

<b>INTRODUCTION .....</b>	<b>3</b>
<b>QDRIVER FILES .....</b>	<b>4</b>
QDRIVER.EXE .....	4
QDRIVER.CMD .....	4
QDRIVER.LOG .....	4
QDRIVER.END .....	5
QDRIVER.TXT .....	5
<b>QDLL FILES .....</b>	<b>7</b>
QDLL.DLL .....	7
QDLL.LIB .....	7
<b>QDRIVER COMMANDS .....</b>	<b>8</b>
PROCESS .....	8
VISIBLE .....	8
ABORT .....	8
BAUDRATE .....	8
CLEARLOG .....	8
ENDFILE .....	8
COMMAND .....	9
<i>REPORTING</i> .....	9
<i>DOWNLOADING</i> .....	11
<i>PLU MAINTENANCE</i> .....	11
<i>SYSTEM CLOCK</i> .....	11
<i>PASSWORD COMMANDS (LI, LO, LS)</i> .....	12
DIAL .....	12
FILE .....	12
DFILE .....	13
BFILE .....	13
DBFILE .....	13
HANGUP .....	13
MODEMINIT .....	13
NEWFILE .....	14
BNEWFILE .....	14
PORT .....	14
REGISTER .....	15
RESULT .....	15
WAIT .....	15
FAIL .....	15
<b>RECORD FORMATS .....</b>	<b>16</b>
DATA RECORD FORMATS .....	16
<i>KEY FUNCTION RECORD</i> .....	16
<i>RECEIPT HEADER RECORD</i> .....	16
<i>RECEIPT TRAILER RECORD</i> .....	16
<i>SLIP HEADER RECORD</i> .....	16
<i>SLIP TRAILER RECORD</i> .....	16
<i>KEYBOARD TEXT RECORD (Qtouch and Concerto only)</i> .....	16
<i>TABLE INFO RECORD</i> .....	17
<i>ROOM INFO RECORD</i> .....	17
<i>ACCOUNT INFO RECORD</i> .....	17
<i>KEYBOARD RECORD</i> .....	17
<i>TAX RECORD</i> .....	17
<i>SPECIAL ITEMIZER RECORD</i> .....	17
<i>ACTION TABLE RECORD</i> .....	18
<i>TENDER RECORD</i> .....	18
<i>DRAWER RECORD</i> .....	18
<i>PORA RECORD</i> .....	18

<i>DISCOUNT RECORD</i> .....	18
<i>FOREIGN CURRENCY RECORD</i> .....	19
<i>CONDIMENT TABLE RECORD</i> .....	19
<i>MACRO RECORD</i> .....	19
<i>WINDOW LOOK UP RECORD</i> .....	19
<i>GROUP RECORD</i> .....	20
<i>CLERK RECORD</i> .....	20
<i>SALESPERSON RECORD</i> .....	20
<i>MODIFIER RECORD</i> .....	20
<i>DEPARTMENT RECORD</i> .....	20
<i>PLU RECORD</i> .....	21
<b>REPORT RECORD FORMAT</b> .....	22
<i>FIELD 1 - PERIOD</i> .....	22
<i>FIELD 2 - POINTERTYPE</i> .....	22
<i>FIELD 3 - POINTERNUMBER</i> .....	22
<i>FIELD 4 - FILETYPE</i> .....	22
<i>FIELD 5 - NUMBER</i> .....	23
<i>FIELD 6 - TEXT</i> .....	24
<i>FIELD 7 - CUSTOMER COUNT</i> .....	24
<i>FIELD 8 - QUANTITY</i> .....	24
<i>FIELD 9 - AMOUNT</i> .....	24
<i>FIELD 10 - RETURN QUANTITY</i> .....	24
<i>FIELD 11 - DISCOUNT</i> .....	24
<i>FIELD 12 - SURCHARGE</i> .....	24
<i>FIELD 13 - TAX</i> .....	25
<i>FIELD 14 - COST</i> .....	25
<i>FIELD 15 - PROFIT</i> .....	25
<i>FIELD 16 - PERCENTAGE</i> .....	25
<i>FIELD 17 - QTY/CC</i> .....	25
<i>FIELD 18 - AMT/CC</i> .....	25
<b>ELECTRONIC JOURNAL RECORD</b> .....	26
<i>TEXT FORMAT (101)</i> .....	26
<i>BINARY FORMAT (103)</i> .....	26
<i>E-JOURNAL RECORD</i> .....	26
<b>BALANCE ITEM RECORD</b> .....	26
<i>BALANCE RECORD</i> .....	26
<b>PLU MAINTENANCE WHEN USING SCAN CODES</b> .....	27
DELETE PLU FILE .....	27
INSTALL PLU FILE .....	27
<i>INSTALL USING CREATION OF NEW PLU's</i> .....	27
<i>INSTALL USING DIRECT PLU FILE ACCESS</i> .....	27
CREATING OR UPDATING A PLU .....	28
DELETING A PLU .....	28
SCANNING PLU ON KEYBOARD .....	28
<b>MODEM CONNECTION</b> .....	29
MODEM TYPE .....	29
<i>COMPUTER PARAMETERS</i> .....	29
<i>CASH REGISTER PARAMETERS</i> .....	29
MODEM CABLE .....	30
<b>USING ETHERNET CONVERTERS</b> .....	31

## INTRODUCTION

QDRIVER is a MS-Windows program which provides an easy interface to QUORION CR3XXX type of cash registers. It can be used to exchange program and report data in an easy manner. The program reads commands from a command file (which is passed on the command line) and handles the communication with the cash register to ensure proper execution of the commands. Because QDRIVER is a program which can be setup in such a way that it doesn't require user interaction it can be started automatically using the standard MS-Windows scheduler. This can be useful if the same action should be done on a regular basis.

The current release is 1.18 and the release of the cash register program must be of 8 february 2002 or newer. From release 1.07 on there is also a DLL available see QDLL FILES for more details.

## QDRIVER FILES

### QDRIVER.EXE

The driver program. The driver ALWAYS reads the command it should process from a command file which can be specified on the command line. When no command file is specified the driver will automatically look for the command file QDRIVER.CMD in the directory where the driver is located. The command file is a text file containing a command on each line.

*Example:*

QDRIVER TEST.CMD -> processes commands from the file TEST.CMD

*Example:*

QDRIVER -> processes commands from the file QDRIVER.CMD

### QDRIVER.CMD

Default command file. When no command file is specified on the command line QDRIVER tries to open QDRIVER.CMD in the directory where it is located. This can be useful when there is only one command file required.

*Example:*

CLEARLOG=1	-> Clear QDRIVER.LOG file
ABORT=1	-> Show Abort Button
RESULT=1	-> Show result window when finished
BAUDRATE=57600	-> Set Baud rate to 57600
PORT=1	-> Open COM1
NEWFILE=report.csv	-> Create NEW file "report.csv"
COMMAND=RSX0001001	-> Take system report 1 (stored in report.csv)
NEWFILE=group.csv	-> Create new file "group.csv"
COMMAND=D1900	-> Download Group File (stored in group.csv)
NEWFILE=dept.csv	-> Create new file "dept.csv"
COMMAND=D5000	-> Download Department File (stored in dept.csv)
NEWFILE=ejournal.csv	-> Create new file "ejournal.csv"
COMMAND=RUX101001001	-> Take user report 101
NEWFILE=plu.csv	-> Create new file "plu.csv"
COMMAND=D10000	-> Download PLU file (stored in plu.csv)

### QDRIVER.LOG

The LOG file in which the results are stored with date and time of the command separated by a semi-colon. Each line starts with the system date and time (in system format) followed by the command and result as status number and text. When the session is started QDRIVER will also write its version number ("*<1.03>*") in the start line.

In the END line the status is always OK unless the user pressed the ABORT button in this case status 6 "User Break" will be shown to signal that not all commands were processed! See example.

