is located in the SWUP Working directory.

Variables

- unsigned char [szNewUnknownProject](#) [200]
saves the name of a new unknown project
- unsigned char [szDerivedFromKnownProject](#) [200]
saves the name of the project from which the new unknown was derived

- **BOOL *fSimulateKnownProject***

*holds the information if user called *WSwup_PrepateForUpdateWithUnknownProject()* and therefore on loading the DLL it must be checked*

8.2.1 Detailed Description

Responsible for the dynamical loading of the project-dependant DLLs, Part of the Main-DLL.

8.2.2 Define Documentation

8.2.2.1 **#define *DLL_FILE_PATTERN* "wsw*.dll"**

8.2.3 Function Documentation

8.2.3.1 **char* *GetDllPath* (char * *pszDllName*)**

get the DLL path from the DLL name (with working directory) (used for new configurable DLL name from Hexer feature)

Parameters:

void

Return values:

the dll path as string

8.2.3.2 **char* *GiveDllNameToLoad* (char * *szProject*, unsigned char *bSvn*, BOOL *fSynchStationNecessary*)**

returns the first DLL name among all DLLS stored where the right project is found in one DLL in the working direcorey, and fills all conditions otherwise try to simulate the project from a new unknown project (from registry)

Parameters:

szProject project Name to find in any DLL

bSvn Software version

fSynchStationNecessary way of boot (cable or synch station)

Return values:

8.2.3.3 **char* *InternGiveDllNameToLoad* (char * *szProject*, unsigned char *bSvn*, BOOL *fSynchStationNecessary*)**

Looks for the project name into all DLLs given.

Returns the first DLL name (per alphabetical order) when the project name is found in one DLL, if the sync station is the same as asked, and if the Software version is newer than the one in the DLL (otherwise DLL version too new for the SW)

Parameters:

szProject project Name to find in any DLL

bSvn Software version

fSynchStationNecessary way of boot (cable or synch station)

Return values:

the right DLL name, otherwise NULL string if not found

8.2.3.4 BOOL IsDllInWorkingDirectory (char * *pszDllPath*)

Verifies if the DLL Name given is located in the SWUP Working directory.

Returns a boolean

Parameters:

pszDllName

Return values:

TRUE -> DLL is into working directory

FALSE -> no DLL found into working directory or no given DLL name

8.2.3.5 BOOL ScanDirectoryForAvailableProjects (void)

stores all dlls (in *pInfoAboutAllDlls*) which contain projects in the working directory => the same dll is stored x times the x projects are in this dll

Parameters:

void

Return values:

TRUE -> project found in any DLL of working directory

FALSE -> no dll found in working directory

8.2.4 Variable Documentation**8.2.4.1 BOOL *fSimulateKnownProject***

holds the information if user called *WSwup.PrepareForUpdateWithUnknownProject()* and therefore on loading the DLL it must be checked

8.2.4.2 unsigned char *szDerivedFromKnownProject*[200]

saves the name of the project from which the new unknown was derived

8.2.4.3 unsigned char *szNewUnknownProject*[200]

saves the name of a new unknown project

8.3 err_text.c File Reference

```
#include <windows.h>
#include "werrenum.h"
#include "err_text.h"
```

Defines

- #define [ERR_TEXT_VERS_MAJOR](#) 0
- #define [ERR_TEXT_VERS_MINOR](#) 55
- #define [ERR_TEXT_DEVELOPER](#) "Dittmer"

Variables

- [t_enumErrorDescription](#) [ErrorDescription](#) []
- size_t [ErrorDescriptionSize](#) = sizeof ([ErrorDescription](#))
- size_t [t_enumErrorDescriptionSize](#) = sizeof ([t_enumErrorDescription](#))

8.3.1 Define Documentation

8.3.1.1 #define [ERR_TEXT_DEVELOPER](#) "Dittmer"

8.3.1.2 #define [ERR_TEXT_VERS_MAJOR](#) 0

8.3.1.3 #define [ERR_TEXT_VERS_MINOR](#) 55

8.3.2 Variable Documentation

8.3.2.1 [t_enumErrorDescription](#) [ErrorDescription](#) []

8.3.2.2 size_t [ErrorDescriptionSize](#) = sizeof ([ErrorDescription](#))

8.3.2.3 size_t [t_enumErrorDescriptionSize](#) = sizeof ([t_enumErrorDescription](#))

8.4 **err_text.h** File Reference

Data Structures

- struct [t_enumErrorDescription](#)

Variables

- [t_enumErrorDescription](#) [ErrorDescription](#) []
- [size_t](#) [ErrorDescriptionSize](#)
- [size_t](#) [t_enumErrorDescriptionSize](#)

8.4.1 Variable Documentation

8.4.1.1 [t_enumErrorDescription](#) [ErrorDescription](#) []

8.4.1.2 [size_t](#) [ErrorDescriptionSize](#)

8.4.1.3 [size_t](#) [t_enumErrorDescriptionSize](#)

8.5 fiwrap.c File Reference

Contains a wrapper for the former file-handling-library, part of the Main-DLL.

```
#include <windows.h>
#include "wswuplib.h"
#include "wbfbtool.h"
#include "wswlibv1.h"
#include "werrenum.h"
#include "wsw_seri.h"
#include <sys/stat.h>
#include <stdio.h>
#include "fiha.h.h"
```

Defines

- #define [SECURITY_RESERVE](#) 0x4000
- #define [BIN_FORMAT_OVERHEAD_HASH](#) 9

Functions

- T_HashListData * [GetHashData](#) (void)
- T_SignatureItem * [GetSignature](#) (unsigned long ulKeyID)
- unsigned char * [GivePtrToBinData](#) (void)
- BOOL [JumpOverControlBlocks](#) (void)
- void [GiveSwInfoStruct](#) (t_SwInformation *pSwInformation)
- BOOL [IsLibraryInitialised](#) (void)
returns the boolean to know if Library is initialized
- BOOL SWUPLIBDLLLEXIMPORT [WSwup_ReadXbiFile](#) (t_SwInformation *pSwInformation, FileReadProgressCallBack pfnReadProgress, char *pszXbiName)
- void SWUPLIBDLLLEXIMPORT [WSwup_CloseXbiFile](#) (void)
- char SWUPLIBDLLLEXIMPORT * [WSwup_GetLastFileError](#) (void)

Variables

- t_ExtendedInfo SWUPLIBDLLLEXIMPORT [XbiHeaderInfo](#)
- unsigned int SWUPLIBDLLLEXIMPORT [unTotalHeapBytes](#)
- unsigned int SWUPLIBDLLLEXIMPORT [unTotalAmountOfBlocks](#)
- HGLOBAL [hGITSig](#) = NULL
- HGLOBAL [hGIToMem](#) = NULL
- HGLOBAL [hGIToHash](#) = NULL
- char [szFileErrorString](#) [200]

8.5.1 Detailed Description

Contains a wrapper for the former file-handling-library, part of the Main-DLL.

Todo

Doxygen-Documentation !

8.5.2 Define Documentation

8.5.2.1 `#define BIN_FORMAT_OVERHEAD_HASH 9`

8.5.2.2 `#define SECURITY_RESERVE 0x4000`

8.5.3 Function Documentation

8.5.3.1 `T_HashListData* GetHashData (void)`

8.5.3.2 `T_SignatureItem* GetSignature (unsigned long ulKeyID)`

8.5.3.3 `unsigned char* GivePtrToBinData (void)`

8.5.3.4 `void GiveSwInfoStruct (t_SwInformation * pSwInformation)`

8.5.3.5 `BOOL IsLibraryInitialised (void)`

returns the boolean to know if Library is initialized

Parameters:

void

Return values:

TRUE -> Library is initialised

FALSE -> Library is not initialised

8.5.3.6 **BOOL** JumpOverControlBlocks (void)

8.5.3.7 **void** SWUPLIBDLLEXIMPORT WSwup_CloseXbiFile (void)

8.5.3.8 **char** SWUPLIBDLLEXIMPORT* WSwup_GetLastFileError (void)

8.5.3.9 **BOOL** SWUPLIBDLLEXIMPORT WSwup_ReadXbiFile (t_SwInformation *
pSwInformation, FileReadProgressCallback *pfnReadProgress*, char * *pszXbiName*)

8.5.4 Variable Documentation

8.5.4.1 **HGLOBAL** [hGltToHash](#) = NULL

8.5.4.2 **HGLOBAL** [hGltToMem](#) = NULL

8.5.4.3 **HGLOBAL** [hGltToSig](#) = NULL

8.5.4.4 **char** [szFileErrorString](#)[200]

8.5.4.5 **unsigned int** SWUPLIBDLLEXIMPORT [unTotalAmountOfBlocks](#)

8.5.4.6 **unsigned int** SWUPLIBDLLEXIMPORT [unTotalHeapBytes](#)

8.5.4.7 [t_ExtendedInfo](#) SWUPLIBDLLEXIMPORT [XbiHeaderInfo](#)

8.6 filehand.c File Reference

Contains all necessary routines to deal with XBI-Files and complete EXE-Files for traders, part of the MAIN-DLL.

```
#include <io.h>
#include <string.h>
#include <sys/stat.h>
#include <stdio.h>
#include <malloc.h>
#include "boot_hex.h"
#include "wswlibv1.h"
#include "fiha.h.h"
#include "wsw_seri.h"
#include "werrenum.h"
#include "pc_mob.h"
```

Data Structures

- struct [FileInfo](#)
a structure which holds all informations of the WSWUP input File
- struct [t_Block](#)
a structure which holds information of one XBI Header block
- struct [T_HashBinBlock](#)
a structure used to save the hash entries or the signature like bin blocks.

Defines

- #define [KENNUNG_SIZE](#) ((sizeof(exe_endung)-1))
- #define [OLD_EXE_ENDUNG_SIZE](#) (4 + KENNUNG_SIZE)
- #define [EXE_ENDUNG_SIZE](#) (OLD_EXE_ENDUNG_SIZE)
- #define [MAX_HASH_BIN_BLOCK_LENGTH](#) 64
- #define [NO_INIT](#) -1

Functions

- int [GetFileInfo](#) (struct [FileInfo](#) *pstFileDaten, unsigned char *pszFileName)
Get the information about the distribution of contents in file.
- void [PrintErrorMessage](#) (void)
prints on the standard output the error (defined as global) obtained

- unsigned int [ReadUndCheckXbiDaten](#) (int *pnEof, unsigned int *punGesamtHeapBytes, unsigned int *punGesamtBloেকে, char *pszFileName, unsigned char FAR *pfucStartHeap, unsigned int unHeapToUse, unsigned int unReserve, void(*funcPtr)(unsigned long ulAlleBytes, unsigned long ulGelesenBytes))

can be repeatedly called, till the File-end is reached.

- unsigned int [ReadUndCheckAnbootDaten](#) (unsigned long *pulAnzChar, unsigned long *pulAnzBinBloেকে, unsigned char FAR *pfucStartHeap, unsigned int unMaximum, unsigned int unReserve, char *pszFileName)

Read and check File Data in the Heap.

- unsigned int [ReadKennung](#) (t_XbiOrExe *ptWelcheSw, char *pszFilename)

Reads from the .exe File the identification and checks it.

- unsigned int [BelegeFileInit](#) (char *pszFileName)

This function must be called , before functions from the Library are being used.

- int [GetSignatureSize](#) (unsigned long *ulSignatureSize)

Returns bytes of signature in heap.

- unsigned long [GetMotorolaLong](#) (unsigned char *pucBuffer)

Reads out a 32-Bit-Word from the given address in the Buffer.

- unsigned short [GetMotorolaWord](#) (unsigned char *pucBuffer)

Reads out a 16-Bit-Word from the given address in the Buffer.

- unsigned int [GiveExtFileInfo](#) (char *pszFilename, unsigned long *pulAnzBinBytes, unsigned long *pulAnzXbiHeaderBytes, unsigned int *punExeEndungBytes)

Returns information to a Mobile-DataFile.

- unsigned int [ReadAndFormatSigData](#) (T_SignatureData *pSigData, unsigned char *pszSigFileName)

Reads the Signature file and save the data it in binary format.

- unsigned int [ReadUndFormatHashCluster](#) (T_HashListData *pwhereto, unsigned char *pszFileName, unsigned long Offset, unsigned long Length)

Reads the whole hash cluster table and save it in bin format.

- int [ReadXbiHeader](#) (long *plAnzXbiBytes, t_ExtendedInfo *ptrWohin, char *pszBinFileName)

Determine Exe-/Xbi-File Open File Read Header Info Close File.

- void [ReleaseXbiHeaderData](#) (t_ExtendedInfo *ptrToInfo)

Deallocates all the lists into t_ExtendedInfo Info structure.

8.6.1 Detailed Description

Contains all necessary routines to deal with XBI-Files and complete EXE-Files for traders, part of the MAIN-DLL.

Todo

Throw away the old-style traces or re-implement a better trace-functionality in the Main-DLL (and the GUI ???). Make comments doxygen-conform. Decide if comments of exported functions should be done in the interface- header fiha.h OR here, not in both parts, that leads to errors.

8.6.2 Define Documentation

8.6.2.1 `#define EXE_ENDUNG_SIZE ( OLD_EXE_ENDUNG_SIZE )`

8.6.2.2 `#define KENNUNG_SIZE ( (sizeof(exe_endung)-1) )`

8.6.2.3 `#define MAX_HASH_BIN_BLOCK_LENGTH 64`

8.6.2.4 `#define NO_INIT -1`

8.6.2.5 `#define OLD_EXE_ENDUNG_SIZE ( 4 + KENNUNG_SIZE )`

8.6.3 Function Documentation

8.6.3.1 `unsigned int BelegeFileInit (char * pszFileName)`

This function must be called , before functions from the Library are being used.

It allocates the FileInit structure with the Data of the given Exe/Xbi File

Parameters:

pszFileName pointer on FileName

Return values:

- 0:** Structure initialised successfully
- 1:** [GibXbiLen\(\)](#) returns Error
- 2:** Error on stat-call
- 3:** ReadXbiHeader Error
- 4:** [ReadKennung\(\)](#) Error
- 5:** fseek-Error

Called functions : [InitFileDaten\(\)](#) [EraseExtendedInformationBlock\(\)](#) [ReadKennung\(\)](#) [GibXbiLen\(\)](#) [ReadXbiHeader\(\)](#)

8.6.3.2 `int GetFileInfo (struct FileInfo * pstFileDaten, unsigned char * pszFileName)`

Get the information about the distribution of contents in file.

Parameters:

pstFileDaten pointer on [FileInfo](#) structure

pszFileName pointer on FileName

Return values:

- 0:** Data init
- 1:** Error

8.6.3.3 unsigned long GetMotorolaLong (unsigned char * *pucBuffer*)

Reads out a 32-Bit-Word from the given address in the Buffer.

Parameters:

pucBuffer - pointer to a Motorola-Format Buffer

Return values:

the 32-Bit long value

8.6.3.4 unsigned short GetMotorolaWord (unsigned char * *pucBuffer*)

Reads out a 16-Bit-Word from the given address in the Buffer.

Parameters:

pucBuffer - pointer to a Motorola-Format Buffer

Return values:

the 16-Bit value

8.6.3.5 int GetSignatureSize (unsigned long * *ulSignatureSize*)

Returns bytes of signature in heap.

Parameters:

ulSignatureSize - Size in bytes.

Return values:

-1: error ?

otherwise *ulSignatureSize

8.6.3.6 unsigned int GiveExtFileInfo (char * *pszFilename*, unsigned long * *pulAnzBinBytes*, unsigned long * *pulAnzXbiHeaderBytes*, unsigned int * *punExeEndungBytes*)

Returns information to a Mobile-DataFile.

Parameters:

pszFilename pointer on FileName

pulAnzBinBytes number of Bytes of the "real" Mobile-SW without XBI-Header bytes

pulAnzXbiHeaderBytes number of bytes of external Info (XBI-Header)

punExeEndungBytes number of bytes of the EXE-identification

Return values:

0 : successfull

>0: Error

8.6.3.7 void PrintErrorMessage (void)

prints on the standard output the error (defined as global) obtained

Parameters:

void

Return values:

void

8.6.3.8 unsigned int ReadAndFormatSigData (T_SignatureData * pSigData, unsigned char * pszSigFileName)

Reads the Signature file and save the data it in binary format.

Prepared to be send to mobile.

Parameters:

pSigData Pointer to save

pszSigFileName pointer on Signature Pathname

Return values:

0 : successfull

1 : String id error

2 : Open file error

3 : Memory error

4 : Error reading signature entry

8.6.3.9 unsigned int ReadKennung (t_XbiOrExe * ptWelcheSw, char * pszFilename)

Reads from the .exe File the identification and checks it.

Parameters:

ptWelcheSw Sw-type. for ex: which kind of Update (Trader/Developer)

pszFilename pointer on Filename

Return values:

0: No error

1: Error on open File

2: fseek-Error

3: Error on stat-call

8.6.3.10 **unsigned int ReadUndCheckAnbootDaten (unsigned long * *pulAnzChar*, unsigned long * *pulAnzBinBloেকে*, unsigned char FAR * *pfucStartHeap*, unsigned int *unMaximum*, unsigned int *unReserve*, char * *pszFileName*)**

Read and check File Data in the Heap.

Maximum maxheap-Reserve bytes allocated, in order to append information (for ex. EOT-Block) at the End when the last Heap-filling occurs

Parameters:

pulAnzChar Number of checked characters
pulAnzBinBloেকে Number of bin blocks
pfucStartHeap Pointer on allocated Memory
unMaximum Maximum standing available Memory
unReserve Memory-Rest, which must no be written
pszFileName Pointer on Filename

Return values:

0: Data in Heap ok.
1: Error open File/File not found
2: fread() - Error
3: Error in [CheckXbiBlock\(\)](#)

Called functions : [CheckXbiBlock\(\)](#)

8.6.3.11 **unsigned int ReadUndCheckXbiDaten (int * *pnEof*, unsigned int * *punGesamtHeapBytes*, unsigned int * *punGesamtBloেকে*, char * *pszFileName*, unsigned char FAR * *pfucStartHeap*, unsigned int *unHeapToUse*, unsigned int *unReserve*, void(**funcPtr*)(unsigned long *ulAlleBytes*, unsigned long *ulGelesenBytes*))**

can be repeatedly called, till the File-end is reached.

Read and check File Data in the Heap. Maximum maxheap-Reserve bytes allocated, in order to append information (for ex. EOT-Block) at the End when the last Heap-filling occurs

Parameters:

pnEof File End reached on reading
punGesamtHeapBytes Number of valid bits in Heap
punGesamtBloেকে Number of valid Blocks
pszFileName Filename
pfucStartHeap Pointer on Heap-Start
unHeapToUse maximum Heap to read
unReserve place, which must not be allocated in Heap
(funcPtr*)** function pointer

Return values:

0: Data in Heap ok.
1: Error in [BelegeFileInit\(\)](#)

- 2: fseek-Error
- 3: Error open File/File not found
- 4: fread-Error
- 5: Error in [CheckXbiBlock\(\)](#)
- 6: Error on Heap-Check
- 7: [BelegeFileInit\(\)](#) not called

Called functions : [CheckXbiBlock\(\)](#)

8.6.3.12 unsigned int ReadUndFormatHashCluster (T_HashListData * *pwhere*to, unsigned char * *pszFileName*, unsigned long *Offset*, unsigned long *Length*)

Reads the whole hash cluster table and save it in bin format.

Prepared to be send to mobile.

Parameters:

- pwhere*to Pointer to save place
- pszFileName* pointer on File name
- Offset* Offset from the begining of the file where hash information starts
- Length* Hash Table length

Returns:

Error level

Return values:

- 0 : No error
- 1 : String id error. Not a Siemens hash
- 2 : Read error
- 3 : seek-Error

8.6.3.13 int ReadXbiHeader (long * *plAnzXbiBytes*, [t_ExtendedInfo](#) * *ptrWohin*, char * *pszBinFileName*)

Determine Exe-/Xbi-File Open File Read Header Info Close File.

Parameters:

- plAnzXbiBytes* Number of XBI-Header-Bytes
- ptrWohin* t_ExtendedInfo-Pointer on InfoStructure
- pszBinFileName* Pointer on FileName

Return values:

- 1: CHK-Error on reading Header
- 0 : XBI-Header successfully read
- 1 : Error on opening File
- 2 : XBI-Length 0 in the EXE-Identification

- 3 : stat-error
- 4 : fseek-Error
- 5 : SwitchCase-Error
- 6 : Error on ReadKennung (read identification)
- 7 : Error on List Management

Called function : SimpleVerschluesselung

8.6.3.14 void ReleaseXbiHeaderData ([t_ExtendedInfo](#) * *ptrToInfo*)

Deallocates all the lists into [t_ExtendedInfo](#) Info structure.

Parameters:

ptrToInfo [t_ExtendedInfo](#) Info structure

Returns:

void

8.7 pc_mob.h File Reference

Interface between PC-Part and Mobile-Part of SWUP, Part of the MAIN-DLL.

Defines

- #define [ACK](#) 0x06
- #define [NAK](#) 0x15
- #define [ACKNOWLEDGE_A5](#) 0xA5
- #define [NAK_HOLD](#) 0x9B
- #define [STEUER](#) 0xFF
- #define [ACK_B0](#) 0xB0
- #define [ACK_B1](#) 0xB1
- #define [REC_ERROR](#) 2
- #define [AUSZEIT](#) -1
- #define [BUFFER_OVERFLOW](#) -2
- #define [BLOCK_TIMEOUT_VAL](#) 200
- #define [BYTE_TIMEOUT_VAL](#) 30
- #define [MI_STATISTIK_DATA](#) 0x00
- #define [MI_REQUEST_TEST_SW](#) 0x01
- #define [MI_REQUEST_MASTER_SW](#) 0x02
- #define [MI_END_OF_TRANSMISSION](#) 0x04
- #define [MI_SELECT_MOBILE_MODE](#) 0x10
- #define [MI_MOBILE_ERROR](#) 0x18
- #define [MI_PROMMER_VERSION](#) 0x1D
- #define [MI_READING_OUT_MASTER](#) 0x1E
- #define [MI_FLASH_CODE](#) 0x1F
- #define [MI_TEXTSTRING](#) 0x20
- #define [MI_GIVE_FLASH_SIZE](#) 0x21
- #define [MI_MOBILE_FLASH_SIZE](#) 0x22
- #define [MI_NEW_FLASH_CODE](#) 0x30
- #define [MI_ACK_END_OF_TRANSMISSION](#) 0x31
- #define [MI_GIVE_FLASH_CODE](#) 0x32
- #define [MI_ACK_GIVE_FLASH_CODE](#) 0x33
- #define [MI_SIMULATE_FLASH](#) 0x34
- #define [MI_GIVE_EXT_FLASH_INFO](#) 0x35
- #define [MI_ACK_GIVE_EXT_FLASH_INFO](#) 0x36
- #define [MI_NEW_BYTE_PROG_ERR](#) 0x40
- #define [MI_FLASH_CHECK_OK](#) 0x41
- #define [MI_MEMORY_TESTS](#) (0x47)
- #define [MI_ERASE_FLASH](#) 0x50
- #define [MI_ACK_ERASE_FLASH](#) 0x51
- #define [MI_TEST_AREA](#) 0x53
- #define [MI_TEST_AREA_ACK](#) 0x54
- #define [MI_ERASE_FLASH_AREA](#) 0x55
- #define [MI_ERASE_FLASH_AREA_ACK](#) 0x56
- #define [MI_ENABLE_FLASHWRITE_BLNR](#) 0x57
- #define [MI_ENABLE_FLASHWRITE_ALL](#) 0x58
- #define [MI_ERASE_FLASH_UPGRADE_CONCEPT](#) 0x59

- #define [MI_ERASE_FLASH_UPGRADE_CONCEPT_RUNNING](#) 0x5A
- #define [MI_ERASE_FLASH_UPGRADE_CONCEPT_ACK](#) 0x5B
- #define [MI_FLASH_ERASE_ADDRESS](#) 0x5C
- #define [MI_CONTROL_COMMAND](#) 0x5D
- #define [MI_EXT_CONTROL_COMMAND](#) 0x5E
- #define [MI_SW_SIGNATURE_NEGOTIATION](#) 0x60
- #define [MI_SW_SIGNATURE_NEGOTIATION_ANSWER](#) 0x61
- #define [MI_DUMMY](#) 0x66
- #define [MI_STATISTIK_DATA_REST](#) 0x70
- #define [MI_STATISTIK_DATA_REST_ANSWER](#) 0x71
- #define [MI_COMPRESSION_ON](#) 0x80
- #define [MI_COMPRESSION_ON_ANSWER](#) 0x81
- #define [MI_COMPRESSION_OFF](#) 0x82
- #define [MI_COMPRESSION_OFF_ANSWER](#) 0x83
- #define [MI_CURR_BAUDRATE](#) 0x90
- #define [MI_CURR_BAUDRATE_ACK](#) 0x91
- #define [MI_MAPPING_SEG_INFO](#) 0x92
- #define [MI_MAPPING_SEG_END](#) 0x93
- #define [MI_ADDITIONAL_MAP_INFO_LEN](#) 0x94
- #define [MI_ADDITIONAL_MAP_INFO](#) 0x95
- #define [MI_ADDITIONAL_MAP_INFO_END](#) 0x96
- #define [MI_STARTUP_INFO](#) 0x97
- #define [MI_FINISH_BOTBLOCK_HANDLING](#) 0x98
- #define [MI_ALIGNMENT](#) 0xA0
- #define [MI_GET_SERIAL_NUMBER](#) 0xB0
- #define [MI_GET_SERIAL_NUMBER_ANSWER](#) 0xB1
- #define [MI_GET_32BIT_NUMBER](#) 0xB2
- #define [MI_GET_32BIT_NUMBER_ANSWER](#) 0xB3
- #define [MI_CALC_SPLIT_INFO](#) 0xB8
- #define [MI_CALC_SPLIT_INFO_ANSWER](#) 0xB9
- #define [MI_GET_UPDATE_SW_VERSION](#) 0xC0
- #define [MI_UPDATE_SW_VERSION_ANSWER](#) 0xC1
- #define [MI_GET_MOBILE_SW_VERSION](#) 0xC2
- #define [MI_MOBILE_SW_VERSION_ANSWER](#) 0xC3
- #define [MI_GET_MEMORY_CONTENT](#) 0xC8
- #define [MI_GET_MEMORY_CONTENT_ANSWER](#) 0xC9
- #define [MI_GET_MEMORY_CONTENT_END](#) 0xCA
- #define [MI_CALC_CHECKSUM](#) 0xCC
- #define [MI_CALC_CHECKSUM_ANSWER](#) 0xCD
- #define [MI_LANGGROUP_INFO](#) 0xCE
- #define [MI_LANGGROUP_INFO_ANSWER](#) 0xCF
- #define [MI_PC_TIMEOUT_SET](#) 0xD0
- #define [MI_PC_TIMEOUT_DEFAULT](#) 0xD1
- #define [MI_PC_BYTE_TIMEOUT_SET](#) 0xD2
- #define [MI_MMISWUP_NORM_PROGRAM](#) 0xE0
- #define [MI_MMISWUP_NORM_PROGRAM_ACK](#) 0xE1
- #define [MI_MMISWUP_RESET_EPROM](#) 0xE2
- #define [MI_MMISWUP_TEST_EPROM](#) 0xE3
- #define [MI_MMISWUP_MASTER_SEEK](#) 0xE4
- #define [MI_MMISWUP_MASTER_SEEK_ACK](#) 0xE5

- #define [MI_MMISWUP_SPEZ_PROGRAM](#) 0xE6
- #define [MI_MMISWUP_SPEZ_PROGRAM_ACK](#) 0xE7
- #define [MI_MMISWUP_TEST_EPROM_ACK](#) 0xE8
- #define [MI_SWITCH_OFF_MOBILE](#) 0xF0
- #define [EI_WHOLE_FLASH](#) 0x00
- #define [EI_CODE_PART](#) 0x10
- #define [EI_EEPROM_SIMULATION](#) 0x20
- #define [EI_EEPROM_SIMULATION_BLOCK](#) 0x21
- #define [EI_VOICE_MEMO](#) 0x30
- #define [EI_VOICE_MEMO_BLOCK](#) 0x31
- #define [EI_VOICE_DIAL](#) 0x40
- #define [EI_VOICE_DIAL_BLOCK](#) 0x41
- #define [EI_FLASH_DEPENDENT_BLOCK](#) 0x50
- #define [MFI_GET_ANZ_FLASHES](#) 0x00
- #define [MFI_GET_FLASH_CODE](#) 0x10
- #define [ERR_RAM_TEST_FF](#) 0x11
- #define [ERR_RAM_TEST_MUSTER](#) 0x12
- #define [ERR_RAM_TEST_00](#) 0x13
- #define [ERR_RAM_TEST_UNKNOWN](#) 0x14
- #define [ERR_NO_EPROM](#) 0x30
- #define [ERR_BLOCK_ERASE](#) 0x60
- #define [ERR_BYTE_PROG](#) 0x70
- #define [ERR_CHECKSUM](#) 0x86
- #define [ERR_UNKNOWN_FLASH](#) 0x87
- #define [ERR_FLASH_BLOCK_ADR](#) 0x88
- #define [ERR_HASHING](#) 0x89
- #define [ERR_SW_REJECTED](#) 0x8A
- #define [ERR_FUNC_NOT_IMPLEMENTED](#) 0xFF
- #define [PH_ID_SIGNATURE_QUERY](#) 0x01
- #define [PH_ID_SIGNUM_SELECTION](#) 0x02
- #define [PH_ID_SIGNUM_TRANSMISSION](#) 0x03
- #define [PH_ID_SIGNUM_TRANSMISSION_END](#) 0x04
- #define [PH_ID_HASHLIST_TRANSMISSION](#) 0x05
- #define [PH_ID_HASHLIST_TRANSMISSION_END](#) 0x06
- #define [PH_ID_START_HASHING](#) 0x07
- #define [SIGACK_NOT_REQUESTED](#) 0x01
- #define [SIGACK_REQUESTED_AND_HASH_ACCEPTED](#) 0x02
- #define [SIGACK_REQUESTED_HASH_NOT_SUPPORTED](#) 0x03
- #define [SIGACK_SIGNATURE_ACCEPTED](#) 0x04
- #define [SIGACK_SIGNATURE_REJECTED](#) 0x05
- #define [SIGACK_HASHLIST_ACCEPTED](#) 0x06
- #define [SIGACK_HASHLIST_REJECTED](#) 0x07
- #define [SIGACK_VERIFICATION_COMPLETED](#) 0x08
- #define [SIGACK_PROTOCOL_ERROR](#) 0x09
- #define [SIGACK_DATABOUNDARY_VIOLATION](#) 0x0A
- #define [SIGACK_WRONGBOOT_CORE](#) 0x0B
- #define [IPC_UPDATE_DATA](#) 0x29
- #define [IPC_UPDATE_CMD](#) 0x2A
- #define [M_MODE_SELECTION](#) 0x40
- #define [M_PC_CONNECTED](#) 0x23

- #define `M_MODEM_UPDATE_STARTED` 0x90

PC-> A : Sent when modem-over-usb update is started.

- #define `M_MODEM_UPDATE_RUNNING` 0x91

PC-> A : Sent during modem-over-usb update.

- #define `M_MODEM_UPDATE_FINISHED` 0x92

PC-> A : Sent when modem-over-usb update is finished.

Enumerations

- enum `t_Flash` { `enOk` = 1, `enKnownButBad` = 2, `enUnknown` = 3, `enSimulated` = 4 }

8.7.1 Detailed Description

Interface between PC-Part and Mobile-Part of SWUP, Part of the MAIN-DLL.

Versions must be the same on both development-environments, means `boot_xxx` for the Mobile-Part in continuous and the PC-SWUP-Tool. All messages with Message-Identifiers described in here can occur on the serial line between Mobile and PC.

Todo

- Comment all the Message-Identifiers ! -Make sure it's the same version on Mobile and PC-Side !

