

ELECTRONIC CASH REGISTERS

CHD5510, CHD5010T
CHD3550, CHD3010T
CHD2300, CHD2200



FISCAL DRIVER DESCRIPTION



Computer Hardware Design SIA

Revision history:

Version	Date	Description of modifications	Author
1.2		1. Revision history started	NN
1.3	25.07.2003	1. <i>Tender</i> command description extended [6.9] 2. New commands added: <i>TOTAL</i> , <i>SHOW_TOTAL</i> [6.15], <i>DISPLAY_ON</i> [6.16], <i>DISPLAY_OFF</i> [6.17], <i>PAUSE</i> [6.18].	NN,VlaR
1.4	27.08.2003	1. New commands added: <i>CONTINUE</i> [6.19], <i>CONTINUE_MANUALLY</i> [6.20].	NN
1.5	14.10.2003	1. Notes about when drawer opens added 2. Misprint found and corrected (DERTNR instead of DEPTNR)	VlaR
1.6	28.10.2003	1. New command added: <i>EXIT</i> [2.21] 2. Reference to "chd_drv.bat" is added [3]	NN VlaR
1.7	06.02.2004	1. New command added: <i>LAST_RECEIPT_COPY</i>	NN
1.8	25.02.2005	1. Driver is ported to Linux. See description of directory structure [10]	OP
1.9	05.10.2005	1. New result file format added [5] 2. New advanced INI-file parameters added [9]: <i>OldResultFileFormat</i> <i>KeepEcrBlocked</i> <i>KeepEcrBlockedAfterError</i>	NN
1.10	2.11.2005	1. New command added: <i>GET_STATUS</i> [6.22]	NN
1.11	3.11.2005	1. Notes about recovery after a printer error added [6.22]	NN
1.12	6.04.2006	1. New <i>BARCODE</i> command added [6.23]	NN
1.13	13.07.2006	1. Logo bitmap send for CHD3550 and CHD5510 added 2. Advanced INI-file parameters added: <i>CmdTimeout_Default</i> , <i>CmdTimeout_Report</i> , <i>CmdTimeout_Tender</i> 3. Connecting ECR through Ethernet section added [8]	NN
1.14	20.07.2006	1. Fiscal command usage examples added	NN
1.15	1.08.2006	1. Advanced INI-parameters made consistent	NN
1.16	08.08.2006	1. Item <i>SALE / RETURN</i> command syntax extended	NN
1.17.	22.12.2006	1. ECR CHD2300 and CHD2200 support added	NN
1.18	14.11.2007	1. New commands added: <i>TICKET</i> [6.24], <i>COUPON</i> [2.25]	NN

TABLE OF CONTENTS

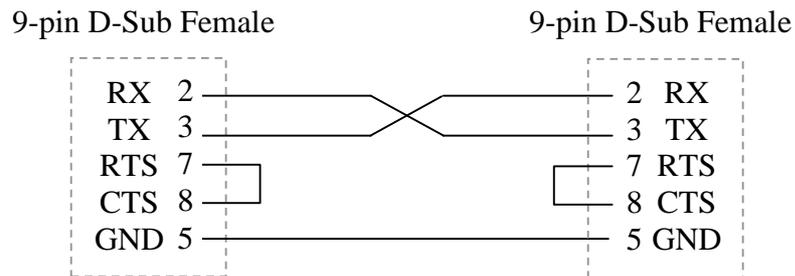
1. Connecting to ECR.....	4
2. Driver usage.....	5
3. Driver return codes (errorlevel).....	6
4. Driver execution modes.....	7
5. Result file format	8
6. Fiscal commands	10
6.1. Print comment	10
6.2. Item sale.....	10
6.3. Item return	11
6.4. Department item sale.....	11
6.5. Department item return	12
6.6. Discount	12
6.7. Markup	13
6.8. Total	13
6.9. Tender	14
6.10. Receive on account	14
6.11. Paid out	15
6.12. Open cash drawer	15
6.13. Last receipt copy.....	15
6.14. Print report.....	16
6.15. Show total.....	16
6.16. Enable showing of information on displays	17
6.17. Disable showing of information on displays.....	17
6.18. Pause	17
6.19. Continuing retail transaction.....	18
6.20. Continuing retail transaction manually	18
6.21. Exit from server mode.....	19
6.22. Read ECR status	19
6.23. Print barcode	20
6.24. Ticket.....	21
6.25. Coupon.....	21
7. Standard INI-file settings	22
8. Connecting to ECR through Ethernet	23
8.1. Configuring ECR.....	23
8.2. Network related INI-file settings	23
9. Advanced INI-file settings.....	24
10. Directory structure	26