Line Format: *Date,time;CMD;CMD text;Status;Status Text*

Status	Text Number	Message
0	9	OK
1	10	Error Opening Log File
2	11	Error Opening Command File
3	12	Error Opening COM port
4	14	Invalid Command
5	18	Communication Error
6	13	User Break
7	19	Error Opening File
8	20	Error Writing to File
9	21	File Not Open

10	22	Com port not open
11	23	Unknown Report
12	24	Invalid Function
13	25	Still in Transaction
14	26	Invalid Command Parameter
15	27	No Connection
16	28	PLU File is FULL
17	30	LOGIN ERROR

*Example:*

```
26-11-01;11:26:01;START;"<1.00> Session Started";0;"OK"
26-11-01;11:26:01;ABORT;"1";0;"OK"
26-11-01;11:26:01;RESULT;"1";0;"OK"
26-11-01;11:26:01;BAUDRATE;"57600";0;"OK"
26-11-01;11:26:01;PORT;"1";0;"OK"
26-11-01;11:26:01;NEWFILE;"report.csv";0;"OK"
26-11-01;11:26:01;COMMAND;"RSX0001001";0;"OK"
26-11-01;11:26:02;NEWFILE;"group.csv";0;"OK"
26-11-01;11:26:02;COMMAND;"D1900";0;"OK"
26-11-01;11:26:02;NEWFILE;"dept.csv";0;"OK"
26-11-01;11:26:02;COMMAND;"D5000";0;"OK"
26-11-01;11:26:03;NEWFILE;"ejournal.csv";0;"OK"
26-11-01;11:26:03;COMMAND;"RUX101001001";0;"OK"
26-11-01;11:26:08;NEWFILE;"plu.csv";0;"OK"
26-11-01;11:26:08;COMMAND;"D10000";0;"OK"
26-11-01;11:26:23;END;"Session Ended";0;"OK"
```

**QDRIVER.END**

When in the command file the option ENDFILE=1 is set Qdriver will create a file called QDRIVER.END when it has processed all commands in the command file or when it exits because of an error. The result code of the last command is stored as 3 digits in this file. The file could be used to check the status of Qdriver. When the calling program deletes this file before calling Qdriver it can check if the file has been created again to check if Qdriver is finished. When the file doesn't contain "000" there was an error and the file QDRIVER.LOG can be checked for more information.

**QDRIVER.TXT**

File containing all texts and messages used by the driver. When the file is not there it will be automatically created in English language. The format of the file is a number specifying the text and the text string. At the moment the following text are used:

1=QDriver	-> Window title (used on screen)
2=Receiving	-> Shown as status when receiving data (used on screen)
3=Sending	-> Shown as status when sending data (used on screen)
4=Retry	-> Shown when a retry is done (used on screen)
5=SYNC	-> Shown when a synchronize message is received (used on screen)
6=Command :	-> Active command (used on screen)
7=Status :	-> Current status (used on screen)
8=Records :	-> Records received/transmitted (used on screen)
9=OK	-> Command has been processed without errors (used in LOG file)
10=Error Opening Log File	-> File error (used in LOG file)
11=Error Opening Command File	-> File error (used in LOG file)
12=Error Opening COM port	-> COMM error (used in LOG file)
13=User Break	-> Abort Button has been pressed (used on screen and LOG file)
14=Invalid Command	-> Command Error (used in LOG file)
15=Finished !	-> All commands processed (used on screen)
16=Session Started	-> Start of session (used in LOG file)
17=Session Ended	-> End of session (used in LOG file)
18=No Answer	-> Communication Time Out (used on screen and in LOG file)

19=Error Opening File	-> File open Error (used on screen and in LOG file)
20=Error Writing to File	-> File write Error (used on screen and in LOG file)
21=File Not Open	-> No file opened when command started (used on screen and in LOG file)
22=COM port not open	-> No COM port opened when command started (used on screen and in LOG file)
23=Unknown Report	-> Unknown Report (used on screen and in LOG file)
24=Invalid Function	-> Invalid File (used on screen and in LOG file)
25=Still in Transaction	-> Register Still in Transaction (used on screen and in LOG file)
26=Invalid Command Parameter	-> The command parameter is invalid (used on screen and in LOG file)
27=No Connection	-> No connection made after DIAL command (used on screen and in LOG file)
28=PLU File Full	-> PLU file in register is full (used on screen and in LOG file)
29=Register	-> Shown when network register selected (used on screen)
30=LOGIN ERROR	-> Login Required (used on screen and in LOG file)

## QDLL FILES

### **QDLL.DLL**

The DLL file is called QDLL.DLL and contains the “C” function:

**“int qdll(char \*cmdfile, HANDLE chandle);”**

which is declared in QDLL.LIB. The function will execute the command file which is passed as an argument. It is possible to pass the COMPORT handle as a parameter. HANDLE is a declaration to a void pointer (“VOID \*”). When chandle is set to a NON ZERO value the driver will use this handle as com port handle and will also NOT close the port when it exists. Note that even when you pass the HANDLE you MUST set the BAUDRATE in the COMMAND file because the DRIVER also uses the BAUDRATE for calculating timers. In the file TESTDLL.ZIP you can find an example on how to use the DLL. This test program and the DLL are made with Borland CBUILDER release 4. It is also tested with CBUILDER release 6.

Note1:

The DLL will store/read the TEXTS used in the program from the file QDLL.TXT instead of QDRIVER.TXT. From release 1.07 the file is located in the directory of the calling program.

Note2:

When no PROCESS name is specified the LOG and END file are called QDLL.LOG and QDLL.END. From release 1.07 the files are located in the directory of the calling program.

Note3:

It has been reported that when used with Delphi the function must be called with:

**“int \_qdll(char \*cmdfile, HANDLE chandle);”** (note underscore!!)

### **QDLL.LIB**

Library file which the “C” function: **“int qdll(char \*cmdfile, HANDLE chandle);”** is declared.

## QDRIVER COMMANDS

You can mark a line as comment by putting a ';' (semi-colon) at the first position of a line. Note the a command line must be terminated by CR/LF so the last line must be empty!

### PROCESS

With this command it is possible to specify a PROCESS name. The parameter is a string containing the name. This name is used as filename for the LOG and END file and can also contain a path. This can be used when you want to run multiple instances of the driver or dll at the same time.

Note1:

This command is ONLY VALID when it is the FIRST command in the command file in all other cases it will result in INVALID COMMAND.

Note2:

When this command is NOT present the LOG and END file are called QDRIVER.LOG and QDRIVER.END when using QDRIVER. When QDLL is used the files will be called QDLL.LOG and QDLL.END.

*Example:* PROCESS=C:\TEST\process1

This command will store the files PROCESS1.LOG and PROCESS1.END in the directory C:\TEST.

### VISIBLE

With this command it is possible to suppress all output to the screen.

Note1:

This command is ONLY VALID when it is the FIRST or SECOND (after PROCESS) command in the command file in all other cases it will result in INVALID COMMAND.

Note2:

When set this command will also automatically suppress the result window.

### ABORT

When ABORT is set to 1 an ABORT button will be shown in the QDRIVER communication window. Default the button is not shown. When the button is pressed the session is aborted. Note that remaining commands in the command file are not processed!!

*example:* ABORT=1

### BAUDRATE

This command is used to specify the baud rate to which the specified port should be set. When no baud rate is specified in the command file the current baud rate is used. Valid baud rates are 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200. Note that the baud rate used must match the baud rate used in the register. The baud rate 115200 is not supported by the registers yet.

*example:* BAUDRATE=57600

### CLEARLOG

Clear the QDRIVER.LOG file.

*example:* CLEARLOG=1

### ENDFILE

When set QDRIVER will create a file QDRIVER.END which contains a 3 digit Result Code.

*example:* ENDFILE=1

## COMMAND

This command is used to specify a command string which is sent to the register. This command string can be a command to get a User Report, System Report or download program information. Note that during a report or a download the cash-register cannot be used until the report or download is finished.

*example:* COMMAND=RUX001001001

## REPORTING

Reports can be taken as USER or SYSTEM reports and in X, Z or C mode.

- An X report sends the active totals and doesn't clear the total.
- A Z report sends the active totals and clears the total after it has been transmitted.
- A C report only clears the active totals.

