



Open Interface Handbook

barSTOCK[®] Exchange Software

Stuart W.J. McMillan, Stuart@barstock.com, www.barstock.com
barSTOCK Entertainment Products Ltd

July 2000

1. GENERAL	3
1.1. INTRODUCTION	3
1.2. PRINCIPLE	3
1.3. QUALITY ASSURANCE AND UPDATE CONTROL	4
1.4. BARSTOCK DATA REQUIREMENTS	4
1.5. BARSTOCK SYSTEM REQUIREMENTS	5
2. EPOS RECOMMENDATION	6
2.1. EPOS PRICE CHANGING	6
2.2. EPOS VOLUME SALES READING	6
2.3. EPOS REPORTING	7
3. INTERFACE SPECIFICATION	8
3.1. SIMPLE INTEGRATED ASCII FILE (NETWORKING)	8
3.2. SIMPLE EXTERNAL ASCII FILE (NETWORKING)	9
3.3. INTERNAL BETRIEVE(ODBC) (NETWORKING)	9
3.4. EXTERNAL BETRIEVE (NETWORKING)	10
3.5. ODBC/IDAPI/SQL (NETWORKING) WINDOWS	10
3.6. RS232/485 (EXTERNAL DOS BOX WITH SERIAL COMMUNICATION UTILITIES)	10
3.7. PROPRIETARY EPOS NETWORK	12
3.8. OTHER COMMUNICATION POSSIBILITIES	12
3.9. BARSTOCK DEVELOPMENT KIT	12
4. APPENDIX	14
4.1. BARSTOCK INTERFACE FILE SPECIFICATION	14
4.2. DOS BOX PROCESS SPECIFICATION	15
4.3. WINDOWS PROCESS SPECIFICATION	17
4.4. BARSTOCK.INI FILE	17
4.4.1. EPOS ENTRIES IN THE BARSTOCK.INI FILE	17
4.4.2. WALL DISPLAY ENTRIES IN THE BARSTOCK.INI FILE	22
4.4.3. CRAWL-TEXT ENTRIES IN THE BARSTOCK.INI FILE	22

1. General

1.1. Introduction

Introduction

This Handbook describes the various interface possibilities that can be used to transfer data to and from the barSTOCK system. If this handbook does not cover a preferred interface, please contact us to investigate the requirements for establishing the desired method.

BarSTOCK Entertainment Products Ltd.

German Office:

Technical enquiries
Stuart W.J. McMillan
Postfach 1147
D-64319 Pfungstadt
Germany
Tel+49 (6157) 991162
Fax +49 (6157) 991163
Mobile: (Int + 49) 177 7977485
E-Mail: stuart@barstock.com

U.K. Office: (Uk and Ireland)

Sales enquiries
Ivan McCormick
306 Hither Green Lane
London SE16 615
Tel. +44 (0) 207 207 5041
Mobile:+44 (0) 7958 350 802
E-mail Ivan@barstock.com

U.K. Office (Northern)

Sales enquiries
Peter Godfrey
Graphi-Text
23, Devonshire St
Keighley
West Yorkshire BD21 2BH
E-mail Graphitext@aol.com
Tel or Fax 01535 / 606140

1.2. Principle

Principle

The barSTOCK system is designed to monitor and respond in **real time** to data characterising the movement of transactions from within the whole EPoS system or from designated tills in the EPoS system, as preferred by the local or corporate management. At predetermined intervals the barSTOCK Stock Market software changes the prices of selected products on all, or on selected, tills. Up-to-the-minute information on prices is relayed to customers on bright and prominently positioned wall displays video screens or projection systems. The prices of drinks float up and down automatically, based on customer demand (just like a real Stock Exchange), reflecting actual sales at the tills.

The barSTOCK software does not affect any EPoS data except those relating to the prices of the featured products. All other forms of data capture or data management (i.e. personnel data, stock records, corporate or on site information and analysis systems) are **totally unaffected** by the operations of barSTOCK system.

1.3. Quality Assurance and Update Control

Quality Assurance and Update Control

Throughout the design and development of the product, great attention has been given to the mission critical nature of its operation. It is therefore of the up-most priority that the quality and functional stability of any such data interface to an EPoS system is ensured.