For questions and suggestions please write to support@chd.lv.

1. CONNECTING TO ECR

CHD_FDRV can be connected to ECR via serial cable (see picture below) or via standard network cable. Connecting ECR to network is described in details in section 8.

NOTE: When using CHD2300 or CHD2200 ECRs driver INI-file must contain a parameter *ProtocolFormatting* set to 1 (see section 7 for details).



Typical ECR serial port settings.

CHD5510, CHD5010T, CHD3550, CHD3010T

Baudrate	19200 bps
Data length	8 bit
Parity	Odd
Stop bit	1

CHD2300, CHD2200

Baudrate	9600 bps
Data length	8 bit
Parity	None
Stop bit	1

2. DRIVER USAGE

chd_drv.exe filename [option]

Options:

- /S Server mode
- /S2 Old server mode (not recommended for new applications!)

For more information see “Driver execution modes”.

3. DRIVER RETURN CODES (ERRORLEVEL)

Fiscal driver has following return codes:

0	OK
1	Input file syntax error
2	Communication error. No fiscal commands from the input file were send yet
101	Communication or fiscal command execution error. Some fiscal commands were already send. It is possible that some recovery actions are necessary.

The batch file “chd_fdrv.bat” (located in “Examples” directory) shows a simple example of how to process drivers return codes. Section [6.22] describes how to automate ECR recovery after a receipt or journal printer error (for example, printer paper end).

4. DRIVER EXECUTION MODES

Single file execution mode

In this mode fiscal driver executes a single file with fiscal commands and quits. The file name is supplied via the driver command line. Result of the execution is returned via the program return code. If INI-file contains a parameter *ResultFileName* execution result is also written to the specified. If parameter is not present the result file will not be created.

This mode allows constructing of a single retail transaction from subsequent fiscal command files (See CONTINUE command). Retail transaction can be finished either by using *TENDER* command or manually on ECR (See CONTINUE_MANUALLY command).

Server mode

In this mode fiscal driver is running all the time as a hidden process waiting for input files. File name pattern is supplied via the driver command line. For example, if driver is executed using command line "*chd_drv.exe /s request.txt*" it will search for "*request*.txt*" input files (i.e. "*request1.txt*", "*request2.txt*" etc.). When input file is found the driver executes fiscal commands from the file and writes result to file specified by an INI-file *ResultFileName* parameter. If an input file name is "*request2.txt*" and *ResultFileName* is set to "*result.tmp*" then a result file "*result2.tmp*" will be created.

By creating of a result file the driver shows that processing of an input file has been finished and driver is ready for the next one (if return code is less than 100).

Driver quits on an input file containing an "*EXIT*" command.

5. RESULT FILE FORMAT

Fiscal driver supports two different result file formats: INI-file style format (new style) and plain text format (old style). By default a new style result file format is used. To force driver create result file in old style format use *OldResultFileFormat* INI-file parameter ([8]).

New style result file format

Result file consists of a single section named *[CHD_FDRV]*, which contains following name-values pairs:

ResultCode = driver return code

Message = result message

LastReceiptNo = last receipt number. Appears in case if receipt was successfully finished.

CurrentReceiptNo = current receipt number. Appears in case if receipt was left unfinished using a corresponding fiscal command (see sections [6.19] and [6.20]).