### USER REPORTS:

The command for the USER report is : RUM###CCCSSS;S#;E#

- R -> Fixed for Reports
- U -> Fixed for User Reports. When in LOWER CASE the register will automatically take a CONSOLIDATED NETWORK report when a network is active.
- M -> Mode X, Z, or C. When the mode is in UPPERCASE the register will automatically close the transaction when it is still open. When the mode is in LOWERCASE the register will NOT close an open transaction and report an error.
- ### -> User Report number depends on available User Report in the application. Special numbers are 100 for INVOICE REPORT and 101 for Electronic Journal.
- CCC -> Clerk Number. Can be set to 1 only when the User Report is a single clerk report you can specify the clerk you want.
- SSS -> Salesperson Number. Can be set to 1 only when the User Report is a single salesperson report you can specify the salesperson you want,
- ;S# -> Optional Start Number (or Scancode). Note that it only applies to the FIRST LIST in the User Report.
- ;E# -> Optional End Number (or ScanCode). Note that it only applies to the FIRST LIST in the User Report.

*example:*

User Report 1X with close Transaction: COMMAND=RUX001001001  
User Report 1Z with close Transaction: COMMAND=RUZ001001001  
User Report 1C with close Transaction: COMMAND=RUC001001001

*example:*

User Report 2X without close Transaction: COMMAND=RUX002001001  
User Report 2Z without close Transaction: COMMAND=RUZ002001001  
User Report 2C without close Transaction: COMMAND=RUC002001001  
(Note small letter x or z rest must be capital !!)

*example:*

User Report 1X with only record 10 : COMMAND=RUX001001001;10  
User Report 1X with record 10 to 20: COMMAND=RUX001001001;10;20  
(Note that when using SCANCODICES the register will ONLY check the BASE file so a MERGE command must be given if you want to check the complete file)

### SYSTEM REPORTS:

The command for the SYSTEM report is : RSMnnnTP##;S#;E

- R -> Fixed for Reports
- S -> Fixed for System Reports. When in LOWER CASE the register will automatically take a CONSOLIDATED NETWORK report when a network is active.
- M -> Mode X, Z, or C. When the mode is in UPPERCASE the register will automatically close the transaction when it is still open. When the mode is in LOWERCASE the register will NOT close an open transaction and report an error.

nnn -> Pointer Number. If non zero and T is also non zero then the report for the specified pointer requested.

T -> Pointer Type. When non zero the report is for the requested pointer type else it is a standard report.

0 -> Standard report

1 -> Clerk report

2 -> Salesperson report

3 -> Time Zone report

4 -> Day of the week Zone Report

5 -> Day of the month Zone Report

6 -> Date to Date Zone report.

P -> Report Period.

0 -> Report Period 1 (Daily)

1 -> Report Period 2 (Weekly)

2 -> Report Period 3 (Monthly)

3 -> Report Period 4 (Yearly)

## -> Report File Number

1 -> Total Sales File

2 -> Group File

3 -> Department File

4 -> PLU File

5 -> Tax File

6 -> Tender File

7 -> Drawer File

8 -> Paid Out & Received On Account File

9 -> Discount File

10 -> Correction File

11 -> Currency File

12 -> Balance Function File

13 -> Balance Totals File

14 -> Balance Items File (future use)

15 -> Menu File (future use)

16 -> Price Level File

17 -> Transaction Type File (future use)

18 -> Special Itemizer File

19 -> Special Action File (future use)

20 -> PLU Inventory File

21 -> Room Totals File

22 -> Room Items File (future use)

23 -> Account Totals File

24 -> Account Items File (future use)

;S# -> Optional Start Number (or Scancode).

;E# -> Optional End Number (or ScanCode).

*example:*

System Report 1X with close Transaction: COMMAND=RSX0000001  
 System Report 1Z with close Transaction: COMMAND=RSZ0000001  
 System Report 1C with close Transaction: COMMAND=RSC0000001

*example:*

System Report 2X without close Transaction: COMMAND=RSx0000002  
 System Report 2Z without close Transaction: COMMAND=RSz0000002  
 System Report 2C without close Transaction: COMMAND=RSc0000002  
 (Note small letter x or z rest must be capital !!)

*example:*

System Report 1X with only record 10 : COMMAND=RSX0000001;10  
 System Report 1X with record 10 to 20 : COMMAND=RSX0000001;10;20  
 (Note that when using SCANCODES the register will ONLY check the BASE file so a MERGE command must be given if you want to check the complete file)

## DOWNLOADING

Note that during a download the cash-register cannot be used until the download is finished !!  
It is possible to download specific files. The command which needs to be sent to the register is D with the key code for that file. It is possible to specify a range for downloading. At the moment the following files can be sent.

### AVAILABLE DOWNLOAD COMMANDS:

COMMAND=D8	-> Download Receipt Header
COMMAND=D9	-> Download Receipt Trailer
COMMAND=D10	-> Download Slip Header
COMMAND=D11	-> Download Slip Trailer
COMMAND=D12	-> Download Keyboard text records (Qtouch Only!)
COMMAND=D249	-> Download Table Info File
COMMAND=D250	-> Download Room Info File
COMMAND=D251	-> Download Account Info File
COMMAND=D252	-> Download Keyboard File
COMMAND=D260	-> Download Tax File
COMMAND=D280	-> Download Special Itemizer File
COMMAND=D500	-> Download Action Table File
COMMAND=D700	-> Download Tender File
COMMAND=D800	-> Download Drawer File
COMMAND=D900	-> Download PoRa File
COMMAND=D1200	-> Download Foreign Currency File
COMMAND=D1400	-> Download Condiment Table File
COMMAND=D1500	-> Download Macro File
COMMAND=D1600	-> Download Window Look Up File
COMMAND=D1900	-> Download Group File
COMMAND=D2000	-> Download Clerk File
COMMAND=D3000	-> Download Salesperson File
COMMAND=D4000	-> Download Modifier File
COMMAND=D5000	-> Download Department File
COMMAND=D10000	-> Download PLU File

### *example:*

Download PLU 10 only : COMMAND=D10000;10

Download PLU Range 10 to 20: COMMAND=D10000;10;20

(Note that when using SCANCODES the register will ONLY check the BASE file so a MERGE command must be given if you want to check the complete file)

## PLU MAINTENANCE

When using PLU scan codes in the register there are special procedures required for PLU file maintenance. The following commands are available.

COMMAND=CPX	-> Clear Complete PLU File (same as keyboard command 255)
COMMAND=CPD	-> Remove deleted PLU's from file (same as keyboard command 102)
COMMAND=CPM	-> Merge BASE and UPDATE Plu File (same as keyboard command 101)
COMMAND=CPB	-> Remove deleted PLU and MERGE BASE and UPDATE file (same as keyboard command 100)

## SYSTEM CLOCK

It is possible set the clock of the register to a specific time and date or to the system Time and date.

COMMAND=TSYSTEM -> Set Register clock to System Time and Date

COMMAND=THHMMSSDDMMYYYY -> Set Register clock to specified time (HHMMSS, 24hour) and date (DDMMYYYY). Note that the format is fixed!

*example:*

COMMAND=T22150014122001 (set clock to 22:00:00 14-12-2001)

## PASSWORD COMMANDS (LI, LO, LS)

When the register is PASSWORD protected (from release XX040730) you will have to LOGIN FIRST before any other commands are processed by the register. When an older release is installed in the cash register all login commands will result in ERROR# 4 (INVALID COMMAND) When you are NOT LOGGED IN all commands will give ERROR# 17 (Login Error). It is possible to LOGIN (LI), LOGOUT(LO) or CHANGE PASSWORD (LS) using QDRIVER. Note that when you are logged in and there is no communication for more then 20 seconds the register will automatically log out. Note that you can also set the password manually on the register by PROGRAM MODE 98 with LOCK in P-mode. The computer password is PASSWORD#2.

### LIxxxxxx – LOGIN COMMAND

With this command it is possible to LOGIN. The password is maximum 9 DIGITS (Numeric). When the password is wrong ERROR# 17 (LOGIN ERROR) is given. When the register doesn't have a password active ERROR# 12 (INVALID FUNCTION) is given.

*example:*

COMMAND=LI123456789 -> Login with password 123456789

### LO – LOGOUT COMMAND

With this command it is possible to LOGOUT. This command is always accepted even when not logged in.