BarSTOCK Entertainment Products Ltd. are continuously improving and refining the barSTOCK package and upgrades are only released after strict testing and documentation. Should any changes effect an interface with a certain make of EPoS product, all efforts will be made to inform the manufacturer of the changes being implemented.

1.4. barSTOCK Data Requirements

barSTOCK Data requirements

For each data transfer direction various methods of interface can be implemented, in combination if required.

EPoS to barSTOCK

barSTOCK stores in its database all PLU numbers and or strings required for products in the Stock Market. In order for barSTOCK to carry out calculations on turnover, the PLU numbers and or strings and quantities sold are required from the EPoS system. These

turnover values can either be the accumulative value since barSTOCK has started or the actual amount in each period. barSTOCK can be pre-programmed in cycles of 1, 3, 5, 6 or more minutes between price changes.

BarSTOCK to EPoS

barSTOCK delivers a list of PLU numbers and or strings and the newly adjusted prices for the product. The format of this ASCII file is described later.

The software also provides the possibility (Configuration menu) when the Exchange starts to set the product sales price to the nominal (on-sale), minimum or maximum. Likewise when the system is brought down, a similar option can be set and the sales price will be reset to the pre-decided value.

Important:

The number of products being read or changed should not exceed 80 items (active plus reserve). However, if there is a limit imposed by the EPoS system, an appropriate notice should be given.

1.5. barSTOCK System Requirements

4 barSTOCK System requirements

The barSTOCK system has been developed in a Borland® Delphi environment and uses Paradox/Dbase file formats for the databases. The software will run on any IBM compatible PC and requires Windows 3.xx or Win95 for operation with a minimum of 8Mb RAM. The main Exchange PC should preferably be a Pentium® or equivalent, ideally rated at 133 MHz or faster and equipped with a CD-ROM reader to support the input of bitmaps and externally produced advertisements. The graphics PC should be of a similar and should be equipped with a graphics card with minimum 8MB RAM with a TV-composite output signal.

Some sites have a back-office PC in place either capturing EPoS data on line or acting as a file server for client terminals. Depending on the nature of the software running on this environment it may be possible to run BarSTOCK parallel on this machine, assuming Windows 95/NT is installed. However this configuration should only be applied after testing. In sites where one or more display boards are in place COM1 and/or COM2 may be required for the communication.

2. EPOS Recommendation

2.1. Epos Price Changing

The procedure for changing prices.

The duration for a price change (comprising changes of up to 80 PLU's, normally between 20 and 60) should not exceed 10 seconds. Ideally this should occur in the background without interruption of till operation. It must be stressed that any delay in till operation interrupts trading, and will have a negative impact on turnover.

In some systems when an attempt is made to either change or access data from tills and the till is busy the EPoS returns a code and a re-attempt is tried. Notice should be given of these codes and in which form they are produced to enable the programmed interface to react correctly.

NOTE:

Some till manufacturers require that their till be closed down before a price change is carried out. A few of the current barSTOCK solutions had to go through a small software alteration in order to satisfy the dynamic price changing requirements.

2.2. Epos Volume Sales reading

Cash Desk Transaction/Turnover recording

Every transaction should read the actual product price from a dynamic memory allocation or active price database file, i.e. when barSTOCK™ changes the price levels, the new price will be reflected in any new transaction journal.

Method of EPoS Turnover Storage

Reference value in memory or data-file

Most of the cash desk's barSTOCK at present interfaces store for each product the recorded turnover in memory. This is an accumulative value increasing every time a sale is registered. barSTOCK has an option to cater for this in the form of a reference value. This initial EPoS log-in on system start-up records the present turnover on each terminal

for the selected active and reserve products, and stores them in the central database. On subsequent reading this reference value is deducted and a new value calculated. This option is set in the barSTOCK Configuration software.

Journal

Secondly if a journal reading is used a pointer will be necessary which records the last recorded position of the scan, and barSTOCK can store this value if required.