Subtotal = Current subtotal. Appears in case if receipt was left unfinished using a corresponding fiscal command (see sections [6.19] and [6.20]).

ErrorCode = Error code. Appears in case of error (see error code tables below).

Result file examples:

```
[CHD_FDRV]
ResultCode=0
Message=OK
LastReceiptNo=456
```

```
[CHD_FDRV]
ResultCode=0
Message=Receipt not finished
CurrentReceiptNo=457
Subtotal=3.34
```

```
[CHD_FDRV]
ResultCode=2
Message=Receipt paper end
ErrorCode=201
```

Error Codes

Input errors

1	Input file syntax error
2	Command line error
3	INI-file error

Driver errors

50	System error (Win32 API error)
99	Internal error

ECR state errors

100	ECR is missing or in OFF state
101	ECR is busy or has incorrect network ID
102	ECR has bad activation code
103	Unsupported ECR type
104	Incorrect key-lock position (must be REG)
105	Previous receipt not finished
106	Current receipt not finished
107	Receipt not started
108	Receipt already finished

Communication errors

150	ECR response lost
151	Send error
152	ECR is sending garbage all the time

ECR errors

201	Receipt paper end
202	Journal paper end
203	Printer error
250	Fiscal command error
299	Other errors

Old style result file format

First result file line contains a driver return code. If there were no errors second result file line will contain "OK", but the third – last receipt number: "ReceiptNo: n". In case of error second line will contain an error message.

6. FISCAL COMMANDS

NOTE: symbol “ can be printed inside of text field using C-syntax: \”.

6.1. Print comment

Command: *COMMENT*

Parameter	Description
<i>TEXT</i>	Text line to be printed. Maximal comment length is 36 characters for CHD5010T, 32 characters for CHD5510 / CHD3550 and 20 characters for CHD3010T / CHD2200 / CHD2300.

Syntax example:

COMMENT: TEXT = "SIA \"ABC\""

Usage example:

COMMENT: TEXT = "CD-RW 700Mb"
SALE: DEPTNR = 1; QNTY = 1; PRICE = 2.45
TENDER

6.2. Item sale

Command: *SALE*

Parameter	Description
<i>PLUNR</i>	PLU number
<i>QNTY</i>	Item quantity. If not specified 1 is used. Must be a number between 0.001 and 999.999.
<i>PRICE</i>	Item price to use. If not specified default price is used. Must be a number between 0.01 and 999999.99.

NOTE: PRICE is supported by ECR firmware v7.2.4 and higher.

Syntax examples:

SALE: PLUNR = 1; QNTY = 13.3
SALE: PLUNR = 1; PRICE = 1.50

Usage example:

SALE: PLUNR = 1; QNTY = 13.3
TENDER

6.3. Item return

Command: *RETURN*

Command has the same syntax as “item sale” command.

Syntax example:

RETURN: PLUNR = 1; QNTY = 13.3

Usage example:

RETURN: PLUNR = 1; QNTY = 13.3
TENDER

6.4. Department item sale

Command: *SALE*

Parameter	Description
<i>DEPTNR</i>	Department number
<i>NAME</i>	Item name to be printed on a receipt. Maximal name length is 18 characters. Parameter is optional.
<i>QNTY</i>	Item quantity. Can contain any number of decimals (see note below)
<i>PRICE</i>	Item price. Can contain any number of decimals (see note below)
<i>AMOUNT</i>	Total amount of items. Number of decimals must be 2 or less

NOTE: $QNTY \times PRICE$ must be a valid amount

Parameters *AMOUNT*, *PRICE* and *QNTY* can be present in following combinations only:

QNTY	PRICE	AMOUNT	Result
+	+	-	Specified <i>QNTY</i> and <i>PRICE</i> values are used.
-	-	+	<i>QNTY = 1</i> and <i>PRICE = AMOUNT</i> are used.
+	+	+	Specified <i>QNTY</i> and <i>PRICE</i> values are used. <i>AMOUNT</i> is used to check the result of <i>QNTY</i> and <i>PRICE</i> multiplication according to current rounding settings on ECR.