*example:*

COMMAND=LO -> Log out.

### LSxxxxxx – CHANGE PASSWORD

With this command it is possible to change the PASSWORD in the register. The password is maximum 9 DIGITS (Numeric). Before changing the password and the register already has a password you MUST be logged in with the old password first. Only when there is no password set the command is accepted without logging in first. When the new password is INVALID it will not be changed and ERROR# 12 will be given. You can also remove the password by only sending LS or LS0 to the register.

*example:*

COMMAND=LS1234 -> Set new password to 1234 (Must be logged in FIRST!)

COMMAND=LS0 -> Remove password (Must be logged in FIRST!)

## DIAL

This command is used to dial the specified number. The driver expects CONNECT xxx as answer so the modem must respond with message strings (ATQ0X4). Note that a MODEM must be connected else all commands result in an error when no connection is made.

*example:* DIAL=ATD12345

## FILE

This command is used to open or create a file in which the driver will store received report or download data. The file specified (including path) will be opened. When the file exists the driver will append the received data at the end of the file. Note that when uploading data the filename in which the upload data is stored is specified with the UPLOAD command!

*example:* FILE=TEST.CSV

## **DFILE**

This command is similar to the FILE command except for the following two points:

- The driver will automatically append the DATE and TIME in the fixed format “\_YYMMDD\_HHMMSS” to the filename before it creates the file.
- When the file size is zero the file will automatically be deleted when closed.

*example:* DFILE=TEST.CSV will create file TEST\_041214\_145433.CSV (14 december 2004, 14:54:33)

## **BFILE**

This command is used to open or create a file in which the driver will store received report or download data in BINARY format. The file specified (including path) will be opened. When the file exists the driver will append the received data at the end of the file. This can be used if you don't want the enduser to open and/or change the file using a texteditor. It will subtract '0' (H'30') from every character before it is written to disk.

Note:

The UPLOAD command will check if the file is BINARY by checking the first BYTE.

*example:* BFILE=TEST.BIN

## **DBFILE**

This command is similar to the BFILE command except for the following two points:

- The driver will automatically append the DATE and TIME in the fixed format “\_YYMMDD\_HHMMSS” to the filename before it creates the file.
- When the file size is zero the file will automatically be deleted when closed.

*example:* DBFILE=TEST.BIN will create file TEST\_041214\_145433.BIN (14 december 2004, 14:54:33)

## **HANGUP**

This command is used to disconnect or leave the line open when modem connection is used. From QDRIVER release 1.04 both DTR and the escape sequence “+++” with ATH0 are used to hangup.

Note1: The driver ALWAYS uses DTR for disconnection so the modem must be set to allow this (AT&D2). When you want to leave the line open you MUST disable this option (AT&D0)

Note2: When an external modem is used the cable must support the DTR signal when the modem is programmed to hangup on DTR.

Note3: When no HANGUP command is placed in the command file QDRIVER will ALWAYS hangup when the command file is processed.

HANGUP=0

With the command HANGUP=0 on the last line of a command file QDRIVER will NOT hangup. Note that this only works with a 'REAL' modem because a SOFTMODEM like SMARTLINK will disconnect when the serial port is closed.

HANGUP=1

With this command you can hangup at the end of a session or when you want to dial another number in the same command file.

## **MODEMINIT**

This command is used to send an initialization string to a modem. The driver expects OK as answer.

Note that a MODEM must be connected else all command result in an error when no connection is made.

*example:* MODEMINIT=AT&FE0Q0V1X4&D2

## **NEWFILE**

This command is used to create a file in which the driver will store received report or download data. The file specified by "filename" (including path) will be created. The driver will overwrite the file if it already exists. Note that when uploading data the filename in which the upload data is stored is specified with the UPLOAD command!

*example:* FILE=TEST.CSV

## **BNEWFILE**

This command is used to create a file in which the driver will store received report or download data in BINARY format. The file specified by "filename" (including path) will be created. The driver will overwrite the file if it already exists. This can be used if you don't want the enduser to open and/or change the file using a texteditor. It will subtract '0' (H'30') from every character before it is written to disk.

Note:

The UPLOAD command will check if the file is BINARY by checking the first BYTE.

*example:* FILE=TEST.CSV

## **PORT**

This command is used to specify the com port which should be opened. The maximum COM PORT number which can be specified is 999. The driver will try to open the specified port. In this way it is possible to use "virtual" communication ports like USB to RS232 converters. Note that when you are using QDLL.DLL you can also pass the COMPORT handle from the calling program. When PORT is set to 9999 or another value larger then 999 an ethernet connection is opened. You must also set the IP ADDRESS!!

*example:* PORT=2

Note:

When using Ethernet connection it is possible to specify another PORT than the default port2030 which is used when PORT# is set 9999. When the PORT command is set to a larger value then 999 the driver will use that port number. In order to stay compatible PORT# 9999 is special and will still use 2030. This can be used when port forwarding is needed.

## **IP**

This command is used to set the IP address of the register (network) which needs to be accessed. You must also set PORT to 9999 or a value larger then 999 to open the connection. Note that the IP ADDRESS is the address of the NETWORK so when you want to access another register in the network you only need to set the REGISTER number and the driver will add it to the IP ADDRESS. See Q\_ETHER.DOC for more information on the cash register settings.

*example:*

```
PORT=9999
IP=192.168.2.100
```

Note:

When using an Ethernet connection it is possible to specify web address instead of an IP address. The driver will try to retrieve the IP address using DNS. See our WAN tutorial for more information.

## **UPLOAD**

This command can be used to sent updated program or report information to the register. The command opens the specified file and reads records from this file. Each line in this text file must be terminated by CR/LF and contains one record which is sent to the register. Unknown records are sent but not processed by the register!! For the record format see FILE RECORD FORMATS. Note that during an upload the

cash-register can be used because the records are sent one at a time. When uploading report data the total records are only updated when they are active in the cash-register. The report records which only contain information (field 1 is 0) are sent but are not processed by the register except for the records which contain the general X-counter, general Z-counter and Receipt Number. Is not possible to upload the INVOICE REPORT (100) or detailed Open Balance report.

Note:

The UPLOAD command will check if the file is BINARY by checking the first BYTE. In textmode the first character is always a DIGIT so when the first BYTE has a value between H'00' and H'09' it knows the file is BINARY and will add the '0' (H'30') before sending the record to the register.

*example:* UPLOAD=TEST.CSV

### **REGISTER**

With this command QDRIVER is put in NETWORK mode. All following commands will be sent to the selected register until the next REGISTER command is found. When you set REGISTER to ZERO QDRIVER is put back in direct mode and the commands are only sent to the register to which the computer is directly connected.

*example:* REGISTER=1

### **RESULT**

When RESULT is set to 1 QDRIVER will show a message window with the session result when the last command has been processed. The user MUST press the OK button to close the driver. This means that when automatic processing like with a scheduler is done this option should NOT be set because the driver will not continue until the OK button is pressed.

*example:* RESULT=1

### **WAIT**

With the WAIT command you can suspend the QDRIVER operation for the specified number of milliseconds. The driver will call the MS-Windows SLEEP command with the specified number of milliseconds.

*example:* WAIT=1500 will result in a delay of 1500 milliseconds.

### **FAIL**

When FAIL is set to 1 QDRIVER will stop execution of the driver immediately when a command is not executed so the calling program knows which command wasn't executed. Default FAIL=0 so the driver continues with the next command.

*example:* FAIL=1 will activate, FAIL=0 will deactivate

## RECORD FORMATS

### DATA RECORD FORMATS

The cash register up and downloads the file records as text strings. The different fields in these text string based records are separated by a fixed field separator character which is a ';' (semi-colon). The lay-out and length of the records depend on the type of file being accessed. Note that the fields MUST be sent in this sequence!! When downloading from the register the fields in the file which are not active or when a FLAG field is not used will be empty else it will contain data. When uploading a record to the cash register only the fields which need to be changed are required the ones which should remain unchanged can be empty. When all fields at the end of the record are empty it is NOT required to include all the field separators (semi-colon) for these fields. The first field in the record always contains the File Type. This type is specified by the code which is also used as key code.