8.7.2 Define Documentation

8.7.2.1 **#define ACK 0x06**

8.7.2.2 **#define ACK_B0 0xB0**

8.7.2.3 **#define ACK_B1 0xB1**

8.7.2.4 **#define ACKNOWLEDGE_A5 0xA5**

8.7.2.5 **#define AUSZEIT -1**

8.7.2.6 **#define BLOCK_TIMEOUT_VAL 200**

8.7.2.7 **#define BUFFER_OVERFLOW -2**

8.7.2.8 **#define BYTE_TIMEOUT_VAL 30**

8.7.2.9 **#define EI_CODE_PART 0x10**

8.7.2.10 **#define EI_EEPROM_SIMULATION 0x20**

8.7.2.11 **#define EI_EEPROM_SIMULATION_BLOCK 0x21**

8.7.2.12 **#define EI_FLASH_DEPENDENT_BLOCK 0x50**

8.7.2.13 **#define EI_VOICE_DIAL 0x40**

8.7.2.14 **#define EI_VOICE_DIAL_BLOCK 0x41**

8.7.2.15 **#define EI_VOICE_MEMO 0x30**

8.7.2.16 **#define EI_VOICE_MEMO_BLOCK 0x31**

8.7.2.17 **#define EI_WHOLE_FLASH 0x00**

8.7.2.18 **#define ERR_BLOCK_ERASE 0x60**

8.7.2.19 **#define ERR_BYTE_PROG 0x70**

8.7.2.20 **#define ERR_CHECKSUM 0x86**

8.7.2.21 **#define ERR_FLASH_BLOCK_ADR 0x88**

8.7.2.22 **#define ERR_FUNC_NOT_IMPLEMENTED 0xFF**

8.7.2.23 **#define ERR_HASHING 0x89**

8.7.2.24 **#define ERR_NO_EPROM 0x30**

8.7.2.25 **#define ERR_RAM_TEST_00 0x13**

8.7.2.26 **#define ERR_RAM_TEST_FF 0x11**

8.7.2.27 **#define ERR_RAM_TEST_MUSTER 0x12**

8.7.2.28 **#define ERR_RAM_TEST_UNKNOWN 0x14**

8.7.2.29 **#define ERR_SW_REJECTED 0x8A**

8.7.2.30 **#define ERR_UNKNOWN_FLASH 0x87**

No payload

8.7.2.35 #define M_MODEM_UPDATE_RUNNING 0x91

PC-> A : Sent during modem-over-usb update.

1. byte percentage payload

8.7.2.36 #define M_MODEM_UPDATE_STARTED 0x90

PC-> A : Sent when modem-over-usb update is started.

No payload

8.7.2.37 `#define M_PC_CONNECTED 0x23`

8.7.2.38 `#define MFI_GET_ANZ_FLASHES 0x00`

8.7.2.39 `#define MFI_GET_FLASH_CODE 0x10`

8.7.2.40 `#define MI_ACK_END_OF_TRANSMISSION 0x31`

8.7.2.41 `#define MI_ACK_ERASE_FLASH 0x51`

8.7.2.42 `#define MI_ACK_GIVE_EXT_FLASH_INFO 0x36`

8.7.2.43 `#define MI_ACK_GIVE_FLASH_CODE 0x33`

8.7.2.44 `#define MI_ADDITIONAL_MAP_INFO 0x95`

8.7.2.45 `#define MI_ADDITIONAL_MAP_INFO_END 0x96`

8.7.2.46 `#define MI_ADDITIONAL_MAP_INFO_LEN 0x94`

8.7.2.47 `#define MI_ALIGNMENT 0xA0`

8.7.2.48 `#define MI_CALC_CHECKSUM 0xCC`

8.7.2.49 `#define MI_CALC_CHECKSUM_ANSWER 0xCD`

8.7.2.50 `#define MI_CALC_SPLIT_INFO 0xB8`

8.7.2.51 `#define MI_CALC_SPLIT_INFO_ANSWER 0xB9`

8.7.2.52 `#define MI_COMPRESSION_OFF 0x82`

8.7.2.53 `#define MI_COMPRESSION_OFF_ANSWER 0x83`

8.7.2.54 `#define MI_COMPRESSION_ON 0x80`

8.7.2.55 `#define MI_COMPRESSION_ON_ANSWER 0x81`

8.7.2.56 `#define MI_CONTROL_COMMAND 0x5D`

8.7.2.57 `#define MI_CURR_BAUDRATE 0x90`

8.7.2.58 `#define MI_CURR_BAUDRATE_ACK 0x91`

8.7.2.59 `#define MI_DUMMY 0x66`

8.7.2.60 `#define MI_ENABLE_FLASHWRITE_ALL 0x58`

8.7.2.61 `#define MI_ENABLE_FLASHWRITE_BLNR 0x57`

8.7.2.62 `#define MI_END_OF_TRANSMISSION 0x04`

8.7.2.63 `#define MI_ERASE_FLASH 0x50`

8.7.2.64 `#define MI_ERASE_FLASH_AREA 0x55`

8.7.2.65 `#define MI_ERASE_FLASH_AREA_ACK 0x56`

8.7.2.66 `#define MI_ERASE_FLASH_UPGRADE_CONCEPT 0x59`

8.7.2.67 `#define MI_ERASE_FLASH_UPGRADE_CONCEPT_ACK 0x5B`

enKnownButBad**enUnknown****enSimulated**

8.8 swupwork.c File Reference

Contains the "working" routine for swup, here is done "everything", Part of the MAIN-DLL.

```
#include <windows.h>
#include <stdlib.h>
#include <string.h>
#include <process.h>
#include "wswuplib.h"
#include "wsw_ser1.h"
#include "wswserl3.h"
#include "wsw_err.h"
#include "wswlibv1.h"
#include "wsw_proj.h"
#include "pc_mob.h"
#include "werrenum.h"
#include "wswbcbdef.h"
#include "IBoot.h"
```

Data Structures

- struct [tag_PINstruct](#)
a structure which holds PIN information
- struct [tag_ThreadInfo](#)
a structure which holds the information of Threads (COM, speed, Update Nr)

Defines

- #define [SWUPLIBDLL_VERS_MAJOR](#) 2
- #define [SWUPLIBDLL_VERS_MINOR](#) 41
- #define [SWUPLIBDLL_DEVELOPER](#) "Schmittu"
- #define [MAX_SIZE_OF_PIN](#) SIZE_OF_PIN
- #define [SEGM_SIZE](#) 0x010000uL
- #define [SIZE_IN_BYTES](#)(a) ((unsigned long)a * 2 * SEGM_SIZE)

Typedefs

- typedef [tag_ThreadInfo](#) [t_ThreadInfo](#)
a structure which holds the information of Threads (COM, speed, Update Nr)
- typedef [tag_PINstruct](#) [t_PINStruct](#)
a structure which holds PIN information

- typedef IBOOT_API BOOL(* [PFN_USBOMAPSWUP_GETDEVICE](#))(WORD wUsbPort, BOOL fForceSavedPort)
- typedef IBOOT_API BOOL(* [PFN_USBOMAPSWUP_RELEASEDEVICE](#))(WORD wUsbPort)

Functions

- BOOL [InitMobileErrorHandler](#) (void)
- void [SoftwareUpdateThread](#) (PVOID pvoid)
Updates the Software Thread (But can not tell more because too long => unreadable !!!).
- BOOL [BootCoreStartUpdateSW_LastHope](#) (t_StartUpdateCoreReturnValues *enRetVal, WORD wUpdateNr, t_SwupCom CurrCom, unsigned long ulSpeed, BootstrapProgressCallBack pfProgress)
This function always indicate that only last hope behaviour is possible (if "boot gut jmp" in mobile not set and BSL is not used bootcore activate update after power on) Last hope behaviour is a replacement for device depending callback function if no project can be determined and update with customer update is requested, or if "complete swup heap library" is used, this is the for project depending dll.
- void [InitGlobalMemory](#) (void)
Does Memory Initialization, i.e for : BootCoreStartUpdateSW (last hope when no BSL and no "boot gut jmp") PIN erasefunctions Functionality.
- BOOL [IsLibraryInitialised](#) (void)
returns the boolean to know if Library is initialized
- BOOL SWUPLIBDLLLEXIMPORT [WSwup_InitLibrary](#) (WORD wMajorNumber, WORD wMinorNumber)
This function must be called by every user of the library once on starting.
- char * [GiveSwupLibDate](#) (void)
gives the compilation date of the Swup Library
- BOOL [IsSwupLibVersionNewEnough](#) (unsigned short nNeededVersion)
compares the current Swup Library version with the one needed if needed > current, return FALSE otherwise TRUE
- void [SetErrorForAllComPorts](#) (DWORD dwError)
*Trace and store error code when Library and Executable have different Interface-Versions *.*
- BOOL SWUPLIBDLLLEXIMPORT [WSwup_GiveVersionInformation](#) (t_VersionInformation *p-Information, t_InfoSelector WhichInfo)
Returns Version Information about the different Parts of WSWUPLIB.
- void SWUPLIBDLLLEXIMPORT [WSwup_PrepareForUpdateWithSynchStation](#) (BOOL fSynchStationPresent)
Prepares for performing the update via Synch-Station.
- void SWUPLIBDLLLEXIMPORT [WSwup_PrepareForUpdateWithUnknownProject](#) (char *pszUnknownProjectName, char *pszDerivedFromKnownProjectName)
Prepares the WSWUP-Library for updating an "unknown" project.

- **BOOL** SWUPLIBDLLEXIMPORT [WSwup_SetBootPIN](#) (WORD wUpdateNr, unsigned int unPINSize, unsigned char *pucBootPIN)
Save a PIN for an Update, This PIN must be provided by data stored in PICS and is sended during BSL connect attempt.
- **BOOL** [InitProjectDependentBootstrapDll](#) (char *szProject, unsigned char bSvn)
initialises the bootstrap Library (DLL) dependent from the current Project depends on the 3 modes : COMPLETE_SWUP_HEAP_LIBRARY COMPLETE_SERVICE_SWUP or other solution
- **BOOL** SWUPLIBDLLEXIMPORT [WSwup_ReloadUpdateDll](#) (void)
Reloads the DLL that is responsible for performing the Update.
- **void** SWUPLIBDLLEXIMPORT [WSwup_InstallSoftwareInfoCallBackFunction](#) (MobileSwUpdateInfoCallBack pfnMobileSwUpdateInfoCallBack, t_InfoCallBackTime WhichCallbackTime)
Installs the MobileSwUpdateInfoCallBack-Function.
- **BOOL** [CheckIfThisDiffSwUpdateIsOk](#) (t_SwupCom WhichCom, t_ExtendedInfo *pXbiInfo, unsigned char *pSwInfoBuffer)
Checks if the OLD XBI dates from XBI Header and Mobile SW are the same.
- **void** SWUPLIBDLLEXIMPORT [WSwup_InstallSoftwareUpdateCallBackFunctions](#) (BootstrapProgressCallBack pfnBootstrapProgress, UpdateSwTransmissionStartCallBack pfnUpdateSwTransmissionStart, UpdateSwTransmissionProgressCallBack pfnUpdateSwTransmissionProgress, FlashTypeCallBack pfnFlashType, EraseProgressCallBack pfnEraseProgress, MobileSwTransmissionStartCallBack pfnMobileSwTransmissionStart, MobileSwTransmissionProgressCallBack pfnMobileSwTransmissionProgress, MobileSwUpdateSuccessCallBack pfnMobileSwUpdateSuccess, MobileSwUpdateErrorCallBack pfnMobileSwUpdateError, MobileSwUpdateConsistenceProblemCallBack pfnMobileConsistenceProblem)
Installs all the needed callbackfunctions for the whole software-update.
- **BOOL** SWUPLIBDLLEXIMPORT [WSwup_CheckComPortAndSpeed](#) (t_SwupCom WhichCom, DWORD dwBaudrate)
This function only tries to open the selected comport with the given baudrate.
- **BOOL** SWUPLIBDLLEXIMPORT [WSwup_SetUsbParams](#) (WORD wUpdateNr, t_SwupCom WhichUsb, BOOL Force)
Sets parameters for update according Sipc protocol.
- **void** SWUPLIBDLLEXIMPORT [WSwup_ResetUsbParams](#) (WORD wUpdateNr)
Sets parameters for update according Sipc protocol.
- **BOOL** SWUPLIBDLLEXIMPORT [WSwup_PerformSoftwareUpdate](#) (WORD wUpdateNr, t_SwupCom WhichCom, unsigned long ulSpeed)
Performs the Software-Update on the given ComPort with the given Baudrate.
- **void** [ConvertStreamToSwInformationStruct](#) (t_SwInformation *pInfoStruct, unsigned char *pabData)
Copies information from string to t_SwInformatio fields.
- **unsigned int** [IsThisADownGrade](#) (char *pszUpdateSwDate, char *pszMobileSwDate)

Compares dates from Mobile SW and XBI Header.

- **BOOL** **SWUPLIBDLLEXIMPORT** **WSwup_SetUpdateConceptMode** (t_UpdateMode Which-Mode)
Tells the library how to perform the Software-Update.
- **t_UpdateMode** **GiveGeneralUpdateMode** (void)
Gives the value which defines the library how to perform the Software-Update (between old behaviour and new SW update, or auto-detect).
- **void** **NewUpdateFailed_UseBsl** (WORD wUpdateNr)
Sets the boolean array for Update Failed with number UpdateNb to TRUE.
- **void** **NewUpdateSucceeded** (WORD wUpdateNr)
Sets the boolean array for Update Failed with number UpdateNb to FALSE.
- **void** **SetPreCheckInfos** (WORD wUpdateNr, t_PreCheckInfos info)
Fills the FlashErasure Table for Update number with info to erase Flash.
- **t_UpdateMode** **GiveUpdateModeForThisUpdate** (WORD wUpdateNr)
*Gives the value which defines the library how to perform the Software-Update *.*
- **void** **FreeList** (void *pHead)
deallocates a list of any type
- **BOOL** **Invalidate_BGJ** (t_SwupCom ComPort, t_ProcType ProcInfo)
Invalidates Boot Good Jump in Flash (when a new Part (for ex.
- **void** **CloseDevice** (WORD wUpdateNr, t_SwupCom device)
closes device, either ComPort or Usb

Variables

- **t_UsbInfo** **UsbSettings** []
- **t_ExtendedInfo** **XbiHeaderInfo**
- **unsigned int** **unTotalHeapBytes**
- **MobileSwUpdateInfoCallBack** **MobileSwInfoFunction** = NULL
tells caller about SW-Version etc in Mobile
- **t_InfoCallBackTime** **GlobalWhichInfoCallBackTime**
- **BOOL** **fCallbacksInstalled** = FALSE
are all the above callbacks installed by the caller ?
- **t_ThreadInfo** **ThreadInfo** [AMOUNT_OF_UPDATES+1]
some information must be held "global"
- **HINSTANCE** **hUpdateLib** = NULL
Handle to the project-dependent UPDATE-Library.

- char [szNewUnknownProject](#) [200]
saves the name of a new unknown project
- char [szDerivedFromKnownProject](#) [200]
saves the name of the project from which the new unknown was derived
- BOOL [fSimulateKnownProject](#) = FALSE
holds the information if user called [WSwup_PrepareForUpdateWithUnknownProject\(\)](#) and therefore on loading the DLL it must be checked
- PFN_GiveUpdateSwPtrAndLen [GiveUpdatePtrAndLenFunc](#)
for retrieving update-sw from inside DLL
- PFN_GiveUpdateMobileSwVersion [GiveUpdateMobileSwVersion](#)
Version-Information Mobile-Update-SW contained in Update-DLL.
- PFN_GiveUpdateDllPcSwVersion [GiveUpdateDllPcSwVersion](#)
Version-Information Update-DLL.
- PFN_BootStrapLoaderWithPIN [BootStrapLoaderFunc](#)
The Bootstrap-Loader itself.
- PFN_StartUpdateSoftwareViaBSL [BootCoreStartUpdateSWFunc](#)
- BOOL [fLibraryIsInitialised](#)
Library is not initialized, must be initialized [WSwup_InitLibrary\(\)](#).
- unsigned char * [abUpdateSwData](#)
pointer to the Update-SW that is transmitted to Mobile
- unsigned long [ulAmountBytesUpdateData](#)
len of Update-SW
- t_UpdateMode [GeneralUpdateMode](#) = enForceBSLBehaviour
- BOOL [NewUpdateFailed](#) [AMOUNT_OF_UPDATES]
- t_UpdateMode [UpdateModes](#) [AMOUNT_OF_UPDATES]
- PFN_USBOMAPSWUP_GETDEVICE [pfnUsbOmapSwup_GetDevice](#)
- PFN_USBOMAPSWUP_RELEASEDEVICE [pfnUsbOmapSwup_ReleaseDevice](#)

8.8.1 Detailed Description

Contains the "working" routine for swup, here is done "everything", Part of the MAIN-DLL.

Todo

Is nearly completely undocumented ...

8.8.2 Define Documentation

8.8.2.1 `#define MAX_SIZE_OF_PIN SIZE_OF_PIN`

8.8.2.2 `#define SEGM_SIZE 0x010000uL`

8.8.2.3 `#define SIZE_IN_BYTES(a) ((unsigned long)a * 2 * SEGM_SIZE)`

8.8.2.4 `#define SWUPLIBDLL_DEVELOPER "Schmittu"`

8.8.2.5 `#define SWUPLIBDLL_VERS_MAJOR 2`

8.8.2.6 `#define SWUPLIBDLL_VERS_MINOR 41`

8.8.3 Typedef Documentation

8.8.3.1 `typedef IBOOT_API BOOL(* PFN_USBOMAPSWUP_GETDEVICE)( WORD wUsbPort, BOOL fForceSavedPort )`

8.8.3.2 `typedef IBOOT_API BOOL(* PFN_USBOMAPSWUP_RELEASEDEVICE)( WORD wUsbPort)`

8.8.3.3 `typedef struct tag_PINstruct t_PINStruct`

a structure which holds PIN information

8.8.3.4 `typedef struct tag_ThreadInfo t_ThreadInfo`

a structure which holds the information of Threads (COM, speed, Update Nr)

8.8.4 Function Documentation

8.8.4.1 **BOOL BootCoreStartUpdateSW_LastHope (t_StartUpdateCoreReturnValues * enRetVal, WORD wUpdateNr, t_SwupCom CurrCom, unsigned long ulSpeed, BootstrapProgressCallBack pfProgress)**

This function always indicate that only last hope behaviour is possible (if "boot gut jmp" in mobile not set and BSL is not used bootcore activate update after power on) Last hope behaviour is a replacement for device depending callback function if no project can be determined and update with customer update is requested, or if "complete swup heap library" is used, this is the for project depending dll.

Parameters:

enRetVal Core return value
wUpdateNr Thread Update number (for each swup)
CurrCom one of all possible Comports
ulSpeed Thread speed
pfProgress variable used for progress during bootstrap

Return values:

TRUE -> mobile was OFF
FALSE -> mobile was ON, or can not open the given Com-Port

8.8.4.2 **BOOL CheckIfThisDiffSwUpdateIsOk (t_SwupCom WhichCom, t_ExtendedInfo * pXbiInfo, unsigned char * pSwInfoBuffer)**

Checks if the OLD XBI dates from XBI Header and Mobile SW are the same.

Parameters:

WhichCom one of all possible Comports

pXbiInfo pointer on XBI-Header structure

pSwInfoBuffer string which contains the old location date (of OLD XBI SW)

Return values:

TRUE Everything is okay

FALSE Some date is wrong

8.8.4.3 **void CloseDevice (WORD wUpdateNr, t_SwupCom device)**

closes device, either ComPort or Usb

Parameters:

device in case of serial update ComPort in case of Usb update virtual UsbPort (=updateNr)

Return values:

void

8.8.4.4 **void ConvertStreamToSwInformationStruct (t_SwInformation * pInfoStruct, unsigned char * pabData)**

Copies information from string to t_SwInformatio fields.

Parameters:

pInfoStruct Structure with Info about the SW in the XBI-File or the Mobile itself

pabData string which contains all informations

Return values:

void

8.8.4.5 **void FreeList (void * pHead)**

deallocates a list of any type

Parameters:

pHead list of type void (but can be of any type)

8.8.4.6 t_UpdateMode GiveGeneralUpdateMode (void)

Gives the value which defines the library how to perform the Software-Update (between old behaviour and new SW update, or auto-detect).

Parameters:

void

Return values:

the update mode as enumeration

8.8.4.7 char* GiveSwupLibDate (void)

gives the compilation date of the Swup Library

Parameters:

void

Return values:

string : the swup library date

8.8.4.8 t_UpdateMode GiveUpdateModeForThisUpdate (WORD wUpdateNr)

Gives the value which defines the library how to perform the Software-Update *.

Parameters:

wUpdateNr the update number (<=> swup tool number)

Return values:

Behaviour structure of the Library concerning the new Update-Concept (xx45)

8.8.4.9 void InitGlobalMemory (void)

Does Memory Initialization, i.e for : BootCoreStartUpdateSW (last hope when no BSL and no "boot gut jmp") PIN erasefunctions Functionality.

• * - * -

Parameters:

void

Return values:

void

8.8.4.10 BOOL InitMobileErrorHandler (void)**8.8.4.11 BOOL InitProjectDependentBootstrapDll (char * *szProject*, unsigned char *bSvn*)**

initialises the bootstrap Library (DLL) dependent from the current Project depends on the 3 modes : COMPLETE_SWUP_HEAP_LIBRARY COMPLETE_SERVICE_SWUP or other solution

Parameters:

- *szProject* the project name
- bSvn* the project SW version

Return values:

TRUE -> depends on each mode ???
FALSE -> ???

8.8.4.12 BOOL Invalidate_BGJ (t_SwupCom *ComPort*, t_ProcType *ProcInfo*)

Invalidates Boot Good Jump in Flash (when a new Part (for ex.

XDI File or Language, Tegic File) is booted with the SW-Update Tool (Winswup), the BGJ is first invalidated. Then, the new part is booted and when the checksum is OK, the BGJ is booted new)

Parameters:

ComPort the configured Com-Port
ProcInfo the kind of processor we have

Return values:

TRUE -> Statistik address range in XBI File is not located at address 0
FALSE -> no possibility to write in the whole flash or not possible to send data into bin block via serial line

8.8.4.13 BOOL IsLibraryInitialised (void)

returns the boolean to know if Library is initialized

Parameters:

void

Return values:

TRUE -> Library is initialised
FALSE -> Library is not initialised

8.8.4.14 BOOL IsSwupLibVersionNewEnough (unsigned short *nNeededVersion*)

compares the current Swup Library version with the one needed if needed > current, return FALSE otherwise TRUE

Parameters:

nNeededVersion : the needed SWUP Library version

Return values:

TRUE -> current Swup Library version is new enough

FALSE -> current Swup Library version is too old

8.8.4.15 unsigned int IsThisADownGrade (char * *pszUpdateSwDate*, char * *pszMobileSwDate*)

Compares dates from Mobile SW and XBI Header.

Parameters:

pszUpdateSwDate string which contains date from the XBI Header

pszMobileSwDate string which contains date from the Mobile SW

Return values:

1 -> dates are the same

0 -> dates are not the same

8.8.4.16 void NewUpdateFailed_UseBsl (WORD *wUpdateNr*)

Sets the boolean array for Update Failed with number UpdateNb to TRUE.

Parameters:

wUpdateNr the Thread Update number (<=>swup tool number)

Return values:

void

8.8.4.17 void NewUpdateSucceeded (WORD *wUpdateNr*)

Sets the boolean array for Update Failed with number UpdateNb to FALSE.

Parameters:

wUpdateNr the Thread Update number (<=>swup tool number)

Return values:

void

8.8.4.18 void SetErrorForAllComPorts (DWORD *dwError*)

Trace and store error code when Library and Executable have different Interface-Versions *.

Parameters:

dwError : the error code (unsigned int)

Return values:

void

8.8.4.19 void SetPreCheckInfos (WORD wUpdateNr, t_PreCheckInfos info)

Fills the FlashErasure Table for Update number with info to erase Flash.

Parameters:

wUpdateNr the Thread Update number (<=>swup tool number)
info erasefunctions Functionality responsible for Flash-Erasure

Return values:

void

8.8.4.20 void SoftwareUpdateThread (PVOID pvoid)

Updates the Software Thread (But can not tell more because too long => unreadable !!!).

Parameters:

void

Return values:

void

ggg change the call order first error to window, second try to swchit off

8.8.4.21 BOOL SWUPLIBDLLEXIMPORT WSwup_CheckComPortAndSpeed (t_SwupCom WhichCom, DWORD dwBaudrate)

This function only tries to open the selected comport with the given baudrate.

In all cases (besides comport already open) the comport will be closed after this try. It returns the result from opening the comport, so its possible to determine if the win-system allows this baudrate on this comport. There is no guaranty, that the update with this baudrate will work !

Parameters:

WhichCom a t_SwupCom-enum, which comport should be tried
dwBaudrate a dword meaning the Baudrate at which the comport should be tried

Return values:

TRUE It is possible (from WIN-System-side) to open the Comport at this baudrate
FALSE Either Comport not available or this baudrate not possible

8.8.4.22 BOOL SWUPLIBDLLEXIMPORT WSwup_GiveVersionInformation (t_VersionInformation * pInformation, t_InfoSelector WhichInfo)

Returns Version Information about the different Parts of WSWUPLIB.

Since the SWUPLIB-functionality is contained in 3 different DLLs, and one of them is dynamically loaded for different processor-types, it is possible to retrieve some information about the DLLs.

Parameters:

WhichInfo a t_InfoSelector , about which part the information is to be retrieved

pInformation a ptr to a t_VersionInformation , that will be filled with the requested information.
 Attention: pInformation->nStructSize must be filled from the caller with sizeof(t_Version-Information) to avoid conflicts on future enhancements

Return values:

TRUE Information-struct is filled with info

FALSE Information not available, ex. the DLL is not yet loaded

8.8.4.23 **BOOL SWUPLIBDLLEXIMPORT WSwup_InitLibrary (WORD wMajorNumber, WORD wMinorNumber)**

This function must be called by every user of the library once on starting.

This intention of this function is to make a interface-compatibility-check at runtime, so every user of the library has to call this function once before using the library.

The user has to pass the two defines:

- WSWUPLIB_INTERFACE_VERSION_MAJOR
- WSWUPLIB_INTERFACE_VERSION_MINOR to the function. If the given version is different from the one which the library expects, (the one it had on its own compile-time) the function will return FALSE and the library will not work. This is to prevent library-users from unexpected runtime-errors in their executables due to a new interface and new DLLs but an old Executable !

Parameters:

wMajorNumber : must be the given define WSWUPLIB_INTERFACE_VERSION_MAJOR

wMinorNumber : must be the given define WSWUPLIB_INTERFACE_VERSION_MINOR

Return values:

TRUE -> Library is initialised, version-information is ok

FALSE -> version-check failed, library does not work

8.8.4.24 **void SWUPLIBDLLEXIMPORT WSwup_InstallSoftwareInfoCallbackFunction (MobileSwUpdateInfoCallback pfnMobileSwUpdateInfoCallback, t_InfoCallbackTime WhichCallbackTime)**

Installs the MobileSwUpdateInfoCallback-Function.

Parameters:

pfnMobileSwUpdateInfoCallback -> Pointer to a CallbackRoutine that is called to give info about the SW already in Mobile

pass a NULL if you want to deinstall a previously installed callbackfunction.

This is the only place in the LIB where NULL as func-ptr is allowed in the moment !

WhichCallbackTime -> a enum of type , where you must select, on which time you want this callbackfunction called. There are two possibilities , either you can have this function called during connection-setup with BFB-Communication or you can select to have this function called later, when update is already running.

The reason to have these two different times is : on the one hand you have to deal with timeouts when already really in updatemode, this leads to the early test. On the other hand the early test is less "hacker-proof" because the mobile is in a totally different mode and there is at least a very little chance to exchange two mobiles between the two mobile-states. So its up to the user of the library at which time he wants the CallBack-Function.

Returns:

void

8.8.4.25 void SWUPLIBDLLEXIMPORT WSwup_InstallSoftwareUpdateCallBackFunctions (BootstrapProgressCallBack *pfnBootstrapProgress*, UpdateSwTransmissionStartCallBack *pfnUpdateSwTransmissionStart*, UpdateSwTransmissionProgressCallBack *pfnUpdateSwTransmissionProgress*, FlashTypeCallBack *pfnFlashType*, EraseProgressCallBack *pfnEraseProgress*, MobileSwTransmissionStartCallBack *pfnMobileSwTransmissionStart*, MobileSwTransmissionProgressCallBack *pfnMobileSwTransmissionProgress*, MobileSwUpdateSuccessCallBack *pfnMobileSwUpdateSuccess*, MobileSwUpdateErrorCallBack *pfnMobileSwUpdateError*, MobileSwUpdateConsistenceProblemCallBack *pfnMobileConsistenceProblem*)

Installs all the needed callbackfunctions for the whole software-update.

Parameters:

pfnBootstrapProgress -> Pointer to a CallbackRoutine that is called on Bootstrap

pfnUpdateSwTransmissionStart -> Pointer to a CallbackRoutine that is called on UpdateSw-TransmissionStart

pfnUpdateSwTransmissionProgress -> Pointer to a CallbackRoutine that is called during Update-SwTransmission

pfnFlashType -> Pointer to a CallbackRoutine that is called for Flash-Type

pfnEraseProgress -> Pointer to a CallbackRoutine that is called on during Erasure of Flash

pfnMobileSwTransmissionStart -> Pointer to a CallbackRoutine that is called on Start of MobileSw-Transmission

pfnMobileSwTransmissionProgress -> Pointer to a CallbackRoutine that is called during Progress of MobileSw-Transmission

pfnMobileSwUpdateSuccess -> Pointer to a CallbackRoutine that is called if SW-Update was successfull

pfnMobileSwUpdateError -> Pointer to a CallbackRoutine that is called if SW-Update had an error

pfnMobileConsistenceProbleme -> Pointer to a CallbackRoutine that is called if there is a consistence-problem

Returns:

void

8.8.4.26 BOOL SWUPLIBDLLEXIMPORT WSwup_PerformSoftwareUpdate (WORD *wUpdateNr*, t_SwupCom *WhichCom*, unsigned long *ulSpeed*)

Performs the Software-Update on the given ComPort with the given Baudrate.

ATTENTION: this function creates a thread, so the function itself will terminate very fast, so that further things can be done. The communication with the main-program will be done via the above Callback-Functions

Parameters:

wUpdateNr The current number for the update, up to four are allowed

WhichCom The Comport on which the update shall be progressed

ulSpeed the max speed that can be driven on this PC , (use ADDIDATA-Baudrates even if you have a fastboot-card installed, the mapping is done internally via the INI-FILE

Return values:

TRUE -> SW-Update is Running (not finished !)

FALSE -> No Callbacks Installed

8.8.4.27 void SWUPLIBDLLEXIMPORT WSwup_PrepareForUpdateWithSynchStation (BOOL fSynchStationPresent)

Prepares for performing the update via Synch-Station.

The Bootstrap-Loader for a Synch-Station is different to a normal BSL. Since the Bootstraploader-DLL is dynamically loaded on File-Reading, this function must be called in advance, to force loading a different synch-station-specific DLL on FileReading

Parameters:

fSynchStationPresent Boolean Flag for presence/absence of Synch-Station

Returns:

void

8.8.4.28 void SWUPLIBDLLEXIMPORT WSwup_PrepareForUpdateWithUnknownProject (char * pszUnknownProjectName, char * pszDerivedFromKnownProjectName)

Prepares the WSWUP-Library for updating an "unknown" project.

This function must be called before calling of [WSwup_ReadXbiFile\(\)](#) in case the development-project is not known to the library. This only happens during development-phase on introduction of new projects. Normally the Library must know all projects to load the right Update-DLL depending on the read in XBI-File. So if a new project is derived from an already known project, this function can be called to tell the library which update-dll shall be loaded, if an specific unknown project happens to be in the XBI-File.

Parameters:

pszUnknownProjectName Name of the project that is unknown to library up to now.

ulCurrentRead Name of the project that is known to library and who's update-DLL should be used.

Returns:

void

8.8.4.29 BOOL SWUPLIBDLLEXIMPORT WSwup_ReloadUpdateDll (void)

Reloads the DLL that is responsible for performing the Update.

If - after reading the file - the decision is made, that this update is made via a synch-station, the DLL must be reloaded, in order to get the correct Bootstraploader. =>

- call WSwup_PrepareForUpdateWithSynchStation(TRUE)
- call this function

Parameters:*none***Return values:****TRUE** Everything is okay**FALSE** something went wrong**8.8.4.30 void SWUPLIBDLLEXIMPORT WSwup_ResetUsbParams (WORD wUpdateNr)**

Sets parameters for update according Sipc protocol.

Parameters:**wUpdateNr** The current number for the update, up to four are allowed**8.8.4.31 BOOL SWUPLIBDLLEXIMPORT WSwup_SetBootPIN (WORD wUpdateNr, unsigned int unPINSize, unsigned char * pucBootPIN)**

Save a PIN for an Update, This PIN must be provided by data stored in PICS and is sended during BSL connect attempt.

This PIN is used for disabled BSL mode devices. If the PIN is correct the mobile will accept the update sw. A PIN can be reseted by calling this function with unPinSize==0 or pucBootPIN==NULL

Parameters:**wUpdateNr** The current number for the update, up to four are allowed**unPINSize** Number of data bytes of Boot PIN (16 Byte(128Bit) in the moment)**pucBootPIN** Pointer to PIN Data**Return values:****TRUE** -> PIN was set /reseted**FALSE** -> unPINSize or wUpdateNr data to large/ not supported**8.8.4.32 BOOL SWUPLIBDLLEXIMPORT WSwup_SetUpdateConceptMode (t_UpdateMode WhichMode)**

Tells the library how to perform the Software-Update.