Syntax examples:

SALE: DEPTNR = 1; AMOUNT = 1.23

SALE: DEPTNR = 2; PRICE = 20.34; QNTY = 2.00; NAME = "CD-ROM DRIVE"

Usage example:

SALE: DEPTNR = 1; AMOUNT = 1.23
TENDER

6.5. Department item return

Command: *RETURN*

Command has the same syntax as "department sale" command.

Syntax example:

RETURN: DEPTNR = 1; AMOUNT = 1.23

Usage example:

RETURN: DEPTNR = 1; AMOUNT = 1.23
TENDER

6.6. Discount

Command: *DISCOUNT*

Parameter	Description
<i>TYPE</i>	Can be one of the following values: ITEM - for discount on last item sold SALE - for discount on all current transaction items.
<i>AMOUNT</i>	Total amount of discount. Number of decimals must be 2 or less.
<i>PERCENTS</i>	Discount percents. Must be a number between 0.01 and 99.99

NOTE: Only one of AMOUNT / PERCENTS parameters must be present.

Syntax examples:

DISCOUNT: TYPE = ITEM; PERCENTS = 25.00

DISCOUNT: TYPE = ITEM; AMOUNT = 5.23

DISCOUNT: TYPE = SALE; PERCENTS = 25.00
DISCOUNT: TYPE = SALE; AMOUNT = 5.23

Usage example:

SALE: DEPTNR = 1; QNTY = 1; PRICE = 2.45
DISCOUNT: TYPE = ITEM; AMOUNT = 0.10
SALE: DEPTNR = 1; QNTY = 2; PRICE = 1.50
DISCOUNT: TYPE = ITEM; AMOUNT = 0.10
DISCOUNT: TYPE = SALE; PERCENTS = 20
TENDER

6.7. Markup

Command: *MARKUP*

Command has the same syntax as “discount” command.

Syntax examples:

MARKUP: TYPE = ITEM; PERCENTS = 25.00
MARKUP: TYPE = ITEM; AMOUNT = 5.23
MARKUP: TYPE = SALE; PERCENTS = 5.00
MARKUP: TYPE = SALE; AMOUNT = 5.23

Usage example:

SALE: DEPTNR = 1; QNTY = 1; PRICE = 2.45
MARKUP: TYPE = ITEM; AMOUNT = 0.05
SALE: DEPTNR = 1; QNTY = 2; PRICE = 1.50
TENDER

6.8. Total

Command: *TOTAL*

Command has no parameters.

Usage example:

TOTAL

6.9. Tender

Command: *TENDER*

This command registers money tendered for sale/return transaction and opens a cash drawer.

NOTE: Old command name *PAYMENT* can be used too.

Parameter	Description
<i>NAME</i>	Tender name. (Old parameter name <i>MEDIA</i> can be used too.) Can be one of the following values: <i>CASH</i> – for tender in cash (same as <i>TENDER1</i>) <i>CARD</i> – for card tender (same as <i>TENDER3</i>) <i>TENDER1-TENDER8</i> – to directly specify ECR tender type. If parameter is not specified then <i>CASH</i> is used.
<i>AMOUNT</i>	Total amount. Parameter is optional. If not present whole transaction will be paid. Number of decimals must be 2 or less.
<i>FOREIGNNR</i>	Foreign currency number. Parameter is optional.

Syntax example:

TENDER
TENDER: NAME = CARD; AMOUNT = 5.36
TENDER: FOREIGNNR = 1; AMOUNT = 1.00

Usage example:

SALE: DEPTNR = 1; AMOUNT = 10.45
TENDER: AMOUNT = 10.50

6.10. Receive on account

Command: *RCV_ON_ACC*

Parameter	Description
<i>TENDER</i>	Name of tender. The same as <i>TENDER</i> command <i>NAME</i> parameter.
<i>AMOUNT</i>	Received amount. . Number of decimals must be 2 or less.
<i>FOREIGNNR</i>	Foreign currency number. Parameter is optional.

