The barSTOCK Exchange Suite

EPoS Interface Specification

Release Version 11.0, 2011

Table of Contents

1 Important Legal Information

1.1 Copyright, Disclamer, Trademarks

Copyright

Disclaimers

barSTOCK Entertainment Products. makes no representations or warranties with respect to the design and documentation herein described and especially disclaims any implied warranties of merchantability or fitness for any particular purpose. barSTOCK Entertainment Products. reserves the right to revise this design and associated documentation and to make changes from time to time in the content without obligation barSTOCK Entertainment Products. to notify any person of such revisions or changes.

Trademarks

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this document, and barSTOCK Entertainment Products. was aware of a trademark claim, the designations have been printed in initial caps or all caps. References may be made in this manual to barSTOCK and APP which are trademarks barSTOCK Entertainment Products. References may be made in this manual to Windows, Windows XP and Windows Vista or Windows 7 which are trademarks of Microsoft Corporation.

2 General

2.1 Principle

The barSTOCK system is designed to monitor and respond in **real time** to data from EPoS (Point of Sale) system or from designated tills in the EPoS (Point of Sale) system. At predetermined intervals the barSTOCK Stock Market software will read the sales data from within a given time period (3 up-to 10minutes), calculate the new prices based on various parameters and change 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 (Point of Sale) 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.

2.2 Inroduction

This Handbook describes the various interface possibilities that can be use 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.

Technical enquiries
Stuart W.J. McMillan
Grenzweg 3d
D-64342 Seeheim-Jugenheim
Germany
Mobile: (Int + 49) 177 7977485
E-Mail: stuart@barstock.com

2.3 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 (Point of Sale) system is ensured.

BarSTOCK Entertainment Products 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 (Point of Sale) product, all efforts will be made to inform the manufacturer of the changes being implemented.

2.4 barSTOCK Data Requirements

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.

2.4.1 Data Required from EPoS (Point of Sale)

barSTOCK stores in its database all PLU (item numbers) numbers products in the Drinks Market. In order for barSTOCK to carry out calculations on the sales figures, the PLU (item) numbers 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 requires in cycles of 1, 3, 5, 6 up-to every 12 minutes.

2.4.2 Data Sent to EPoS (Point of Sale)

barSTOCK delivers a list of PLU numbersand thier newly adjusted prices for each 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.

2.5 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 minimum Windows XP for operation with a minimum of 2GB RAM. The main Exchange PC should preferably be a Pentium® or equivalent and equipped with a CD-ROM/USB 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 1GB RAM with a TV-composite output signal.

Some sites have a back-office PC in place for 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 XP/Vista or 7 is installed. However this configuration should only be applied after testing.

3 EPOS Recommendation

3.1 EpoS Price Changing

The duration for a price change (comprising changes of up to 80 PLU's, normally between 20 and 60) should not exceed 30 seconds. Ideally this should occur in the background without interruption of till (Point of Sale) 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.

3.2 EpoS Volume Sales reading

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

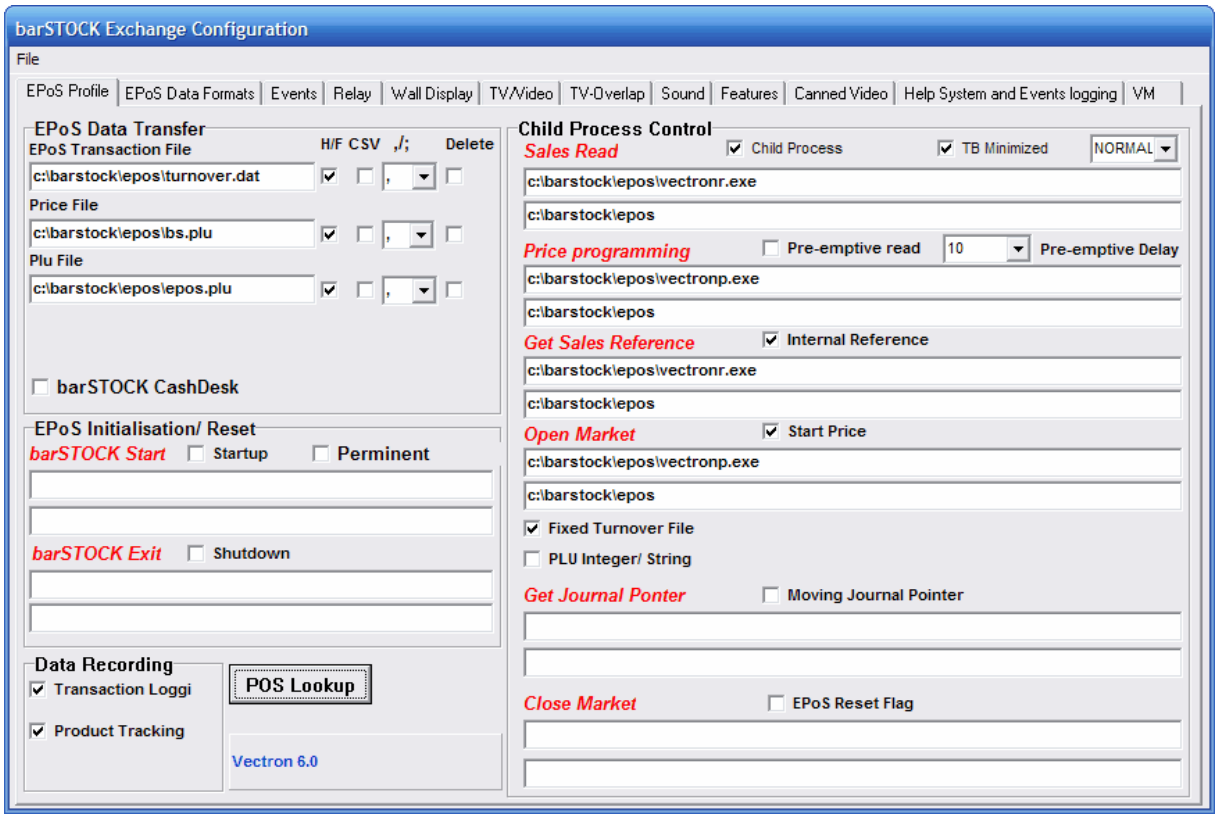
Secondly if a journal reading is used a pointer will be necessary which records the last recorded position of the scan, and barSTOCK can store this value if required.