For testing there is a special record (KEY FUNCTION RECORD) available for simulating key functions on the cash-register.

### KEY FUNCTION RECORD

FIELD 1 -> Fixed to 5 which is the ID for a key function record  
FIELD 2 -> Key code as programmed in the key table.

### RECEIPT HEADER RECORD

FIELD 1 -> Fixed to 8 which is the ID for a receipt header record  
FIELD 2 -> Receipt Header Line Number (1 – 9)  
FIELD 3 -> Receipt Header Line Font Number (0 – 7)  
FIELD 4 -> Receipt Header Line Text String

### RECEIPT TRAILER RECORD

FIELD 1 -> Fixed to 9 which is the ID for a receipt trailer record  
FIELD 2 -> Receipt Trailer Line Number (1 – 9)  
FIELD 3 -> Receipt Trailer Line Font Number (0 – 7)  
FIELD 4 -> Receipt Trailer Line Text String

### SLIP HEADER RECORD

FIELD 1 -> Fixed to 10 which is the ID for a slip header record  
FIELD 2 -> Slip Header Line Number (1 – 9)  
FIELD 3 -> Slip Header Line Font Number (0 – 7)  
FIELD 4 -> Slip Header Line Text String

### SLIP TRAILER RECORD

FIELD 1 -> Fixed to 11 which is the ID for a slip trailer record  
FIELD 2 -> Slip Trailer Line Number (1 – 9)  
FIELD 3 -> Slip Trailer Line Font Number (0 – 7)  
FIELD 4 -> Slip Trailer Line Text String

### KEYBOARD TEXT RECORD (Qtouch and Concerto only)

FIELD 1 -> Fixed to 12 which is the ID for a keyboard text record  
FIELD 2 -> Key Number (1 – 56)  
FIELD 3 -> Keyboard Level (1 – 8)  
FIELD 4 -> Text size: 0 = standard, 1 = double width, 2 = double height, 3 = double width & height.  
FIELD 5 -> Text Color (0 –255)  
FIELD 6 -> Background Color (0- 255)  
FIELD 7 -> Key text maximum 30 characters.

Note: It is important that after changing keys a record with Key Number ZERO (12;0) is sent to the register so it writes the new key texts to the flash memory else they will be lost after power off.

### TABLE INFO RECORD

FIELD 1 -> Fixed to 249 which is the ID for a table info record  
FIELD 2 -> Table Record Number  
FIELD 3 -> Random code (optional)  
FIELD 4 -> Amount Limit (optional)  
FIELD 5 – 10 -> Max 6 Text lines (optional)

### ROOM INFO RECORD

FIELD 1 -> Fixed to 250 which is the ID for a room info record  
FIELD 2 -> Room Record Number  
FIELD 3 -> Random code (optional)  
FIELD 4 -> Amount Limit (optional)  
FIELD 5 – 10 -> Max 6 Text lines (optional)

### ACCOUNT INFO RECORD

FIELD 1 -> Fixed to 251 which is the ID for an account info record  
FIELD 2 -> Account Record Number  
FIELD 3 -> Random code (optional)  
FIELD 4 -> Amount Limit (optional)  
FIELD 5 – 10 -> Max 6 Text lines (optional)

### KEYBOARD RECORD

FIELD 1 -> Fixed to 252 which is the ID for a registration keyboard record  
FIELD 2 -> Keyboard Number  
FIELD 3 -> Keyboard Row Number (8 columns each)  
FIELD 4-11 -> 8 key codes for each column in this row. When sending a code starting with B is interpreted as a scan code and the register will insert to correct PLU key code. When receiving the scan code is always stored when active.

### TAX RECORD

FIELD 1 -> Fixed to 260 -> Tax File ID  
FIELD 2 -> Tax Record number  
FIELD 3 -> Tax Descriptor max length 25 characters  
FIELD 4 -> Percentage  
FIELD 5 -> Exempt Amount  
FIELD 6 -> Option1 (1 to 8)  
FIELD 7 -> Option2 (1 to 8)  
FIELD 8 -> Print Option1 (1 to 8)  
FIELD 9 -> Print Option2 (1 to 8)

Note that break point tables can't be changed with this record !!!!

### SPECIAL ITEMIZER RECORD

FIELD 1 -> Fixed to 280 -> Special Itemizer File ID  
FIELD 2 -> Special Itemizer Record number  
FIELD 3 -> Special Itemizer Descriptor max length 25 characters  
FIELD 4 -> Discount Percentage  
FIELD 5 -> Exempt Amount  
FIELD 6 -> Option1 (1 to 8)  
FIELD 7 -> Option2 (1 to 8)  
FIELD 8 -> Print Option1 (1 to 8)  
FIELD 9 -> Print Option2 (1 to 8)

**ACTION TABLE RECORD**

FIELD 1	-> Fixed to 500 -> ActionTable File ID
FIELD 2	-> Action Table Record number
FIELD 3	-> Action Table Descriptor max length 25 characters
FIELD 4	-> Option1 (1 to 8)
FIELD 5	-> Option2 (1 to 8)
FIELD 6	-> Print Option1 (1 to 8)
FIELD 7	-> Print Option2 (1 to 8)
FIELD 8	-> Action Table Match (depend on Option 1-1)
FIELD 9	-> Action Table Discount (depends on Option 1-3)
FIELD 10	-> Maximum Discount in Amount
FIELD 11	-> Number of Items in the List.
FIELD 12-31	-> PLU and Optional Match max 10 items (depends on Option 1-1 and 1-8).

**TENDER RECORD**

FIELD 1	-> Fixed to 700 -> Tender File ID
FIELD 2	-> Tender Record number
FIELD 3	-> Tender Descriptor max length 25 characters
FIELD 4	-> Option1 (1 to 8)
FIELD 5	-> Option2 (1 to 8)
FIELD 6	-> Entry Limitation
FIELD 7	-> Print Option1 (1 to 8)
FIELD 8	-> Print Option2 (1 to 8)
FIELD 9	-> Drawer Selection
FIELD 10	-> Overtender Drawer Selection

**DRAWER RECORD**

FIELD 1	-> Fixed to 800 -> Drawer File ID
FIELD 2	-> Drawer Record number
FIELD 3	-> Drawer Descriptor max length 25 characters
FIELD 4	-> Option1 (1 to 8)
FIELD 5	-> Option2 (1 to 8)
FIELD 6	-> Print Option1 (1 to 8)
FIELD 7	-> Print Option2 (1 to 8)
FIELD 8	-> Currency Selection

**PORA RECORD**

FIELD 1	-> Fixed to 900 -> PoRa File ID
FIELD 2	-> PoRa Record number
FIELD 3	-> PoRa Descriptor max length 25 characters
FIELD 4	-> Option1 (1 to 8)
FIELD 5	-> Option2 (1 to 8)
FIELD 6	-> Entry Limitation
FIELD 7	-> Print Option1 (1 to 8)
FIELD 8	-> Print Option2 (1 to 8)
FIELD 9	-> Drawer Selection

**DISCOUNT RECORD**

FIELD 1	-> Fixed to 1000 -> Discount File ID
FIELD 2	-> Discount Record number
FIELD 3	-> Discount Descriptor max length 25 characters
FIELD 4	-> Discount (Rate or Amount)
FIELD 5	-> Exempt Amount
FIELD 6	-> Option1 (1 to 8)
FIELD 7	-> Option2 (1 to 8)
FIELD 8	-> Special Itemizer Selection (Zero or empty when not active)

FIELD 9 -> Entry Limitation  
FIELD 10 -> Print Option1 (1 to 8)  
FIELD 11 -> Print Option2 (1 to 8)

## FOREIGN CURRENCY RECORD

FIELD 1 -> Fixed to 1200 -> Foreign Currency File ID  
FIELD 2 -> Foreign Currency Record number  
FIELD 3 -> Foreign Currency Descriptor max length 25 characters  
FIELD 4 -> Foreign Current Prefix max 3 characters  
FIELD 5 -> Conversion Rate max 10 digits no decimals because is fixed  
FIELD 6 -> Option1 (1 to 8)  
FIELD 7 -> Option2 (1 to 8)  
FIELD 8 -> Print Option1 (1 to 8)  
FIELD 9 -> Print Option2 (1 to 8)  
FIELD 10 -> Drawer Selection

## CONDIMENT TABLE RECORD

FIELD 1 -> Fixed to 1400 -> Condiment Table File ID  
FIELD 2 -> Condiment Table Record number  
FIELD 3 -> Condiment Table Descriptor max length 25 characters  
FIELD 4 -> Option1 (1 to 8)  
FIELD 5 -> Option2 (1 to 8)  
FIELD 6 -> Print Option1 (1 to 8)  
FIELD 7 -> Print Option2 (1 to 8)  
FIELD 8 -> Number of Condiments in the List (maximum depends configuration) .  
FIELD 9-107 -> KeyCodes of Condiments. Only keycodes for Modifiers (4XXX) are accepted.