2.3. Epos Reporting

Reporting

Turnover reporting should be based on the journal recordings and **not** an article database with no history of price change.

3. Interface Specification

3.1. Simple integrated ASCII File (Networking)

Simple integrated ASCII File (Networking)

This environment requires that both barSTOCK Exchange process and the EPoS software share the PLU, turnover or price file (ASCII) on a network and that barSTOCK does not establish a DOS-box to read the files.

The following interface method has been implemented by a number European EPoS Company's, however software changes had to be made on the EPoS side to accommodate the data-transfer.

This interface is normally for the EPoS systems programmed in Pascal, C, C++ or Visual Basic. Normally an attempt is made by the barSTOCK and/or EPoS process to open the file to read or write. If no other process has opened the file a DOS error 0 is returned. If the file is currently being scanned or filled by another process a DOS error 5 is returned. The barSTOCK process will continue (and the EPoS programme should continue) to attempt an opening until the file is free to be read.

Reading turnover

(Optional information for EPoS)

At the initialisation of barSTOCK and at the end of each cycle, new price calculations are carried out on selected products based on their turnover. To this effect barSTOCK produces a file called *EPOS.PLU* in the "C:\barstock\epos" directory listing the PLU data required. This file is ASCII and has the following format.

```
HEADER
0001
0002
0003
0004
...
...
4 bytes PLU(0000-9999)
ENDOFFILE
```

The EPoS system produces a file called *TURNOVER.DAT*, which contains the PLU's and quantities of all or (using the option file), selected products. barSTOCK will read this file at the end of each Stock Market period and re-create it for further trickling or dumping from the EPoS. This file will have the following format:

```
HEADER
000100001
```

4 Bytes PLU 5 Bytes Quantity (right adjust).

New Prices

At the initialisation of barSTOCK and at the end of each cycle, a new price file called ***BS.PLU*** is produced in the "C:\barstock\epos" directory listing the PLU and the new prices. This file is in ASCII and has the following format

```
HEADER: New Data/No New Data
000102.50
4 Bytes Price 5 Bytes (5,2 right-adjust).
...
...
ENDOFFILE
```

The EPoS software should be programmed to scan this file every 5 seconds. If barSTOCK has just completed a Stock Market period it will create the new price file. The EPoS will pick up the file, find the 'New Data' header and read the new prices in until the string "ENDOFFILE" is found. The EPoS will then re-create the file and insert the header 'No New Data'.

3.2. Simple External ASCII File (Networking)

Simple External ASCII File (Networking)

This system is based on barSTOCK creating a DOS box to read/write new data. Very commonly the EPoS manufacturer provides a **utility** which can be run with a reference input file which extracts the required PLU data to/from the EPoS. If the formats of the returned/to-send data are not identical with the ***TURNOVER.DAT*** and ***BS.PLU*** a barSTOCK utility can be provided/adapted (formats must be supplied by the manufacturer) to suit the format, assuming it is based on ASCII.

3.3. Internal Retrieve(ODBC) (Networking)

Internal Retrieve(ODBC) (Networking)

In later versions of Retrieve an internal ODBC interface is possible. However notice must be given of database definitions and socket version of ODBC driver.

3.4. External Betrieve (Networking)

External Betrieve (Networking)

Read transactions

Assuming the EPoS holds one or more Betrieve-format journal(s) of each transaction centrally, a Windows® or DOS-based process using TSR driver can access this (these) file(s) in read-only mode and scan it (them) for the required "new" transactions. However, the file pointer must be recorded for the next entry into the file, as to avoid scanning old data.

Important

Further information such as till/terminal number could be relevant for scanning if the venue has multiple bars/levels and requires only certain sections of the location to be involved in the Stock Market.

New Prices

Assuming a central Betrieve file format is used and that each transaction reads this file on-line a Windows® or DOS based process could open the file in read/write mode, and change the required PLU/prices combination for the relevant till(s).

3.5. ODBC/IDAPI/SQL (Networking) Windows

ODBC/IDAPI/SQL (Networking) Windows

Depending on the method of storing journal data and prices files an internal/external access is possible. To realise this interface a number of Windows (see Appendix) programs can be triggered to carry out the required collection or writing of data.