In a development-environment it must be still possible do use the old BSL-behaviour, but for a customer-update its necessary to switch to the new update-mode. For development environment it's also possible to do an autodetect, first the new concept is tried and if this fails (e.x. because its an old project) then the old BSL-Mode is activated. If you want to use the new mode (enForceNewConcept, enAutoDetect) you MUST call this function before [WSwup_CheckUpdateVoltage\(\)](#) and you also MUST call function [WSwup_CheckUpdateVoltage\(\)](#) !

Parameters:**WhichMode** an , which tells the library how to behave**ulSpeed** the max speed that can be driven on this PC , (use ADDIDATA-Baudrates even if you have a fastboot-card installed)**Return values:****TRUE** -> everything prepared**FALSE** ->

8.8.4.33 **BOOL SWUPLIBDLLEXIMPORT WSwup_SetUsbParams (WORD wUpdateNr, t_SwupCom WhichUsb, BOOL Force)**

Sets parameters for update according Sipc protocol.

Parameters:

wUpdateNr The current number for the update, up to four are allowed

WhichUsb The UsbPort on which the update shall be progressed

Force TRUE: force virtual UsbPort to this UpdateNr

8.8.5 Variable Documentation

8.8.5.1 unsigned char* **abUpdateSwData**

pointer to the Update-SW that is transmitted to Mobile

8.8.5.2 PFN_StartUpdateSoftwareViaBSL **BootCoreStartUpdateSWFunc**

8.8.5.3 PFN_BootStrapLoaderWithPIN **BootStrapLoaderFunc**

The Bootstrap-Loader itself.

8.8.5.4 **BOOL fCallbacksInstalled = FALSE**

are all the above callbacks installed by the caller ?

8.8.5.5 **BOOL fLibraryIsInitialised**

Library is not initialized, must be initialized **WSwup_InitLibrary()**.

8.8.5.6 **BOOL fSimulateKnownProject = FALSE**

holds the information if user called **WSwup.PrepareForUpdateWithUnknownProject()** and therefore on loading the DLL it must be checked

8.8.5.7 t_UpdateMode **GeneralUpdateMode = enForceBSLBehaviour**

8.8.5.8 PFN_GiveUpdateDllPcSwVersion **GiveUpdateDllPcSwVersion**

Version-Information Update-DLL.

8.8.5.9 PFN_GiveUpdateMobileSwVersion **GiveUpdateMobileSwVersion**

Version-Information Mobile-Update-SW contained in Update-DLL.

8.8.5.10 PFN_GiveUpdateSwPtrAndLen **GiveUpdatePtrAndLenFunc**

for retrieving update-sw from inside DLL

8.8.5.11 `t_InfoCallbackTime` [GlobalWhichInfoCallbackTime](#)

8.8.5.12 `HINSTANCE` [hUpdateLib](#) = NULL

Handle to the project-dependent UPDATE-Library.

8.8.5.13 `MobileSwUpdateInfoCallback` [MobileSwInfoFunction](#) = NULL

tells caller about SW-Version etc in Mobile

8.8.5.14 `BOOL` [NewUpdateFailed](#)[AMOUNT_OF_UPDATES]

8.8.5.15 `PFN_USBOMAPSWUP_GETDEVICE` [pfnUsbOmapSwup_GetDevice](#)

8.8.5.16 `PFN_USBOMAPSWUP_RELEASEDEVICE` [pfnUsbOmapSwup_ReleaseDevice](#)

8.8.5.17 `char` [szDerivedFromKnownProject](#)[200]

saves the name of the project from which the new unknown was derived

8.8.5.18 `char` [szNewUnknownProject](#)[200]

saves the name of a new unknown project

8.8.5.19 `t_ThreadInfo` [ThreadInfo](#)[AMOUNT_OF_UPDATES + 1]

some information must be held "global"

8.8.5.20 `unsigned long` [ulAmountBytesUpdateData](#)

len of Update-SW

8.8.5.21 `unsigned int` [unTotalHeapBytes](#)

8.8.5.22 `t_UpdateMode` [UpdateModes](#)[AMOUNT_OF_UPDATES]

8.8.5.23 `t_UsbInfo` [UsbSettings](#)[]

8.8.5.24 `t_ExtendedInfo` [XbiHeaderInfo](#)

8.9 wfbttool.c File Reference

Contains the bfb-access-functions , part of the MAIN-DLL.

```
#include <windows.h>
#include "wswuplib.h"
#include "wsw_ser1.h"
#include "wsw_err.h"
#include "wswlibv1.h"
#include "wfbttool.h"
#include "fiha_h.h"
#include "werrenum.h"
```

Data Structures

- struct [tag_BfbThreadInfo](#)

Defines

- #define [MAX_VOLTAGE_CALLBACKS](#) 6
- #define [MAX_SUCCESS_CALLBACKS](#) 4
- #define [W_SWUPHELPSAPI](#) 0x0e
- #define [M_SWH_IGNITION](#) 0x01
- #define [M_SWH_VOLTAGE](#) 0x02
- #define [M_SWH_VERSION_NUMBER](#) 0x03
- #define [M_SWH_MOBILE_OFF](#) 0x04
- #define [M_SWH_GET_POWER](#) 0x05
- #define [M_SWH_CUSTOMER_REBOOT](#) 0x20
- #define [M_SWH_GET_PUTCOUNT](#) 0x21
- #define [DllExImport](#) __declspec(dllimport)

Typedefs

- typedef DllExImport HANDLE(* [PFN_WCOMOPEN](#))(int Device, DWORD [dwBaudrate](#))
This are the functionpointers to the BFB-Library.
- typedef DllExImport BOOL(* [PFN_WCOMCLOSE](#))(HANDLE hDevice)
- typedef DllExImport BOOL(* [PFN_WPINGMOBILE](#))(HANDLE hDevice)
- typedef DllExImport BOOL(* [PFN_WSETBFBMODE](#))(HANDLE hDevice)
- typedef DllExImport BOOL(* [PFN_WMOBILEON](#))(HANDLE hDevice)
- typedef DllExImport BOOL(* [PFN_WCOMWRITE](#))(HANDLE hDevice, BYTE bSapi, BYTE bLength, BYTE *pbBuffer)
- typedef DllExImport BOOL(* [PFN_WCOMRECEIVE](#))(HANDLE hDevice, BYTE bFilterSapi, BYTE bFilterMsgType, BYTE *pbSapi, BYTE *pbLength, BYTE *pbMsg)
- typedef DllExImport BOOL(* [PFN_WCOMREGISTRY](#))(BYTE *ComPort, DWORD *dwBaudrate)
- typedef DllExImport BOOL(* [PFN_WMOBILEOFF](#))(HANDLE hDevice)
- typedef [tag_BfbThreadInfo](#) [t_BfbThreadInfo](#)

Enumerations

- enum [t_BfbStates](#) {
[enComOpen](#), [enPingFirst](#), [enPingSecond](#), [enPingThird](#),
[enMobileOn](#), [enGetVoltage](#), [enSwitchOff](#), [enError](#),
[enReady](#), [enWaitALittleBit](#) }

Functions

- unsigned short [GetActVoltage](#) (HANDLE hCom)
- BOOL [SwitchMobileToNewBootMode](#) (HANDLE hCom, unsigned long ulUpdateSpeed)
Sends the Mobile the switch-to-boot-message via SWUPHELP-Sapi.
- BOOL [GivePutcountAndDirectory](#) (HANDLE hCom, unsigned long *pulPutCount, unsigned char *pabDirectory)
- void SWUPLIBDLLEXIMPORT [WSwup_InstallBfbCallBackFunctions](#) (VoltageCheckCallBack pfnVoltageCheck, UpdateSuccessCallBack pfnSuccessCheck)
- BOOL [InitBfbLibrary](#) (t_BfbLib WhichLib)
Loads the BFB-DLL and retrieves pointers to all necessary functions.
- BOOL [DeactBfbLibrary](#) (void)
Frees the loaded BFB-DLL.
- BOOL SWUPLIBDLLEXIMPORT [WSwup_CheckUpdateSuccess](#) (WORD wUpdateNumber, t_-SwupCom WhichCom)
- void [BfbCheckThread](#) (PVOID pvoid)
- BOOL SWUPLIBDLLEXIMPORT [WSwup_CheckUpdateVoltage](#) (WORD wUpdateNumber, t_-SwupCom WhichCom, unsigned short *punVoltage, unsigned long ulSpeed)

Variables

- HINSTANCE [hBfbLib](#) = NULL
Handle to the BFB-Library.
- [PFN_WPINGMOBILE](#) [pfnWPingMobile](#)
pointer to Wping
- [PFN_WSETBFBMODE](#) [pfnWSetBFBMode](#)
pointer to WSetBfb
- [PFN_WMOBILEON](#) [pfnWMobileOn](#)
pointer to WMon
- [PFN_WCOMOPEN](#) [pfnWComOpen](#)
pointer to WOpen
- [PFN_WCOMCLOSE](#) [pfnWComClose](#)
pointer to WClose

- [PFN_WCOMREGISTRY](#) [pfnWComReadRegistry](#)

pointer to WRegistry

- [PFN_WCOMWRITE](#) [pfnWComWrite](#)

pointer to Write

- [PFN_WCOMRECEIVE](#) [pfnWComReceive](#)

pointer to Receive

- [PFN_WMOBILEOFF](#) [pfnWMobileOff](#)

pointer to Mobile Off

- [DWORD](#) [dwBaudrate](#) = 57600

Baudrate , fix to 57600, no registry-check !

- [VoltageCheckCallBack](#) [pfnVoltageCheckCallback](#) = NULL

- [UpdateSuccessCallBack](#) [pfnUpdateCheckCallback](#) = NULL

- [t.UpdateMode](#) [MyUpdateMode](#)

- [BOOL](#) [NewUpdateFailed](#) []

- [t.BfbThreadInfo](#) [BfbThreadInfo](#) [AMOUNT_OF_UPDATES+1]

- [t.ExtendedInfo](#) [XbiHeaderInfo](#)

8.9.1 Detailed Description

Contains the bfb-access-functions , part of the MAIN-DLL.

Todo

Make some comments.

8.9.2 Define Documentation

8.9.2.1 `#define DllExport __declspec(dllexport)`

8.9.2.2 `#define M_SWH_CUSTOMER_REBOOT 0x20`

8.9.2.3 `#define M_SWH_GET_POWER 0x05`

8.9.2.4 `#define M_SWH_GET_PUTCOUNT 0x21`

8.9.2.5 `#define M_SWH_IGNITION 0x01`

8.9.2.6 `#define M_SWH_MOBILE_OFF 0x04`

8.9.2.7 `#define M_SWH_VERSION_NUMBER 0x03`

8.9.2.8 `#define M_SWH_VOLTAGE 0x02`

8.9.2.9 `#define MAX_SUCCESS_CALLBACKS 4`

8.9.2.10 `#define MAX_VOLTAGE_CALLBACKS 6`

8.9.2.11 `#define W_SWUPHELPSAPI 0x0e`

8.9.3 Typedef Documentation

8.9.3.1 `typedef DllExport BOOL( * PFN\_WCOMCLOSE)( HANDLE hDevice )`

8.9.3.2 `typedef DllExport HANDLE( * PFN\_WCOMOPEN)( int Device, DWORD dwBaudrate )`

This are the functionpointers to the BFB-Library.

Since we do no statical linking against the export-lib (we support BFB95 and BFB95EG), instead we do a dynamical loading, we have to declare a FunctionPointer for each used functions. These functionpointers will be resolved after loading the DLL Here is always in the first line the original prototype from bfb95.h as comment and in the second line there is the from this driven functionpointer

8.9.3.3 typedef DllExport BOOL(* [PFN_WCOMRECEIVE](#))(HANDLE hDevice, BYTE bFilterSapi, BYTE bFilterMsgType, BYTE *pbSapi, BYTE *pbLength, BYTE *pbMsg)

8.9.3.4 typedef DllExport BOOL(* [PFN_WCOMREGISTRY](#))(BYTE *ComPort, DWORD *dwBaudrate)

8.9.3.5 typedef DllExport BOOL(* [PFN_WCOMWRITE](#))(HANDLE hDevice, BYTE bSapi, BYTE bLength, BYTE *pbBuffer)

8.9.3.6 typedef DllExport BOOL(* [PFN_WMOBILEOFF](#))(HANDLE hDevice)

8.9.3.7 typedef DllExport BOOL(* [PFN_WMOBILEON](#))(HANDLE hDevice)

8.9.3.8 typedef DllExport BOOL(* [PFN_WPINGMOBILE](#))(HANDLE hDevice)

8.9.3.9 typedef DllExport BOOL(* [PFN_WSETBFBMODE](#))(HANDLE hDevice)

8.9.3.10 typedef struct [tag_BfbThreadInfo](#) t_BfbThreadInfo

8.9.4 Enumeration Type Documentation

8.9.4.1 enum [t_BfbStates](#)

Enumeration values:

enComOpen

enPingFirst

enPingSecond

enPingThird

enMobileOn

enGetVoltage

enSwitchOff

enError

enReady

enWaitALittleBit

8.9.5 Function Documentation

8.9.5.1 void BfbCheckThread (PVOID pvoid)

8.9.5.2 BOOL DeactBfbLibrary (void)

Frees the loaded BFB-DLL.

8.9.5.3 unsigned short GetActVoltage (HANDLE *hCom*)

8.9.5.4 BOOL GivePutcountAndDirectory (HANDLE *hCom*, unsigned long * *pulPutCount*, unsigned char * *pabDirectory*)

8.9.5.5 BOOL InitBfbLibrary (t_BfbLib *WhichLib*)

Loads the BFB-DLL and retrieves pointers to all necessary functions.

8.9.5.6 BOOL SwitchMobileToNewBootMode (HANDLE *hCom*, unsigned long *ulUpdateSpeed*)

Sends the Mobile the switch-to-boot-message via SWUPHELP-Sapi.

We have three possibilities : 1. 0x20 0x00 baudrate as little-endian long (20 00 F0 31 06 00 = 406000 Bd) 2. 0x20 0x01 3. 0x20 0x02 default to 57600kBaud

8.9.5.7 BOOL SWUPLIBDLLEXIMPORT WSwup_CheckUpdateSuccess (WORD *wUpdateNumber*, t_SwupCom *WhichCom*)

8.9.5.8 BOOL SWUPLIBDLLEXIMPORT WSwup_CheckUpdateVoltage (WORD *wUpdateNumber*, t_SwupCom *WhichCom*, unsigned short * *punVoltage*, unsigned long *ulSpeed*)

8.9.5.9 void SWUPLIBDLLEXIMPORT WSwup_InstallBfbCallBackFunctions (VoltageCheckCallBack *pfnVoltageCheck*, UpdateSuccessCallBack *pfnSuccessCheck*)

8.9.6 Variable Documentation

8.9.6.1 t_BfbThreadInfo BfbThreadInfo[AMOUNT_OF_UPDATES + 1]

8.9.6.2 DWORD dwBaudrate = 57600

Baudrate , fix to 57600, no registry-check !

8.9.6.3 HINSTANCE hBfbLib = NULL

Handle to the BFB-Library.

8.9.6.4 t_UpdateMode MyUpdateMode

8.9.6.5 BOOL NewUpdateFailed[]

8.9.6.6 UpdateSuccessCallBack pfnUpdateCheckCallback = NULL

8.9.6.7 VoltageCheckCallBack pfnVoltageCheckCallback = NULL

8.9.6.8 PFN_WCOMCLOSE pfnWComClose

pointer to WClose

8.9.6.9 PFN_WCOMOPEN pfnWComOpen

pointer to WOpen

8.9.6.10 PFN_WCOMREGISTRY pfnWComReadRegistry

pointer to WRegistry

8.9.6.11 PFN_WCOMRECEIVE pfnWComReceive

pointer to Receive

8.9.6.12 PFN_WCOMWRITE pfnWComWrite

pointer to Write

8.9.6.13 PFN_WMOBILEOFF pfnWMobileOff

pointer to Mobile Off

8.9.6.14 PFN_WMOBILEON pfnWMobileOn

pointer to WMon

8.9.6.15 PFN_WPINGMOBILE pfnWPingMobile

pointer to Wping

8.9.6.16 PFN_WSETBFBMODE pfnWSetBFBMode

pointer to WSetBfb

8.9.6.17 t_ExtendedInfo XbiHeaderInfo

8.10 wmob_err.c File Reference

Responsible for decoding errors sent from mobile, contained in the MAIN-DLL.

```
#include <windows.h>
#include "wswuplib.h"
#include "wsw_ser_i.h"
#include "pc_mob.h"
#include "wsw_err.h"
#include "werrenum.h"
```

Functions

- BOOL [MobileErrorHandler](#) (t_SwupCom ComPort, unsigned char ucMi, unsigned int unLen, unsigned char *abBuffer)
- BOOL [InitMobileErrorHandler](#) (void)

8.10.1 Detailed Description

Responsible for decoding errors sent from mobile, contained in the MAIN-DLL.

8.10.2 Function Documentation

8.10.2.1 BOOL InitMobileErrorHandler (void)

8.10.2.2 BOOL MobileErrorHandler (t_SwupCom ComPort, unsigned char ucMi, unsigned int unLen, unsigned char * abBuffer)

8.11 wseril1.c File Reference

Serial Communication : Layer1 -functionality, contained in WSWUPSER-DLL.

```
#include <windows.h>
#include "wswuplib.h"
#include "wsw_ser_i.h"
#include "wsw_err.h"
#include "werrenum.h"
```

Data Structures

- struct [t_ComConfig](#)
This describes one serial connection.

Defines

- #define [ADDIFAST_INI_NAME](#) "addifast.ini"
INI-File that selects if an ADDI-Data-Card or an old-Fastboot-Card is present.
- #define [DEFAULT_COMTYPE](#) "fastboot"
Type in INI-File for selecting an old fastboot-card.
- #define [ADDI_COMTYPE](#) "addidata"
Type in INI-File for selecting an ADDIDATA-Card.

Functions

- unsigned long [ConvertBaudRatesForAddiData](#) (t_SwupCom WhichCom, unsigned long ulBaud-Rate)
Converts Baudrates depending on the (via INI-File) selected card.
- BOOL SWUPSERILOWEXIMPORT [WSwupComOpen](#) (t_SwupCom ComPort, DWORD [dw-Baudrate](#), [t_SwupParity](#) WhichParity)
Opens the given Com-Port with the given Baud-Rate and Parity.
- BOOL SWUPSERILOWEXIMPORT [WSwupComConfigure](#) (t_SwupCom ComPort, DWORD [dw-Baudrate](#))
Changes the Baudrate of the Given ComPort.
- BOOL SWUPSERILOWEXIMPORT [WSwupComClose](#) (t_SwupCom ComPort)
Closes the given Com-Port.
- BOOL SWUPSERILOWEXIMPORT [WSwupComWriteByte](#) (t_SwupCom ComPort, BYTE b-Value)
Writes a single Byte to the given ComPort.

- **BOOL SWUPSERILOWEXIMPORT WSwupComWriteDataBlock** (t_SwupCom ComPort, BYTE *pbValue, DWORD len)
Writes a Data-Block to the given comport.
- **BOOL SWUPSERILOWEXIMPORT WSwupComReadByteTimeOut** (t_SwupCom ComPort, LPBYTE lpbByte, DWORD dwTimeOut)
Reads a Byte from the given serial port with timeout.
- **BOOL SWUPSERILOWEXIMPORT WSwupComReadByte** (t_SwupCom ComPort, LPBYTE lpbByte)
Reads a Byte from the given serial port.
- **BOOL SWUPSERILOWEXIMPORT WSwupSetIgnition** (t_SwupCom ComPort)
Sets the Ignition-Port on the serial port (DTR-Line).
- **BOOL SWUPSERILOWEXIMPORT WSwupClearIgnition** (t_SwupCom ComPort)
Clears the Ignition-Port on the serial port (DTR-Line).
- **HANDLE SWUPSERILOWEXIMPORT WSwupGiveHandleForComport** (t_SwupCom ComPort)
Returns the windows handle for a given ComPort.

Variables

- **t_ComConfig ComPorts** [AMOUNT_OF_SWUP_PORTS]
Array of Information for each possible connection.

8.11.1 Detailed Description

Serial Communication : Layer1 -functionality, contained in WSWUPSER-DLL.

8.11.2 Define Documentation

8.11.2.1 #define ADDI_COMTYPE "addidata"

Type in INI-File for selecting an ADDIDATA-Card.

8.11.2.2 #define ADDIFAST_INI_NAME "addifast.ini"

INI-File that selects if an ADDI-Data-Card or an old-Fastboot-Card is present.

8.11.2.3 #define DEFAULT_COMTYPE "fastboot"

Type in INI-File for selecting an old fastboot-card.

8.11.3 Function Documentation

8.11.3.1 HANDLE SWUPSERILOWEXIMPORT WSwupGiveHandleForComport (t_SwupComComPort)

Returns the windows handle for a given ComPort.

This function is necessary because there is an external interface to INFINEON-DLL: SG_RAM_Loader.dll.

Parameters:

ComPort The ComPort , see t_SwupCom

8.11.4 Variable Documentation

8.11.4.1 [t_ComConfig ComPorts](#)[AMOUNT_OF_SWUP_PORTS]

Array of Information for each possible connection.

8.12 wseril2.c File Reference

Serial Communication : Layer2 -functionality, contained in WSWUPSER-DLL.

```
#include <windows.h>
#include "wswuplib.h"
#include "wsw_ser_i.h"
#include "wsw_err.h"
#include "werrenum.h"
#include "pc_mob.h"
#include "IBoot.h"
```

Data Structures

- struct [t_TableEntry](#)

For each Message-Identifier it is possible to define a default-Handler.

Defines

- #define [SWUPSERIDLL_VERS_MAJOR](#) 1
Major-Version of the serial DLL.
- #define [SWUPSERIDLL_VERS_MINOR](#) 43
Minor-Version of the serial DLL.
- #define [SWUPSERIDLL_DEVELOPER](#) "schmittu"
Maintainer or programmer that did the last major change in the serial DLL.
- #define [ACK_LEN](#) 1
Sends ACK/NAK in SIPC format.
- #define [ACK_LEN_TO_SEND](#) 4

Typedefs

- typedef IBOOT_API BOOL(* [PFN_USBOMAPSWUP_WRITETODATAPIPE](#))(WORD wUsbPort, BYTE *OutBuffer, long OutBufferLen, void(*UpdateTransferStatus)(WORD, short))
- typedef IBOOT_API BOOL(* [PFN_USBOMAPSWUP_READFROMDATAPIPE](#))(WORD wUsbPort, BYTE *InBuffer, long InBufferLen, ULONG *nBytesRead)

Functions

- [t_RecResult ReceiveCmdBlockInBinMode](#) (t_SwupCom ComPort, [t_MessageBlock](#) *pMessageBlock, BOOL fReadFirstChar, DWORD dwTimeOut)
Receives a CMD-Block in BIN-Mode.

- **BOOL WSwSendDataBlockInBinFormat** (t_SwupCom ComPort, unsigned char *pabBuffer, DWORD dwRecTimeOut, unsigned short *pAmountSent)
Sends a Data-Block explicitley in BIN-Format.
- **BOOL WSwSendDataBlockInSipcFormat** (WORD UsbPort, unsigned char *pabBuffer, unsigned short *pAmountSent)
Sends a Data-Block explicitley in SIPC-Format.
- **t_RecResult ReceiveBlockInSipcMode** (WORD wUpdateNr, t_MessageBlock *pMessageBlock)
Receives a CMD-Block in SipcMode.
- **BOOL SWUPSERILOWEXIMPORT WSwSipcWriteToDataPipe** (WORD wUpdateNr, unsigned char *ucOutBuffer, t_RecResult *RecResult, unsigned char *ucRetBuffer)
- **void SendAckNakInSipcMode** (WORD wUpdateNr, unsigned char ucValue)
- **BOOL SWUPSERILOWEXIMPORT WSwupSeriLowGiveVersion** (t_VersionInformation *pInfo)
Returns information about the Serial-Low-DLL.
- **BOOL InstallDefaultHandlerFunction** (unsigned char ucThisMi, pfnDefaultHandler ThisDefaultHandler)
Installs a defaulthandler-function for a specific Message-Identifier.
- **void SWUPSERILOWEXIMPORT WSwupSendBotBlock** (t_SwupCom ComPort, unsigned char *pucSendBuf, unsigned int unDatLen)
Sends a so called BOT-Block to the Mobile.
- **void SWUPSERILOWEXIMPORT WSwupSendSGoldBotBlock** (t_SwupCom ComPort, unsigned char *pucSendBuf, unsigned int unDatLen)
Sends a so called BOT-Block to the SGOLD-Mobile.
- **t_RecResult SWUPSERILOWEXIMPORT WSwupReceiveValueFromSerial** (t_SwupCom ComPort, unsigned char *pucRecBuf, DWORD dwTimeOutMilli)
Read a single byte from the serial Port.
- **BOOL SWUPSERILOWEXIMPORT WSwupReceiveCmdBlock** (t_SwupCom ComPort, t_MessageBlock *pMessageBlock, unsigned char ucExpectedMi, DWORD dwTimeOutMilli)
Waites for a CMD-Block with the given Message-Identifier on the serial port.
- **BOOL SWUPSERILOWEXIMPORT WSwupSendMessageBlock** (t_SwupCom ComPort, unsigned char ucMi, unsigned short unLen, unsigned char *pabBuffer)
Sends a CMD-Block (currently only in BIN-Mode).
- **void SWUPSERILOWEXIMPORT WSwupSetCurrentReceiveMode** (t_SwupCom ComPort, t_KnownReceiveModes WhichMode)
Sets the current receive-mode .
- **t_KnownReceiveModes SWUPSERILOWEXIMPORT WSwupWSwupGetCurrentReceiveMode** (t_SwupCom ComPort)
Retrieves the current receive-mode .

- **BOOL SWUPSERILOWEXIMPORT WSwSipcWriteToDataPipe_CoproMode** (WORD wUsbPort, unsigned char *ucOutBuffer, unsigned char ucExpectedSapi, unsigned char *ucRetBuffer, **t_RecResult** *RecResult)
sends a bfb message to modem's CoproMode via Usb same layer
- **t_RecResult CheckSipcMessage** (WORD wUsbPort, unsigned int ReceivedBytes, **t_MessageBlock** *pMessageBlock, unsigned char *CheckBuffer, unsigned char *Anzahl)
- **void SWUPSERILOWEXIMPORT SendSimpleMessageInSipcMode** (WORD wUsbPort, unsigned char ucMi, unsigned char ucPayload)
Sends message in SIPC format without waiting for response.

Variables

- **PFN_USBOMAPSWUP_WRITETODATAPIPE** pfnUsbOmapSwup_WriteToDataPipe_Low
- **PFN_USBOMAPSWUP_READFROMDATAPIPE** pfnUsbOmapSwup_ReadFromDataPipe_Low
- **t_KnownReceiveModes CurrentReceiveMode** [AMOUNT_OF_SWUP_PORTS]
for each possible open connection the current receiveand send-mode
- **t_MessageBlock ActualMessage** [AMOUNT_OF_SWUP_PORTS]
for each possible open connection the last received Message-Block
- **t_TableEntry DefaultHandlerTable** [255]
for each Message-Identifier a default-handler
- **HINSTANCE hIBootLibLow** = NULL

8.12.1 Detailed Description

Serial Communication : Layer2 -functionality, contained in WSWUPSER-DLL.

8.12.2 Define Documentation

8.12.2.1 #define ACK_LEN_TO_SEND 4

8.12.2.2 #define SWUPSERIDLL_DEVELOPER "schmittu"

Maintainer or programmer that did the last major change in the serial DLL.

8.12.2.3 #define SWUPSERIDLL_VERS_MAJOR 1

Major-Version of the serial DLL.

8.12.2.4 #define SWUPSERIDLL_VERS_MINOR 43

Minor-Version of the serial DLL.

8.12.3 Typedef Documentation

8.12.3.1 **typedef IBOOT_API BOOL**(* [PFN_USBOMAPSWUP_READFROMDATAPIPE](#))(
WORD wUsbPort, BYTE *InBuffer, long InBufferLen, ULONG *nBytesRead)

8.12.3.2 **typedef IBOOT_API BOOL**(* [PFN_USBOMAPSWUP_WRITETODATAPIPE](#))(WORD
wUsbPort, BYTE *OutBuffer, long OutBufferLen,void (*UpdateTransferStatus)(WORD,
short))

8.12.4 Function Documentation

8.12.4.1 **t_RecResult** CheckSipcMessage (WORD wUsbPort, unsigned int *ReceivedBytes*,
[t_MessageBlock](#) * *pMessageBlock*, unsigned char * *CheckBuffer*, unsigned char * *Anzahl*)

8.12.4.2 **void** SendAckNakInSipcMode (WORD wUpdateNr, unsigned char *ucValue*)

8.12.4.3 **BOOL** SWUPSERILOWEXIMPORT WSwSendDataBlockInBinFormat (t_SwupCom
ComPort, unsigned char * *pabBuffer*, DWORD *dwRecTimeOut*, unsigned short *
pAmountSent)

Sends a Data-Block explicitly in BIN-Format.

This routine forms a BIN-Block with the given Message-Identifier and the given date, calculates the check-sum and sends it out via serial line. Then it waits for the answer of the Mobile via function WSwupReceive-ValueFromSerial which normally waits for a single char, but can handle Message-Blocks if they occur. If the received value is a single char and an ACK, this routine returns TRUE, if it is not ACK, the Block is sent once more, this up to the value of MAX_RETRY_ON_NAK

Parameters:

ComPort : the comport to send the message-block

ucMi : the Message-Identifier

unLen : the amount of data following in

pabBuffer : the buffer to the data to send

Return values:

TRUE -> Message-Block was sent

FALSE -> an error occurred

8.12.4.4 **BOOL** SWUPSERILOWEXIMPORT WSwSendDataBlockInSipcFormat (WORD
UsbPort, unsigned char * *pabBuffer*, unsigned short * *pAmountSent*)

Sends a Data-Block explicitly in SIPC-Format.

Then it waits for the answer of the Mobile via function ??? which normally waits for a single char, but can handle Message-Blocks if they occur. If the received value is a single char and an ACK, this routine returns TRUE, if it is not ACK, the Block is sent once more, this up to the value of MAX_RETRY_ON_NAK

Parameters:

UsbPort : where to send

pabBuffer : the buffer of the data to send

Return values:*TRUE* -> Data-Block was sent*FALSE* -> an error occurred

!!???!!!

8.12.4.5 `BOOL SWUPSERILOWEXIMPORT WSwSipcWriteToDataPipe (WORD wUsbPort, unsigned char * ucOutBuffer, t_RecResult * RecResult, unsigned char * ucRetBuffer)`

!!???!!!

!!???!!!