## MACRO RECORD

FIELD 1 -> Fixed to 1500 -> Macro File ID  
FIELD 2 -> Macro Record number  
FIELD 3 -> Macro Descriptor max length 25 characters  
FIELD 4 -> Option1 (1 to 8)  
FIELD 5 -> Option2 (1 to 8)  
FIELD 6 -> Print Option1 (1 to 8)  
FIELD 7 -> Print Option2 (1 to 8)  
FIELD 8 -> Number of Entries in the List (maximum depends configuration).  
FIELD 9-107 -> Keynumbers or KeyCode of functions when Option 1-2 is set. When sending a keycode a code starting with B is interpreted as a scan code and the register will insert to correct PLU key code. When receiving a macro with keycodes the scan code is always stored when active. Note the the maximum record length is 500 bytes so it is not possible to send 99 scancodes!

## WINDOW LOOK UP RECORD

FIELD 1 -> Fixed to 1600 -> Window Look Up File ID  
FIELD 2 -> Window Look Up Record number  
FIELD 3 -> Window Look Up Descriptor max length 25 characters  
FIELD 4 -> Option1 (1 to 8)  
FIELD 5 -> Option2 (1 to 8)  
FIELD 6 -> Print Option1 (1 to 8)  
FIELD 7 -> Print Option2 (1 to 8)  
FIELD 8 -> Number of Functions in the List (maximum depends configuration) .  
FIELD 9-107 -> KeyCodes of functions. When sending a code starting with B is interpreted as a scan code and the register will insert to correct PLU key code. When receiving the scan code is always stored when active. Note the the maximum record length is 500 bytes so it is not possible to send 99 scancodes!

## GROUP RECORD

- FIELD 1 -> Fixed to 1900 -> Group File ID
- FIELD 2 -> Group Record number
- FIELD 3 -> Group Descriptor max length 25 characters
- FIELD 4 -> Option1 (1 to 8)
- FIELD 5 -> Option2 (1 to 8)

## CLERK RECORD

- FIELD 1 -> Fixed to 2000 -> Clerk File ID
- FIELD 2 -> Clerk Record number
- FIELD 3 -> Clerk Descriptor max length 25 characters
- FIELD 4 -> Option1 (1 to 8)
- FIELD 5 -> Option2 (1 to 8)
- FIELD 6 -> Secret Code 8 digits hexadecimal (Optional)
- FIELD 7 -> Sign On Code 8 digits hexadecimal (future) (Optional)
- FIELD 8 -> Commission Rate (future) (Optional)

## SALESPERSON RECORD

- FIELD 1 -> Fixed to 3000 -> Salesperson File ID
- FIELD 2 -> Salesperson Record number
- FIELD 3 -> Salesperson Descriptor max length 25 characters
- FIELD 4 -> Option1 (1 to 8)
- FIELD 5 -> Option2 (1 to 8)
- FIELD 6 -> Secret Code 8 digits hexadecimal (Optional)
- FIELD 7 -> Commission Rate (future) (Optional)

## MODIFIER RECORD

- FIELD 1 -> Fixed to 4000 -> Modifier File ID
- FIELD 2 -> Modifier Record number
- FIELD 3 -> Modifier Descriptor max length 25 characters
- FIELD 4 -> Option1 (1 to 8)
- FIELD 5 -> Option2 (1 to 8)
- FIELD 6 -> Print Option1 (1 to 8)
- FIELD 7 -> Print Option2 (1 to 8)

## DEPARTMENT RECORD

- FIELD 1 -> Fixed to 5000 -> Department File ID
- FIELD 2 -> Department Record Number
- FIELD 3 -> Department Descriptor max length 25 characters
- FIELD 4 -> Group Selection
- FIELD 5 -> Group 2 Selection (optional)
- FIELD 6 -> Option1 (1 to 8)
- FIELD 7 -> Option2 (1 to 8)
- FIELD 8 -> Entry Limitation
- FIELD 9 -> Print Option1 (1 to 8)
- FIELD 10 -> Print Option2 (1 to 8)
- FIELD 11 -> Tax Selection (1 to 8) (optional)
- FIELD 12 -> Special Itemizer Selection (1 to 8) (optional)
- FIELD 13 -> Kitchen Printer Selection (1 to 8) (optional)
- FIELD 14 -> Action Table Link (optional)
- FIELD 15 -> Condiment Table (1-8) (1 to 8) (optional)
- FIELD 16 -> Condiment Table (9-16) 1 to 8 (optional)
- FIELD 17 -> Condiment Table (17-24) (1 to 8) (optional)
- FIELD 18 -> Condiment Table (25-32) (1 to 8) (optional)
- FIELD 19 -> Tare Weight (future use)(optional)

## PLU RECORD

- FIELD 1 -> Fixed to 10000 -> PLU File ID or 10000+REC# when used for installing PLU file when scan codes are used.
- FIELD 2 -> PLU Record Number or Scan code when active. When the scan code is preceded by '-' the PLU is deleted.
- FIELD 3 -> PLU Descriptor 1 max length 25 characters.
- FIELD 4 -> PLU Descriptor 2 max length 25 characters(optional)
- FIELD 5 -> Department Selection
- FIELD 6 -> Department 2 Selection (optional)
- FIELD 7 -> Group Selection (optional)
- FIELD 8-27 -> Plu Prices maximum 20 depends on application.
- FIELD 28 -> Cost Price (optional)
- FIELD 29 -> PLU Link (optional)
- FIELD 30 -> Tare Weight (future use)(optional)
- FIELD 31 -> Action Table Link (optional)
- FIELD 32 -> Condiment Table (1-8) (1 to 8) (optional)
- FIELD 33 -> Condiment Table (9-16) 1 to 8) (optional)
- FIELD 34 -> Condiment Table (17-24) (1 to 8) (optional)
- FIELD 35 -> Condiment Table (25-32) (1 to 8) (optional)
- FIELD 36 -> Inventory Quantity (optional) When up or downloading the PLU file and the PLU inventory field is active you can program a PLU inventory link in this field. When you have software version 050105 or newer in the cash register it will accept xxxA as entry. This entry is processed by the cash register as a PLU INVENTORY LINK which means that this PLU will link to inventory PLU xxx.
- FIELD 37 -> Inventory Amount (optional)
- FIELD 38 -> Inventory Minimum Level (optional)
- FIELD 39 -> Option1 (1 to 8)
- FIELD 40 -> Option2 (1 to 8)
- FIELD 41 -> Entry Limitation
- FIELD 42 -> Print Option1 (1 to 8)
- FIELD 43 -> Print Option2 (1 to 8)
- FIELD 44 -> Tax Selection (1 to 8) (optional)
- FIELD 45 -> Special Itemizer Selection (1 to 8) (optional)
- FIELD 46 -> Kitchen Printer Selection (1 to 8) (optional)

## REPORT RECORD FORMAT

The cash register sends the report records as text string. The different fields in these text string based records are separated by a fixed field separator character which is ';' (semi-colon). The lay-out and length of the records is variable and depend on the type of record. Every report record always contains 6 fields (Period, Pointer Type, Pointer Number, File Type, Number and text field). Depending on the file type and the report construction it can contain additional fields (Customer Count, Quantity, Amount; Return Quantity, Discount, Surcharge, Tax, Cost, Profit, Percentage, Qty/CC and Amt/CC). When a field is empty it is not used.