3.6. RS232/485 (External DOS box with serial communication utilities)

RS232/485 (External DOS box with serial communication utilities)

An example solution

The communication consists of the combination of two instructions sent in a master/slave (serial 9600, 8,1,Even) environment (i.e. PC: Master, Till: Slave):

- (Command 1) transmitting the new price of a product
- (Command 2) request for the quantities sold of a product

Protocol format

Command 1 PC->TERMINAL

SI, INSTRUCTION, PLU+US, PRICE, ETX
INSTRUCTION: 2 BYTES (Dec 49: Send;Dec 50 Receive)
PLU 4 BYTES (MAX) or PLU
US Unit separator (Dec 31)
PREIS 6 Position without comma (e.g. 000240 = DM 2,40)
ETX Dec 3

Cash-Desk-PC ACK : OK; NAK: not OK re-send

Command 2 PC-> TERMINAL

SI, INSTRUCTION, PLU+US, ETX
INSTRUCTION: 2 BYTES (Dec 49: Send;Dec 50 Receive)
PLU 4 BYTES (MAX) or PLU
US Unit separator (Dec 31)
ETX Dec 3

CASH-DESK->PCSTX, QUANTITY, ETX or NAK

STX 1 BYTE (Dec 2)
Quantity 4 Position

In the BarSTOCK™ software there are two programs which carry out these operations:

RS232G.EXE

and,

RS232S.EXE

RS232G.EXE Get Quantities

The utility uses two files generated by BarSTOCK

EPOS.PLU: Contains the PLU for which the RS232G.exe should request the quantities.

TURNOVER.DAT: The result of the transmission for every PLU will be loaded into this file.

RS232S.EXE Send Process

This utility uses one file generated by barSTOCK: bs.plu. This file contains each PLU with its new price to be fired to the EPOS.

These two utilities handle all RS232/485 communications.

3.7. Proprietary EPoS Network

Proprietary EPoS Network

Many EPoS systems are based on their own LAN twisted-pair network type. However, their manufacturers almost certainly provide a utility to extract/insert data from/to each terminal on the network. This can be used in combination with interface '**Simple External ASCII File (Networking)**'.

3.8. Other communication possibilities

Other communication possibilities

Pseudo kitchen printer

The barSTOCK PC can be set up to operate a background process which simulates a remote receipt printer and each till must be configured to send all takings to a secondary printer device.

The background monitoring process will use the *EPOS.PLU* file with the list of active products to filter out the required products and produce the *TURNOVER.DAT* format file. This file will then be internally shared by barSTOCK using the method described in 4.1.

3.9. barSTOCK Development Kit

barSTOCK Development Kit

A barSTOCK EPoS development kit is also included in the package (on a sub-directory on the demo CD). This software is the full version without display interfaces, and should be used to develop test the interface. Use the setup.exe to install the kit, however you

barSTOCK Exchange POS Interface Handbook

must first remove the demo software from your PC if this has already been installed.

4. Appendix

4.1. BarSTOCK Interface File Specification

BarSTOCK Interface File Specification

Article information EPOS.PLU

(Optional information for EPoS)

At the initialisation of barSTOCK and at the end of each cycle, new price calculations are carried out on selected products based on their turnover. To this effect barSTOCK produces a file called *EPOS.PLU* in the "C:\barstock\epos" directory listing the PLU data required. This file is ASCII and has the following format.

```
HEADER
0001
0002
0003
0004
...
...
4 bytes PLU(0000-9999)
ENDOFFILE
```

and for POS systems using product string identifiers instead of numeric identifiers:

```
HEADER
0001Product1
0002Product2
0003Product3
0004Product4
...
...
4 bytes PLU(0000-9999) 20 bytes (XXXXXXXXXXXXXXXXXXXX)
ENDOFFILE
```

Turnover TURNOVER.PLU

The EPoS system produces a file called *TURNOVER.DAT*, which contains the PLU's and quantities of all or (using the option file), selected products. barSTOCK will read this file at the end of each Stock Market period and re-create it for further trickling or dumping from the EPoS. This file will have the following format:

```
HEADER
000100001
4 Bytes PLU 5 Bytes Quality (right adjust).
```