8.12.4.6 `BOOL SWUPSERILOWEXIMPORT WSwSipcWriteToDataPipe_CoproMode (WORD wUsbPort, unsigned char * ucOutBuffer, unsigned char ucExpectedSapi, unsigned char * ucRetBuffer, t_RecResult * RecResult)`

sends a bfb message to modem's CoproMode via Usb same layer

Then it waits for the answer of the Mobile via function ????? If the received value is a single char and an ACK, this routine returns TRUE, if it is not ACK, the Block is sent once more, this up to the value of MAX_RETRY_ON_NAK

Parameters:*wUsbPort*: the UsbPort to receive the message-block, same as UpdateNr*ucMi* : the Message-Identifier*unLen* : the amount of data following in*pabBuffer* : the buffer to the data to send**Return values:***TRUE* -> Message-Block was sent*FALSE* -> an error occurred**8.12.5 Variable Documentation****8.12.5.1** `t_MessageBlock ActualMessage[ AMOUNT_OF_SWUP_PORTS ]`

for each possible open connection the last received Message-Block

8.12.5.2 `t_KnownReceiveModes CurrentReceiveMode[ AMOUNT_OF_SWUP_PORTS ]`

for each possible open connection the current receiveand send-mode

8.12.5.3 `t_TableEntry DefaultHandlerTable[255]`

for each Message-Identifier a default-handler

8.12.5.4 **HINSTANCE** **hIBootLibLow** = NULL

8.12.5.5 **PFN_USBOMAPSWUP_READFROMDATAPIPE** **pfnUsbOmapSwup_ReadFromDataPipe_Low**

8.12.5.6 **PFN_USBOMAPSWUP_WRITETODATAPIPE** **pfnUsbOmapSwup_WriteToDataPipe_Low**

8.13 wseril3.c File Reference

Layer-3-functionality for the serial communication, Part of the MAIN-DLL.

```
#include <windows.h>
#include "wswuplib.h"
#include "wsw_ser_i.h"
#include "wsw_err.h"
#include "pc_mob.h"
#include "wswlibv1.h"
#include "wswserl3.h"
#include "werrenum.h"
#include "IBoot.h"
```

Defines

- #define [W_SWUPHELPSAPI](#) 0x0e
- #define [CONTROL_BIN_BLOCKS_SUPPORTED](#) 0xAA
- #define [ERASE_TIME_FLASH_SEGMENT](#) 10000
- #define [ERASE_AREA_OK](#) 0x00
- #define [ERASE_AREA_NOT_OK](#) 0xFF
- #define [M_SWH_VOLTAGE](#) 0x02

Typedefs

- typedef IBOOT_API BOOL(* [PFN_USBOMAPSWUP_GETDEVICE](#))(WORD wUsbPort, BOOL fForceSavedPort)
- typedef IBOOT_API BOOL(* [PFN_USBOMAPSWUP_RELEASEDEVICE](#))(WORD wUsbPort)
- typedef IBOOT_API BOOL(* [PFN_USBOMAPSWUP_WRITETOCONTROLPIPE](#))(WORD wUsbPort, BYTE *OutBuffer, short OutBufferLen)
- typedef IBOOT_API BOOL(* [PFN_USBOMAPSWUP_WRITETODATAPIPE](#))(WORD wUsbPort, BYTE *OutBuffer, long OutBufferLen, void(*UpdateTransferStatus)(WORD, short))
- typedef IBOOT_API BOOL(* [PFN_USBOMAPSWUP_READFROMDATAPIPE](#))(WORD wUsbPort, BYTE *InBuffer, long InBufferLen, ULONG *nBytesRead)

Functions

- [t_KnownReceiveModes](#) SWUPSERILOWEXIMPORT [WSwupWSwupGetCurrentReceiveMode](#) (t_SwupCom ComPort)
Retrieves the current receive-mode .
- BOOL [DummyUsbOmapSwup_GetDevice](#) (WORD wUsbPort, BOOL fForceSavedPort)
- BOOL [DummyUsbOmapSwup_ReleaseDevice](#) (WORD wUsbPort)
- BOOL [DummyUsbOmapSwup_WriteToControlPipe](#) (WORD wUsbPort, BYTE *OutBuffer, short OutBufferLen)
- BOOL [DummyUsbOmapSwup_WriteToDataPipe](#) (WORD wUsbPort, BYTE *OutBuffer, long OutBufferLen, void(*UpdateTransferStatus)(WORD, short))

- BOOL [DummyUsbOmapSwup_ReadFromDataPipe](#) (WORD wUsbPort, BYTE *InBuffer, long InBufferLen, ULONG *nBytesRead)
- BOOL [InitIBootLibrary](#) (void)
load functionpointers of iBootLib static Lib impossible because NT does not support USB same layer
- BOOL [DeactIBootLibrary](#) (void)
unloads functionpointers of iBootLib same layer
- BOOL [WSwupSendAlignement](#) (t_SwupCom ComPort, unsigned short unAlignement)
- BOOL [MobileUpdateConceptRunningHandler](#) (t_SwupCom ComPort, unsigned char ucMi, unsigned int unLen, unsigned char *abBuffer)
- BOOL [WSwupSendData](#) (t_SwupCom ComPort, unsigned char *pabBuffer, unsigned long ulLenToSend, DWORD dwRecTimeOut, WORD wNumber, void(*Progress)(WORD, unsigned long))
- BOOL [WSwupSendCommandData](#) (t_SwupCom ComPort, unsigned char **ppabSource, unsigned long *pulLenToSend, DWORD dwRecTimeOut, WORD wNumber)
- BOOL [WSwupSendEotBlock](#) (t_SwupCom ComPort)
- BOOL [WSwupSelectMobileMode](#) (t_SwupCom ComPort, t_MsmLowMobileStationMode NextMode, t_MsmLowMobileStationMode PrevMode)
- BOOL [WSwupRequestFlashCode](#) (t_SwupCom ComPort, unsigned char HowToWrite)
- BOOL [WSwupSendBlockTimeOut](#) (t_SwupCom ComPort, unsigned int unTicks)
- BOOL [WSwupSendByteTimeOut](#) (t_SwupCom ComPort, unsigned int unSek)
- BOOL [WSwupRequestFlashSize](#) (t_SwupCom ComPort)
- BOOL [WSwupRequestEraseBlocks](#) (t_SwupCom ComPort, unsigned long ulFrom, unsigned long ulTo)
- BOOL [WSwupEraseOneFlashBlock](#) (t_SwupCom ComPort, unsigned int unNumber)
- BOOL [WSwupEraseFlashBlocks](#) (t_SwupCom ComPort, unsigned int unFirst, unsigned int unLast)
- BOOL [WSwupSendSwitchOff](#) (t_SwupCom ComPort)
- BOOL [WSwupSendCompressionInfo](#) (t_SwupCom ComPort, unsigned short unAlgorithmNumber, unsigned char ucFormatBefore, unsigned char ucFormatAfter, unsigned short unAddInfo1, unsigned short unAddInfo2, unsigned short unAddInfo3)
- BOOL [WSwupSendSwitchOffCompression](#) (t_SwupCom ComPort)
- BOOL [WSwupGiveMobileSwVersion](#) (t_SwupCom ComPort)
- BOOL [WSwupSendAdditionalInfo](#) (t_SwupCom ComPort, unsigned char *pucData, unsigned short unLenData)
- BOOL [WSwupSendSplitIDInfo](#) (t_SwupCom ComPort, unsigned long ulIdAdress)
- BOOL [WSwupSendLanguageGroupInfo](#) (t_SwupCom ComPort, unsigned long ulStart, unsigned long ulTo, unsigned long ulXorAdr, unsigned long ulPatchAdr)
- BOOL [WSwupSendNewSplitInfo](#) (t_SwupCom ComPort, unsigned long ulSplitStart, unsigned long ulSplitEnd, unsigned long ulSplitCheckAdd)
- BOOL [EnableWriteFlashBlock](#) (t_SwupCom ComPort, unsigned long BlockToUnlock)
- BOOL [WSwupEnableFlashProgrammingWholeFlash](#) (t_SwupCom ComPort)
- BOOL [WSwupSendEraseCodeAreaNewUpdateConcept](#) (t_SwupCom ComPort)
- BOOL [EraseWithNewUpdateConcept](#) (t_SwupCom ComPort, WORD wUpdateNr, EraseProgress-Callback Progress)
- BOOL [EraseFlashBlocksWithAcknowledge](#) (t_SwupCom ComPort, t_EraseBlocks *pEraseBlocks, WORD wUpdateNr, WORD wMaxAmount, EraseProgressCallback Progress)
- BOOL [RequestBlockNumbersFromMobile](#) (t_SwupCom ComPort, t_EraseBlocks **ppEraseBlocks, t_Area *pEraseArea, unsigned short *punAmountBlocks)
- BOOL [WSwupSendStartHashing](#) (t_SwupCom ComPort, t_SignatureUpdateMode *pSignMode)
- void [SignatureTransmissionProgress](#) (WORD wUpdateNr, unsigned long ulBytes)
- void [HashListTransmissionProgress](#) (WORD wUpdateNr, unsigned long ulBytes)

- BOOL [NegotiateSignatureParameter](#) (t_SwupCom ComPort, t_SignatureUpdateMode *pSignMode)
- BOOL [WSwup_USB_ModeSwitch](#) (WORD WhichUsb, unsigned char UsbControl)
forces OMAP to switch ReceiveMode same layer
- void [dummy](#) (unsigned short UpdateNr, short percent)
- unsigned int [WSwup_Usb_GetVoltage](#) (WORD wUsbPort)
asks the modem's voltage via SwupHelpTsap via Usb same layer
- BOOL [SipcLoadUpdateSw](#) (WORD wUsbPort, unsigned char ucLoadParam)
initiates modem to leave CoproMode, restart in SwupMode, load its UpdateSw and start it same layer
- BOOL [WSwup_UsbReconnected](#) (WORD wUsbPort)
transmits a SipcFrame to the Omap to tell it that the Usb connection is successfully reinstalled after mobiles reset same layer

Variables

- [t_UsbInfo UsbSettings](#) [AMOUNT_OF_UPDATES] = {0}
- [PFN_USBOMAPSWUP_GETDEVICE](#) [pfnUsbOmapSwup_GetDevice](#) = DummyUsbOmapSwup_-GetDevice
- [PFN_USBOMAPSWUP_RELEASEDEVICE](#) [pfnUsbOmapSwup_ReleaseDevice](#) = DummyUsbOmapSwup_ReleaseDevice
- [PFN_USBOMAPSWUP_WRITETOCONTROLPIPE](#) [pfnUsbOmapSwup_WriteToControlPipe](#) = DummyUsbOmapSwup_WriteToControlPipe
- [PFN_USBOMAPSWUP_WRITETODATAPIPE](#) [pfnUsbOmapSwup_WriteToDataPipe](#) = DummyUsbOmapSwup_WriteToDataPipe
- [PFN_USBOMAPSWUP_READFROMDATAPIPE](#) [pfnUsbOmapSwup_ReadFromDataPipe](#) = DummyUsbOmapSwup_ReadFromDataPipe
- HINSTANCE [hIBootLib](#) = NULL

Loads the IBootDll and retrieves pointers to all necessary functions.

8.13.1 Detailed Description

Layer-3-functionality for the serial communication, Part of the MAIN-DLL.

Contains all functions that fill a message-block and send this message-block via Layer2-functions. Some receive-decoding routines are also in here.

Todo

- Make some comments either here or in the export-header wswserl3.h. -Make a decision about where to put the comments.

8.13.2 Define Documentation

8.13.2.1 **#define** CONTROL_BIN_BLOCKS_SUPPORTED 0xAA

8.13.2.2 **#define** ERASE_AREA_NOT_OK 0xFF

8.13.2.3 **#define** ERASE_AREA_OK 0x00

8.13.2.4 **#define** ERASE_TIME_FLASH_SEGMENT 10000

8.13.2.5 **#define** M_SWH_VOLTAGE 0x02

8.13.2.6 **#define** W_SWUPHELPSAPI 0x0e

8.13.3 Typedef Documentation

8.13.3.1 **typedef** IBOOT_API BOOL(* [PFN_USBOMAPSWUP_GETDEVICE](#))(WORD wUsbPort, BOOL fForceSavedPort)

8.13.3.2 **typedef** IBOOT_API BOOL(* [PFN_USBOMAPSWUP_READFROMDATAPIPE](#))(WORD wUsbPort, BYTE *InBuffer, long InBufferLen, ULONG *nBytesRead)

8.13.3.3 **typedef** IBOOT_API BOOL(* [PFN_USBOMAPSWUP_RELEASEDEVICE](#))(WORD wUsbPort)

8.13.3.4 **typedef** IBOOT_API BOOL(* [PFN_USBOMAPSWUP_WRITETOCONTROLPIPE](#))(WORD wUsbPort, BYTE *OutBuffer, short OutBufferLen)

8.13.3.5 **typedef** IBOOT_API BOOL(* [PFN_USBOMAPSWUP_WRITETODATAPIPE](#))(WORD wUsbPort, BYTE *OutBuffer, long OutBufferLen, void (*UpdateTransferStatus)(WORD, short))

8.13.4 Function Documentation

8.13.4.1 **BOOL** DeactIBootLibrary (void)

unloads functionpointers of iBootLib same layer

Parameters:

none

Returns:

success/nosuccess

- 8.13.4.2 void dummy (unsigned short *UpdateNr*, short *percent*)
- 8.13.4.3 BOOL DummyUsbOmapSwup_GetDevice (WORD *wUsbPort*, BOOL *fForceSavedPort*)
- 8.13.4.4 BOOL DummyUsbOmapSwup_ReadFromDataPipe (WORD *wUsbPort*, BYTE * *InBuffer*, long *InBufferLen*, ULONG * *nBytesRead*)
- 8.13.4.5 BOOL DummyUsbOmapSwup_ReleaseDevice (WORD *wUsbPort*)
- 8.13.4.6 BOOL DummyUsbOmapSwup_WriteToControlPipe (WORD *wUsbPort*, BYTE * *OutBuffer*, short *OutBufferLen*)
- 8.13.4.7 BOOL DummyUsbOmapSwup_WriteToDataPipe (WORD *wUsbPort*, BYTE * *OutBuffer*, long *OutBufferLen*, void(* *UpdateTransferStatus*)(WORD, short))
- 8.13.4.8 BOOL EnableWriteFlashBlock (t_SwupCom *ComPort*, unsigned long *BlockToUnlock*)
- 8.13.4.9 BOOL EraseFlashBlocksWithAcknowledge (t_SwupCom *ComPort*, t_EraseBlocks * *pEraseBlocks*, WORD *wUpdateNr*, WORD *wMaxAmount*, EraseProgressCallBack *Progress*)
- 8.13.4.10 BOOL EraseWithNewUpdateConcept (t_SwupCom *ComPort*, WORD *wUpdateNr*, EraseProgressCallBack *Progress*)
- 8.13.4.11 void HashListTransmissionProgress (WORD *wUpdateNr*, unsigned long *ulBytes*)
- 8.13.4.12 BOOL InitIBootLibrary (void)

load functionpointers of iBootLib static Lib impossible because NT does not support USB same layer

Parameters:

none

Returns:

success/nosuccess

- 8.13.4.13 BOOL MobileUpdateConceptRunningHandler (t_SwupCom *ComPort*, unsigned char *ucMi*, unsigned int *unLen*, unsigned char * *abBuffer*)
- 8.13.4.14 BOOL NegotiateSignatureParameter (t_SwupCom *ComPort*, t_SignatureUpdateMode * *pSignMode*)
- 8.13.4.15 BOOL RequestBlockNumbersFromMobile (t_SwupCom *ComPort*, t_EraseBlocks ** *ppEraseBlocks*, t_Area * *pEraseArea*, unsigned short * *punAmountBlocks*)
- 8.13.4.16 void SignatureTransmissionProgress (WORD *wUpdateNr*, unsigned long *ulBytes*)
- 8.13.4.17 BOOL SipcLoadUpdateSw (WORD *wUpdateNr*, unsigned char *ucLoadParam*)

initiates modem to leave CoproMode, restart in SwupMode, load its UpdateSw and start it same layer

Parameters:

wUpdateNr

8.13.4.18 unsigned int WSwup_Usb_GetVoltage (WORD *wUpdateNr*)

asks the modem's voltage via SwupHelpTsap via Usb same layer

Parameters:

wUpdateNr

Returns:

voltage

8.13.4.19 BOOL WSwup_USB_ModeSwitch (WORD *wUpdateNr*, unsigned char *ucUsbControl*)

forces OMAP to switch ReceiveMode same layer

Parameters:

wUpdateNr

ReceiveMode

Returns:

success/nosuccess

8.13.4.20 BOOL WSwup_UsbReconnected (WORD *wUpdateNr*)

transmits a SipcFrame to the Omap to tell it that the Usb connection is successfully reinstalled after mobiles reset same layer

Parameters:

wUpdateNr

- 8.13.4.21 **BOOL** WSwupEnableFlashProgrammingWholeFlash (t_SwupCom ComPort)
- 8.13.4.22 **BOOL** WSwupEraseFlashBlocks (t_SwupCom ComPort, unsigned int *unFirst*, unsigned int *unLast*)
- 8.13.4.23 **BOOL** WSwupEraseOneFlashBlock (t_SwupCom ComPort, unsigned int *unNumber*)
- 8.13.4.24 **BOOL** WSwupGiveMobileSwVersion (t_SwupCom ComPort)
- 8.13.4.25 **BOOL** WSwupRequestEraseBlocks (t_SwupCom ComPort, unsigned long *ulFrom*, unsigned long *ulTo*)
- 8.13.4.26 **BOOL** WSwupRequestFlashCode (t_SwupCom ComPort, unsigned char *HowToWrite*)
- 8.13.4.27 **BOOL** WSwupRequestFlashSize (t_SwupCom ComPort)
- 8.13.4.28 **BOOL** WSwupSelectMobileMode (t_SwupCom ComPort, t_MsmLowMobileStationMode *NextMode*, t_MsmLowMobileStationMode *PrevMode*)
- 8.13.4.29 **BOOL** WSwupSendAdditionalInfo (t_SwupCom ComPort, unsigned char * *pucData*, unsigned short *unLenData*)
- 8.13.4.30 **BOOL** WSwupSendAlignement (t_SwupCom ComPort, unsigned short *unAlignement*)
- 8.13.4.31 **BOOL** WSwupSendBlockTimeOut (t_SwupCom ComPort, unsigned int *unTicks*)
- 8.13.4.32 **BOOL** WSwupSendByteTimeOut (t_SwupCom ComPort, unsigned int *unSek*)
- 8.13.4.33 **BOOL** WSwupSendCommandData (t_SwupCom ComPort, unsigned char ** *ppabSource*, unsigned long * *pulLenToSend*, DWORD *dwRecTimeOut*, WORD *wNumber*)
- 8.13.4.34 **BOOL** WSwupSendCompressionInfo (t_SwupCom ComPort, unsigned short *unAlgorithmNumber*, unsigned char *ucFormatBefore*, unsigned char *ucFormatAfter*, unsigned short *unAddInfo1*, unsigned short *unAddinfo2*, unsigned short *unAddInfo3*)
- 8.13.4.35 **BOOL** WSwupSendData (t_SwupCom ComPort, unsigned char * *pabBuffer*, unsigned long *ulLenToSend*, DWORD *dwRecTimeOut*, WORD *wNumber*, void(* *Progress*)(WORD, unsigned long))
- 8.13.4.36 **BOOL** WSwupSendEotBlock (t_SwupCom ComPort)
- 8.13.4.37 **BOOL** WSwupSendEraseCodeAreaNewUpdateConcept (t_SwupCom ComPort)
- 8.13.4.38 **BOOL** WSwupSendLanguageGroupInfo (t_SwupCom ComPort, unsigned long *ulStart*, unsigned long *ulTo*, unsigned long *ulXorAdr*, unsigned long *ulPatchAdr*)
- 8.13.4.39 **BOOL** WSwupSendNewSplitInfo (t_SwupCom ComPort, unsigned long *ulSplitStart*, unsigned long *ulSplitEnd*, unsigned long *ulSplitCheckAdd*)
- 8.13.4.40 **BOOL** WSwupSendSplitIDInfo (t_SwupCom ComPort, unsigned long *ulIdAdress*)
- 8.13.4.41 **BOOL** WSwupSendStartHashing (t_SwupCom ComPort, t_SignatureUpdateMode * *pSignMode*)
- 8.13.4.42 **BOOL** WSwupSendSwitchOff (t_SwupCom ComPort)
- 8.13.4.43 **BOOL** WSwupSendSwitchOffCompression (t_SwupCom ComPort)

8.13.5 Variable Documentation

- 8.13.5.1 **HINSTANCE** [hIBootLib](#) = NULL

- 8.13.5.2 **PFN_USBOMAPSWUP_GETDEVICE** `pfnUsbOmapSwup_GetDevice = DummyUsbOmapSwup_GetDevice`
- 8.13.5.3 **PFN_USBOMAPSWUP_READFROMDATAPIPE** `pfnUsbOmapSwup_ReadFromDataPipe = DummyUsbOmapSwup_ReadFromDataPipe`
- 8.13.5.4 **PFN_USBOMAPSWUP_RELEASEDEVICE** `pfnUsbOmapSwup_ReleaseDevice = DummyUsbOmapSwup_ReleaseDevice`
- 8.13.5.5 **PFN_USBOMAPSWUP_WRITETOCONTROLPIPE** `pfnUsbOmapSwup_WriteToControlPipe = DummyUsbOmapSwup_WriteToControlPipe`
- 8.13.5.6 **PFN_USBOMAPSWUP_WRITETODATAPIPE** `pfnUsbOmapSwup_WriteToDataPipe = DummyUsbOmapSwup_WriteToDataPipe`
- 8.13.5.7 **t_UsbInfo** `UsbSettings[AMOUNT_OF_UPDATES] = {0}`

8.14 wsw_seri.h File Reference

Data Structures

- struct [t_ErrorField](#)
A struct containing all errors from the indicated update, bIndex is a pointer to the next free t_ErrorStruct.
- struct [tag_UsbInfo](#)
A struct containing all information needed for UsbUpdate.
- struct [tagMessageBlock](#)
A structure containing a message-block with Message-Identifier, Len and additional Data.

Defines

- #define [SWUPSERILOWEXIMPORT](#) __declspec(dllimport)
- #define [ArrayEntries\(x\)](#) (sizeof(x) / sizeof(x[0]))
- #define [DEFAULT_TIMEOUT_SINGLE_CHAR](#) 10
Milli-Seconds.
- #define [DEFAULT_TIMEOUT_BLOCK_REC](#) 100
Milli-Seconds.
- #define [ACK](#) 0x06
Standard-Communication: ACK.
- #define [ACK_HOLD](#) 0x9B
Standart-Comunication: ACK_HOLD.
- #define [NAK](#) 0x15
Standard-Communication: NAK.
- #define [A5_ACK](#) 0xA5
EGOLD-Mask-Communication: Okay.
- #define [MAX_DATA_IN_BIN_BLOCK](#) 64
Max amount of Data in one BIN-Block.
- #define [ADRESS_LEN_BIN_BLOCK](#) 3
Amount of Bytes that form the adress of a BIN-Block.
- #define [MAX_RETRY_ON_NAK](#) 5
Max amount of Retries.
- #define [USB_SIPC_MODE](#) 0xFD
force OMAP into Sipc-protocol
- #define [USB_NORMAL_MODE](#) 0xFE

force OMAP into normal USB-protocol

- #define [WSwInternalSetLastError\(x, y\)](#) [WSwInternalSetLastErrorWithFileInfo\(\(x\), \(y\), __FILE__, __LINE__ \)](#)

Typedefs

- typedef BOOL(* [pfnDefaultHandler](#))(t_SwupCom ComPort, unsigned char ucMi, unsigned int unLen, unsigned char *abBuffer)
- typedef [tagMessageBlock](#) [t_MessageBlock](#)
A structure containing a message-block with Message-Identifier, Len and additional Data.
- typedef [tag_UsbInfo](#) [t_UsbInfo](#)
A struct containing all information needed for UsbUpdate.

Enumerations

- enum [t_SwupParity](#) { [SwupNoParity](#), [SwupEvenParity](#), [SwupOddParity](#) }
Different enumeration-values for the parity of the serial communication.
- enum [t_RecResult](#) {
[enRecOk](#), [enRecTimeout](#), [enMobileError](#), [enSerialError](#),
[enCmdBlockReceived](#), [enChkErr](#), [enUnknownChar](#), [enIncomplete](#),
[enSipcCmd](#) }
Different enumeration-values for the result of receiving data.
- enum [t_KnownReceiveModes](#) { [enBinMode](#), [enLenChkMode](#), [enSipcMode](#) }
Different enumeration-values for send and reception-protocol.

Functions

- BOOL SWUPSERILOWEXIMPORT [WSwupSeriLowGiveVersion](#) (t_VersionInformation *pInfo)
Returns information about the Serial-Low-DLL.
- BOOL SWUPSERILOWEXIMPORT [WSwupComOpen](#) (t_SwupCom nComPort, DWORD [dwBaudrate](#), [t_SwupParity](#) WhichParity)
Opens the given Com-Port with the given Baud-Rate and Parity.
- BOOL SWUPSERILOWEXIMPORT [WSwupComConfigure](#) (t_SwupCom nComPort, DWORD [dwBaudrate](#))
Changes the Baudrate of the Given ComPort.
- BOOL SWUPSERILOWEXIMPORT [WSwupComClose](#) (t_SwupCom nComPort)
Closes the given Com-Port.
- BOOL SWUPSERILOWEXIMPORT [WSwupComWriteByte](#) (t_SwupCom nComPort, BYTE b-Value)

Writes a single Byte to the given ComPort.

- **BOOL** SWUPSERILOWEXIMPORT [WSwupComWriteDataBlock](#) (t_SwupCom ComPort, BYTE *pbValue, DWORD len)

Writes a Data-Block to the given comport.

- **BOOL** SWUPSERILOWEXIMPORT [WSwupComReadByte](#) (t_SwupCom nComPort, LPBYTE lpbByte)

Reads a Byte from the given serial port.

- **BOOL** SWUPSERILOWEXIMPORT [WSwupComReadByteTimeout](#) (t_SwupCom nComPort, LPBYTE lpbByte, DWORD dwTimeout)

Reads a Byte from the given serial port with timeout.

- **BOOL** SWUPSERILOWEXIMPORT [WSwupSetIgnition](#) (t_SwupCom ComPort)

Sets the Ignition-Port on the serial port (DTR-Line).

- **BOOL** SWUPSERILOWEXIMPORT [WSwupClearIgnition](#) (t_SwupCom ComPort)

Clears the Ignition-Port on the serial port (DTR-Line).

- **void** SWUPSERILOWEXIMPORT [WSwupSendBotBlock](#) (t_SwupCom nComPort, unsigned char *pucSendBuf, unsigned int unDatLen)

Sends a so called BOT-Block to the Mobile.

- **void** SWUPSERILOWEXIMPORT [WSwupSendSGoldBotBlock](#) (t_SwupCom ComPort, unsigned char *pucSendBuf, unsigned int unDatLen)

Sends a so called BOT-Block to the SGOLD-Mobile.

- **t_RecResult** SWUPSERILOWEXIMPORT [WSwupReceiveValueFromSerial](#) (t_SwupCom nComPort, unsigned char *pucRecBuf, DWORD dwTimeout)

Read a single byte from the serial Port.

- **void** SWUPSERILOWEXIMPORT [WSwupSetCurrentReceiveMode](#) (t_SwupCom nComPort, [t_KnownReceiveModes](#) WhichMode)

Sets the current receive-mode .

- [t_KnownReceiveModes](#) SWUPSERILOWEXIMPORT [WSwupGetCurrentReceiveMode](#) (t_SwupCom nComPort)

- **BOOL** SWUPSERILOWEXIMPORT [WSwupReceiveCmdBlock](#) (t_SwupCom ComPort, [t_MessageBlock](#) *pMessageBlock, unsigned char ucExpectedMi, DWORD dwTimeout)

Waites for a CMD-Block with the given Message-Identifier on the serial port.

- **BOOL** SWUPSERILOWEXIMPORT [WSwupSendMessageBlock](#) (t_SwupCom ComPort, unsigned char ucMi, unsigned short unLen, unsigned char *pabBuffer)

Sends a CMD-Block (currently only in BIN-Mode).

- **BOOL** SWUPSERILOWEXIMPORT [InstallDefaultHandlerFunction](#) (unsigned char ucThisMi, [pfn_DefaultHandler](#) ThisDefaultHandler)

Installs a default handler-function for a specific Message-Identifier.

- `BOOL SWUPSERILOWEXIMPORT WSwSendDataBlockInBinFormat` (`t_SwupCom ComPort`, unsigned char *pabBuffer, `DWORD dwRecTimeOut`, unsigned short *pAmountSent)
- `BOOL SWUPSERILOWEXIMPORT WSwSendDataBlockInSipcFormat` (`WORD UsbPort`, unsigned char *pabBuffer, unsigned short *pAmountSent)
- `void SWUPSERILOWEXIMPORT WSwupDebugString` (`WORD wUpdateNr`, `t_DebugLevel Show-OnDebugLevel`, `WORD wDebugGroup`, const char *pszFormat,...)
Debugging to files and to standard-debug-port.
- `void SWUPSERILOWEXIMPORT WSwupInt_EnableOnlineDebugging` (`t_DebugLevel Requested-DebugLevel`, `WORD wRequestedDebugGroups`)
Enables Debugging.
- `void SWUPSERILOWEXIMPORT WSwupInt_DisableOnlineDebugging` (`void`)
Disables Debugging.
- `void SWUPSERILOWEXIMPORT WSwupInt_EnableDebuggingToFile` (`t_DebugLevel Requested-DebugLevel`, `WORD wRequestedDebugGroups`, char *pszFileNamePrefix)
Enables Debugging to File.
- `void SWUPSERILOWEXIMPORT WSwupInt_DisableDebuggingToFile` (`void`)
Disables Debugging to File.
- `WORD SWUPSERILOWEXIMPORT WSwupInt_GiveUpdateNrFromComPort` (`t_SwupCom ComPort`)
Gives UpdateNumber in dependency with ComPort.
- `void SWUPSERILOWEXIMPORT WSwupInt_SetUpdateNrToComPort` (`t_SwupCom ComPort`, `WORD wUpdateNr`)
Sets UpdateNumber in dependency from ComPort.
- `void SWUPSERILOWEXIMPORT WSwInternalSetLastErrorWithFileInfo` (`t_SwupCom ComPort`, `DWORD dwLastError`, char *pszFileName, int nLine)
Used to trace and store error-codes and descriptions.
- `void SWUPSERILOWEXIMPORT WSwSetLastErrorText` (`t_SwupCom ComPort`, const char *pszFormat,...)
Sets an error-text for for the last error.
- `DWORD SWUPSERILOWEXIMPORT WSwupInt_GetLastError` (`t_SwupCom ComPort`)
Returns last error-value for this comport.
- `char SWUPSERILOWEXIMPORT * WSwupInt_GetLastErrorString` (`t_SwupCom ComPort`)
Returns last error-string for this comport.
- `void SWUPSERILOWEXIMPORT WSwupIntSetError` (`WORD wUpdateNr`, unsigned int en_Error, unsigned char *ucInfo,...)
Sets an error-number for the last error into a global variable, defined in wsw_seri.h.
- `void SWUPSERILOWEXIMPORT WSwupIntSetErrorAllUpdates` (`WORD wUpdateNr`, unsigned int en_Error, unsigned char *ucInfo, va_list point)
Sets an error-number for the last error into a global variable, defined in wsw_seri.h.

- void SWUPSERILOWEXIMPORT [WSwupIntForAllUpdatesSetError](#) (unsigned int enError, unsigned char *ucInfo,...)
Sets an error-number for the last error into a global variable, defined in [wsw_seri.h](#), valid for updates.
- SWUPSERILOWEXIMPORT t_ErrorStruct * [WSwupIntGiveLastError](#) (WORD wUpdateNr)
Returns the last Error for the indicated updatenumber with its specific parameters declaration of t_ErrorStruct see above.
- SWUPSERILOWEXIMPORT t_ErrorField * [WSwupIntGiveErrorList](#) (WORD wUpdateNr)
Returns the complete Errorlist for the indicated updatenumber with its specific parameters declaration of [t_ErrorField](#) : typedef struct { BYTE bIndex; // the index for enum field next to filled t_ErrorStruct Error [MAX_ENTRIES]; }t_ErrorField; declaration of t_ErrorStruct see WSwupGiveLastError maximum entries : MAX_ENTRIES for each updatenumber.
- HANDLE SWUPSERILOWEXIMPORT [WSwupGiveHandleForComPort](#) (t_SwupCom ComPort)
Returns the windows handle for a given ComPort.
- BOOL [SipcLoadUpdateSw](#) (WORD wUpdateNr, unsigned char ucLoadParam)
initiates modem to leave CoproMode, restart in SwupMode, load its UpdateSw and start it same layer
- BOOL [WSwup.UsbReconnected](#) (WORD wUpdateNr)
transmits a SipcFrame to the Omap to tell it that the Usb connectionis successfully reinstalled after mobiles reset same layer
- unsigned int [WSwup.Usb.GetVoltage](#) (WORD wUpdateNr)
asks the modem's voltage via SwupHelpTsap via Usb same layer
- BOOL [WSwup.USB.ModeSwitch](#) (WORD wUpdateNr, unsigned char ucUsbControl)
forces OMAP to switch ReceiveMode same layer
- BOOL SWUPSERILOWEXIMPORT [WSwSipcWriteToDataPipe_CoproMode](#) (WORD wUpdateNr, unsigned char *ucOutBuffer, unsigned char ucExpectedSapi, unsigned char *ucRetBuffer, [t_RecResult](#) *RecResult)
sends a bfb message to modem's CoproMode via Usb same layer
- void SWUPSERILOWEXIMPORT [SendSimpleMessageInSipcMode](#) (WORD wUsbPort, unsigned char ucMi, unsigned char ucPayload)
Sends message in SIPC format without waiting for response.
- BOOL [InitIBootLibrary](#) (void)
load functionpointers of iBootLib static Lib impossible because NT does not support USB same layer
- BOOL [DeactIBootLibrary](#) (void)
unloads functionpointers of iBootLib same layer

8.14.1 Define Documentation

8.14.1.1 #define A5_ACK 0xA5

EGOLD-Mask-Communication: Okay.

8.14.1.2 #define ACK 0x06

Standard-Communication: ACK.

8.14.1.3 #define ACK_HOLD 0x9B

Standart-Communication: ACK_HOLD.

8.14.1.4 #define ADRESS_LEN_BIN_BLOCK 3

Amount of Bytes that form the adress of a BIN-Block.

8.14.1.5 #define ArrayEntries(x) (sizeof(x) / sizeof(x[0]))**8.14.1.6 #define DEFAULT_TIMEOUT_BLOCK_REC 100**

Milli-Seconds.

8.14.1.7 #define DEFAULT_TIMEOUT_SINGLE_CHAR 10

Milli-Seconds.

8.14.1.8 #define MAX_DATA_IN_BIN_BLOCK 64

Max amount of Data in one BIN-Block.