Syntax examples:

RCV_ON_ACC: AMOUNT = 1.00

RCV_ON_ACC: NAME = CARD; AMOUNT = 5.00

Usage example:

RCV_ON_ACC: AMOUNT = 5.00

6.11. Paid out

Command: *PAID_OUT*

Command has the same syntax as “receive on account” command.

Syntax examples:

PAID_OUT: AMOUNT = 1.00

PAID_OUT: NAME = CARD; AMOUNT = 5.00

Usage example:

PAID_OUT: AMOUNT = 1.00

6.12. Open cash drawer

Command: *OPEN_DRAWER*

Opens a cash drawer outside a sale/return transaction.
Command has no parameters.

Usage example:

OPEN_DRAWER

6.13. Last receipt copy

Command: *LAST_RECEIPT_COPY*

Prints a last receipt copy.
Command has no parameters.

Usage example:

LAST_RECEIPT_COPY

6.14. Print report

Command: *REPORT*

Parameter	Description
<i>TYPE</i>	Name of report to print: <i>X</i> – for a financial X-report <i>Z</i> – for a financial Z-report <i>X_CashInDrawer</i> – for cash in drawer X-report.
<i>MODE</i>	Report mode: “X” or “Z”. Must be used with <i>NUMBER</i> parameter.
<i>NUMBER</i>	Report number. For detailed information about report numbers see ECR documentation. Must be used with <i>MODE</i> parameter.

Only one of *TYPE* or *MODE+NUMBER* combination must be present.

Syntax example:

REPORT: TYPE = Z
REPORT: MODE = X; NUMBER = 13

Usage example:

REPORT : TYPE = X

6.15. Show total

Command: *SHOW_TOTAL*

This command suspends an execution of following fiscal commands and shows current transaction total on customer and operator displays for a specified amount of time. Command is functionally equal to: “DISPLAY_ON, TOTAL, PAUSE (SECONDS), DISPLAY_OFF”

Parameter	Description
<i>SECONDS</i>	Number of seconds to show subtotal. Parameter is optional. By default is 0.8 second.

Syntax example:

SHOW_TOTAL: SECONDS = 1.5

Usage example:

SALE: DEPTNR = 1; QNTY = 1; PRICE = 2.45
SHOW_TOTAL
SALE: DEPTNR = 1; QNTY = 1; PRICE = 2.45
SHOW_TOTAL: SECONDS = 2
TENDER

6.16. Enable showing of information on displays

Command: *DISPLAY_ON*

This command allows showing of subsequent commands on both customer and operator displays.

Command has no parameters.

Usage example:

DISPLAY_ON
SALE: DEPTNR = 1; QNTY = 1; PRICE = 2.45
SALE: DEPTNR = 1; QNTY = 1; PRICE = 2.55
TENDER

6.17. Disable showing of information on displays

Command: *DISPLAY_OFF*

Command has no parameters.

Usage example:

DISPLAY_OFF
SALE: DEPTNR = 1; QNTY = 1; PRICE = 2.45
TENDER

6.18. Pause

Command: *PAUSE*

Pauses fiscal command file execution.

Parameter	Description
<i>SECONDS</i>	Number of seconds before next command execute

Syntax example:

```
PAUSE  
PAUSE: SECONDS = 1.5
```

Usage example:

```
DISPLAY_ON  
SALE: DEPTNR = 1; QNTY = 1; PRICE = 2.45  
SALE: DEPTNR = 1; QNTY = 1; PRICE = 2.55  
TOTAL  
PAUSE: SECONDS = 2  
TENDER
```

6.19. Continuing retail transaction

NOTE: Following command can be used in driver *single execution mode* only.

Command: *CONTINUE*

Command is used to construct single retail transaction consisting of multiple subsequent files with fiscal commands. Command can be used at the beginning or at the end of fiscal file, or both at the beginning and at the end of the file.

CONTINUE command at the beginning of file: continue previously started retail transaction.