3.3 Epos Reporting

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

4 Interface Process Triggers

The backbone of the barSTOCK EPoS Interface is its ability to trigger of external processes to interact with eth POS system. The details of these are descibed in this section.

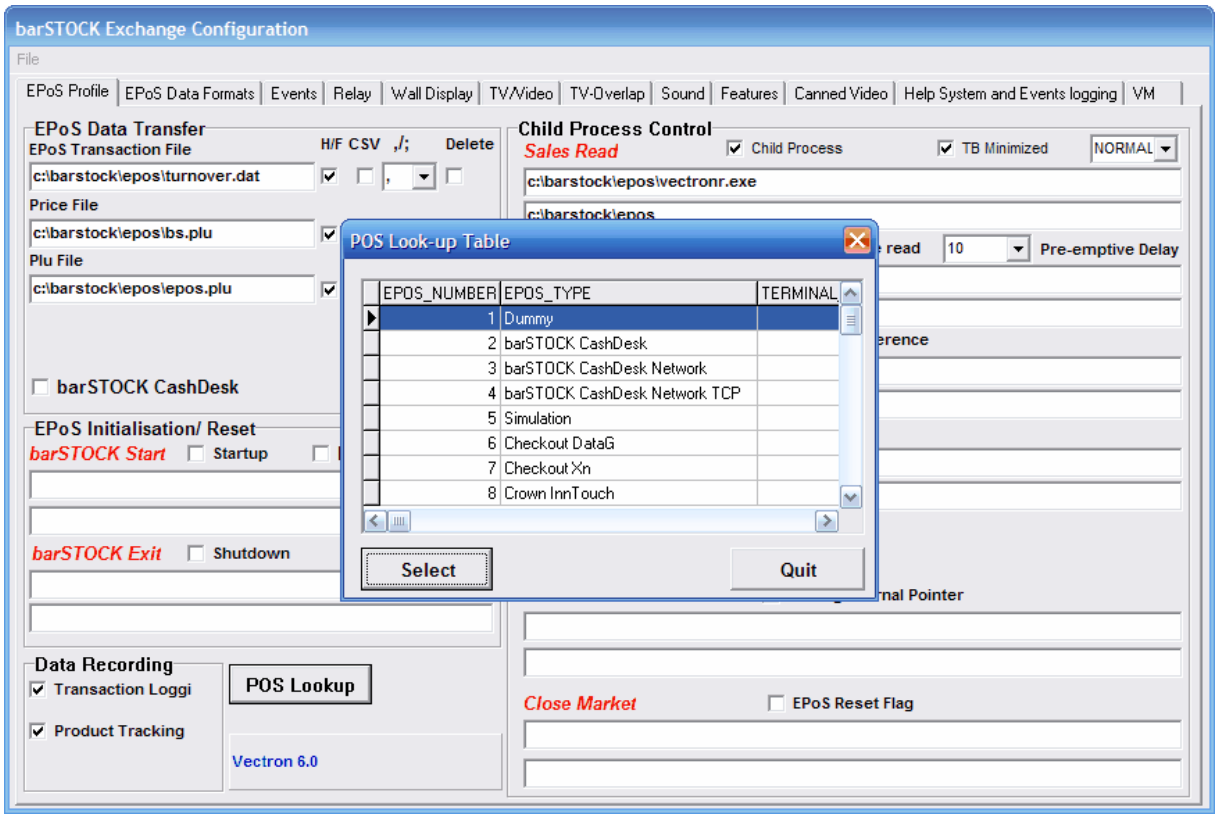


Process with (*) afterwards are manditory.