8.14.1.9 #define MAX_RETRY_ON_NAK 5

Max amount of Retries.

8.14.1.10 #define NAK 0x15

Standard-Communication: NAK.

8.14.1.11 #define SWUPSERILOWEXIMPORT __declspec(dllimport)**8.14.1.12 #define USB_NORMAL_MODE 0xFE**

force OMAP into normal USB-protocol

8.14.1.13 #define USB_SIPC_MODE 0xFD

force OMAP into Sipc-protocol

8.14.1.14 `#define WSwInternalSetLastError(x, y) WSwInternalSetLastErrorWithFileInfo( (x), __FILE__, __LINE__ )`

8.14.2 Typedef Documentation

8.14.2.1 `typedef BOOL( * pfnDefaultHandler)( t_SwupCom ComPort, unsigned char ucMi, unsigned int unLen, unsigned char * abBuffer )`

8.14.2.2 `typedef struct tagMessageBlock t_MessageBlock`

A structure containing a message-block with Message-Identifier, Len and additional Data.

8.14.2.3 `typedef struct tag\_UsbInfo t_UsbInfo`

A struct containing all information needed for UsbUpdate.

8.14.3 Enumeration Type Documentation

8.14.3.1 `enum t\_KnownReceiveModes`

Different enumeration-values for send and reception-protocol.

Only BIN-Mode is currently fully implemented, LENCHK-Mode is only prepared in a few cases.

Enumeration values:

enBinMode normal BIN-Mode : 3 Bytes Adr, 1 Byte Len, Data, 1 Byte Checksum

enLenChkMode new mode, not fully implemented, 1 Byte Len, Data, 1 Byte Checksum

enSipcMode used for SIPC-protocol via USB (ep55k1)

8.14.3.2 `enum t\_RecResult`

Different enumeration-values for the result of receiving data.

Enumeration values:

enRecOk Receive was okay.

enRecTimeout a timeout occurred during trying to receive data

enMobileError an error in the mobile occurred during trying to receive data

enSerialError a serial error occurred during trying to receive data

enCmdBlockReceived a Message-Block was received, maybe instead of an answer to the sent data

enChkErr a checksum-error occurred during trying to receive data

enUnknownChar unspected character received

enIncomplete incoming message via ReadFromDataPipe (USB) is not complete

enSipcCmd SipcCmd message.

8.14.3.3 enum [t_SwupParity](#)

Different enumeration-values for the parity of the serial communication.

Enumeration values:

- SwupNoParity** Communication without parity.
- SwupEvenParity** Communication with even parity.
- SwupOddParity** Communication with odd parity.

8.14.4 Function Documentation

8.14.4.1 BOOL DeactIBootLibrary (void)

unloads functionpointers of iBootLib same layer

Parameters:

none

Returns:

success/nosuccess

8.14.4.2 BOOL InitIBootLibrary (void)

load functionpointers of iBootLib static Lib impossible because NT does not support USB same layer

Parameters:

none

Returns:

success/nosuccess

8.14.4.3 BOOL SipcLoadUpdateSw (WORD wUpdateNr, unsigned char ucLoadParam)

initiates modem to leave CoproMode, restart in SwupMode, load its UpdateSw and start it same layer

Parameters:

wUpdateNr

8.14.4.4 void SWUPSERILOWEXIMPORT WSwInternalSetLastErrorWithFileInfo (t_SwupCom ComPort, DWORD dwLastError, char * pszFileName, int nLine)

Used to trace and store error-codes and descriptions.

Dont call this function direct, use instead macro : [WSwInternalSetLastError\(\)](#), so that you must not deal with File-Name and Code Line ! This function must be called before [WSwSetLastErrorText\(\)](#) , because the text is set to StandardText

Parameters:

ComPort : the comport

dwLastError : Error-Number
pszFileName : FileName of Source-File (via Macro)
nLine : Line-Number in Source-File (via Macro)

Returns:
 void

8.14.4.5 **BOOL SWUPSERILOWEXIMPORT WSwSendDataBlockInBinFormat (t_SwupCom ComPort, unsigned char * *pabBuffer*, DWORD *dwRecTimeOut*, unsigned short * *pAmountSent*)**

8.14.4.6 **BOOL SWUPSERILOWEXIMPORT WSwSendDataBlockInSipcFormat (WORD *UsbPort*, unsigned char * *pabBuffer*, unsigned short * *pAmountSent*)**

8.14.4.7 **void SWUPSERILOWEXIMPORT WSwSetLastErrorText (t_SwupCom *ComPort*, const char * *pszFormat*, ...)**

Sets an error-text for for the last error.

Only call this function if you cant use the standardtext. Must be called after WSwInternalSetLastError, otherwise this Text will be overwritten. Can be used like printf.

Parameters:

ComPort : the comport
pszFormat : the Format-String (like printf)
 ... : Additional parameters (like printf)

Returns:
 void

8.14.4.8 **BOOL SWUPSERILOWEXIMPORT WSwSipcWriteToDataPipe_CoproMode (WORD *wUsbPort*, unsigned char * *ucOutBuffer*, unsigned char *ucExpectedSapi*, unsigned char * *ucRetBuffer*, **t_RecResult** * *RecResult*)**

sends a bfb message to modem's CoproMode via Usb same layer

Then it waits for the answer of the Mobile via function ????? If the received value is a single char and an ACK, this routine returns TRUE, if it is not ACK, the Block is sent once more, this up to the value of MAX_RETRY_ON_NAK

Parameters:

wUsbPort: the UsbPort to receive the message-block, same as UpdateNr
ucMi : the Message-Identifier
unLen : the amount of data following in
pabBuffer : the buffer to the data to send

Return values:

TRUE -> Message-Block was sent
FALSE -> an error occurred

8.14.4.9 unsigned int WSwup_Usb_GetVoltage (WORD wUpdateNr)

asks the modem's voltage via SwupHelpTsap via Usb same layer

Parameters:

wUpdateNr

Returns:

voltage

8.14.4.10 BOOL WSwup_USB_ModeSwitch (WORD wUpdateNr, unsigned char ucUsbControl)

forces OMAP to switch ReceiveMode same layer

Parameters:

wUpdateNr

ReceiveMode

Returns:

success/nosuccess

8.14.4.11 BOOL WSwup_UsbReconnected (WORD wUpdateNr)

transmits a SipcFrame to the Omap to tell it that the Usb connection is successfully reinstalled after mobiles reset same layer

Parameters:

wUpdateNr

8.14.4.12 t_KnownReceiveModes SWUPSERILOWEXIMPORT WSwupGetCurrentReceiveMode (t_SwupCom nComPort)**8.14.4.13 HANDLE SWUPSERILOWEXIMPORT WSwupGiveHandleForComport (t_SwupCom ComPort)**

Returns the windows handle for a given ComPort.

This function is necessary because there is an external interface to INFINEON-DLL: SG_RAM_Loader.dll.

Parameters:

ComPort The ComPort , see t_SwupCom

8.14.4.14 DWORD SWUPSERILOWEXIMPORT WSwupInt_GetLastError (t_SwupCom ComPort)

Returns last error-value for this comport.

This is only the error-value, in the moment this value does not make any sense outside this module.

Parameters:

ComPort : Last Error is to be retrieved for this comport

Returns:

Last error-value for this comport

8.14.4.15 **char SWUPSERILOWEXIMPORT* WSwupInt_GetLastErrorString (t_SwupCom ComPort)**

Returns last error-string for this comport.

This returns a pointer to a descriptive error-string. Parameters-In: ComPort

Parameters:

ComPort : Last Error is to be retrieved for this comport

Returns:

Last error-string for this comport

8.14.4.16 **void SWUPSERILOWEXIMPORT WSwupIntForAllUpdatesSetError (unsigned int enError, unsigned char * ucInfo, ...)**

Sets an error-number for the last error into a global variable, defined in [wsw_seri.h](#), valid for updates.

Parameters:

enError : the Error-number

ucInfo : indicates the kind of the following parameters, e.g. s for string, x,d,i,o,u for numerals, no l...! e.g. "sxsd", maximum SIZE_OF_PINFO Entries (10 at the moment), upper and lower chars allowed maximum 5 strings and 5 non-strings just like d,X... in printf, without it's assumed that every string parameter ends with '0' params : maximum 5 LONG values and maximum 5 char pointers(for max 2000 chars) please don't forget to cast the non-strings to LONG in every call

Returns:

void

8.14.4.17 **SWUPSERILOWEXIMPORT t_ErrorField* WSwupIntGiveErrorList (WORD wUpdateNr)**

Returns the complete Errorlist for the indicated updatenumber with its specific parameters declaration of [t_ErrorField](#) : typedef struct { BYTE bIndex; // the index for enum field next to filled t_ErrorStruct Error [MAX_ENTRIES]; } [t_ErrorField](#); declaration of t_ErrorStruct see WSwupGiveLastError maximum entries : MAX_ENTRIES for each updatenumber.

Parameters:

wUpdateNr : the current Update-number

Returns:

error-number

8.14.4.18 **SWUPSERILOWEXIMPORT t_ErrorStruct* WSwupIntGiveLastError (WORD wUpdateNr)**

Returns the last Error for the indicated updatenumber with its specific parameters declaration of t_ErrorStruct see above.

Parameters:

wUpdateNr : the current Update-number

Returns:

error-number

8.14.4.19 **void SWUPSERILOWEXIMPORT WSwupIntSetError (WORD wUpdateNr, unsigned int en_Error, unsigned char * ucInfo, ...)**

Sets an error-number for the last error into a global variable, defined in [wsw_seri.h](#).

variable parameters

Parameters:

wUpdateNr : the current Update-number

en_Error : the Error-number

ucInfo : indicates the kind of the following parameters, e.g. s for string, x,d,i,o,u for numerals, no l...! e.g. "sxs d", maximum SIZE_OF_PINFO Entries (10 at the moment), upper and lower chars allowed maximum 5 strings and 5 non-strings just like d,X... in printf, without it's assumed that every string parameter ends with '0' params : maximum 5 LONG values and maximum 5 char pointers(for max 2000 chars) please don't forget to cast the non-strings to LONG in every call

Returns:

void

for further information :typedef struct { DWORD dwErrorNumber; // the error-enum unsigned char ucParamInfo[SIZE_OF_PINFO]; // info about the validity of the following params LONG ulParamValue[5]; // variables that will be inserted in the indicated string at d,x,... unsigned char ucParamString[5][SIZE_OF_PSTRING]; // variables that will be inserted in the indicated string at s } t_ErrorStruct;

8.14.4.20 **void SWUPSERILOWEXIMPORT WSwupIntSetErrorAllUpdates (WORD wUpdateNr, unsigned int en_Error, unsigned char * ucInfo, va_list point)**

Sets an error-number for the last error into a global variable, defined in [wsw_seri.h](#).

for all updates called by WSwupIntForAllUpdatesSetError pointer to variable parameterlist

Parameters:

wUpdateNr : the current Update-number

en_Error : the Error-number

ucInfo : indicates the kind of the following parameters, e.g. s for string, x,d,i,o,u for numerals, no l...! e.g. "sxs d", maximum SIZE_OF_PINFO Entries (10 at the moment), upper and lower chars allowed maximum 5 strings and 5 non-strings just like d,X... in printf, without it's assumed that every string parameter ends with '0'

point : has to be defined as va_list

Returns:

void

8.15 wsw_tool.c File Reference

Some tools like GetWord and the stuff, part of the MAIN-DLL.

```
#include <windows.h>
```

Functions

- unsigned short [GetIntelShort](#) (unsigned char *abBuffer)
- void [InsertLongToStringIntel](#) (unsigned char *pucBuffer, unsigned long ulVal)
- void [InsertWordToStringIntel](#) (unsigned char *pucBuffer, unsigned short unVal)
- void [InsertLongToStringMotorola](#) (unsigned char *pucBuffer, unsigned long ulVal)
- unsigned long [GetMotorolaLongFromBuffer](#) (unsigned char *pucBuffer)

8.15.1 Detailed Description

Some tools like GetWord and the stuff, part of the MAIN-DLL.

Todo

Make at least some comments.

8.15.2 Function Documentation

8.15.2.1 unsigned short GetIntelShort (unsigned char * *abBuffer*)

8.15.2.2 unsigned long GetMotorolaLongFromBuffer (unsigned char * *pucBuffer*)

8.15.2.3 void InsertLongToStringIntel (unsigned char * *pucBuffer*, unsigned long *ulVal*)

8.15.2.4 void InsertLongToStringMotorola (unsigned char * *pucBuffer*, unsigned long *ulVal*)

8.15.2.5 void InsertWordToStringIntel (unsigned char * *pucBuffer*, unsigned short *unVal*)

8.16 wswdbgex.c File Reference

The external-debug-interfaces to the internal functions, part of the MAIN-DLL.

```
#include <windows.h>
#include "wswuplib.h"
#include "wsw_seri.h"
```

Functions

- void SWUPLIBDLLEXIMPORT [WSwup_EnableOnlineDebugging](#) (t_DebugLevel RequestedDebugLevel, WORD wRequestedDebugGroups)
- void SWUPLIBDLLEXIMPORT [WSwup_DisableOnlineDebugging](#) (void)
- void SWUPLIBDLLEXIMPORT [WSwup_EnableFileDebugging](#) (t_DebugLevel RequestedDebugLevel, WORD wRequestedDebugGroups, char *FilePrefix)
- void SWUPLIBDLLEXIMPORT [WSwup_DisableFileDebugging](#) (void)

8.16.1 Detailed Description

The external-debug-interfaces to the internal functions, part of the MAIN-DLL.

Todo

Already commented bot not doxygen-conform.

8.16.2 Function Documentation

8.16.2.1 void SWUPLIBDLLEXIMPORT WSwup_DisableFileDebugging (void)

8.16.2.2 void SWUPLIBDLLEXIMPORT WSwup_DisableOnlineDebugging (void)

8.16.2.3 void SWUPLIBDLLEXIMPORT WSwup_EnableFileDebugging (t_DebugLevel RequestedDebugLevel, WORD wRequestedDebugGroups, char * FilePrefix)

8.16.2.4 void SWUPLIBDLLEXIMPORT WSwup_EnableOnlineDebugging (t_DebugLevel RequestedDebugLevel, WORD wRequestedDebugGroups)

8.17 wswdbgin.c File Reference

Debugging-Functionality contained in WSWUPSER-DLL.

```
#include <windows.h>
#include <stdio.h>
#include <stdarg.h>
#include "wswuplib.h"
#include "wsw_ser_i.h"
```

Data Structures

- struct [tagDebugFiles](#)

Typedefs

- typedef [tagDebugFiles](#) t_DebugFiles

Functions

- void [WSwupDebugString](#) (WORD wUpdateNr, t_DebugLevel ShowOnDebugLevel, WORD w-DebugGroup, const char *pszFormat,...)
Debugging to files and to standard-debug-port.
- void SWUPSERILOWEXIMPORT [WSwupInt_EnableOnlineDebugging](#) (t_DebugLevel Requested-DebugLevel, WORD wRequestedDebugGroups)
Enables Debugging.
- void SWUPSERILOWEXIMPORT [WSwupInt_DisableOnlineDebugging](#) (void)
Disables Debugging.
- void SWUPSERILOWEXIMPORT [WSwupInt_EnableDebuggingToFile](#) (t_DebugLevel Requested-DebugLevel, WORD wRequestedDebugGroups, char *pszFileNamePrefix)
Enables Debugging to File.
- void SWUPSERILOWEXIMPORT [WSwupInt_DisableDebuggingToFile](#) (void)
Disables Debugging to File.
- WORD SWUPSERILOWEXIMPORT [WSwupInt_GiveUpdateNrFromComPort](#) (t_SwupCom Com-Port)
Gives UpdateNumber in dependency with ComPort.
- void SWUPSERILOWEXIMPORT [WSwupInt_SetUpdateNrToComPort](#) (t_SwupCom ComPort, WORD wUpdateNr)
Sets UpdateNumber in dependency from ComPort.

Variables

- `t_DebugFiles DebugFiles` [AMOUNT_OF_UPDATES]
- `t_DebugLevel CurrentOnlineDebugLevel` = `enNoDebug`
CurrentDebug-Level for Online-Debugging, initailaised to not debug.
- WORD `wEnabledOnlineDebugGroups`
the different groups that are enabled for online debugging
- `t_DebugLevel CurrentFileDebugLevel` = `enNoDebug`
CurrentDebug-Level for File-Debugging, initailaised to not debug.
- WORD `wEnabledFileDebugGroups`
the different groups that are enabled for file-debugging
- BOOL `fDebugToFile` = `FALSE`
Bool, No debug to file as default.
- BOOL `fDebugToWindow` = `FALSE`
Bool, No debug to window as default.

8.17.1 Detailed Description

Debugging-Functionality contained in WSWUPSER-DLL.

8.17.2 Typedef Documentation

8.17.2.1 typedef struct `tagDebugFiles t_DebugFiles`

8.17.3 Variable Documentation

8.17.3.1 `t_DebugFiles DebugFiles`[AMOUNT_OF_UPDATES]

8.18 wswerrex.c File Reference

The external-error-interfaces to the internal functions, part of the MAIN-DLL.

```
#include <windows.h>
#include "wswuplib.h"
#include "wsw_ser_i.h"
#include "werrenum.h"
#include "err_text.h"
```

Functions

- `DWORD SWUPLIBDLLEXIMPORT WSwup_GetLastError (t_SwupCom ComPort)`
- `char SWUPLIBDLLEXIMPORT * WSwup_OldGetLastErrorString (t_SwupCom ComPort)`
- `SWUPLIBDLLEXIMPORT t_ErrorStruct * WSwup_GiveLastError (WORD wUpdateNr)`
Returns the last error in dependency from the updatenumber.
- `void SWUPLIBDLLEXIMPORT WSwup_SetErrorTextBehaviourToEnduser (BOOL fEndUser)`
- `void SpecialSprintf (char *pszFinalString, char *pszErrorDescription, t_ErrorStruct *pErrStruct)`
Does a special sprintf with the given information from the new internal error-information.
- `char SWUPLIBDLLEXIMPORT * WSwup_GetLastErrorString (t_SwupCom ComPort)`
Generates an english errortext, retrieved from the new internal function WSwupIntGiveLastError ().

8.18.1 Detailed Description

The external-error-interfaces to the internal functions, part of the MAIN-DLL.

Todo

Already commented bot not doxygen-conform.

8.18.2 Function Documentation

8.18.2.1 void SpecialSprintf (char * pszFinalString, char * pszErrorDescription, t_ErrorStruct * pErrStruct)

Does a special sprintf with the given information from the new internal error-information.

Parameters:

- pszFinalString** : pointer to the destination-string like in sprintf
- pszErrorDescription** : pointer to the format-string like in sprintf
- pErrStruct** : pointer to the additional error-information

Returns:

void

8.18.2.2 DWORD SWUPLIBDLLEXIMPORT WSwup_GetLastError (t_SwupCom ComPort)**8.18.2.3 char SWUPLIBDLLEXIMPORT* WSwup_GetLastErrorString (t_SwupCom ComPort)**

Generates an english errortext, retrieved from the new internal function WSwupIntGiveLastError ().

Therefore it retrieves a pointer to the new error-struct. With the error-number it loops through the additionally given array with the errortexts to find the right error-text. If an alternate error-number is given (and behaviour is set to enduser), it loops once more through the array to find the alternate errortext. This Error-Text is (via definition) parameter-free, so the text can be returned. If there is no alternate error-number given there must be a parameter-check and substitution

Parameters:

ComPort : the Com-Port on which the error occurred

Returns:

pointer to the decoded error-string

8.18.2.4 SWUPLIBDLLEXIMPORT t_ErrorStruct* WSwup_GiveLastError (WORD wUpdateNr)

Returns the last error in dependency from the updatenumber.

Parameters:

wUpdateNr : the current Update-number

Returns:

error-number

8.18.2.5 char SWUPLIBDLLEXIMPORT* WSwup_OldGetLastErrorString (t_SwupCom ComPort)**8.18.2.6 void SWUPLIBDLLEXIMPORT WSwup_SetErrorTextBehaviourToEnduser (BOOL fEndUser)**

8.19 wswerrin.c File Reference

Error-Logging-Functionality contained in WSWUPSER-DLL.

```
#include <windows.h>
#include <stdio.h>
#include <stdarg.h>
#include "wswuplib.h"
#include "wsw_seri.h"
#include "wsw_err.h"
#include "werrenum.h"
```

Data Structures

- struct [t_ErrorCodeDescription](#)
Structure containing the error-code and a description.
- struct [t_ErrorInfo](#)
A struct containing all info about an error.

Enumerations

- enum [t_ErrTextInfo](#) { [enStandardErrText](#), [enAdditionalErrText](#) }
An enumeration containing information about the error-text, some errors have predefined texts, some errors have dynamical error-texts.

Functions

- DWORD SWUPSERILOWEXIMPORT [WSwupInt_GetLastError](#) (t_SwupCom ComPort)
Returns last error-value for this comport.
- char SWUPSERILOWEXIMPORT * [WSwupInt_GetLastErrorString](#) (t_SwupCom ComPort)
Returns last error-string for this comport.
- void SWUPSERILOWEXIMPORT [WSwInternalSetLastErrorWithFileInfo](#) (t_SwupCom ComPort, DWORD dwLastError, char *pszFileName, int nLine)
Used to trace and store error-codes and descriptions.
- void SWUPSERILOWEXIMPORT [WSwSetLastErrorText](#) (t_SwupCom ComPort, const char *psz-Format,...)
Sets an error-text for for the last error.
- void SWUPSERILOWEXIMPORT [WSwupIntSetError](#) (WORD wUpdateNr, unsigned int en_Error, unsigned char *ucInfo,...)
Sets an error-number for the last error into a global variable, defined in [wsw_seri.h](#).

- void SWUPSERILOWEXIMPORT [WSwupIntSetErrorAllUpdates](#) (WORD wUpdateNr, unsigned int en_Error, unsigned char *ucInfo, va_list point)
Sets an error-number for the last error into a global variable, defined in [wsw_seri.h](#).
- void SWUPSERILOWEXIMPORT [WSwupIntForAllUpdatesSetError](#) (unsigned int enError, unsigned char *ucInfo,...)
Sets an error-number for the last error into a global variable, defined in [wsw_seri.h](#), valid for updates.
- SWUPSERILOWEXIMPORT t_ErrorStruct * [WSwupIntGiveLastError](#) (WORD wUpdateNr)
Returns the last Error for the indicated updatenumber with its specific parameters declaration of t_ErrorStruct see above.
- SWUPSERILOWEXIMPORT t_ErrorField * [WSwupIntGiveErrorList](#) (WORD wUpdateNr)
Returns the complete Errorlist for the indicated updatenumber with its specific parameters declaration of [t_ErrorField](#) : typedef struct { BYTE bIndex; // the index for enum field next to filled t_ErrorStruct Error [MAX_ENTRIES]; }t_ErrorField; declaration of t_ErrorStruct see WSwupGiveLastError maximum entries : MAX_ENTRIES for each updatenumber.

8.19.1 Detailed Description

Error-Logging-Functionality contained in WSWUPSER-DLL.

Todo

In the moment all error-strings are hardcoded in english ! This is very bad for multilanguage-GUIs. So decide about another implementation, but be aware, that some errors have additional information which is generated during runtime ! How can this be handled ?

8.19.2 Function Documentation

8.19.2.1 void SWUPSERILOWEXIMPORT WSwInternalSetLastErrorWithFileInfo (t_SwupCom ComPort, DWORD dwLastError, char * pszFileName, int nLine)

Used to trace and store error-codes and descriptions.

Dont call this function direct, use instead macro : [WSwInternalSetLastError\(\)](#), so that you must not deal with File-Name and Code Line ! This function must be called before [WSwSetLastErrorText\(\)](#) , because the text is set to StandardText

Parameters:

ComPort : the comport

dwLastError : Error-Number

pszFileName : FileName of Source-File (via Macro)

nLine : Line-Number in Source-File (via Macro)

Returns:

void

8.19.2.2 void SWUPSERILOWEXIMPORT WSwSetLastErrorText (t_SwupCom ComPort, const char * pszFormat, ...)

Sets an error-text for for the last error.

Only call this function if you cant use the standardtext. Must be called after WSwInternalSetLastError, otherwise this Text will be overwritten. Can be used like printf.

Parameters:

ComPort : the comport

pszFormat : the Format-String (like printf)

... : Additional parameters (like printf)

Returns:

void

8.19.2.3 DWORD SWUPSERILOWEXIMPORT WSwupInt_GetLastError (t_SwupCom ComPort)

Returns last error-value for this comport.

This is only the error-value, in the moment this value does not make any sense outside this module.

Parameters:

ComPort : Last Error is to be retrieved for this comport

Returns:

Last error-value for this comport

8.19.2.4 char SWUPSERILOWEXIMPORT* WSwupInt_GetLastErrorString (t_SwupCom ComPort)

Returns last error-string for this comport.

This returns a pointer to a descriptive error-string. Parameters-In: ComPort

Parameters:

ComPort : Last Error is to be retrieved for this comport

Returns:

Last error-string for this comport

8.19.2.5 void SWUPSERILOWEXIMPORT WSwupIntForAllUpdatesSetError (unsigned int enError, unsigned char * ucInfo, ...)

Sets an error-number for the last error into a global variable, defined in [wsw_seri.h](#), valid for updates.

Parameters:

enError : the Error-number

ucInfo : indicates the kind of the following parameters, e.g. s for string, x,d,i,o,u for numerals, no l...! e.g. "sxsd", maximum SIZE_OF_PINFO Entries (10 at the moment), upper and lower chars allowed maximum 5 strings and 5 non-strings just like d,X... in printf, without it's assumed that every string parameter ends with '0' params : maximum 5 LONG values and maximum 5 char pointers(for max 2000 chars) please don't forget to cast the non-strings to LONG in every call

Returns:

void

8.19.2.6 SWUPSERILOWEXIMPORT [t_ErrorField](#)* WSwupIntGiveErrorList (WORD *wUpdateNr*)

Returns the complete Errorlist for the indicated updatenumber with its specific parameters declaration of [t_ErrorField](#) : typedef struct { BYTE bIndex; // the index for enum field next to filled t_ErrorStruct Error [MAX_ENTRIES]; } [t_ErrorField](#); declaration of t_ErrorStruct see WSwupGiveLastError maximum entries : MAX_ENTRIES for each updatenumber.

Parameters:

wUpdateNr : the current Update-number

Returns:

error-number

8.19.2.7 SWUPSERILOWEXIMPORT [t_ErrorStruct](#)* WSwupIntGiveLastError (WORD *wUpdateNr*)

Returns the last Error for the indicated updatenumber with its specific parameters declaration of t_ErrorStruct see above.

Parameters:

wUpdateNr : the current Update-number

Returns:

error-number

8.19.2.8 void SWUPSERILOWEXIMPORT WSwupIntSetError (WORD *wUpdateNr*, unsigned int *en_Error*, unsigned char * *ucInfo*, ...)

Sets an error-number for the last error into a global variable, defined in [wsw_seri.h](#).

variable parameters

Parameters:

wUpdateNr : the current Update-number

en_Error : the Error-number

ucInfo : indicates the kind of the following parameters, e.g. s for string, x,d,i,o,u for numerals, no l...! e.g. "sxsd", maximum SIZE_OF_PINFO Entries (10 at the moment), upper and lower chars allowed maximum 5 strings and 5 non-strings just like d,X... in printf, without it's assumed that every string parameter ends with '0' params : maximum 5 LONG values and maximum 5 char pointers(for max 2000 chars) please don't forget to cast the non-strings to LONG in every call

Returns:

void

for further information :typedef struct { DWORD dwErrorNumber; // the error-enum unsigned char ucParamInfo[SIZE_OF_PINFO]; // info about the validity of the following params LONG ulParamValue[5];
 // variables that will be inserted in the indicated string at d,x,... unsigned char ucParamString[5][SIZE_OF_PSTRING]; // variables that will be inserted in the indicated string at s } t_ErrorStruct;

8.19.2.9 void SWUPSERILOWEXIMPORT WSwupIntSetErrorAllUpdates (WORD *wUpdateNr*, unsigned int *en_Error*, unsigned char * *ucInfo*, va_list *point*)

Sets an error-number for the last error into a global variable, defined in [wsw_seri.h](#).

for all updates called by WSwupIntForAllUpdatesSetError pointer to variable parameterlist

Parameters:

wUpdateNr : the current Update-number

en_Error : the Error-number

ucInfo : indicates the kind of the following parameters, e.g. s for string, x,d,i,o,u for numerals, no l...! e.g. "sxsds", maximum SIZE_OF_PINFO Entries (10 at the moment), upper and lower chars allowed maximum 5 strings and 5 non-strings just like d,X... in printf, without it's assumed that every string parameter ends with '0'

point : has to be defined as va_list

Returns:

void

8.20 wswupprog.dox File Reference

8.21 xbi_info.h File Reference

Interface between the HEX-BIN-Converter and SWUP, part of the MAIN-DLL.

Data Structures

- struct [t_ExtendedInfo](#)
The "Master"-stuct, containing all info in the XBI-Header All arrays containg text (especially dates) are 1 Byte longer to allow the storage of the .
- struct [t_LangCheckInfo](#)
Information about Adresses on Language-Group-Booting.
- struct [tag_AllLibInfo](#)
A linked list containing the library-infos.
- struct [tag_Area](#)
Definition of a listenelement of the Erase/Not-Erase/Generate/And-so-on - Table.
- struct [tag_CompressInfo](#)
A (complete useles) information about parameters on compressing.
- struct [tag_LocaterRecord](#)
A new structure for "locating" Up to now only implemented in the HEX-BIN-Converter.
- struct [tagNewSplitInfo](#)
Definition of an element of the New SPlit information (can be Tegic, Language, FileSystem,...) Implemented as a linked list.

Defines

- #define [MAX_SPLIT_COMMENT_LENGTH](#) 20
what do we allow as maximum for a split-comment

Typedefs

- typedef [tag_LocaterRecord](#) [t_LocaterRecord](#)
A new structure for "locating" Up to now only implemented in the HEX-BIN-Converter.
- typedef [tag_AllLibInfo](#) [t_AllLibInfo](#)
A linked list containing the library-infos.
- typedef [tag_CompressInfo](#) [t_CompressInfo](#)
A (complete useles) information about parameters on compressing.
- typedef [tag_Area](#) [t_Area](#)
Definition of a listenelement of the Erase/Not-Erase/Generate/And-so-on - Table.

- typedef [tagNewSplitInfo](#) [t_NewSplitInfo](#)

Definition of an element of the New Split information (can be Tegic, Language, FileSystem,...) Implemented as a linked list.

Enumerations

- enum [t_SwType](#) {
[enMobSw](#) = 0, [enEesimu](#) = 1, [enVoice_Memo](#) = 2, [enCodeOnly](#) = 3,
[enLangOnly](#) = 4, [enCodeAndLang](#) = 5, [enDiffFile](#) = 6, [enExtendedNewSplit](#) = 7 }
What kind of software do we have ?
- enum [t_AsicTyp](#) { [enStuttgart](#) = 0, [enStarnBerg](#) }
What kind of asic do we have ?
- enum [t_WriteTyp](#) {
[enHighActive](#) = 0, [enLowActive](#), [enNotUsed](#), [enWriteTypeV4](#),
[enWriteTypeEgoldPlusOnly](#), [enWriteTypeEgoldPlusLow](#), [enWriteTypeEgoldPlusHigh](#), [enWriteTypeEgoldPlusOldLow](#),
[enWriteTypeEgoldPlusOldHigh](#) }
How do we switch on writing to Flash in this Mobile.
- enum [t_XbiOrExe](#) { [enXbi](#), [enExe](#) }
Only for SWUP : What Kind of file do we have, an XBI or a trader-EXE.
- enum [t_ProcType](#) {
[enV0toV36](#) = 0, [enV4C7](#), [enV4C9](#), [enEgoldV12](#),
[enEgoldV2](#), [enEgoldPlusV12](#), [enEgoldPlusV3](#), [enTiHercules](#) = 0x80 }
What kind of processor do we have .
- enum [t_IgnType](#) { [enIgnitionPort](#) = 0, [enIgnitionAccessory](#) }
How can we switch the ignition, via Port or via accessory-bus .
- enum [t_FileFormat](#) { [enBin](#), [enLenChk](#), [enRaw](#), [enCompressed](#) }
What kind of File-Format do we have for the DATA, the XBI-Header will always remain the same.
- enum [t_NewSplitIdentifier](#) { [enSplitOnly](#) = 0, [enSplitAndRest](#) = 1 }
What kind of DATA do we have for New Split Information.

8.21.1 Detailed Description

Interface between the HEX-BIN-Converter and SWUP, part of the MAIN-DLL.

Must be the complete same file in both development environments ! Contains the structure that represents the content of the XBI-Header. The HEX-Bin-Converter fills this structure with the content of the Mobile-Memory, that he has in his memory during the conversion. This structure is then on writing the XBI-File written into the XBI-Header, using all the defines from [boot_hex::h](#) . The SWUP-Programm reads the XBI-Header using also this defines and then filling the struct below !