CONTINUE command at the end of file: current retail transaction will be continued by subsequent file with fiscal commands (this file must have *CONTINUE* as a first command).

Usage example:

```
File1:  
SALE: DEPTNR=1; NAME="ITEM1"; AMOUNT=1.23  
CONTINUE
```

```
File2:  
CONTINUE  
TENDER: NAME=CASH; AMOUNT=2.00
```

6.20. Continuing retail transaction manually

NOTE: Following command can be used only in driver *single execution mode*.

Command: *CONTINUE_MANUALLY*

Command is used to continue current retail transaction manually on ECR. Allowed only at the end of input file. Usually this command is used to finalize transaction manually on ECR keyboard. Command has no parameters.

Usage example:

```
SALE: DEPTNR=1; NAME="ITEM1"; AMOUNT=1.23  
CONTINUE_MANUALLY
```

6.21. Exit from server mode

NOTE: Following command can be used only in driver *server mode*.

Command: *EXIT*

Command has no parameters.

Usage example:

```
EXIT
```

6.22. Read ECR status

Command: *GET_STATUS*

This command reads ECR status and creates a corresponding result file. It can be used to automate ECR recovery after errors. Below you can see a recovery after a journal printer paper end scenario example.

During a fiscal command file execution a journal printer paper end error occurred. Driver created the following result file:

```
[CHD_FDRV]  
ResultCode=101  
Message= Journal paper end  
ErrorCode=203
```

1. Put new paper roll into the journal printer and press [CLEAR]. Wait for ECR to finish printing.
2. Read ECR status (using *GET_STATUS* command) and analyze the created result file.

3a. In case of:

```
[CHD_FDRV]
ResultCode=0
Message=OK
LastReceiptNo=456
```

the receipt was successfully finished. No recovery actions are required.

3b. In case of:

```
[CHD_FDRV]
ResultCode=0
Message=Receipt not finished
CurrentReceiptNo=457
Subtotal=3.34
```

using the subtotal value find out how many fiscal command file commands were actually executed and then either continue execution from the right position (**NOTE:** this can be done only if ECR is still in fiscal mode, i.e. is blocked) or finish the receipt and then void all executed commands (using *RETURN*).

6.23. Print barcode

Command: *BARCODE*

Parameter	Description
<i>DATA</i>	Barcode data.
<i>TYPE</i>	Barcode data type. Parameter is optional. If not specified driver will try to automatically detect it from a barcode data. Currently supported barcode types are: UPC, EAN, CODE39, ITF, CODEBAR, CODE93 and CODE128

Syntax examples:

```
BARCODE: DATA = 012345678901
BARCODE: DATA = 012345678901; TYPE = EAN
```

Usage example:

```
SALE: DEPTNR = 1; NAME = "2x4 WOOD"; PRICE = 1.29; QNTY = 2.12
BARCODE: DATA="2612345667800"; TYPE="EAN"
TENDER
```

6.24. Ticket

NOTE: This command works with Greek ECR versions only.

Command: *TICKET*

Parameter	Description
<i>AMOUNT</i>	Amount of discount. Number of decimals must be 2 or less.

Syntax examples:

TICKET: AMOUNT = 5.00

Usage example:

SALE: DEPTNR = 1; QNTY = 1; PRICE = 2.45
TOTAL
TICKET: AMOUNT = 0.10
TENDER

6.25. Coupon

NOTE: This command works with Greek ECR versions only.

Command: *COUPON*

Parameter	Description
<i>AMOUNT</i>	Amount of discount. Number of decimals must be 2 or less.

Syntax examples:

COUPON: AMOUNT = 5.00

Usage example:

SALE: DEPTNR = 1; QNTY = 1; PRICE = 2.45
COUPON: AMOUNT = 0.10
TENDER

7. STANDARD INI-FILE SETTINGS

CHD_FDRV driver looks for configuration INI-file named “*chd_drv.ini*” in driver directory. INI-file should contain only one section named [*CHD_FDRV*]. This section may contain following settings:

ComNumber

Serial port number.