All these files are setup in the barSTOCK configuration utility.
The barSTOCK process PROC_SMP.EXE is programmed to wait until the Windows child process has completed and therefore there are no API calls required.

4.1 EPoS Profiles

For most available interfaces we have created a POS profile which stores all the nessessery settings for the interface including the process names. Once the interface is finalized, a new Profile will be created under the POS name. This makes installtion much more simple:



The barSTOCK Installer package also includes some interfaces as a secondary installer integrated in the main package. If it is possible to package these utilise this will be done.

4.2 barSTOCK Startup

This is when the Exchange Manager is started, ***not*** when the Drinks Markets is opened.

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.

4.3 barSTOCK Shutdown

This process holder is for when the barSTOCK Exchange Manager is exited and ***not*** when the Drinks Market is closed.

4.4 Market Open

Process will be launched when the Drinks Market is openned.

4.5 Drinks Market Close

Similar ro the Shutdown function, as soon as the barSTOCK Drinks Market is closed this process will be spawned and re-set any parameters or modes on the POS system that may be requried for normal operation. However unlike the the Start-Up, Shutdown functions, The

Exchange Manager does not have to be stopped and re-launched!

4.6 Get Journal Pointer

Where the Read Sales Interface is based on monitoring the poistion of a Journal, this process holder is used to read the current poition of this Journal.

4.7 Get Sales Reference

Where the item sales number is accumulative, this process holder is used to gather the initial value and use as a reference for future reads.

4.8 Sales Read (*)

At the end of each barSTOCK Drinks Market cylce the process is launched ot gather the sales information. This process is manditory.

4.9 Price Programming (*)

After the item sales information has been read in. the new prices are calculated. Afterwhich this process is triggered to communicate with eth POS and programm the new prices. This process is mandatory.

5 Interface Medthods

5.1 barSTOCK/POS ASCII FIle Sharing

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 barSTOC does not spawn an external process 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 the their turnover. To this effect barSTOCK produces a file called ***EPOS.PLU***[1] 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***[1], 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***[2] 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'.

5.2 External Spawned Process (ASCII File)

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.

5.3 ODBC/IDAPI/SQL

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.

5.4 RS232/RS485/USB

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->PC STX, 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.

5.5 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 **'External Spawned Process (ASCII File)'**.

5.6 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.

5.7 barSTOCK Development Kit

barSTOCK Trial Software

To help the development of an interface to a specific POS system a trial version of the barSTOCK Exchange is available for down load on our WEB site:

www.barstock.de/barSTOCKExchange.zip

This is the full functional bar STOCK package is only limited by time and the number if starts of the system.

Please contact barSTOCK (stuart@barstock.com) to receive the authorization code to install this software.

6 Appendix

6.1 BarSTOCK Interface File Specification

File format is csv and can be either , or ; seperated.

6.1.1 PLU Reference File (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 the their turnover. To this effect barSTOCK produces a file called ***EPOS.PLU***[1] in the "C:\barstock\epos" directory listing the PLU data required. This file is ASCII and has the following format.

1
2
3
4

6.1.2 PLU Sales File (turnover.plu)

The EPoS system produces a file called ***TURNOVER.DAT***[1], 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:

1, 1
2, 5
3, 10
4, 25

6.1.3 PLU Price Information (bs.plu)

At the initialisation of barSTOCK and at the end of each cycle, a new price file called ***BS.PLU***[2] 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

1, 2.50
2, 1.50
3, 4.90
4, 20.22

6.2 barstock.ini FIle

6.2.1 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. **IMPORTANT:** 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). **NOTE:** If this option is chosen it is automatically assumed that for **each** product a reference value is always used by each read.
Ref_directory Start_price_directory=c:\windows	Directory in which either the 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_headfoor_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_headfoor_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_headfoor_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.

6.2.2 Wall Display entries in the BarSTOCK.INI file

If an electronic wall display or other such large display medium is used, barSTOCK provides the possibilty 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.

6.2.3 Crawl-text entries in the BarSTOCK.INI file

If TV's or video projector systems are in place, barSTOCK provides this possibilty. 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.