8.21.2 Define Documentation

8.21.2.1 #define MAX_SPLIT_COMMENT_LENGTH 20

what do we allow as maximum for a split-comment

8.21.3 Typedef Documentation

8.21.3.1 typedef struct tag_AllLibInfo t_AllLibInfo

A linked list containing the library-infos.

8.21.3.2 typedef struct tag_Area t_Area

Definition of a listenelement of the Erase/Not-Erase/Generate/And-so-on - Table.

Implemented as a linked list.

8.21.3.3 typedef struct tag_CompressInfo t_CompressInfo

A (complete useles) information about parameters on compressing.

Was never used, is intended to allow different parameters on compression in HEX-BIN-Converter and then to transmit to mobile, so that mobile can adapt it's decompressor.

8.21.3.4 typedef struct tag_LocaterRecord t_LocaterRecord

A new structure for "locating" Up to now only implemented in the HEX-BIN-Converter.

switched on via Flag in the NHK-File.

8.21.3.5 typedef struct tagNewSplitInfo t_NewSplitInfo

Definition of an element of the New SPlit information (can be Tegic, Language, FileSystem,...) Implemented as a linked list.

8.21.4 Enumeration Type Documentation

8.21.4.1 enum t_AsicTyp

What kind of asic do we have ?

Enumeration values:

enStuttgart Stuttgart-ASIC, Watchdog-times between 20 and 100ms.

enStarnBerg Starnberg-ASIC, Watchdog-times between 400 and 2500ms.

8.21.4.2 enum [t_FileFormat](#)

What kind of File-Format do we have for the DATA, the XBI-Header will always remain the same.

Enumeration values:

- enBin** Bin-Format, 3 Byte Adr, 1 Byte Len max 64Byte Data, 1 Byte CHKSum.
- enLenChk** Len-Chk-Format, 1 Byte Len max 0x7F Byte Data, 1 Byte CHKSum, Address in Special-Block.
- enRaw** RAW content of the Flash.
- enCompressed** Compressed Data.

8.21.4.3 enum [t_IgnType](#)

How can we switch the ignition, via Port or via accessory-bus .

Enumeration values:

- enIgnitionPort** Ignition is detected via Port.
- enIgnitionAccessory** Ignition is detected via Accessory-BUS => we cant check Ignition before booting.

8.21.4.4 enum [t_NewSplitIdentifier](#)

What kind of DATA do we have for New Split Information.

Enumeration values:

- enSplitOnly** Only Split informations (Tegic or/and Language) in BIN File.
- enSplitAndRest** Split-Info with-code-rest in BIN-Part of file.

8.21.4.5 enum [t_ProcType](#)

What kind of processor do we have .

Enumeration values:

- enV0toV36** a very old High-Gold-Processor with 166 Kernel
- enV4C7** a High-Gold-V4-Processor in C7 Technolgy with 163 Kernel (only used during evaluation)
- enV4C9** a High-Gold-V4-Processor in C9 Technolgy with 163 Kernel
- enEgoldV12** EGOLD V1.2.
- enEgoldV2** EGOLD V2.
- enEgoldPlusV12** EGOLD-Plus currently Version-independant.
- enEgoldPlusV3** EGOLD+ V3.
- enTiHercules** A TI-Processor named Hercules.

8.21.4.6 enum [t_SwType](#)

What kind of software do we have ?

Enumeration values:

- enMobSw** the normal Mobile-SW
- enEesimu** the EESIMU content in BIN-Format ...
- enVoice_Memo** one Voice-Memo in BIN-Format ..
- enCodeOnly** clean Code area
- enLangOnly** only Language area
- enCodeAndLang** Code and Language.
- enDiffFile** Difference between two XBI-Files.
- enExtendedNewSplit** New extended Split Info (Code+Language, Code+Tegic, ...).

8.21.4.7 enum [t_WriteTyp](#)

How do we switch on writing to Flash in this Mobile.

Enumeration values:

- enHighActive** Port to High => writing allowed.
- enLowActive** Port to Low => writing allowed.
- enNotUsed** no write-protection at all
- enWriteTypeV4** HighGoldV4 with Flashin-Register.
- enWriteTypeEgoldPlusOnly** EGOLD with Flashin-Register.
- enWriteTypeEgoldPlusLow** EGOLD with Flashin-Register and Port on High.
- enWriteTypeEgoldPlusHigh** EGOLD with Flashin-Register and Port on Low.
- enWriteTypeEgoldPlusOldLow** old EGOLD (I have no idea ..
) with Flashin-Register and Port on Low
- enWriteTypeEgoldPlusOldHigh** old EGOLD (I have no idea ..
) with Flashin-Register and Port on High

8.21.4.8 enum [t_XbiOrExe](#)

Only for SWUP : What Kind of file do we have, an XBI or a trader-EXE.

Enumeration values:

- enXbi** its an XBI-File
- enExe** its a Trader-EXE-File

Chapter 9

Programmers Guide for the Winswup-Library Page Documentation

9.1 Todo List

File [fhawrap.c](#) Doxygen-Documentation !

File [filehand.c](#) Throw away the old-style traces or re-implement a better trace-functionality in the Main-DLL (and the GUI ???). Make comments doxygen-conform. Decide if comments of exported functions should be done in the interface- header `fiha.h` OR here, not in both parts, that leads to errors.

File [pc_mob.h](#) -Comment all the Message-Identifiers ! -Make sure it's the same version on Mobile and PC-Side !

File [swupwork.c](#) Is nearly completely undocumented ...

File [wbfbtool.c](#) Make some comments.

File [wseril3.c](#) -Make some comments either here or in the export-header `wswserl3.h`. -Make a decision about where to put the comments.

File [wsw_tool.c](#) Make at least some comments.

File [wsbdbgex.c](#) Already commented but not doxygen-conform.

File [wsverrex.c](#) Already commented but not doxygen-conform.

File [wsverrin.c](#) In the moment all error-strings are hardcoded in english ! This is very bad for multilanguage-GUIs. So decide about another implementation, but be aware, that some errors have additional information which is generated during runtime ! How can this be handled ?

Index

- A5_ACK
 - ws_w seri.h, 152
- abInfo
 - tag_AllLibInfo, 54
- abUpdateSwData
 - swupwork.c, 120
- ACK
 - pc_mob.h, 100
 - ws_w seri.h, 152
- ACK_B0
 - pc_mob.h, 100
- ACK_B1
 - pc_mob.h, 100
- ACK_HOLD
 - ws_w seri.h, 153
- ACK_LEN
 - layer2func, 21
- ACK_LEN_TO_SEND
 - wseril2.c, 135
- ACKNOWLEDGE_A5
 - pc_mob.h, 100
- ActualMessage
 - wseril2.c, 137
- ADDI_COMTYPE
 - wseril1.c, 131
- ADDIFAST_INI_NAME
 - wseril1.c, 131
- Address
 - T_HashBinBlock, 51
- ADDRESS_LEN_BIN_BLOCK
 - ws_w seri.h, 153
- ArrayEntries
 - ws_w seri.h, 153
- aucBootPIN
 - tag_PINstruct, 60
- aucDataBuf
 - t_Block, 34
- AUSZEIT
 - pc_mob.h, 100
- BelegeFileInit
 - filehand.c, 89
- BfbCheckThread
 - wbfbtool.c, 126
- BfbThreadInfo
 - wbfbtool.c, 127
- BIN_FORMAT_OVERHEAD_HASH
 - fiwrap.c, 85
- bIndex
 - t_ErrorField, 38
- BLOCK_TIMEOUT_VAL
 - pc_mob.h, 100
- boot_hex.h, 67
 - CONVERTER_VERSION_MAJOR, 72
 - CONVERTER_VERSION_MINOR, 72
 - EXE_KENNUNG, 72
 - HASH_ID_STRING, 72
 - HEXUNIX_VERSION_MAJOR, 72
 - HEXUNIX_VERSION_MINOR, 72
 - ID_ADDITIONAL_INFO_END, 72
 - ID_ADDITIONAL_INFO_ENTRY, 72
 - ID_ADDITIONAL_INFO_LEN, 72
 - ID_ALIGNEMENT, 72
 - ID_ALL_LIB_INFO_P1, 72
 - ID_ALL_LIB_INFO_P2, 73
 - ID_ASIC_TYPE, 73
 - ID_COMPRESS_INFO, 73
 - ID_CONSTANTS, 73
 - ID_DEVELOPER, 73
 - ID_END_LOCATER_ENTRY, 73
 - ID_END_OF_HEADER, 73
 - ID_EPROM_SIZE, 73
 - ID_ERASE_INFO, 73
 - ID_ERTEC_SUM, 73
 - ID_EXT_LIB_1_DATE, 73
 - ID_EXT_LIB_2_DATE, 74
 - ID_FLASH_WRITE_TYPE, 74
 - ID_FORMAT_INFO, 74
 - ID_HASH_TABLE_INFO, 74
 - ID_HEXER_VERSION, 74
 - ID_IGNITION_INFO, 74
 - ID_INTERNAL_VERSION, 74
 - ID_LANG_CHECK_INFO, 74
 - ID_LANG_SPLIT_ID, 74
 - ID_LEAST_SWUP_VERSION, 74
 - ID_LOCATER_ENTRY, 74
 - ID_MOB_SW_LOC_DATE, 75
 - ID_MOB_SW_RBM_DATE, 75
 - ID_NEW_SPLIT_ADD_INFO_TEXT, 75
 - ID_NEW_SPLIT_INFO, 75

- ID_NEW_SPLIT_INFO_TEXT, 75
- ID_NO_CHECK, 75
- ID_OLD_DEVELOPER, 75
- ID_OLD_LOC_DATE, 75
- ID_OLD_PROJ_NAME, 75
- ID_OLD_PROJECT, 75
- ID_OLD_RBM_DATE, 75
- ID_PROC_TYPE, 76
- ID_PROJECT_ID, 76
- ID_PROJECT_NAME, 76
- ID_RAM_SIZE, 76
- ID_RBC_PUT_COUNT, 76
- ID_RBM_PUT_COUNT, 76
- ID_SPEECHGROUP_ID, 76
- ID_SPEECHGROUP_NAME, 76
- ID_SW_GENERATION, 76
- ID_SW_PRODUCT_NAME, 76
- ID_SW_STATE, 77
- ID_SW_TYPE, 77
- ID_SW_VENDOR_NAME, 77
- ID_SW_VERSION_NUMBER, 77
- ID_SWUP_DLL, 77
- ID_TEGIC_GROUP, 77
- ID_TEXT, 77
- ID_TEXT_END, 77
- ID_TEXT_LENGTH, 77
- LEAST_LANGSWUP_VERSION_-
MAJOR, 77
- LEAST_LANGSWUP_VERSION_-
MINOR, 78
- LEAST_SWUP_VERSION_MAJOR, 78
- LEAST_SWUP_VERSION_MINOR, 78
- LEAST_ZIPSWUP_VERSION_MAJOR,
78
- LEAST_ZIPSWUP_VERSION_MINOR,
78
- MEMO2XBI_VERSION_MAJOR, 78
- MEMO2XBI_VERSION_MINOR, 78
- SIG_ID_STRING, 78
- SIG_ID_STRING_SGOLD, 78
- XBI_KENNUNGS_STRING, 78
- XBI_KENNUNGS_STRING_SGOLD, 78
- BootCoreStartUpdateSW_LastHope
swupwork.c, 110
- BootCoreStartUpdateSWFunc
swupwork.c, 120
- BootStrapLoaderFunc
swupwork.c, 120
- BUFFER_OVERFLOW
pc_mob.h, 100
- BYTE_TIMEOUT_VAL
pc_mob.h, 100
- CheckIfThisDiffSwUpdateIsOk
swupwork.c, 110
- CheckSipcMessage
wseril2.c, 136
- Checksum
T_HashBinBlock, 51
- CloseDevice
swupwork.c, 111
- CompFormatAfter
t_ExtendedInfo, 44
- CompFormatBefore
t_ExtendedInfo, 44
- ComPort
tagDebugFiles, 63
- ComPorts
wseril1.c, 132
- CompressionInfo
t_ExtendedInfo, 44
- CONTROL_BIN_BLOCKS_SUPPORTED
wseril3.c, 142
- ConvertBaudRatesForAddiData
layer1func, 16
- CONVERTER_VERSION_MAJOR
boot_hex.h, 72
- CONVERTER_VERSION_MINOR
boot_hex.h, 72
- ConvertStreamToSwInformationStruct
swupwork.c, 111
- ctmo
t_ComConfig, 35
- CurrentFileDebugLevel
prog_debugfun, 29
- CurrentOnlineDebugLevel
prog_debugfun, 29
- CurrentReceiveMode
wseril2.c, 137
- Data
T_HashBinBlock, 51
- dcb
t_ComConfig, 35
- DeactBfbLibrary
wbfbtool.c, 126
- DeactIBootLibrary
wseril3.c, 142
wsw_seri.h, 155
- DebugFiles
wsbdbgin.c, 163
- DEFAULT_COMTYPE
wseril1.c, 131
- DEFAULT_TIMEOUT_BLOCK_REC
wsw_seri.h, 153
- DEFAULT_TIMEOUT_SINGLE_CHAR
wsw_seri.h, 153
- DefaultHandler

- t_TableEntry, 53
- DefaultHandlerTable
 - wseril2.c, 137
- DLL_FILE_PATTERN
 - dyna_dll.c, 80
- DllExport
 - wfbftool.c, 125
- dummy
 - wseril3.c, 142
- DummyUsbOmapSwup_GetDevice
 - wseril3.c, 143
- DummyUsbOmapSwup_ReadFromDataPipe
 - wseril3.c, 143
- DummyUsbOmapSwup_ReleaseDevice
 - wseril3.c, 143
- DummyUsbOmapSwup_WriteToControlPipe
 - wseril3.c, 143
- DummyUsbOmapSwup_WriteToDataPipe
 - wseril3.c, 143
- dwBaudrate
 - wfbftool.c, 127
- dwErrCode
 - t_ErrorCodeDescription, 37
- dwError
 - t_ErrorInfo, 39
- dyna_dll.c, 79
 - DLL_FILE_PATTERN, 80
 - fSimulateKnownProject, 81
 - GetDllPath, 80
 - GiveDllNameToLoad, 80
 - InternGiveDllNameToLoad, 80
 - IsDllInWorkingDirectory, 81
 - ScanDirectoryForAvailableProjects, 81
 - szDerivedFromKnownProject, 81
 - szNewUnknownProject, 81
- EI.CODE.PART
 - pc_mob.h, 100
- EI.EEPROM.SIMULATION
 - pc_mob.h, 100
- EI.EEPROM.SIMULATION.BLOCK
 - pc_mob.h, 100
- EI.FLASH.DEPENDENT.BLOCK
 - pc_mob.h, 100
- EI.VOICE.DIAL
 - pc_mob.h, 100
- EI.VOICE.DIAL.BLOCK
 - pc_mob.h, 100
- EI.VOICE.MEMO
 - pc_mob.h, 100
- EI.VOICE.MEMO.BLOCK
 - pc_mob.h, 100
- EI.WHOLE.FLASH
 - pc_mob.h, 100
- EnableWriteFlashBlock
 - wseril3.c, 143
- enAdditionalErrText
 - prog_errorfun, 30
- enBin
 - xbi_info.h, 175
- enBinMode
 - ws_wseri.h, 154
- enChkErr
 - ws_wseri.h, 154
- enCmdBlockReceived
 - ws_wseri.h, 154
- enCodeAndLang
 - xbi_info.h, 176
- enCodeOnly
 - xbi_info.h, 176
- enComOpen
 - wfbftool.c, 126
- enCompressed
 - xbi_info.h, 175
- enDiffFile
 - xbi_info.h, 176
- enEesimu
 - xbi_info.h, 176
- enEgoldPlusV12
 - xbi_info.h, 175
- enEgoldPlusV3
 - xbi_info.h, 175
- enEgoldV12
 - xbi_info.h, 175
- enEgoldV2
 - xbi_info.h, 175
- enError
 - wfbftool.c, 126
- enExe
 - xbi_info.h, 176
- enExtendedNewSplit
 - xbi_info.h, 176
- enGetVoltage
 - wfbftool.c, 126
- enHighActive
 - xbi_info.h, 176
- enIgnitionAccessory
 - xbi_info.h, 175
- enIgnitionPort
 - xbi_info.h, 175
- enIncomplete
 - ws_wseri.h, 154
- enInitStatus
 - FileInfo, 32
- enKnownButBad
 - pc_mob.h, 103
- enLangOnly
 - xbi_info.h, 176

- enLenChk
 - xbi_info.h, 175
- enLenChkMode
 - wsw_seri.h, 154
- enLowActive
 - xbi_info.h, 176
- enMobileError
 - wsw_seri.h, 154
- enMobileOn
 - wbfbtool.c, 126
- enMobSw
 - xbi_info.h, 176
- enNotUsed
 - xbi_info.h, 176
- enOk
 - pc_mob.h, 103
- enPingFirst
 - wbfbtool.c, 126
- enPingSecond
 - wbfbtool.c, 126
- enPingThird
 - wbfbtool.c, 126
- enRaw
 - xbi_info.h, 175
- enReady
 - wbfbtool.c, 126
- enRecOk
 - wsw_seri.h, 154
- enRecTimeout
 - wsw_seri.h, 154
- enSerialError
 - wsw_seri.h, 154
- enSimulated
 - pc_mob.h, 104
- enSipcCmd
 - wsw_seri.h, 154
- enSipcMode
 - wsw_seri.h, 154
- enSplitAndRest
 - xbi_info.h, 175
- enSplitOnly
 - xbi_info.h, 175
- enStandardErrText
 - prog_errorfun, 30
- enStarnBerg
 - xbi_info.h, 174
- enStuttgart
 - xbi_info.h, 174
- enSwitchOff
 - wbfbtool.c, 126
- enTiHercules
 - xbi_info.h, 175
- enUnknown
 - pc_mob.h, 104
- enUnknownChar
 - wsw_seri.h, 154
- enV0toV36
 - xbi_info.h, 175
- enV4C7
 - xbi_info.h, 175
- enV4C9
 - xbi_info.h, 175
- enVoice_Memo
 - xbi_info.h, 176
- enWaitALittleBit
 - wbfbtool.c, 126
- enWriteTypeEgoldPlusHigh
 - xbi_info.h, 176
- enWriteTypeEgoldPlusLow
 - xbi_info.h, 176
- enWriteTypeEgoldPlusOldHigh
 - xbi_info.h, 176
- enWriteTypeEgoldPlusOldLow
 - xbi_info.h, 176
- enWriteTypeEgoldPlusOnly
 - xbi_info.h, 176
- enWriteTypeV4
 - xbi_info.h, 176
- enXbi
 - xbi_info.h, 176
- ERASE_AREA_NOT_OK
 - wseril3.c, 142
- ERASE_AREA_OK
 - wseril3.c, 142
- ERASE_TIME_FLASH_SEGMENT
 - wseril3.c, 142
- EraseFlashBlocksWithAcknowledge
 - wseril3.c, 143
- EraseWithNewUpdateConcept
 - wseril3.c, 143
- ERR_BLOCK_ERASE
 - pc_mob.h, 100
- ERR_BYTE_PROG
 - pc_mob.h, 100
- ERR_CHECKSUM
 - pc_mob.h, 100
- ERR_FLASH_BLOCK_ADR
 - pc_mob.h, 100
- ERR_FUNC_NOT_IMPLEMENTED
 - pc_mob.h, 100
- ERR_HASHING
 - pc_mob.h, 100
- ERR_NO_EPROM
 - pc_mob.h, 100
- ERR_RAM_TEST_00
 - pc_mob.h, 100
- ERR_RAM_TEST_FF
 - pc_mob.h, 100

- ERR_RAM_TEST_MUSTER
 - pc_mob.h, 100
- ERR_RAM_TEST_UNKNOWN
 - pc_mob.h, 100
- ERR_SW_REJECTED
 - pc_mob.h, 100
- err_text.c, 82
 - ERR_TEXT_DEVELOPER, 82
 - ERR_TEXT_VERS_MAJOR, 82
 - ERR_TEXT_VERS_MINOR, 82
 - ErrorDescription, 82
 - ErrorDescriptionSize, 82
 - t_enumErrorDescriptionSize, 82
- err_text.h, 83
 - ErrorDescription, 83
 - ErrorDescriptionSize, 83
 - t_enumErrorDescriptionSize, 83
- ERR_TEXT_DEVELOPER
 - err_text.c, 82
- ERR_TEXT_VERS_MAJOR
 - err_text.c, 82
- ERR_TEXT_VERS_MINOR
 - err_text.c, 82
- ERR_UNKNOWN_FLASH
 - pc_mob.h, 100
- Error
 - t_ErrorField, 38
- ErrorDescription
 - err_text.c, 82
 - err_text.h, 83
- ErrorDescriptionSize
 - err_text.c, 82
 - err_text.h, 83
- ErrTextInfo
 - t_ErrorInfo, 39
- EXE_ENDUNG_SIZE
 - filehand.c, 89
- EXE_KENNUNG
 - boot_hex.h, 72
- fAdditionalInfoAvail
 - t_ExtendedInfo, 45
- fCallbacksInstalled
 - swupwork.c, 120
- fDebugToFile
 - prog_debugfun, 29
- fDebugToWindow
 - prog_debugfun, 29
- fForceDllNameFromNhk
 - t_ExtendedInfo, 45
- fiwrap.c, 84
 - BIN_FORMAT_OVERHEAD_HASH, 85
 - GetHashData, 85
 - GetSignature, 85
 - GivePtrToBinData, 85
 - GiveSwInfoStruct, 85
 - hGIToHash, 86
 - hGIToMem, 86
 - hGIToSig, 86
 - IsLibraryInitialised, 85
 - JumpOverControlBlocks, 85
 - SECURITY_RESERVE, 85
 - szFileErrorString, 86
 - unTotalAmountOfBlocks, 86
 - unTotalHeapBytes, 86
 - WSwup_CloseXbiFile, 86
 - WSwup_GetLastFileError, 86
 - WSwup_ReadXbiFile, 86
 - XbiHeaderInfo, 86
- filehand.c, 87
 - BelegeFileInit, 89
 - EXE_ENDUNG_SIZE, 89
 - GetFileInfo, 89
 - GetMotorolaLong, 89
 - GetMotorolaWord, 90
 - GetSignatureSize, 90
 - GiveExtFileInfo, 90
 - KENNUNG_SIZE, 89
 - MAX_HASH_BIN_BLOCK_LENGTH, 89
 - NO_INIT, 89
 - OLD_EXE_ENDUNG_SIZE, 89
 - PrintErrorMessage, 90
 - ReadAndFormatSigData, 91
 - ReadKennung, 91
 - ReadUndCheckAnbootDaten, 91
 - ReadUndCheckXbiDaten, 92
 - ReadUndFormatHashCluster, 93
 - ReadXbiHeader, 93
 - ReleaseXbiHeaderData, 94
- FileInfo, 31
- FileInfo
 - enInitStatus, 32
 - KindofFile, 32
 - ulFileSize, 32
 - ulHashClusterLen, 32
 - ulSignatureLen, 32
 - ulSwupExeLen, 32
 - ulXbiDatenEnde, 32
 - ulXbiDatenStart, 32
 - ulXbiHeaderLen, 33
 - ulXbiLen, 33
 - unExeEndungLen, 33
- fInit
 - t_ComConfig, 35
- fLibraryIsInitialised
 - swupwork.c, 120
- ForceUsb
 - tag_UsbInfo, 62

- FormatInfo
 - t_ExtendedInfo, [45](#)
- fpDebug
 - tagDebugFiles, [63](#)
- FreeList
 - swupwork.c, [111](#)
- fSimulateKnownProject
 - dyna_dll.c, [81](#)
 - swupwork.c, [120](#)
- GeneralUpdateMode
 - swupwork.c, [120](#)
- GetActVoltage
 - wbfbtool.c, [126](#)
- GetDllPath
 - dyna_dll.c, [80](#)
- GetFileInfo
 - filehand.c, [89](#)
- GetHashData
 - fhawrap.c, [85](#)
- GetIntelShort
 - wsw_tool.c, [160](#)
- GetMotorolaLong
 - filehand.c, [89](#)
- GetMotorolaLongFromBuffer
 - wsw_tool.c, [160](#)
- GetMotorolaWord
 - filehand.c, [90](#)
- GetSignature
 - fhawrap.c, [85](#)
- GetSignatureSize
 - filehand.c, [90](#)
- GiveDllNameToLoad
 - dyna_dll.c, [80](#)
- GiveExtFileInfo
 - filehand.c, [90](#)
- GiveGeneralUpdateMode
 - swupwork.c, [111](#)
- GivePtrToBinData
 - fhawrap.c, [85](#)
- GivePutcountAndDirectory
 - wbfbtool.c, [127](#)
- GiveSwInfoStruct
 - fhawrap.c, [85](#)
- GiveSwupLibDate
 - swupwork.c, [112](#)
- GiveUpdateDllPcSwVersion
 - swupwork.c, [120](#)
- GiveUpdateMobileSwVersion
 - swupwork.c, [120](#)
- GiveUpdateModeForThisUpdate
 - swupwork.c, [112](#)
- GiveUpdatePtrAndLenFunc
 - swupwork.c, [120](#)
- GlobalWhichInfoCallBackTime
 - swupwork.c, [120](#)
- HASH_ID_STRING
 - boot_hex.h, [72](#)
- HashInfo
 - t_ExtendedInfo, [45](#)
- HashListTransmissionProgress
 - wseril3.c, [143](#)
- hBfbLib
 - wbfbtool.c, [127](#)
- hCom
 - t_ComConfig, [35](#)
- HEXUNIX_VERSION_MAJOR
 - boot_hex.h, [72](#)
- HEXUNIX_VERSION_MINOR
 - boot_hex.h, [72](#)
- hGIToHash
 - fhawrap.c, [86](#)
- hGIToMem
 - fhawrap.c, [86](#)
- hGIToSig
 - fhawrap.c, [86](#)
- hIBootLib
 - wseril3.c, [146](#)
- hIBootLibLow
 - wseril2.c, [137](#)
- HowToWrite
 - t_ExtendedInfo, [45](#)
- hUpdateLib
 - swupwork.c, [121](#)
- ID_ADDITIONAL_INFO_END
 - boot_hex.h, [72](#)
- ID_ADDITIONAL_INFO_ENTRY
 - boot_hex.h, [72](#)
- ID_ADDITIONAL_INFO_LEN
 - boot_hex.h, [72](#)
- ID_ALIGNMENT
 - boot_hex.h, [72](#)
- ID_ALL_LIB_INFO_P1
 - boot_hex.h, [72](#)
- ID_ALL_LIB_INFO_P2
 - boot_hex.h, [73](#)
- ID_ASIC_TYPE
 - boot_hex.h, [73](#)
- ID_COMPRESS_INFO
 - boot_hex.h, [73](#)
- ID_CONSTANTS
 - boot_hex.h, [73](#)
- ID_DEVELOPER
 - boot_hex.h, [73](#)
- ID_END_LOCATER_ENTRY
 - boot_hex.h, [73](#)

- ID_END_OF_HEADER
boot_hex.h, 73
- ID_EPROM_SIZE
boot_hex.h, 73
- ID_ERASE_INFO
boot_hex.h, 73
- ID_ERTEC_SUM
boot_hex.h, 73
- ID_EXT_LIB_1_DATE
boot_hex.h, 73
- ID_EXT_LIB_2_DATE
boot_hex.h, 74
- ID_FLASH_WRITE_TYPE
boot_hex.h, 74
- ID_FORMAT_INFO
boot_hex.h, 74
- ID_HASH_TABLE_INFO
boot_hex.h, 74
- ID_HEXER_VERSION
boot_hex.h, 74
- ID_IGNITION_INFO
boot_hex.h, 74
- ID_INTERNAL_VERSION
boot_hex.h, 74
- ID_LANG_CHECK_INFO
boot_hex.h, 74
- ID_LANG_SPLIT_ID
boot_hex.h, 74
- ID_LEAST_SWUP_VERSION
boot_hex.h, 74
- ID_LOCATER_ENTRY
boot_hex.h, 74
- ID_MOB_SW_LOC_DATE
boot_hex.h, 75
- ID_MOB_SW_RBM_DATE
boot_hex.h, 75
- ID_NEW_SPLIT_ADD_INFO_TEXT
boot_hex.h, 75
- ID_NEW_SPLIT_INFO
boot_hex.h, 75
- ID_NEW_SPLIT_INFO_TEXT
boot_hex.h, 75
- ID_NO_CHECK
boot_hex.h, 75
- ID_OLD_DEVELOPER
boot_hex.h, 75
- ID_OLD_LOC_DATE
boot_hex.h, 75
- ID_OLD_PROJ_NAME
boot_hex.h, 75
- ID_OLD_PROJECT
boot_hex.h, 75
- ID_OLD_RBM_DATE
boot_hex.h, 75
- ID_PROC_TYPE
boot_hex.h, 76
- ID_PROJECT_ID
boot_hex.h, 76
- ID_PROJECT_NAME
boot_hex.h, 76
- ID_RAM_SIZE
boot_hex.h, 76
- ID_RBC_PUT_COUNT
boot_hex.h, 76
- ID_RBM_PUT_COUNT
boot_hex.h, 76
- ID_SPEECHGROUP_ID
boot_hex.h, 76
- ID_SPEECHGROUP_NAME
boot_hex.h, 76
- ID_SW_GENERATION
boot_hex.h, 76
- ID_SW_PRODUCT_NAME
boot_hex.h, 76
- ID_SW_STATE
boot_hex.h, 77
- ID_SW_TYPE
boot_hex.h, 77
- ID_SW_VENDOR_NAME
boot_hex.h, 77
- ID_SW_VERSION_NUMBER
boot_hex.h, 77
- ID_SWUP_DLL
boot_hex.h, 77
- ID_TEGIC_GROUP
boot_hex.h, 77
- ID_TEXT
boot_hex.h, 77
- ID_TEXT_END
boot_hex.h, 77
- ID_TEXT_LENGTH
boot_hex.h, 77
- IgnitionInfo
t_ExtendedInfo, 45
- InitBfbLibrary
wbfbtool.c, 127
- InitGlobalMemory
swupwork.c, 112
- InitIBootLibrary
wseril3.c, 143
wsw_seri.h, 155
- InitMobileErrorHandler
swupwork.c, 112
wmob_err.c, 129
- InitProjectDependentBootstrapDll
swupwork.c, 113
- InsertLongToStringIntel
wsw_tool.c, 160

- InsertLongToStringMotorola
 - ws_w_tool.c, 160
- InsertWordToStringIntel
 - ws_w_tool.c, 160
- InstallDefaultHandlerFunction
 - layer2func, 21
- InternGiveDllNameToLoad
 - dyna_dll.c, 80
- Invalidate_BGJ
 - swupwork.c, 113
- IPC_UPDATE_CMD
 - pc_mob.h, 100
- IPC_UPDATE_DATA
 - pc_mob.h, 100
- IsDllInWorkingDirectory
 - dyna_dll.c, 81
- IsLibraryInitialised
 - fiwrap.c, 85
 - swupwork.c, 113
- IsSwupLibVersionNewEnough
 - swupwork.c, 113
- IsThisADownGrade
 - swupwork.c, 114
- JumpOverControlBlocks
 - fiwrap.c, 85
- KENNUNG_SIZE
 - filehand.c, 89
- KindofFile
 - FileInfo, 32
- LanguageCheckInfo
 - t_ExtendedInfo, 45
- layer1func
 - ConvertBaudRatesForAddiData, 16
 - WSwupClearIgnition, 17
 - WSwupComClose, 17
 - WSwupComConfigure, 17
 - WSwupComOpen, 17
 - WSwupComReadByte, 18
 - WSwupComReadByteTimeOut, 18
 - WSwupComWriteByte, 18
 - WSwupComWriteDataBlock, 19
 - WSwupSetIgnition, 19
- layer2func
 - ACK_LEN, 21
 - InstallDefaultHandlerFunction, 21
 - ReceiveBlockInSipcMode, 21
 - ReceiveCmdBlockInBinMode, 22
 - SendSimpleMessageInSipcMode, 22
 - WSwupReceiveCmdBlock, 22
 - WSwupReceiveValueFromSerial, 23
 - WSwupSendBotBlock, 23
 - WSwupSendMessageBlock, 23
 - WSwupSendSGoldBotBlock, 24
 - WSwupSeriLowGiveVersion, 24
 - WSwupSetCurrentReceiveMode, 24
 - WSwupWSwupGetCurrentReceiveMode, 25
- LEAST_LANGSWUP_VERSION_MAJOR
 - boot_hex.h, 77
- LEAST_LANGSWUP_VERSION_MINOR
 - boot_hex.h, 78
- LEAST_SWUP_VERSION_MAJOR
 - boot_hex.h, 78
- LEAST_SWUP_VERSION_MINOR
 - boot_hex.h, 78
- LEAST_ZIPSWUP_VERSION_MAJOR
 - boot_hex.h, 78
- LEAST_ZIPSWUP_VERSION_MINOR
 - boot_hex.h, 78
- Length
 - T_HashBinBlock, 51
- M_MODE_SELECTION
 - pc_mob.h, 100
- M_MODEM_UPDATE_FINISHED
 - pc_mob.h, 100
- M_MODEM_UPDATE_RUNNING
 - pc_mob.h, 101
- M_MODEM_UPDATE_STARTED
 - pc_mob.h, 101
- M_PC_CONNECTED
 - pc_mob.h, 101
- M_SWH_CUSTOMER_REBOOT
 - wfbttool.c, 125
- M_SWH_GET_POWER
 - wfbttool.c, 125
- M_SWH_GET_PUTCOUNT
 - wfbttool.c, 125
- M_SWH_IGNITION
 - wfbttool.c, 125
- M_SWH_MOBILE_OFF
 - wfbttool.c, 125
- M_SWH_VERSION_NUMBER
 - wfbttool.c, 125
- M_SWH_VOLTAGE
 - wfbttool.c, 125
 - wseril3.c, 142
- MAX_DATA_IN_BIN_BLOCK
 - ws_w_seri.h, 153
- MAX_HASH_BIN_BLOCK_LENGTH
 - filehand.c, 89
- MAX_RETRY_ON_NAK
 - ws_w_seri.h, 153
- MAX_SIZE_OF_PIN
 - swupwork.c, 110