### FIELD 1 - PERIOD

VALUE	MEANING
0	A value of 0 indicates that the record contains information which is NOT PERIOD related.
1	A Value of 1 indicates that the record contains information of PERIOD 1.
2	A Value of 2 indicates that the record contains information of PERIOD 2.
3	A Value of 3 indicates that the record contains information of PERIOD 3.
4	A Value of 4 indicates that the record contains information of PERIOD 4.

### FIELD 2 - POINTERTYPE

VALUE	MEANING
0	A value of 0 indicates that the record contains information which is not POINTER related.
1	A Value of 1 indicates that the record contains STANDARD information.
2	A Value of 2 indicates that the record contains CLERK information.
3	A Value of 3 indicates that the record contains SALESPERSON information.
4	A Value of 4 indicates that the record contains TIME ZONE information.
5	A Value of 5 indicates that the record contains DAY OF WEEK information.
6	A Value of 6 indicates that the record contains DAY OF MONTH information.
7	A Value of 7 indicates that the record contains DATE TO DATE information.

### FIELD 3 - POINTERNUMBER

VALUE	MEANING
XXXXX	Number of the pointer to which the information belongs.

### FIELD 4 - FILETYPE

VALUE	MEANING
0	A value of 0 indicates that the record contains not file related information.
1	Total Sales File.
2	Group File.
3	Department File.
4	PLU File.
5	Tax File.
6	Tender File.
7	Drawer File.
8	Paid Out & Received On Account File
9	Discount File.
10	Correction File.
11	Currency File.
12	Balance Function File.
13	Balance Totals File.
14	Balance Items File.
15	Menu File (future use).
16	Price Level File.
17	Transaction Type File (future use).
18	Special Itemizer File.
19	Special Action File (future use).
20	Article Inventory File.
21	Room Totals File.
22	Room Items File.

23	Account Totals File.
24	Account Items File.
25	Clerk Time File.
100	Invoice File
101	Electronic Journal File

**FIELD 5 - NUMBER**

**A – RECORD NUMBER**

When the File Type is NON ZERO this field contains the File Record Number (like PLU number) to which the sales information belongs. When the OPTION in the report is set that totals for this file are accumulated the File Record Number field will be EMPTY to specify that this record contains the accumulated total for this file.

**B – RECORD TYPE**

When the File Type is ZERO this field contains the record type which specifies what type of information this record contains.

VALUE	CONTENTS
0	Report End Record FIELD 6 -> Status Message. FIELD 7 -> Error Number.
1	System Report Name FIELD 6 -> String with System Report Name FIELD 7 -> Report type X or Z FIELD 8 -> Line 2 of the Receipt Header when active.
2	User Report Name FIELD 6 -> String with User Report Name FIELD 7 -> Report type X or Z FIELD 8 -> Line 2 of the Receipt Header when active.
3	Pointer Name FIELD 6 -> Active Pointer Name
4	X-Counter FIELD 6 -> X-Counter String FIELD 7 -> General X-Counter FIELD 8 -> User Report X-Counter
5	Z-Counter FIELD 6 -> Z-Counter String FIELD 7 -> General Z-Counter FIELD 8 -> User Report Z-Counter
6	Date FIELD 6 -> Date String, in format used by cash register. FIELD 7 -> Time String, in format used by cash register. FIELD 8 -> Current Receipt Number
7	Electronic Journal Percentage Used FIELD 6 -> String with percentage used in Electronic Journal
8	Cash Register Number in Consolidated Reports FIELD 6 -> String with Cash Register Number FIELD 7 -> Cash Register Number

9	Cash Register Number when Network Error FIELD 6 -> String with Cash Register Number FIELD 7 -> Cash Register Number
10	NEW ARTICLES separator used to indicate start of UPDATE PLU file FIELD 6 -> String with MESSAGE used to separate the file
11	Positive Part of Non-Resettable Grand Total FIELD 6 -> String with name "(+)NRGT" FIELD 7 -> Amount
12	Negative Part of Non-Resettable Grand Total FIELD 6 -> String with name "(-)NRGT" FIELD 7 -> Amount
13	Non-Resettable Grand Total FIELD 6 -> String with name "NRGT" FIELD 7 -> Amount

**FIELD 6 – TEXT**

Contains the function or item name when the file type is NON ZERO. When the file type is ZERO check FIELD 5 for detailed information.

**FIELD 7 – CUSTOMER COUNT**

Contains the Customer Count when active in report construction else the field is empty. The field is formatted as a quantity with the thousand separator as setup for amounts in the register (point or comma).

**FIELD 8 – QUANTITY**

Contains the Quantity Count when active in report construction else the field is empty. The field is formatted as a quantity with the thousand separator and decimal point as setup for amounts in the register (point + comma or comma + point).

**FIELD 9 – AMOUNT**

Contains the Amount when active in report construction else the field is empty. The field is formatted as an amount with the thousand separator and decimal point as setup for amounts in the register (point + comma or comma + point).

**FIELD 10 – RETURN QUANTITY**

Contains the Return Quantity Count when active in report construction else the field is empty. The field is formatted as a quantity with the thousand separator and decimal point as setup for amounts in the register (point + comma or comma + point).

**FIELD 11 – DISCOUNT**

Contains the Discount Amount when active in report construction else the field is empty. The field is formatted as an amount with the thousand separator and decimal point as setup for amounts in the register (point + comma or comma + point).

**FIELD 12 – SURCHARGE**

Contains the Surcharge Amount when active in report construction else the field is empty. The field is formatted as an amount with the thousand separator and decimal point as setup for amounts in the register (point + comma or comma + point).

### **FIELD 13 – TAX**

Contains the Tax Amount when active in report construction else the field is empty. The field is formatted as an amount with the thousand separator and decimal point as setup for amounts in the register (point + comma or comma + point).

### **FIELD 14 – COST**

Contains the Cost Amount when active in report construction else the field is empty. The field is formatted as an amount with the thousand separator and decimal point as setup for amounts in the register (point + comma or comma + point).

### **FIELD 15 – PROFIT**

Future Use.

### **FIELD 16 – PERCENTAGE**

Future Use.

### **FIELD 17 – QTY/CC**

Future Use.

### **FIELD 18 – AMT/CC**

Future Use.

---

## ***ELECTRONIC JOURNAL RECORD***

### **TEXT FORMAT (101)**

For the electronic journal the same record fields are used as for a report except that some fields have fixed values. When the electronic journal is in Text format there is one record for each text line. Field 1, 2 & 3 are fixed to 1 and field 4 is fixed 101 (file type). Field 5 contains the line number and field 6 contains the text line.

### **BINARY FORMAT (103)**

For the binary electronic journal the a special record format is used. One transaction consists out of all records with the same date, time and receipt number.

### **E-JOURNAL RECORD**

FIELD 1	-> Fixed to 103 (E_Journal ID).
FIELD 2	-> Sequential Record Number.
FIELD 3	-> Location + Register Number.
FIELD 4	-> Date formatted according to application setting.
FIELD 5	-> Time formatted according to application setting.
FIELD 6	-> Receipt Number.
FIELD 7	-> Salesperson Number when active in application.
FIELD 8	-> Balance Invoice Number. When Invoice without number fixed to 65534.
FIELD 9	-> Balance Type (table=249, room=250, account=251).
FIELD 10	-> Balance Number.
FIELD 11	-> Clerk Number.
FIELD 12	-> Function Type.
FIELD 13	-> Function Number.
FIELD 14	-> Function Text.
FIELD 15	-> Quantity.
FIELD 16	-> Amount.

## ***BALANCE ITEM RECORD***

For the retrieving detailed balance information a special record format is used. All records which have the same balance number belong to the balance.

### **BALANCE RECORD**

FIELD 1	-> Fixed to 14 (Table ID), 22 (Room ID) or 24 (Account ID).
FIELD 2	-> Balance Number.
FIELD 3	-> Clerk Number.
FIELD 4	-> Function Type.
FIELD 5	-> Function Number.
FIELD 6	-> Function Text.
FIELD 7	-> Quantity.
FIELD 8	-> Amount.