New Prices BS.PLU

At the initialisation of barSTOCK and at the end of each cycle, a new price file called ***BS.PLU*** is produced in the "C:\barstock\epos" directory listing the PLU and the new prices. This file is in ASCII and has the following format

```
HEADER: New Data/No New Data
000102.50Product1
4 Bytes Price 5 Bytes (5,2 right adjust).20 Bytes(XXXXXXXXXXXXXXXXXX)
...
...
ENDOFFILE
```

4.2. DOS box process specification

DOS box process specification

At Programm startup

A number of currently installed systems require, before the EPOS system is prepared for operation and the stock market has started, but once the Exchange Manager has been activated, that certain adjustment are made to allow fast stock market operation. These are adjustments are then reversed when the Exchange Manager is exited. The DOS box batch file for the Exchange Manager opening is called ***EPOSSUP.BAT***.

At Programm shutdown

The DOS box batch file is called ***EPOSSHUT.BAT (EPOSSHUT.PIF)***. Again this process will allow any parameters specifically required for Stock Market operation to be set to their original settings. This can also be a Windows program.

At barSTOCK operation close (Reset)

Similar ro the Shutdown function, as soon as the barSTOCK is closed this process will be spawned and re-set any parameters or modes on the POS system that may be required for normal operation. However unlike the the Start-Up, Shutdown functions, The Exchange Manager does not have to be re-launched!

The DOS box batch file is called ***EPOSSHUT.BAT (EPOSSHUT.PIF)***. Again this process will allow any parameters specifically required for Stock Market operation to be set to their original settings. This can also be a Windows program.

Moving Pointer Journal start-up

Where a journal File is used to gather turnover the current last position of the data must be recorded before future sales can be analysed. The DOX box batch file *EPOSREF.BAT (EPOSREF.PIF)* is used to start the required processes. If desired the start-up price setting can also be included in this process, otherwise it must be entered in the *EPOSSTPR.BAT* batch.

At Stock Market start-up with turnover reference

Initially at Stock Market start a DOS box is set up with the batch file *EPOSSTAR.BAT (EPOSSTAR.PIF)*. In this batch file two procedures are normally included:

1. To set all active products to their start price. (can also be done in the *EPOSSTPR.BAT*)
2. If the quantities sold are recorded as accumulative values, barSTOCK has to establish a reference quantity from which to deduct the actual unit sales in future analyses.

Also there are adjustments required on the POS system and they can be carried out during operation they should be entered here. They can of course be entered in the *EPOSSTUP.BAT* process.

Stock Market Price setting

If there is no reference process or Journal file used this process trigger can be used to set the product prices to the correct start-up values. The DOS box batch file is called *EPOSSTPR.BAT*.

During System cycles

During operation after every period the batch file *EPOSREAD.BAT (EPOSREAD.PIF)* is triggered in a DOS box. Any utilities used to extract data and produce the *TURNOVER.DAT* file should be launched here.

Likewise to send new prices to the EPoS system all utilities should be entered in the *EPOSNEW.BAT (EPOSNEW.PIF)* which is triggered before the new period starts.

At system shutdown

An option is available to reset any EPoS adjustments made for the Exchange operation. This process is controlled by the batch file *EPOSEND.BAT (EPOSEND.PIF)*. This could also be replaced by the *EPOSSHUT.BAT*, the difference being that this process is triggered when the stock market is being ended and not when the Exchange Manager is being exited.

Important

All these files are setup in the barSTOCK configuration utility. The barSTOCK process PROC_SMP.EXE is programmed to wait until the DOS and /or

Windows child process has completed and therefore there are no API calls required.

4.3. Windows Process Specification

Windows Process Specification

Similar to the spawning of a DOS box the barSTOCK Exchange process can be configured (barSTOCK Configuration utility) to start a Windows based programme to carry out the required data transfer. Instead of entering a .pif file and its location, simply enter the name of the process and its location respectively.