- MAX_SPLIT_COMMENT_LENGTH
 - xbi_info.h, [174](#)
- MAX_SUCCESS_CALLBACKS
 - wbfbtool.c, [125](#)
- MAX_VOLTAGE_CALLBACKS
 - wbfbtool.c, [125](#)
- MEMO2XBI_VERSION_MAJOR
 - boot_hex.h, [78](#)
- MEMO2XBI_VERSION_MINOR
 - boot_hex.h, [78](#)
- MFI_GET_ANZ_FLASHES
 - pc_mob.h, [103](#)
- MFI_GET_FLASH_CODE
 - pc_mob.h, [103](#)
- MI
 - T_HashBinBlock, [51](#)
- MI_ACK_END_OF_TRANSMISSION
 - pc_mob.h, [103](#)
- MI_ACK_ERASE_FLASH
 - pc_mob.h, [103](#)
- MI_ACK_GIVE_EXT_FLASH_INFO
 - pc_mob.h, [103](#)
- MI_ACK_GIVE_FLASH_CODE
 - pc_mob.h, [103](#)
- MI_ADDITIONAL_MAP_INFO
 - pc_mob.h, [103](#)
- MI_ADDITIONAL_MAP_INFO_END
 - pc_mob.h, [103](#)
- MI_ADDITIONAL_MAP_INFO_LEN
 - pc_mob.h, [103](#)
- MI_ALIGNEMENT
 - pc_mob.h, [103](#)
- MI_CALC_CHECKSUM
 - pc_mob.h, [103](#)
- MI_CALC_CHECKSUM_ANSWER
 - pc_mob.h, [103](#)
- MI_CALC_SPLIT_INFO
 - pc_mob.h, [103](#)
- MI_CALC_SPLIT_INFO_ANSWER
 - pc_mob.h, [103](#)
- MI_COMPRESSION_OFF
 - pc_mob.h, [103](#)
- MI_COMPRESSION_OFF_ANSWER
 - pc_mob.h, [103](#)
- MI_COMPRESSION_ON
 - pc_mob.h, [103](#)
- MI_COMPRESSION_ON_ANSWER
 - pc_mob.h, [103](#)
- MI_CONTROL_COMMAND
 - pc_mob.h, [103](#)
- MI_CURR_BAUDRATE
 - pc_mob.h, [103](#)
- MI_CURR_BAUDRATE_ACK
 - pc_mob.h, [103](#)
- MI_DUMMY
 - pc_mob.h, [103](#)
- MI_ENABLE_FLASHWRITE_ALL
 - pc_mob.h, [103](#)
- MI_ENABLE_FLASHWRITE_BLNR
 - pc_mob.h, [103](#)
- MI_END_OF_TRANSMISSION
 - pc_mob.h, [103](#)
- MI_ERASE_FLASH
 - pc_mob.h, [103](#)
- MI_ERASE_FLASH_AREA
 - pc_mob.h, [103](#)
- MI_ERASE_FLASH_AREA_ACK
 - pc_mob.h, [103](#)
- MI_ERASE_FLASH_UPGRADE_CONCEPT
 - pc_mob.h, [103](#)
- MI_ERASE_FLASH_UPGRADE_CONCEPT_-ACK
 - pc_mob.h, [103](#)
- MI_ERASE_FLASH_UPGRADE_CONCEPT_-RUNNING
 - pc_mob.h, [103](#)
- MI_EXT_CONTROL_COMMAND
 - pc_mob.h, [103](#)
- MI_FINISH_BOTBLOCK_HANDLING
 - pc_mob.h, [103](#)
- MI_FLASH_CHECK_OK
 - pc_mob.h, [103](#)
- MI_FLASH_CODE
 - pc_mob.h, [103](#)
- MI_FLASH_ERASE_ADDRESS
 - pc_mob.h, [103](#)
- MI_GET_32BIT_NUMBER
 - pc_mob.h, [103](#)
- MI_GET_32BIT_NUMBER_ANSWER
 - pc_mob.h, [103](#)
- MI_GET_MEMORY_CONTENT
 - pc_mob.h, [103](#)
- MI_GET_MEMORY_CONTENT_ANSWER
 - pc_mob.h, [103](#)
- MI_GET_MEMORY_CONTENT_END
 - pc_mob.h, [103](#)
- MI_GET_MOBILE_SW_VERSION
 - pc_mob.h, [103](#)
- MI_GET_SERIAL_NUMBER
 - pc_mob.h, [103](#)
- MI_GET_SERIAL_NUMBER_ANSWER
 - pc_mob.h, [103](#)
- MI_GET_UPDATE_SW_VERSION
 - pc_mob.h, [103](#)
- MI_GIVE_EXT_FLASH_INFO
 - pc_mob.h, [103](#)
- MI_GIVE_FLASH_CODE
 - pc_mob.h, [103](#)

- MI_GIVE_FLASH_SIZE
 - pc_mob.h, [103](#)
- MI_LANGGROUP_INFO
 - pc_mob.h, [103](#)
- MI_LANGGROUP_INFO_ANSWER
 - pc_mob.h, [103](#)
- MI_MAPPING_SEG_END
 - pc_mob.h, [103](#)
- MI_MAPPING_SEG_INFO
 - pc_mob.h, [103](#)
- MI_MEMORY_TESTS
 - pc_mob.h, [103](#)
- MI_MMISWUP_MASTER_SEEK
 - pc_mob.h, [103](#)
- MI_MMISWUP_MASTER_SEEK_ACK
 - pc_mob.h, [103](#)
- MI_MMISWUP_NORM_PROGRAM
 - pc_mob.h, [103](#)
- MI_MMISWUP_NORM_PROGRAM_ACK
 - pc_mob.h, [103](#)
- MI_MMISWUP_RESET_EPROM
 - pc_mob.h, [103](#)
- MI_MMISWUP_SPEZ_PROGRAM
 - pc_mob.h, [103](#)
- MI_MMISWUP_SPEZ_PROGRAM_ACK
 - pc_mob.h, [103](#)
- MI_MMISWUP_TEST_EPROM
 - pc_mob.h, [103](#)
- MI_MMISWUP_TEST_EPROM_ACK
 - pc_mob.h, [103](#)
- MI_MOBILE_ERROR
 - pc_mob.h, [103](#)
- MI_MOBILE_FLASH_SIZE
 - pc_mob.h, [103](#)
- MI_MOBILE_SW_VERSION_ANSWER
 - pc_mob.h, [103](#)
- MI_NEW_BYTE_PROG_ERR
 - pc_mob.h, [103](#)
- MI_NEW_FLASH_CODE
 - pc_mob.h, [103](#)
- MI_PC_BYTE_TIMEOUT_SET
 - pc_mob.h, [103](#)
- MI_PC_TIMEOUT_DEFAULT
 - pc_mob.h, [103](#)
- MI_PC_TIMEOUT_SET
 - pc_mob.h, [103](#)
- MI_PROMMER_VERSION
 - pc_mob.h, [103](#)
- MI_READING_OUT_MASTER
 - pc_mob.h, [103](#)
- MI_REQUEST_MASTER_SW
 - pc_mob.h, [103](#)
- MI_REQUEST_TEST_SW
 - pc_mob.h, [103](#)
- MI_SELECT_MOBILE_MODE
 - pc_mob.h, [103](#)
- MI_SIMULATE_FLASH
 - pc_mob.h, [103](#)
- MI_STARTUP_INFO
 - pc_mob.h, [103](#)
- MI_STATISTIK_DATA
 - pc_mob.h, [103](#)
- MI_STATISTIK_DATA_REST
 - pc_mob.h, [103](#)
- MI_STATISTIK_DATA_REST_ANSWER
 - pc_mob.h, [103](#)
- MI_SW_SIGNATURE_NEGOTIATION
 - pc_mob.h, [103](#)
- MI_SW_SIGNATURE_NEGOTIATION_-ANSWER
 - pc_mob.h, [103](#)
- MI_SWITCH_OFF_MOBILE
 - pc_mob.h, [103](#)
- MI_TEST_AREA
 - pc_mob.h, [103](#)
- MI_TEST_AREA_ACK
 - pc_mob.h, [103](#)
- MI_TEXTSTRING
 - pc_mob.h, [103](#)
- MI_UPDATE_SW_VERSION_ANSWER
 - pc_mob.h, [103](#)
- MobileErrorHandler
 - wmob_err.c, [129](#)
- MobileSwInfoFunction
 - swupwork.c, [121](#)
- MobileUpdateConceptRunningHandler
 - wseril3.c, [143](#)
- MyUpdateMode
 - wfbttool.c, [127](#)
- NAK
 - pc_mob.h, [103](#)
 - wsw_seri.h, [153](#)
- NAK_HOLD
 - pc_mob.h, [103](#)
- nAltSeg1
 - tag_LocaterRecord, [58](#)
- nAltSeg2
 - tag_LocaterRecord, [58](#)
- NegotiateSignatureParameter
 - wseril3.c, [143](#)
- nErtecSum
 - t_ExtendedInfo, [45](#)
- NewSplitId
 - tagNewSplitInfo, [65](#)
- NewUpdateFailed
 - swupwork.c, [121](#)
 - wfbttool.c, [127](#)

- NewUpdateFailed_UseBsl
 - swupwork.c, [114](#)
- NewUpdateSucceeded
 - swupwork.c, [114](#)
- nHexVersion
 - t_ExtendedInfo, [45](#)
- nLineNo
 - t_ErrorInfo, [39](#)
- NO_INIT
 - filehand.c, [89](#)
- nReleaseMajor
 - t_ExtendedInfo, [45](#)
- nReleaseMinor
 - t_ExtendedInfo, [45](#)
- nSegInMobile
 - tag_LocaterRecord, [58](#)
- nSegInRom
 - tag_LocaterRecord, [58](#)
- nSwupVersion
 - t_ExtendedInfo, [46](#)
- nTableEntries
 - t_ExtendedInfo, [46](#)
- offset
 - T_HashBinBlock, [51](#)
- OLD_EXE_ENDUNG_SIZE
 - filehand.c, [89](#)
- pc_mob.h
 - enKnownButBad, [103](#)
 - enOk, [103](#)
 - enSimulated, [104](#)
 - enUnknown, [104](#)
- pc_mob.h, [95](#)
 - ACK, [100](#)
 - ACK_B0, [100](#)
 - ACK_B1, [100](#)
 - ACKNOWLEDGE_A5, [100](#)
 - AUSZEIT, [100](#)
 - BLOCK_TIMEOUT_VAL, [100](#)
 - BUFFER_OVERFLOW, [100](#)
 - BYTE_TIMEOUT_VAL, [100](#)
 - EI_CODE_PART, [100](#)
 - EI_EEPROM_SIMULATION, [100](#)
 - EI_EEPROM_SIMULATION_BLOCK, [100](#)
 - EI_FLASH_DEPENDENT_BLOCK, [100](#)
 - EI_VOICE_DIAL, [100](#)
 - EI_VOICE_DIAL_BLOCK, [100](#)
 - EI_VOICE_MEMO, [100](#)
 - EI_VOICE_MEMO_BLOCK, [100](#)
 - EI_WHOLE_FLASH, [100](#)
 - ERR_BLOCK_ERASE, [100](#)
 - ERR_BYTE_PROG, [100](#)
 - ERR_CHECKSUM, [100](#)
 - ERR_FLASH_BLOCK_ADR, [100](#)
 - ERR_FUNC_NOT_IMPLEMENTED, [100](#)
 - ERR_HASHING, [100](#)
 - ERR_NO_EPROM, [100](#)
 - ERR_RAM_TEST_00, [100](#)
 - ERR_RAM_TEST_FF, [100](#)
 - ERR_RAM_TEST_MUSTER, [100](#)
 - ERR_RAM_TEST_UNKNOWN, [100](#)
 - ERR_SW_REJECTED, [100](#)
 - ERR_UNKNOWN_FLASH, [100](#)
 - IPC_UPDATE_CMD, [100](#)
 - IPC_UPDATE_DATA, [100](#)
 - M_MODE_SELECTION, [100](#)
 - M_MODEM_UPDATE_FINISHED, [100](#)
 - M_MODEM_UPDATE_RUNNING, [101](#)
 - M_MODEM_UPDATE_STARTED, [101](#)
 - M_PC_CONNECTED, [101](#)
 - MFI_GET_ANZ_FLASHES, [103](#)
 - MFI_GET_FLASH_CODE, [103](#)
 - MI_ACK_END_OF_TRANSMISSION, [103](#)
 - MI_ACK_ERASE_FLASH, [103](#)
 - MI_ACK_GIVE_EXT_FLASH_INFO, [103](#)
 - MI_ACK_GIVE_FLASH_CODE, [103](#)
 - MI_ADDITIONAL_MAP_INFO, [103](#)
 - MI_ADDITIONAL_MAP_INFO_END, [103](#)
 - MI_ADDITIONAL_MAP_INFO_LEN, [103](#)
 - MI_ALIGNEMENT, [103](#)
 - MI_CALC_CHECKSUM, [103](#)
 - MI_CALC_CHECKSUM_ANSWER, [103](#)
 - MI_CALC_SPLIT_INFO, [103](#)
 - MI_CALC_SPLIT_INFO_ANSWER, [103](#)
 - MI_COMPRESSION_OFF, [103](#)
 - MI_COMPRESSION_OFF_ANSWER, [103](#)
 - MI_COMPRESSION_ON, [103](#)
 - MI_COMPRESSION_ON_ANSWER, [103](#)
 - MI_CONTROL_COMMAND, [103](#)
 - MI_CURR_BAUDRATE, [103](#)
 - MI_CURR_BAUDRATE_ACK, [103](#)
 - MI_DUMMY, [103](#)
 - MI_ENABLE_FLASHWRITE_ALL, [103](#)
 - MI_ENABLE_FLASHWRITE_BLNR, [103](#)
 - MI_END_OF_TRANSMISSION, [103](#)
 - MI_ERASE_FLASH, [103](#)
 - MI_ERASE_FLASH_AREA, [103](#)
 - MI_ERASE_FLASH_AREA_ACK, [103](#)
 - MI_ERASE_FLASH_UPGRADE_-
CONCEPT, [103](#)
 - MI_ERASE_FLASH_UPGRADE_-
CONCEPT_ACK, [103](#)
 - MI_ERASE_FLASH_UPGRADE_-
CONCEPT_RUNNING, [103](#)
 - MI_EXT_CONTROL_COMMAND, [103](#)
 - MI_FINISH_BOTBLOCK_HANDLING,
[103](#)

- MI_FLASH_CHECK_OK, [103](#)
- MI_FLASH_CODE, [103](#)
- MI_FLASH_ERASE_ADDRESS, [103](#)
- MI_GET_32BIT_NUMBER, [103](#)
- MI_GET_32BIT_NUMBER_ANSWER, [103](#)
- MI_GET_MEMORY_CONTENT, [103](#)
- MI_GET_MEMORY_CONTENT_-ANSWER, [103](#)
- MI_GET_MEMORY_CONTENT_END, [103](#)
- MI_GET_MOBILE_SW_VERSION, [103](#)
- MI_GET_SERIAL_NUMBER, [103](#)
- MI_GET_SERIAL_NUMBER_ANSWER, [103](#)
- MI_GET_UPDATE_SW_VERSION, [103](#)
- MI_GIVE_EXT_FLASH_INFO, [103](#)
- MI_GIVE_FLASH_CODE, [103](#)
- MI_GIVE_FLASH_SIZE, [103](#)
- MI_LANGGROUP_INFO, [103](#)
- MI_LANGGROUP_INFO_ANSWER, [103](#)
- MI_MAPPING_SEG_END, [103](#)
- MI_MAPPING_SEG_INFO, [103](#)
- MI_MEMORY_TESTS, [103](#)
- MI_MMISWUP_MASTER_SEEK, [103](#)
- MI_MMISWUP_MASTER_SEEK_ACK, [103](#)
- MI_MMISWUP_NORM_PROGRAM, [103](#)
- MI_MMISWUP_NORM_PROGRAM_-ACK, [103](#)
- MI_MMISWUP_RESET_EPROM, [103](#)
- MI_MMISWUP_SPEZ_PROGRAM, [103](#)
- MI_MMISWUP_SPEZ_PROGRAM_ACK, [103](#)
- MI_MMISWUP_TEST_EPROM, [103](#)
- MI_MMISWUP_TEST_EPROM_ACK, [103](#)
- MI_MOBILE_ERROR, [103](#)
- MI_MOBILE_FLASH_SIZE, [103](#)
- MI_MOBILE_SW_VERSION_ANSWER, [103](#)
- MI_NEW_BYTE_PROG_ERR, [103](#)
- MI_NEW_FLASH_CODE, [103](#)
- MI_PC_BYTE_TIMEOUT_SET, [103](#)
- MI_PC_TIMEOUT_DEFAULT, [103](#)
- MI_PC_TIMEOUT_SET, [103](#)
- MI_PROMMER_VERSION, [103](#)
- MI_READING_OUT_MASTER, [103](#)
- MI_REQUEST_MASTER_SW, [103](#)
- MI_REQUEST_TEST_SW, [103](#)
- MI_SELECT_MOBILE_MODE, [103](#)
- MI_SIMULATE_FLASH, [103](#)
- MI_STARTUP_INFO, [103](#)
- MI_STATISTIK_DATA, [103](#)
- MI_STATISTIK_DATA_REST, [103](#)
- MI_STATISTIK_DATA_REST_ANSWER, [103](#)
- MI_SW_SIGNATURE_NEGOTIATION, [103](#)
- MI_SW_SIGNATURE_NEGOTIATION_-ANSWER, [103](#)
- MI_SWITCH_OFF_MOBILE, [103](#)
- MI_TEST_AREA, [103](#)
- MI_TEST_AREA_ACK, [103](#)
- MI_TEXTSTRING, [103](#)
- MI_UPDATE_SW_VERSION_ANSWER, [103](#)
- NAK, [103](#)
- NAK_HOLD, [103](#)
- PH_ID_HASHLIST_TRANSMISSION, [103](#)
- PH_ID_HASHLIST_TRANSMISSION_-END, [103](#)
- PH_ID_SIGNATURE_QUERY, [103](#)
- PH_ID_SIGNUM_SELECTION, [103](#)
- PH_ID_SIGNUM_TRANSMISSION, [103](#)
- PH_ID_SIGNUM_TRANSMISSION_END, [103](#)
- PH_ID_START_HASHING, [103](#)
- REC_ERROR, [103](#)
- SIGACK_DATABOUNDARY_-VIOLATION, [103](#)
- SIGACK_HASHLIST_ACCEPTED, [103](#)
- SIGACK_HASHLIST_REJECTED, [103](#)
- SIGACK_NOT_REQUESTED, [103](#)
- SIGACK_PROTOCOL_ERROR, [103](#)
- SIGACK_REQUESTED_AND_HASH_-ACCEPTED, [103](#)
- SIGACK_REQUESTED_HASH_NOT_-SUPPORTED, [103](#)
- SIGACK_SIGNATURE_ACCEPTED, [103](#)
- SIGACK_SIGNATURE_REJECTED, [103](#)
- SIGACK_VERIFICATION_-COMPLETED, [103](#)
- SIGACK_WRONGBOOT_CORE, [103](#)
- STEUER, [103](#)
- t_Flash, [103](#)
- PFN_USBOMAPSWUP_GETDEVICE
swupwork.c, [110](#)
wseril3.c, [142](#)
- PFN_USBOMAPSWUP_-READFROMDATAPIPE
wseril2.c, [136](#)
wseril3.c, [142](#)
- PFN_USBOMAPSWUP_RELEASEDEVICE
swupwork.c, [110](#)
wseril3.c, [142](#)
- PFN_USBOMAPSWUP_-WRITETOCONTROLPIPE

- wseril3.c, [142](#)
- PFN_USBOMAPSWUP_WRITETODATAPIPE
 - wseril2.c, [136](#)
 - wseril3.c, [142](#)
- PFN_WCOMCLOSE
 - wbfbtool.c, [125](#)
- PFN_WCOMOPEN
 - wbfbtool.c, [125](#)
- PFN_WCOMRECEIVE
 - wbfbtool.c, [125](#)
- PFN_WCOMREGISTRY
 - wbfbtool.c, [126](#)
- PFN_WCOMWRITE
 - wbfbtool.c, [126](#)
- PFN_WMOBILEOFF
 - wbfbtool.c, [126](#)
- PFN_WMOBILEON
 - wbfbtool.c, [126](#)
- PFN_WPINGMOBILE
 - wbfbtool.c, [126](#)
- PFN_WSETBFBMODE
 - wbfbtool.c, [126](#)
- pfnDefaultHandler
 - wsu_seri.h, [154](#)
- pfnUpdateCheckCallback
 - wbfbtool.c, [127](#)
- pfnUsbOmapSwup_GetDevice
 - swupwork.c, [121](#)
 - wseril3.c, [146](#)
- pfnUsbOmapSwup_ReadFromDataPipe
 - wseril3.c, [147](#)
- pfnUsbOmapSwup_ReadFromDataPipe_Low
 - wseril2.c, [138](#)
- pfnUsbOmapSwup_ReleaseDevice
 - swupwork.c, [121](#)
 - wseril3.c, [147](#)
- pfnUsbOmapSwup_WriteToControlPipe
 - wseril3.c, [147](#)
- pfnUsbOmapSwup_WriteToDataPipe
 - wseril3.c, [147](#)
- pfnUsbOmapSwup_WriteToDataPipe_Low
 - wseril2.c, [138](#)
- pfnVoltageCheckCallback
 - wbfbtool.c, [127](#)
- pfnWComClose
 - wbfbtool.c, [127](#)
- pfnWComOpen
 - wbfbtool.c, [127](#)
- pfnWComReadRegistry
 - wbfbtool.c, [128](#)
- pfnWComReceive
 - wbfbtool.c, [128](#)
- pfnWComWrite
 - wbfbtool.c, [128](#)
- pfnWMobileOff
 - wbfbtool.c, [128](#)
- pfnWMobileOn
 - wbfbtool.c, [128](#)
- pfnWPingMobile
 - wbfbtool.c, [128](#)
- pfnWSetBFBMode
 - wbfbtool.c, [128](#)
- PH_ID_HASHLIST_TRANSMISSION
 - pc_mob.h, [103](#)
- PH_ID_HASHLIST_TRANSMISSION_END
 - pc_mob.h, [103](#)
- PH_ID_SIGNATURE_QUERY
 - pc_mob.h, [103](#)
- PH_ID_SIGNUM_SELECTION
 - pc_mob.h, [103](#)
- PH_ID_SIGNUM_TRANSMISSION
 - pc_mob.h, [103](#)
- PH_ID_SIGNUM_TRANSMISSION_END
 - pc_mob.h, [103](#)
- PH_ID_START_HASHING
 - pc_mob.h, [103](#)
- PhaseID
 - T_HashBinBlock, [51](#)
- pLibInfo
 - t_ExtendedInfo, [46](#)
- pNext
 - tag_Area, [55](#)
 - tagNewSplitInfo, [65](#)
- PrintErrorMessage
 - filehand.c, [90](#)
- ProcessorInfo
 - t_ExtendedInfo, [46](#)
- prog_debugfun
 - CurrentFileDebugLevel, [29](#)
 - CurrentOnlineDebugLevel, [29](#)
 - fDebugToFile, [29](#)
 - fDebugToWindow, [29](#)
 - wEnabledFileDebugGroups, [29](#)
 - wEnabledOnlineDebugGroups, [29](#)
 - WSwupDebugString, [27](#)
 - WSwupInt_DisableDebuggingToFile, [27](#)
 - WSwupInt_DisableOnlineDebugging, [27](#)
 - WSwupInt_EnableDebuggingToFile, [27](#)
 - WSwupInt_EnableOnlineDebugging, [28](#)
 - WSwupInt_GiveUpdateNrFromComPort, [28](#)
 - WSwupInt_SetUpdateNrToComPort, [28](#)
- prog_errorfun
 - enAdditionalErrText, [30](#)
 - enStandardErrText, [30](#)
- prog_errorfun
 - t_ErrTextInfo, [30](#)
- pszDescription

- t_enumErrorDescription, 36
 - t_ErrorCodeDescription, 37
- pszFile
 - t_ErrorInfo, 39
- ptrEraseList
 - t_ExtendedInfo, 46
- ptrLocTable
 - t_ExtendedInfo, 46
- ptrNewSplitInfoList
 - t_ExtendedInfo, 46
- ptrNext
 - tag_AllLibInfo, 54
 - tag_LocaterRecord, 58
- pucText
 - t_ExtendedInfo, 46
- punThreadVoltage
 - tag_BfbThreadInfo, 56
- ReadAndFormatSigData
 - filehand.c, 91
- ReadKennung
 - filehand.c, 91
- ReadUndCheckAnbootDaten
 - filehand.c, 91
- ReadUndCheckXbiDaten
 - filehand.c, 92
- ReadUndFormatHashCluster
 - filehand.c, 93
- ReadXbiHeader
 - filehand.c, 93
- REC_ERROR
 - pc_mob.h, 103
- ReceiveBlockInSipcMode
 - layer2func, 21
- ReceiveCmdBlockInBinMode
 - layer2func, 22
- ReleaseXbiHeaderData
 - filehand.c, 94
- RequestBlockNumbersFromMobile
 - wseril3.c, 143
- ScanDirectoryForAvailableProjects
 - dyna_dll.c, 81
- SECURITY_RESERVE
 - fhawrap.c, 85
- SEGM_SIZE
 - swupwork.c, 110
- SendAckNakInSipcMode
 - wseril2.c, 136
- SendSimpleMessageInSipcMode
 - layer2func, 22
- SetErrorForAllComPorts
 - swupwork.c, 114
- SetPreCheckInfos
 - swupwork.c, 114
- SIG_ID_STRING
 - boot_hex.h, 78
- SIG_ID_STRING_SGOLD
 - boot_hex.h, 78
- SIGACK_DATABOUNDARY_VIOLATION
 - pc_mob.h, 103
- SIGACK_HASHLIST_ACCEPTED
 - pc_mob.h, 103
- SIGACK_HASHLIST_REJECTED
 - pc_mob.h, 103
- SIGACK_NOT_REQUESTED
 - pc_mob.h, 103
- SIGACK_PROTOCOL_ERROR
 - pc_mob.h, 103
- SIGACK_REQUESTED_AND_HASH_-ACCEPTED
 - pc_mob.h, 103
- SIGACK_REQUESTED_HASH_NOT_-SUPPORTED
 - pc_mob.h, 103
- SIGACK_SIGNATURE_ACCEPTED
 - pc_mob.h, 103
- SIGACK_SIGNATURE_REJECTED
 - pc_mob.h, 103
- SIGACK_VERIFICATION_COMPLETED
 - pc_mob.h, 103
- SIGACK_WRONGBOOT_CORE
 - pc_mob.h, 103
- SignatureTransmissionProgress
 - wseril3.c, 143
- SipcLoadUpdateSw
 - wseril3.c, 143
 - wsw_seri.h, 155
- SIZE_IN_BYTES
 - swupwork.c, 110
- SoftwareUpdateThread
 - swupwork.c, 115
- SpecialSprintf
 - wswerrex.c, 164
- SplitCountID
 - tagNewSplitInfo, 65
- STEUER
 - pc_mob.h, 103
- SwitchMobileToNewBootMode
 - wfbttool.c, 127
- SwType
 - t_ExtendedInfo, 46
- SwupEvenParity
 - wsw_seri.h, 155
- SWUPLIBDLL_DEVELOPER
 - swupwork.c, 110
- SWUPLIBDLL_VERS_MAJOR
 - swupwork.c, 110

- SWUPLIBDLL_VERS_MINOR
 - swupwork.c, 110
- SwupNoParity
 - ws_wseri.h, 155
- SwupOddParity
 - ws_wseri.h, 155
- SWUPSERIDLL_DEVELOPER
 - wseril2.c, 135
- SWUPSERIDLL_VERS_MAJOR
 - wseril2.c, 135
- SWUPSERIDLL_VERS_MINOR
 - wseril2.c, 135
- SWUPSERILOWEXIMPORT
 - ws_wseri.h, 153
- swupwork.c, 105
 - abUpdateSwData, 120
 - BootCoreStartUpdateSW_LastHope, 110
 - BootCoreStartUpdateSWFunc, 120
 - BootStrapLoaderFunc, 120
 - CheckIfThisDiffSwUpdateIsOk, 110
 - CloseDevice, 111
 - ConvertStreamToSwInformationStruct, 111
 - fCallbacksInstalled, 120
 - fLibraryIsInitialised, 120
 - FreeList, 111
 - fSimulateKnownProject, 120
 - GeneralUpdateMode, 120
 - GiveGeneralUpdateMode, 111
 - GiveSwupLibDate, 112
 - GiveUpdateDllPcSwVersion, 120
 - GiveUpdateMobileSwVersion, 120
 - GiveUpdateModeForThisUpdate, 112
 - GiveUpdatePtrAndLenFunc, 120
 - GlobalWhichInfoCallBackTime, 120
 - hUpdateLib, 121
 - InitGlobalMemory, 112
 - InitMobileErrorHandler, 112
 - InitProjectDependentBootstrapDll, 113
 - Invalidate_BGJ, 113
 - IsLibraryInitialised, 113
 - IsSwupLibVersionNewEnough, 113
 - IsThisADownGrade, 114
 - MAX_SIZE_OF_PIN, 110
 - MobileSwInfoFunction, 121
 - NewUpdateFailed, 121
 - NewUpdateFailed_UseBsl, 114
 - NewUpdateSucceeded, 114
 - PFN_USBOMAPSWUP_GETDEVICE, 110
 - PFN_USBOMAPSWUP_-RELEASEDEVICE, 110
 - pfnUsbOmapSwup_GetDevice, 121
 - pfnUsbOmapSwup_ReleaseDevice, 121
 - SEGM_SIZE, 110
 - SetErrorForAllComPorts, 114
 - SetPreCheckInfos, 114
 - SIZE_IN_BYTES, 110
 - SoftwareUpdateThread, 115
 - SWUPLIBDLL_DEVELOPER, 110
 - SWUPLIBDLL_VERS_MAJOR, 110
 - SWUPLIBDLL_VERS_MINOR, 110
 - szDerivedFromKnownProject, 121
 - szNewUnknownProject, 121
 - t_PINStruct, 110
 - t_ThreadInfo, 110
 - ThreadInfo, 121
 - ulAmountBytesUpdateData, 121
 - unTotalHeapBytes, 121
 - UpdateModes, 121
 - UsbSettings, 121
 - WSwup_CheckComPortAndSpeed, 115
 - WSwup_GiveVersionInformation, 115
 - WSwup_InitLibrary, 116
 - WSwup_InstallSoftwareInfoCallBackFunction, 116
 - WSwup_InstallSoftwareUpdateCallBackFunctions, 117
 - WSwup_PerformSoftwareUpdate, 117
 - WSwup_PrepareForUpdateWithSynchStation, 118
 - WSwup_PrepareForUpdateWithUnknownProject, 118
 - WSwup_ReloadUpdateDll, 118
 - WSwup_ResetUsbParams, 119
 - WSwup_SetBootPIN, 119
 - WSwup_SetUpdateConceptMode, 119
 - WSwup_SetUsbParams, 119
 - XbiHeaderInfo, 121
- szCommentAboutSplitEntity
 - tagNewSplitInfo, 65
- szDerivedFromKnownProject
 - dyna_dll.c, 81
 - swupwork.c, 121
- szDevIName
 - t_ExtendedInfo, 46
- szErrBuff
 - t_ErrorInfo, 39
- szFileErrorString
 - fiwrap.c, 86
- szLangGroup
 - t_ExtendedInfo, 46
- szLib_1_Date
 - t_ExtendedInfo, 47
- szLib_1_Time
 - t_ExtendedInfo, 47
- szLib_2_Date
 - t_ExtendedInfo, 47
- szLib_2_Time