## PLU MAINTENANCE WHEN USING SCAN CODES

When using scan codes in the register there are some special procedures available for PLU file maintenance. Because the register uses an index file to store the scan codes with an index to the actual PLU record it is not possible to update a PLU record directly but this must be done through the index file using the scan code.

### **DELETE PLU FILE**

When a new PLU file needs to be installed first the command to delete the existing PLU file must be given. When the register receives this command it will clear the COMPLETE PLU file and create an empty INDEX file. Note that all PLU Sales and Inventory information is also deleted!

*Example:*

```
COMMAND=CPX
```

### **INSTALL PLU FILE**

When a PLU file needs to be installed in the register this can be done in 2 ways.

### **INSTALL USING CREATION OF NEW PLU's**

This method uses the standard routines for creation of a new plu in the register. This means that the new plu is stored in the UPDATE file and that this file is re-sorted every time a new plu is created. When the file is full the register will automatically perform the MERGE and REMOVE command which will take some additional time. When the file containing the PLU's is sorted from HIGH to LOW and the file is relatively small (< 500 PLU) the speed of this procedure is acceptable but for complete installation of a NEW PLU file it is advised to use the method with Direct PLU file Access. Note that the standard PLU record with ID-code 10000 is used.

*Example:*

```
COMMAND=CPX
UPLOAD=PLU.CSV
COMMAND=CPM
```

*Contents of PLU.CSV:*

```
10000;1230007;"Article 7";;1;;;0,07
10000;1230006;"Article 6";;1;;;0,06
10000;1230005;"Article 5";;1;;;0,05
10000;1230004;"Article 4";;1;;;0,04
10000;1230003;"Article 3";;1;;;0,03
10000;1230002;"Article 2";;1;;;0,02
10000;1230001;"Article 1";;1;;;0,01
```

### **INSTALL USING DIRECT PLU FILE ACCESS**

This method is especially made for installing a NEW PLU file in the cash register. Because the method which uses the normal creation method can take a lot of time this method was created. Sorting of the PLU file can be done much faster on a computer so when it is possible to provide a SORTED PLU this procedure is the fastest. Note that the ID-code in the PLU record is not standard but 10000 plu the INDEX number of the PLU record.

**IMPORTANT:**

- Only use this method when starting with an EMPTY PLU FILE else the sequence of the file will be corrupted.
- Be sure that the file is SEQUENTIAL and no records are missing!!
- In case direct PLU are used on the keyboard re-program the key-locations with a DIRECT PLU because in the key code the PLU record number is used which will probably be changed.

*Example:*

```
COMMAND=CPX
UPLOAD=PLU.CSV
```

*Contents of PLU.CSV:*

```
10001;1230001;"Article 1";;1;;0,01
10002;1230002;"Article 2";;1;;0,02
10003;1230003;"Article 3";;1;;0,03
10004;1230004;"Article 4";;1;;0,04
10005;1230005;"Article 5";;1;;0,05
10006;1230006;"Article 6";;1;;0,06
10007;1230007;"Article 7";;1;;0,07
```

## CREATING OR UPDATING A PLU

Creation or updating of a PLU is done using the standard PLU record with ID-code 10000. When the PLU doesn't exist it will be automatically created else it will only be updated. When new plu's are created it is advised to close the session with the MERGE command which will merge the BASE and UPDATE file. When the merge command is given on a regular basis the UPDATE file will not grow very large which will improve the speed of creation of new PLU.

*Example:*

```
UPLOAD=PLU.CSV
COMMAND=CPM
```

*Contents of PLU.CSV:*

```
10000;1230001;"Article 1";;1;;0,01
10000;1230002;"Article 2";;1;;0,02
```

## DELETING A PLU

Deleting of a PLU is done using the standard PLU record with ID-code 10000. The only difference is that the scan code is preceded by a '-' sign. When the PLU exists it will be marked for deletion and cannot be used anymore. In order to improve the speed of deletion the space a PLU which is marked for deletion occupies is not released yet! This space is released when the REMOVE command is given. This command will go through the complete PLU file and release all the space occupied by PLU which are marked for deletion.

*Example:*

```
UPLOAD=PLU.CSV
COMMAND=CPD
```

*Contents of PLU.CSV:*

```
10000;-1230001
10000;-1230002
```

## SCANNING PLU ON KEYBOARD

For Direct PLU on the keyboard the register needs the PLU record number. When using scan codes this number is not known so there is a special command in the keyboard record when you also want a PLU as a fixed PLU on the keyboard. You can put the scan code in the KEYBOARD record preceded by the 'B' character. When the register receives the keyboard record and reads a key code which starts with the 'B' character it will automatically replace the scan code which follows with the correct PLU record number in the key table.

*Example, put PLU with code 1230001 on key# 5 and PLU with code 1230002 on key# 6:*

```
252;1;1;1;48;6;29;B1230001;B1230002;701;701
```

## MODEM CONNECTION

When using a modem connection the cash register doesn't know that there is a modem connected. This means that once the connection is made by the computer the communication with the cash register is the same as when direct connection is used. In order to achieve this the modem which is connected to the cash register and the one to the computer must operate at a fixed DTE speed for communicating. Neither the driver nor the cash register will change the speed it uses to communicate with the modem. The actual connection speed depends on the modem and line quality.

### MODEM TYPE

Any Hayes Compatible modem can be used although it might be necessary to re-program some parameters. Nowadays modems have a lot of brand specific parameters which are not standard. When configuring the modem a number of parameters must be checked.

## COMPUTER PARAMETERS

The modem which is connected to the computer actual starts the connection. The modem must be programmed with the following parameters:

- Respond to commands (ATQ0)
- Answer with message strings (ATV1)
- Result Strings (ATX4)
- Don't echo commands (ATE0)
- Drop line when DTR drops (AT&D2)

In most cases the init string : AT&FE0Q0V1&D2 will be ok.

*Example:*

```
CLEARLOG=1  
BAUDRATE=57600  
PORT=1
```

```
MODEMINIT=AT&FE0Q0&D2  
DIAL=ATDT1234
```

NOTE:

When you only want to make a connection using QDRIVER you can do that by using the command HANGUP=0 as the last command in the command file. Don't forget to program the modem NOT to drop the LINE when DTR drops. This can in most cases be done with the command AT&D0. When using a SOFTMODEM like SMARTLINK this cannot be used because the modem always drops the line when QDRIVER closes the serial port. Don't forget to hangup using another command file with HANGUP or HANGUP=1 else the connection stays open.

## CASH REGISTER PARAMETERS

The modem which is connected to the cash register should be programmed so it doesn't send any data to the register unless there is a connection. The modem should be programmed with the following parameters

- Don't respond to commands (ATQ1)
- Don't echo commands (ATE0)
- Pick up line after 'x' rings (ATS0=x)
- Save configuration (AT&W)

In most cases the init string : AT&FE0Q1S0=1&W will be ok.

Configuring of the modem can be done using HYPERTERMINAL in MS-Windows or by using a special command file with QDRIVER. Below you can find an example of a command file which can be used but note that you will get an error in the log file because the modem doesn't respond. After programming the modem it should be powered off/on to check if the configuration is stored correctly.

## QDriver Manual

---

*Example:*

```
CLEARLOG=1  
BAUDRATE=57600  
PORT=1
```

```
MODEMINIT=AT&FE0Q1S0=1&W
```

### **MODEM CABLE**

The cable used for modem connection is a standard modem to computer cable.

---

## USING ETHERNET CONVERTERS

It has been found that when using a device which supports ONLY RTS/CTS handshaking like an ETHERNET CONVERTER the handshaking works better when DTR from the register is connected to CTS of the device and RTS of the register to DSR of the device. This is done because the register actually uses DTR for handshaking.

REGISTER 2 (9 pin male)	DEVICE (9-pin MALE)
2 (RXD) -----	3 (TXD)
3 (TXD) -----	2 (RXD)
5 (GND) -----	5 (SGND)
4 (DTR) -----	8 (CTS)
6 (DSR) -----	4 (DTR)
7 (RTS) -----	6 (DSR)
8 (CTS) -----	8 (RTS)