4.4. barstock.ini File

4.4.1. Epos entries in the BarSTOCK.INI file

Epos entries in the BarSTOCK.INI file

There are a number of Boolean flags under [**EposInterface**] that should be set depending on the type of turnover analysis and price programming made:

ChildProcess=0/1	If the read and write interface uses a DOS based program to communicate with the Epos system this flag should be set to 1.
TurnoverFile=c:\barstock\epos\turnover.dat	No matter which form the interface comprises of, barSTOCK uses this file to read in the turnover. The format is described in section (8.1). The programmed interface can produce a file of another name, however this must then be entered in the field (barSTOCK Configuration Utility).
PriceFile=c:\barstock\epos\bs.plu	On every new price calculation barSTOCK produces this file of article numbers and new prices. The ASCII format is detailed in section (8.1). Again like the turnover file, if the programmed interface uses a different name, this should be entered with full directory in this

	field.
PluFile=c:\barstock\epos\epos.plu	The file is produced by barSTOCK for those interface who wish to filter out the active and reserve articles. The ASCII format is detailed in section (8.1). Again like the turnover file, if the programmed interface uses a different name, this should be entered with full directory in this field.
FixedTurnover=0/1	If the process used to read turnover uses the epos.plu file to produce a set file with an 'ENDOFFILE' marker this flag must be set to 1. Otherwise if a file is continuously filed by the EpoS system and no end of file is set this should be set to 0.
Reference=0/1	If the data collected needs a moving turnover reference to be recorded and the initial BarSTOCK Start-up and deducted form the accumulated value on every price change.
Ref_program	Process name which extracts the current values of each active or reserve products. If the program is Windows then this should be the .exe name. If a Dos program is being used this should be the .pif program. If any EpoS reset procedures are to be triggered at the beginning of the barSTOCK operation they should be set in this process of batch file. <u>IMPORTANT:</u> Furthermore if at the barSTOCK startup the product prices are to be set at values other than the normal levels(adjusted in the barSTOCK Configuration utility), the price change process/program should be started here (using the bs.plu file). <u>NOTE:</u> If this option is chosen it is automatically assumed that for each product a reference value is always used by each read.
Ref_directory	Directory in which either the Windows

Start_price_directory=c:\windows	program or the .pif file resides.
Start_price=0/1	If an independent price setting process is used at the start-up of the stock market then this flag should be 1.
Start_price_program	Name of the DOS pif file or the windows process program name.
Start_price_directory	Location of the DOS pif file or the windows program.
Moving_pointer=0/1	Where the turnover data is extracted from a Journal file/Data-base a moving pointer must be maintained. If this flag is set to 1, barSTOCK will spawn a process at system initialisation to read and store the current position of this pointer.
Pointer_program	Process name which extracts the current value of pointer in the EpoS Journal file. If the program is Windows then this should be the .exe name. If a Dos program is being used this should be the .pif program. If any EpoS reset procedures are to be triggered at the beginning of the barSTOCK operation they should be set in this process of batch file. IMPORTANT: Furthermore if at the barSTOCK start-up the product prices are to be set at values other than the normal levels(adjusted in the barSTOCK Configuration utility), the price change process/program should be started here (using the bs.plu file).
Pointer_directory	Directory in which either the Windows program or the .pif file resides.
Epos_reset=0/1	At the end of barSTOCK operation some EpoS system have to be reset for 'Normal' operation. This process is separate from the price re-setting, and is primarily for the EpoS system to re-adjust it's mode for non-Exchange operation.