NetworkID

ECR network identifier (ECR number within ECR network). By default is 1. On ECR this can be set in PRG mode (for details see ECR documentation).

ProtocolFormatting

When using CHD2300 or CHD2200 ECR this parameter must be set to 1. By default is 0 (corresponds to CHD5XX0 and CHD3XX0 type ECRs).

DbgErrorLog = 0...2

- 0 = do not write debug log file (default)
- 1 = write debug log file only on case of error
- 2 = always write debug log file

ResultFileName

Result file name. For details see “Driver execution modes”.

CashierNumber

Cashier number to use for login when cashier system on ECR is activated. By default is 1.

CodePage = DOS | RIM | TILDE

Code page to use. By default is RIM.

8. CONNECTING TO ECR THROUGH ETHERNET

CHD3550 and CHD5510 have a build-in network card and can be connected to network. Accessing ECR through a network can significantly improve data exchange speed and reliability.

8.1. Configuring ECR

Then connecting to network ECR must be assigned an IP address and UDP port on which it will listen for data exchange requests. Also a gateway address and subnet mask must be set. For security reasons it is required to specify an IP address from which ECR will be accessed.

ECR network settings can be programmed in the following way:

Move key-lock to PRG-position
53 [#NS]
ECR IP address [CASH]
IP address from which ECR will be accessed [CASH]
UDP port [CASH]
Gateway [CASH]
Network mask [CASH]
[#NS]

8.2. Network related INI-file settings

To use an ECR connected through network you must specify following INI-file parameters:

IpAddress

ECR IP address.

UdpPort

ECR data exchange UDP port.

9. ADVANCED INI-FILE SETTINGS

All INI-file settings above are optional. By default it is set to most optimal values. But in certain situations advanced users (for example service personnel) may want to change these settings:

StandardLog = 0...1

Use or do not use standard log file. Standard log file contains started, finished and also error messages. By default is 0.

CmdTimeout_Default

Default fiscal command execution timeout in milliseconds.

CmdTimeout_Report

REPORT command execution timeout in milliseconds.

CmdTimeout_Tender

TENTER command execution timeout in milliseconds.

KeepEcrBlocked = 0...1

Keep on do not keep ECR in blocked state after an executing a fiscal command file. By default is 0.

KeepEcrBlockedAfterError = 0... 1

Keep on do not keep ECR in blocked state after an error. By default is 0.

OldResultFileFormat = 0... 1

Use or do not use an old style result file format. By default is 0.

ComSettings

Serial port configuration string in windows mode.exe utility format (for a details see Windows help). For example: "baud=19200 parity=o stop=1 octs=off"

ComRTTC

Serial port data read total timeout constant in milliseconds.

ComWTTC

Serial port data write total timeout constant in milliseconds.

ComByteTimeMult

This value affects maximum allowed time gap between two sequential bytes.

BsyRetryCount

Retry count on BSY response from ECR.

BsySleepTime

Sleep time in milliseconds on BSY response from ECR.

NakRetryCount

Retry count on NAK response from ECR.

NakSleepTime

Sleep time in milliseconds on NAK response from ECR.

GarbageRetryCount

Retry count on bad response from ECR

GarbageSleepTime

Sleep time in milliseconds on bad response from ECR

LostReponseRetryCount

Retry count on no response from ECR.

10. DIRECTORY STRUCTURE

Input file location is detected from the command line. If it does not contain full path than current working directory is used.

Result file name should be given in configuration file. On both Windows and Linux the *result* file is placed in the same directory as *input* file.

Location of configuration and log files depends on the operating system used:

- Windows version of the driver reads *configuration* INI-file from the directory, where the driver is located. The same directory is used to write *standard* and *debug* log files.
- Linux version of the driver reads *configuration* INI-file from directory */etc/chd_fdrv*. *Standard* and *debug* log files are written to */var/log/chd_fdrv* (driver has to be granted write access to these files).