- t_ExtendedInfo, 47
- szLocDate
 - t_ExtendedInfo, 47
- szLocTime
 - t_ExtendedInfo, 47
- szNewUnknownProject
 - dyna_dll.c, 81
 - swupwork.c, 121
- szOldDevlName
 - t_ExtendedInfo, 47
- szOldLocDate
 - t_ExtendedInfo, 47
- szOldLocTime
 - t_ExtendedInfo, 47
- szOldProjName
 - t_ExtendedInfo, 47
- szOldRbmDate
 - t_ExtendedInfo, 47
- szOldRbmTime
 - t_ExtendedInfo, 48
- szProductName
 - t_ExtendedInfo, 48
- szProjName
 - t_ExtendedInfo, 48
- szRbmDate
 - t_ExtendedInfo, 48
- szRbmTime
 - t_ExtendedInfo, 48
- szSwGeneration
 - t_ExtendedInfo, 48
- szSwupDllName
 - t_ExtendedInfo, 48
- szVendorName
 - t_ExtendedInfo, 48
- t_AllLibInfo
 - xbi_info.h, 174
- t_Area
 - xbi_info.h, 174
- t_AsicTyp
 - xbi_info.h, 174
- t_BfbStates
 - wbfbtool.c, 126
- t_BfbThreadInfo
 - wbfbtool.c, 126
- t_Block, 34
 - aucDataBuf, 34
 - ucId, 34
 - ucLen, 34
- t_ComConfig, 35
- t_ComConfig
 - ctmo, 35
 - dcb, 35
 - fInit, 35
- hCom, 35
- t_CompressInfo
 - xbi_info.h, 174
- t_DebugFiles
 - wswdbgin.c, 163
- t_enumErrorDescription, 36
- t_enumErrorDescription
 - pszDescription, 36
 - ui_enAltError, 36
 - ui_enError, 36
- t_enumErrorDescriptionSize
 - err_text.c, 82
 - err_text.h, 83
- t_ErrorCodeDescription, 37
- t_ErrorCodeDescription
 - dwErrCode, 37
 - pszDescription, 37
- t_ErrorField, 38
- t_ErrorField
 - bIndex, 38
 - Error, 38
- t_ErrorInfo, 39
- t_ErrorInfo
 - dwError, 39
 - ErrTextInfo, 39
 - nLineNo, 39
 - pszFile, 39
 - szErrBuff, 39
- t_ErrTextInfo
 - prog_errorfun, 30
- t_ExtendedInfo, 41
- t_ExtendedInfo
 - CompFormatAfter, 44
 - CompFormatBefore, 44
 - CompressionInfo, 44
 - fAdditionalInfoAvail, 45
 - fForceDllNameFromNhk, 45
 - FormatInfo, 45
 - HashInfo, 45
 - HowToWrite, 45
 - IgnitionInfo, 45
 - LanguageCheckInfo, 45
 - nErtecSum, 45
 - nHexVersion, 45
 - nReleaseMajor, 45
 - nReleaseMinor, 45
 - nSwupVersion, 46
 - nTableEntries, 46
 - pLibInfo, 46
 - ProcessorInfo, 46
 - ptrEraseList, 46
 - ptrLocTable, 46
 - ptrNewSplitInfoList, 46
 - pucText, 46

- SwType, 46
- szDevlName, 46
- szLangGroup, 46
- szLib_1_Date, 47
- szLib_1_Time, 47
- szLib_2_Date, 47
- szLib_2_Time, 47
- szLocDate, 47
- szLocTime, 47
- szOldDevlName, 47
- szOldLocDate, 47
- szOldLocTime, 47
- szOldProjName, 47
- szOldRbmDate, 47
- szOldRbmTime, 48
- szProductName, 48
- szProjName, 48
- szRbmDate, 48
- szRbmTime, 48
- szSwGeneration, 48
- szSwupDllName, 48
- szVendorName, 48
- ucAdditionalMapInfo, 48
- ulConstAdress, 48
- ulEpromSizeInBytes, 48
- ulRbcPutCount, 49
- ulRbmPutCount, 49
- ulSplitID, 49
- ulSplitIdAdress, 49
- ulTextLen, 49
- unAdditionalMapLen, 49
- unAlignement, 49
- unCompAlg, 49
- unCompressRatio, 49
- unRamSize, 49
- WhichAsic, 49
- t_FileFormat
 - xbi_info.h, 174
- t_Flash
 - pc_mob.h, 103
- T_HashBinBlock, 51
- T_HashBinBlock
 - Address, 51
 - Checksum, 51
 - Data, 51
 - Length, 51
 - MI, 51
 - offset, 51
 - PhaseID, 51
- t_IgnType
 - xbi_info.h, 175
- t_KnownReceiveModes
 - wsw_seri.h, 154
- t_LangCheckInfo, 52
 - ulChecksumAdr, 52
 - ulEndAdr, 52
 - ulPatchAdr, 52
 - ulStartAdr, 52
- t_LocaterRecord
 - xbi_info.h, 174
- t_MessageBlock
 - wsw_seri.h, 154
- t_NewSplitIdentifier
 - xbi_info.h, 175
- t_NewSplitInfo
 - xbi_info.h, 174
- t_PINStruct
 - swupwork.c, 110
- t_ProcType
 - xbi_info.h, 175
- t_RecResult
 - wsw_seri.h, 154
- t_SwType
 - xbi_info.h, 175
- t_SwupParity
 - wsw_seri.h, 154
- t_TableEntry, 53
- t_TableEntry
 - DefaultHandler, 53
 - ucMi, 53
- t_ThreadInfo
 - swupwork.c, 110
- t_UsbInfo
 - wsw_seri.h, 154
- t_WriteTyp
 - xbi_info.h, 176
- t_XbiOrExe
 - xbi_info.h, 176
- tag_AllLibInfo, 54
- tag_AllLibInfo
 - abInfo, 54
 - ptrNext, 54
- tag_Area, 55
 - pNext, 55
 - ulEndAdress, 55
 - ulStartAdress, 55
- tag_BfbThreadInfo, 56
- tag_BfbThreadInfo
 - punThreadVoltage, 56
 - ThreadWhichCom, 56
 - wThreadUpdateNr, 56
- tag_CompressInfo, 57
- tag_CompressInfo
 - unInfoOne, 57
 - unInfoThree, 57
 - unInfoTwo, 57
- tag_LocaterRecord, 58

- tag_LocaterRecord
 - nAltSeg1, 58
 - nAltSeg2, 58
 - nSegInMobile, 58
 - nSegInRom, 58
 - ptrNext, 58
- tag_PINstruct, 60
 - aucBootPIN, 60
 - unPINSize, 60
- tag_ThreadInfo, 61
- tag_ThreadInfo
 - ThreadWhichCom, 61
 - ulThreadSpeed, 61
 - wThreadUpdateNr, 61
- tag_UsbInfo, 62
- tag_UsbInfo
 - ForceUsb, 62
 - UsbPort, 62
 - UsbWanted, 62
- tagDebugFiles, 63
- tagDebugFiles
 - ComPort, 63
 - fpDebug, 63
- tagMessageBlock, 64
- tagMessageBlock
 - ucLen, 64
 - ucMessageData, 64
 - ucMi, 64
- tagNewSplitInfo, 65
- tagNewSplitInfo
 - NewSplitId, 65
 - pNext, 65
 - SplitCountID, 65
 - szCommentAboutSplitEntity, 65
 - ulChecksumAdr, 66
 - ulEndAdr, 66
 - ulStartAdr, 66
- ThreadInfo
 - swupwork.c, 121
- ThreadWhichCom
 - tag_BfbThreadInfo, 56
 - tag_ThreadInfo, 61
- ucAdditionalMapInfo
 - t_ExtendedInfo, 48
- ucId
 - t_Block, 34
- ucLen
 - t_Block, 34
 - tagMessageBlock, 64
- ucMessageData
 - tagMessageBlock, 64
- ucMi
 - t_TableEntry, 53
- tagMessageBlock, 64
- ui_enAltError
 - t_enumErrorDescription, 36
- ui_enError
 - t_enumErrorDescription, 36
- ulAmountBytesUpdateData
 - swupwork.c, 121
- ulChecksumAdr
 - t_LangCheckInfo, 52
 - tagNewSplitInfo, 66
- ulConstAddress
 - t_ExtendedInfo, 48
- ulEndAdr
 - t_LangCheckInfo, 52
 - tagNewSplitInfo, 66
- ulEndAddress
 - tag_Area, 55
- ulEpromSizeInBytes
 - t_ExtendedInfo, 48
- ulFileSize
 - FileInfo, 32
- ulHashClusterLen
 - FileInfo, 32
- ulPatchAdr
 - t_LangCheckInfo, 52
- ulRbcPutCount
 - t_ExtendedInfo, 49
- ulRbmPutCount
 - t_ExtendedInfo, 49
- ulSignatureLen
 - FileInfo, 32
- ulSplitID
 - t_ExtendedInfo, 49
- ulSplitIdAddress
 - t_ExtendedInfo, 49
- ulStartAdr
 - t_LangCheckInfo, 52
 - tagNewSplitInfo, 66
- ulStartAddress
 - tag_Area, 55
- ulSwupExeLen
 - FileInfo, 32
- ulTextLen
 - t_ExtendedInfo, 49
- ulThreadSpeed
 - tag_ThreadInfo, 61
- ulXbiDatenEnde
 - FileInfo, 32
- ulXbiDatenStart
 - FileInfo, 32
- ulXbiHeaderLen
 - FileInfo, 33
- ulXbiLen
 - FileInfo, 33

- unAdditionalMapLen
 - t_ExtendedInfo, [49](#)
- unAlignement
 - t_ExtendedInfo, [49](#)
- unCompAlg
 - t_ExtendedInfo, [49](#)
- unCompressRatio
 - t_ExtendedInfo, [49](#)
- unExeEndungLen
 - FileInfo, [33](#)
- unInfoOne
 - tag_CompressInfo, [57](#)
- unInfoThree
 - tag_CompressInfo, [57](#)
- unInfoTwo
 - tag_CompressInfo, [57](#)
- unPINSize
 - tag_PINstruct, [60](#)
- unRamSize
 - t_ExtendedInfo, [49](#)
- unTotalAmountOfBlocks
 - fihawrap.c, [86](#)
- unTotalHeapBytes
 - fihawrap.c, [86](#)
 - swupwork.c, [121](#)
- UpdateModes
 - swupwork.c, [121](#)
- USB_NORMAL_MODE
 - wsu_seri.h, [153](#)
- USB_SIPC_MODE
 - wsu_seri.h, [153](#)
- UsbPort
 - tag_UsbInfo, [62](#)
- UsbSettings
 - swupwork.c, [121](#)
 - wseril3.c, [147](#)
- UsbWanted
 - tag_UsbInfo, [62](#)
- W_SWUPHELPSAPI
 - wfbftool.c, [125](#)
 - wseril3.c, [142](#)
- wfbftool.c, [122](#)
 - BfbCheckThread, [126](#)
 - BfbThreadInfo, [127](#)
 - DeactBfbLibrary, [126](#)
 - DllExImport, [125](#)
 - dwBaudrate, [127](#)
 - enComOpen, [126](#)
 - enError, [126](#)
 - enGetVoltage, [126](#)
 - enMobileOn, [126](#)
 - enPingFirst, [126](#)
 - enPingSecond, [126](#)
 - enPingThird, [126](#)
 - enReady, [126](#)
 - enSwitchOff, [126](#)
 - enWaitALittleBit, [126](#)
 - GetActVoltage, [126](#)
 - GivePutcountAndDirectory, [127](#)
 - hBfbLib, [127](#)
 - InitBfbLibrary, [127](#)
 - M_SWH_CUSTOMER_REBOOT, [125](#)
 - M_SWH_GET_POWER, [125](#)
 - M_SWH_GET_PUTCOUNT, [125](#)
 - M_SWH_IGNITION, [125](#)
 - M_SWH_MOBILE_OFF, [125](#)
 - M_SWH_VERSION_NUMBER, [125](#)
 - M_SWH_VOLTAGE, [125](#)
 - MAX_SUCCESS_CALLBACKS, [125](#)
 - MAX_VOLTAGE_CALLBACKS, [125](#)
 - MyUpdateMode, [127](#)
 - NewUpdateFailed, [127](#)
 - PFN_WCOMCLOSE, [125](#)
 - PFN_WCOMOPEN, [125](#)
 - PFN_WCOMRECEIVE, [125](#)
 - PFN_WCOMREGISTRY, [126](#)
 - PFN_WCOMWRITE, [126](#)
 - PFN_WMOBILEOFF, [126](#)
 - PFN_WMOBILEON, [126](#)
 - PFN_WPINGMOBILE, [126](#)
 - PFN_WSETBFBMODE, [126](#)
 - pfnUpdateCheckCallback, [127](#)
 - pfnVoltageCheckCallback, [127](#)
 - pfnWComClose, [127](#)
 - pfnWComOpen, [127](#)
 - pfnWComReadRegistry, [128](#)
 - pfnWComReceive, [128](#)
 - pfnWComWrite, [128](#)
 - pfnWMobileOff, [128](#)
 - pfnWMobileOn, [128](#)
 - pfnWPingMobile, [128](#)
 - pfnWSetBFBMode, [128](#)
 - SwitchMobileToNewBootMode, [127](#)
 - t_BfbStates, [126](#)
 - t_BfbThreadInfo, [126](#)
 - W_SWUPHELPSAPI, [125](#)
 - WSwup_CheckUpdateSuccess, [127](#)
 - WSwup_CheckUpdateVoltage, [127](#)
 - WSwup_InstallBfbCallBackFunctions, [127](#)
 - XbiHeaderInfo, [128](#)
- wEnabledFileDebugGroups
 - prog_debugfun, [29](#)
- wEnabledOnlineDebugGroups
 - prog_debugfun, [29](#)
- WhichAsic
 - t_ExtendedInfo, [49](#)
- wmob_err.c, [129](#)

- InitMobileErrorHandler, 129
- MobileErrorHandler, 129
- wseril1.c, 130
 - ADDI.COMTYPE, 131
 - ADDIFAST_INI_NAME, 131
 - ComPorts, 132
 - DEFAULT.COMTYPE, 131
 - WSwupGiveHandleForComport, 132
- wseril2.c, 133
 - ACK_LEN_TO_SEND, 135
 - ActualMessage, 137
 - CheckSipcMessage, 136
 - CurrentReceiveMode, 137
 - DefaultHandlerTable, 137
 - hIBootLibLow, 137
 - PFN_USBOMAPSWUP_-
 - READFROMDATAPIPE, 136
 - PFN_USBOMAPSWUP_-
 - WRITETODATAPIPE, 136
 - pfnUsbOmapSwup_ReadFromDataPipe_-
 - Low, 138
 - pfnUsbOmapSwup_WriteToDataPipe_Low, 138
 - SendAckNakInSipcMode, 136
 - SWUPSERIDLL_DEVELOPER, 135
 - SWUPSERIDLL_VERS_MAJOR, 135
 - SWUPSERIDLL_VERS_MINOR, 135
 - WSwSendDataBlockInBinFormat, 136
 - WSwSendDataBlockInSipcFormat, 136
 - WSwSipcWriteToDataPipe, 137
 - WSwSipcWriteToDataPipe_CoproMode, 137
- wseril3.c, 139
 - CONTROL_BIN_BLOCKS_SUPPORTED, 142
 - DeactIBootLibrary, 142
 - dummy, 142
 - DummyUsbOmapSwup_GetDevice, 143
 - DummyUsbOmapSwup_-
 - ReadFromDataPipe, 143
 - DummyUsbOmapSwup_ReleaseDevice, 143
 - DummyUsbOmapSwup_-
 - WriteToControlPipe, 143
 - DummyUsbOmapSwup_WriteToDataPipe, 143
 - EnableWriteFlashBlock, 143
 - ERASE_AREA_NOT_OK, 142
 - ERASE_AREA_OK, 142
 - ERASE_TIME_FLASH_SEGMENT, 142
 - EraseFlashBlocksWithAcknowledge, 143
 - EraseWithNewUpdateConcept, 143
 - HashListTransmissionProgress, 143
 - hIBootLib, 146
 - InitIBootLibrary, 143
 - M_SWH_VOLTAGE, 142
 - MobileUpdateConceptRunningHandler, 143
 - NegotiateSignatureParameter, 143
 - PFN_USBOMAPSWUP_GETDEVICE, 142
 - PFN_USBOMAPSWUP_-
 - READFROMDATAPIPE, 142
 - PFN_USBOMAPSWUP_-
 - RELEASEDEVICE, 142
 - PFN_USBOMAPSWUP_-
 - WRITETOCONTROLPIPE, 142
 - PFN_USBOMAPSWUP_-
 - WRITETODATAPIPE, 142
 - pfnUsbOmapSwup_GetDevice, 146
 - pfnUsbOmapSwup_ReadFromDataPipe, 147
 - pfnUsbOmapSwup_ReleaseDevice, 147
 - pfnUsbOmapSwup_WriteToControlPipe, 147
 - pfnUsbOmapSwup_WriteToDataPipe, 147
 - RequestBlockNumbersFromMobile, 143
 - SignatureTransmissionProgress, 143
 - SipcLoadUpdateSw, 143
 - UsbSettings, 147
 - W_SWUPHELPSAPI, 142
 - WSwup_Usb_GetVoltage, 144
 - WSwup_USB_ModeSwitch, 144
 - WSwup_UsbReconnected, 144
 - WSwupEnableFlashProgrammingWholeFlash, 144
 - WSwupEraseFlashBlocks, 146
 - WSwupEraseOneFlashBlock, 146
 - WSwupGiveMobileSwVersion, 146
 - WSwupRequestEraseBlocks, 146
 - WSwupRequestFlashCode, 146
 - WSwupRequestFlashSize, 146
 - WSwupSelectMobileMode, 146
 - WSwupSendAdditionalInfo, 146
 - WSwupSendAlignement, 146
 - WSwupSendBlockTimeOut, 146
 - WSwupSendByteTimeOut, 146
 - WSwupSendCommandData, 146
 - WSwupSendCompressionInfo, 146
 - WSwupSendData, 146
 - WSwupSendEotBlock, 146
 - WSwupSendEraseCodeAreaNewUpdateConcept, 146
 - WSwupSendLanguageGroupInfo, 146
 - WSwupSendNewSplitInfo, 146
 - WSwupSendSplitIDInfo, 146
 - WSwupSendStartHashing, 146
 - WSwupSendSwitchOff, 146

- WSwupSendSwitchOffCompression, 146
- ws_wseri.h
 - enBinMode, 154
 - enChkErr, 154
 - enCmdBlockReceived, 154
 - enIncomplete, 154
 - enLenChkMode, 154
 - enMobileError, 154
 - enRecOk, 154
 - enRecTimeout, 154
 - enSerialError, 154
 - enSipcCmd, 154
 - enSipcMode, 154
 - enUnknownChar, 154
 - SwupEvenParity, 155
 - SwupNoParity, 155
 - SwupOddParity, 155
- ws_wseri.h, 148
 - A5_ACK, 152
 - ACK, 152
 - ACK_HOLD, 153
 - ADDRESS_LEN_BIN_BLOCK, 153
 - ArrayEntries, 153
 - DeactIBootLibrary, 155
 - DEFAULT_TIMEOUT_BLOCK_REC, 153
 - DEFAULT_TIMEOUT_SINGLE_CHAR, 153
 - InitIBootLibrary, 155
 - MAX_DATA_IN_BIN_BLOCK, 153
 - MAX_RETRY_ON_NAK, 153
 - NAK, 153
 - pfnDefaultHandler, 154
 - SipcLoadUpdateSw, 155
 - SWUPSERILOWEXIMPORT, 153
 - t_KnownReceiveModes, 154
 - t_MessageBlock, 154
 - t_RecResult, 154
 - t_SwupParity, 154
 - t_UsbInfo, 154
 - USB_NORMAL_MODE, 153
 - USB_SIPC_MODE, 153
 - WSwInternalSetLastError, 153
 - WSwInternalSetLastErrorWithFileInfo, 155
 - WSwSendDataBlockInBinFormat, 156
 - WSwSendDataBlockInSipcFormat, 156
 - WSwSetLastErrorText, 156
 - WSwSipcWriteToDataPipe.CoproMode, 156
 - WSwup_Usb_GetVoltage, 156
 - WSwup_USB_ModeSwitch, 157
 - WSwup_UsbReconnected, 157
 - WSwupGetCurrentReceiveMode, 157
 - WSwupGiveHandleForComport, 157
 - WSwupInt_GetLastError, 157
 - WSwupInt_GetLastErrorString, 158
 - WSwupIntForAllUpdatesSetError, 158
 - WSwupIntGiveErrorList, 158
 - WSwupIntGiveLastError, 158
 - WSwupIntSetError, 159
 - WSwupIntSetErrorAllUpdates, 159
- ws_wtool.c, 160
 - GetIntelShort, 160
 - GetMotorolaLongFromBuffer, 160
 - InsertLongToStringIntel, 160
 - InsertLongToStringMotorola, 160
 - InsertWordToStringIntel, 160
- ws_wdbgex.c, 161
 - WSwup_DisableFileDebugging, 161
 - WSwup_DisableOnlineDebugging, 161
 - WSwup_EnableFileDebugging, 161
 - WSwup_EnableOnlineDebugging, 161
- ws_wdbgin.c, 162
 - DebugFiles, 163
 - t_DebugFiles, 163
- ws_werrex.c, 164
 - SpecialPrintf, 164
 - WSwup_GetLastError, 164
 - WSwup_GetLastErrorString, 165
 - WSwup_GiveLastError, 165
 - WSwup_OldGetLastErrorString, 165
 - WSwup_SetErrorTextBehaviourToEnduser, 165
- ws_werrin.c, 166
 - WSwInternalSetLastErrorWithFileInfo, 167
 - WSwSetLastErrorText, 167
 - WSwupInt_GetLastError, 168
 - WSwupInt_GetLastErrorString, 168
 - WSwupIntForAllUpdatesSetError, 168
 - WSwupIntGiveErrorList, 169
 - WSwupIntGiveLastError, 169
 - WSwupIntSetError, 169
 - WSwupIntSetErrorAllUpdates, 170
- WSwInternalSetLastError
 - ws_wseri.h, 153
- WSwInternalSetLastErrorWithFileInfo
 - ws_wseri.h, 155
 - ws_werrin.c, 167
- WSwSendDataBlockInBinFormat
 - ws_wseri2.c, 136
 - ws_wseri.h, 156
- WSwSendDataBlockInSipcFormat
 - ws_wseri2.c, 136
 - ws_wseri.h, 156
- WSwSetLastErrorText
 - ws_wseri.h, 156
 - ws_werrin.c, 167

- WSwSipcWriteToDataPipe
 - wseril2.c, [137](#)
- WSwSipcWriteToDataPipe_CoproMode
 - wseril2.c, [137](#)
 - ws_wseri.h, [156](#)
- WSwup_CheckComPortAndSpeed
 - swupwork.c, [115](#)
- WSwup_CheckUpdateSuccess
 - wfbttool.c, [127](#)
- WSwup_CheckUpdateVoltage
 - wfbttool.c, [127](#)
- WSwup_CloseXbiFile
 - fiwrap.c, [86](#)
- WSwup_DisableFileDebugging
 - wsdbgex.c, [161](#)
- WSwup_DisableOnlineDebugging
 - wsdbgex.c, [161](#)
- WSwup_EnableFileDebugging
 - wsdbgex.c, [161](#)
- WSwup_EnableOnlineDebugging
 - wsdbgex.c, [161](#)
- WSwup_GetLastError
 - wserrx.c, [164](#)
- WSwup_GetLastErrorString
 - wserrx.c, [165](#)
- WSwup_GetLastFileError
 - fiwrap.c, [86](#)
- WSwup_GiveLastError
 - wserrx.c, [165](#)
- WSwup_GiveVersionInformation
 - swupwork.c, [115](#)
- WSwup_InitLibrary
 - swupwork.c, [116](#)
- WSwup_InstallBfbCallBackFunctions
 - wfbttool.c, [127](#)
- WSwup_InstallSoftwareInfoCallBackFunction
 - swupwork.c, [116](#)
- WSwup_InstallSoftwareUpdateCallBackFunctions
 - swupwork.c, [117](#)
- WSwup_OldGetLastErrorString
 - wserrx.c, [165](#)
- WSwup_PerformSoftwareUpdate
 - swupwork.c, [117](#)
- WSwup_PrepareForUpdateWithSynchStation
 - swupwork.c, [118](#)
- WSwup_PrepareForUpdateWithUnknownProject
 - swupwork.c, [118](#)
- WSwup_ReadXbiFile
 - fiwrap.c, [86](#)
- WSwup_ReloadUpdateDll
 - swupwork.c, [118](#)
- WSwup_ResetUsbParams
 - swupwork.c, [119](#)
- WSwup_SetBootPIN
 - swupwork.c, [119](#)
- WSwup_SetErrorTextBehaviourToEnduser
 - wserrx.c, [165](#)
- WSwup_SetUpdateConceptMode
 - swupwork.c, [119](#)
- WSwup_SetUsbParams
 - swupwork.c, [119](#)
- WSwup_Usb_GetVoltage
 - wseril3.c, [144](#)
 - ws_wseri.h, [156](#)
- WSwup_USB_ModeSwitch
 - wseril3.c, [144](#)
 - ws_wseri.h, [157](#)
- WSwup_UsbReconnected
 - wseril3.c, [144](#)
 - ws_wseri.h, [157](#)
- WSwup_ClearIgnition
 - layer1func, [17](#)
- WSwup_ComClose
 - layer1func, [17](#)
- WSwup_ComConfigure
 - layer1func, [17](#)
- WSwup_ComOpen
 - layer1func, [17](#)
- WSwup_ComReadByte
 - layer1func, [18](#)
- WSwup_ComReadByteTimeOut
 - layer1func, [18](#)
- WSwup_ComWriteByte
 - layer1func, [18](#)
- WSwup_ComWriteDataBlock
 - layer1func, [19](#)
- WSwup_DebugString
 - prog_debugfun, [27](#)
- WSwup_EnableFlashProgrammingWholeFlash
 - wseril3.c, [144](#)
- WSwup_EraseFlashBlocks
 - wseril3.c, [146](#)
- WSwup_EraseOneFlashBlock
 - wseril3.c, [146](#)
- WSwup_GetCurrentReceiveMode
 - ws_wseri.h, [157](#)
- WSwup_GiveHandleForComport
 - wseril1.c, [132](#)
 - ws_wseri.h, [157](#)
- WSwup_GiveMobileSwVersion
 - wseril3.c, [146](#)
- WSwupInt_DisableDebuggingToFile
 - prog_debugfun, [27](#)
- WSwupInt_DisableOnlineDebugging
 - prog_debugfun, [27](#)
- WSwupInt_EnableDebuggingToFile
 - prog_debugfun, [27](#)
- WSwupInt_EnableOnlineDebugging

- prog.debugfun, 28
- WSwupInt_GetLastError
 - ws_wseri.h, 157
 - ws_werrin.c, 168
- WSwupInt_GetLastErrorString
 - ws_wseri.h, 158
 - ws_werrin.c, 168
- WSwupInt_GiveUpdateNrFromComPort
 - prog.debugfun, 28
- WSwupInt_SetUpdateNrToComPort
 - prog.debugfun, 28
- WSwupIntForAllUpdatesSetError
 - ws_wseri.h, 158
 - ws_werrin.c, 168
- WSwupIntGiveErrorList
 - ws_wseri.h, 158
 - ws_werrin.c, 169
- WSwupIntGiveLastError
 - ws_wseri.h, 158
 - ws_werrin.c, 169
- WSwupIntSetError
 - ws_wseri.h, 159
 - ws_werrin.c, 169
- WSwupIntSetErrorAllUpdates
 - ws_wseri.h, 159
 - ws_werrin.c, 170
- ws_wupprog.dox, 171
- WSwupReceiveCmdBlock
 - layer2func, 22
- WSwupReceiveValueFromSerial
 - layer2func, 23
- WSwupRequestEraseBlocks
 - ws_wseril3.c, 146
- WSwupRequestFlashCode
 - ws_wseril3.c, 146
- WSwupRequestFlashSize
 - ws_wseril3.c, 146
- WSwupSelectMobileMode
 - ws_wseril3.c, 146
- WSwupSendAdditionalInfo
 - ws_wseril3.c, 146
- WSwupSendAlignement
 - ws_wseril3.c, 146
- WSwupSendBlockTimeOut
 - ws_wseril3.c, 146
- WSwupSendBotBlock
 - layer2func, 23
- WSwupSendByteTimeOut
 - ws_wseril3.c, 146
- WSwupSendCommandData
 - ws_wseril3.c, 146
- WSwupSendCompressionInfo
 - ws_wseril3.c, 146
- WSwupSendData
 - ws_wseril3.c, 146
- WSwupSendEotBlock
 - ws_wseril3.c, 146
- WSwupSendEraseCodeAreaNewUpdateConcept
 - ws_wseril3.c, 146
- WSwupSendLanguageGroupInfo
 - ws_wseril3.c, 146
- WSwupSendMessageBlock
 - layer2func, 23
- WSwupSendNewSplitInfo
 - ws_wseril3.c, 146
- WSwupSendSGoldBotBlock
 - layer2func, 24
- WSwupSendSplitIDInfo
 - ws_wseril3.c, 146
- WSwupSendStartHashing
 - ws_wseril3.c, 146
- WSwupSendSwitchOff
 - ws_wseril3.c, 146
- WSwupSendSwitchOffCompression
 - ws_wseril3.c, 146
- WSWUPSER.DLL : Functionality for Debug-
ging, 26
- WSWUPSER.DLL : Functionality for Error-
Tracing, 30
- WSWUPSER.DLL : Functions concerning block
reception and sending., 20
- WSWUPSER.DLL : Functions concerning serial
lowest layer functionality., 15
- WSwupSeriLowGiveVersion
 - layer2func, 24
- WSwupSetCurrentReceiveMode
 - layer2func, 24
- WSwupSetIgnition
 - layer1func, 19
- WSwupWSwupGetCurrentReceiveMode
 - layer2func, 25
- wThreadUpdateNr
 - tag_BfbThreadInfo, 56
 - tag_ThreadInfo, 61
- xbi_info.h
 - enBin, 175
 - enCodeAndLang, 176
 - enCodeOnly, 176
 - enCompressed, 175
 - enDiffFile, 176
 - enEesimu, 176
 - enEgoldPlusV12, 175
 - enEgoldPlusV3, 175
 - enEgoldV12, 175
 - enEgoldV2, 175
 - enExe, 176
 - enExtendedNewSplit, 176

- enHighActive, [176](#)
- enIgnitionAccessory, [175](#)
- enIgnitionPort, [175](#)
- enLangOnly, [176](#)
- enLenChk, [175](#)
- enLowActive, [176](#)
- enMobSw, [176](#)
- enNotUsed, [176](#)
- enRaw, [175](#)
- enSplitAndRest, [175](#)
- enSplitOnly, [175](#)
- enStarnBerg, [174](#)
- enStuttgart, [174](#)
- enTiHercules, [175](#)
- enV0toV36, [175](#)
- enV4C7, [175](#)
- enV4C9, [175](#)
- enVoice_Memo, [176](#)
- enWriteTypeEgoldPlusHigh, [176](#)
- enWriteTypeEgoldPlusLow, [176](#)
- enWriteTypeEgoldPlusOldHigh, [176](#)
- enWriteTypeEgoldPlusOldLow, [176](#)
- enWriteTypeEgoldPlusOnly, [176](#)
- enWriteTypeV4, [176](#)
- enXbi, [176](#)
- xbi_info.h, [172](#)
 - MAX_SPLIT_COMMENT_LENGTH, [174](#)
 - t_AllLibInfo, [174](#)
 - t_Area, [174](#)
 - t_AsicTyp, [174](#)
 - t_CompressInfo, [174](#)
 - t_FileFormat, [174](#)
 - t_IgnType, [175](#)
 - t_LocaterRecord, [174](#)
 - t_NewSplitIdentifier, [175](#)
 - t_NewSplitInfo, [174](#)
 - t_ProcType, [175](#)
 - t_SwType, [175](#)
 - t_WriteTyp, [176](#)
 - t_XbiOrExe, [176](#)
- XBI_KENNUNGS_STRING
 - boot_hex.h, [78](#)
- XBI_KENNUNGS_STRING_SGOLD
 - boot_hex.h, [78](#)
- XbiHeaderInfo
 - fiwrap.c, [86](#)
 - swupwork.c, [121](#)
 - wbfbtool.c, [128](#)