Reset_program	Process name which resets any parameters on the EpoS , needing restored to the non-barSTOCK levels. If the program is Windows then this should be the .exe name. If a Dos program is being used this should be the .pif program. IMPORTANT: Similar to the barSTOCK start-up the product prices are to be set at values other than the normal levels (adjusted in the barSTOCK Configuration utility), the price change process/program should be started here (using the bs.plu file).
Reset_directory	Directory in which either the Windows program or the .pif file resides.
Read_csv=0/1	If 1 the TURNOVER.DAT (TurnoverFile above) will be expected in CSV format with default comma separators and with a header and footer.
Read_seperator_csv=0/1	If 1 the separator expected will be a ';' semi-cologne, if 0 it'll be default a comma.
Read_headfoot_csv=0	If 0 a header and footer will be expected, if 1 without!
New_csv=0/1	If 1 the BS.PLU (PriceFile above) will be expected in CSV format with default comma separators and with aheader and footer.
New_seperator_csv=0/1	If 1 the separator expected will be a ';' semi-cologne, if 0 it'll be default a comma.
New_headfoot_csv=0	If 0 a header and footer will be expected, if 1 without!
Ref_csv=0/1	If 1 the EPOS.PLU (PluFile above) will be expected in CSV format with default comma separators and with aheader and footer.
Ref_seperator_csv=0/1	If 1 the separator expected will be a ';' semi-cologne, if 0 it'll be default a comma.
Ref_headfoot_csv=0/1	If 0 a header and footer will be expected,

if 1 without!

Processes_minimized=0/1

When set at 1, all child processes required for the POS interface will be spawned in minimized form.

In some venues where opening times are restricted and therefore the POS system is started up every evening, certain adjustments may be required on the POS system before barSTOCK can begin operation. Likewise the POS system may require certain resetting functions to be carried when it is run down. The next two setting allow two separate processes to be triggered at these events.

Startup=0/1

Trigger start-up process once the barSTOCK Exchange Manager is started but the exchange not yet opened.

Startup_Program

Name of program or windows .pif file.

Startup_directory

Directory in which either the Windows program or the .pif file resides.

Shutdown=0/1

Trigger shutdown process once the barSTOCK Exchange Manager is exited to Windows.

Shutdown_program

Name of program or windows .pif file.

Shutdown_directory

Directory in which either the Windows program or the .pif file resides.

Data Formats

Plu_length=4

This is the total length of the new PLU produced in the BS.PLU file.

Plu_decimal=0

This is the after comma number produced. Default 0.

Volume_length=5

The total length of the turnover figure expected. Default 5.

Volume_decimal=0

Number of after-comma number expected. Default 0

Price_length=5

Total length of the price produced for POS update. Default 5.

Price_decimal=2

After-comma length (decimal) for POS update. Default 2.

Price_display_len=5	Total length of the price produced for display medium update. Default 5.
Price_display_dec=2	After-comma length (decimal) for display medium update. Default 2.

4.4.2. Wall Display entries in the BarSTOCK.INI file

Wall Display entries in the BarSTOCK.INI file

If an electronic wall display or other such large display medium is used, barSTOCK provides the possibility by every Stock Market event to alter the product price information to be supplied to these external hardwares. An ASCII file is created containing the event name and product information that should be projected on such displays. In this section of the barstock.ini the process/program name with its residing directory can be entered as well as the file name that should be used.

Present=1/0	Indicates if a wall display is available for driving.
Windows_child_process=0	Windows process active
Win_process	Name of the windows process used to drive the board.
Win_dir	Location of the Windows process.
Dos_child_process=0	DOS process active
Win_process	Name of the DOS process (*.pif) used to drive the board.
Win_dir	Location of the *.pif process
Wall_display_file	Name of the ASCII file containing product information.

4.4.3. Crawl-text entries in the BarSTOCK.INI file

Crawl-text entries in the BarSTOCK.INI file

If TV's or video projector systems are in place, barSTOCK provides this possibility. At every Stock Market event the product price information is supplied to a crawl text generator. An ASCII file is created containing the event name and product information that should be projected on such displays. In this section of the barstock.ini the process/program name with its residing directory can be entered as well as the file name

that should be used.

Present=1/0	Indicates if a crawl text required.
Windows_child_process=0	Windows process active
Win_process	Name of the windows process used to drive the crawl text generator.
Win_dir	Location of the Windows process.
Dos_child_process=0	DOS process active
Win_process	Name of the DOS process (*.pif) used to drive the crawl text generator.
Win_dir	Location of the *.pif process
Crawl_file	Name of the ASCII file containing product

Note

The initialisation for the system is BARSTOCK.INI, which is placed in the c:\WINDOWS directory during installation.