



jBASE Environment Variables

Contents

JBASE ENVIRONMENT VARIABLES	1
Documentation Conventions	1
ENVIRONMENT VARIABLES.....	3
SETTING/GETTING ENVIRONMENT VARIABLES	3
jBASE PROGRAMS.....	4
jBASE Initialization	4
Initial Environment Variables	5
ADDITIONAL JBASE ENVIRONMENT VARIABLES	6
EXECUTION - Environment Variables.....	6
DEVELOPMENT - Environment Variables.....	7
MISCELLANEOUS - Environment Variables	7
TERMINAL - Environment Variables.....	7
EMBEDDED SQL - Environment Variables.....	8
CREATE-FILE - Environment Variables	8
jRFS – Environment Variables.....	8
PUTENV - Command	9
GETENV - Command	10
PATH.....	11
LD_LIBRARY_PATH.....	12
TERM.....	13
TERMINFO.....	14
JBCPORTNO	15
JBCLOGNAME	16
JBCGLOBALDIR	17
JBCRELEASEDIR.....	18
JBCDATADIR	19
JBCDEFDICTS	20
JBCEMULATE	21
JDIAG	22
JEDIFILEPATH.....	23
JEDIFILENAME_MD	24
JEDIFILENAME_SYSTEM.....	25
JEDIENABLEQ2Q	26
JEDI_DISTRIB_DEFOPEN	27
JEDI_SECURE_LEVEL.....	28
JBC_TCLRESTART	30
JBC_ENDRESTART	31
JBCOBJECTLIST	32
JBC_BLOCK_SYSTEM14.....	33
JBASEUNIQUE.....	34
JBASE_ERRMSG_NON_NUMERIC.....	35
JBASE_ERRMSG_TRACE.....	36

JBASE_ERRMSG_ZERO_USED	37
JBASE_WIN_TERM_SVR.....	38
JBASE_SVR_SESSION.....	39
JBC_INVERT_ALPHA_CHARS	40
INTERNATIONALISATION	41
JBASE_I18N.....	41
JBASE_CODEPAGE	42
JBASE_LOCALE	43
JBASE_TIMEZONE	44
DEVELOPMENT.....	45
JBCDEV_BIN.....	45
JBCDEV_LIB	46
JBCTTYNAME	47
JBCERRFILE	48
JBCSPOOLERDIR	49
JBC_DESPOOLSLEEP.....	50
JBC_CRREQ	51
JBCLISTFILE.....	53
JBCSCREEN_WIDTH.....	54
JBCPRINTER_DEPTH	55
JBCPRINTER_WIDTH.....	56
JBASE REMOTE FILE SERVER (JRFS).....	57
JBCNETACCESS	57
JBCNETDIR	58
JRFS_REMOTE_JQL.....	59
JRFS_LOCALPATH_JQL	60
JRFS_SERVERNAME.....	61

Documentation Conventions

This manual uses the following conventions:

Convention	Usage
BOLD	In syntax, bold indicates commands, function names, and options. In text, bold indicates keys to press, function names, menu selections, and MS-DOS commands.
UPPERCASE	In syntax, uppercase indicates JBASE commands, keywords, and options; BASIC statements and functions; and SQL statements and keywords. In text, uppercase also indicates JBASE identifiers such as filenames, account names, schema names, and Windows NT filenames and pathnames.
UPPERCASE <i>Italic</i>	In syntax, italic indicates information that you supply. In text, italic also indicates UNIX commands and options, filenames, and pathnames.
<code>Courier</code>	<code>Courier</code> indicates examples of source code and system output.
Courier Bold	Courier Bold In examples, indicates characters that the user types or keys (for example, <Return>).
[]	Brackets enclose optional items. Do not type the brackets unless indicated.
{ }	Braces enclose nonoptional items from which you must select at least one. Do not type the braces.
ItemA .itemB	A vertical bar separating items indicates that you can choose only one item. Do not type the vertical bar.
...	Three periods indicate that more of the same type of item can optionally follow.
⇒	A right arrow between menu options indicates you should choose each option in sequence. For example, “Choose File ⇒ .Exit ” means you should choose File from the menu bar, and then choose Exit from the File pull-down menu.

Syntax definitions and examples are indented for ease in reading.

All punctuation marks included in the syntax—for example, commas, parentheses, or quotation marks—are required unless otherwise indicated.

Syntax lines that do not fit on one line in this manual are continued on subsequent lines. The continuation lines are indented. When entering syntax, type the entire syntax entry, including the continuation lines, on the same input line.

ENVIRONMENT VARIABLES

jBASE uses a number of environment variables to modify jBASE behavior. Suitable defaults apply to most of them. Although most environment variables can be set any time, the best place to do so is in the .profile script.

Setting/Getting Environment Variables

Windows	UNIX
set variable=value	variable=value export variable
echo %variable%	echo \$variable
Variables can be configured in the System environment for all users, and/or on a per user basis via the user environment. Additional variables for jBASE can also be added to the current user configuration registry.	This works for all shells, although one can do “export variable=value” in ksh, etc.
Win9x variables are usually configured in the AutoExec.bat. Care should be taken that the environment area does not become overwritten on Win9x as it is initially quite small, approximately 512 bytes. Subsequent .bat commands should increase the required environment space. Setting it in the config.sys file can explicitly increase the environment space:	Variables are usually configured in the .profile of the user login directory although global variables can be added to the /etc/profile script.
shell=c:\command.com /e:2048 /p	

jBASE PROGRAMS

The jBASE BASIC functions PUTENV and GETENV can be used to manipulate environment variables. For example:

```
IF PUTENV (envar=x) THEN ELSE NULL
IF GETENV (envar) THEN ELSE NULL
```

jBASE Initialization

Some environment variables can only be set before jBASE initialization. jBASE initialization occurs when the first jBASE program is executed for a particular "PORT".

The jBASE initialization process reads the environment entries looking for possible variables required by jBASE. These environment variables continue to be valid as long as the "PORT" is still active. Some environment variables can be changed by subsequent program execution. The state of these variables is imported back into the local environment after program execution.

For instance:

T-ATT requires a "PORT" against which it saves the tape device assignment

.SP-ASSIGN requires a "PORT" with which to save assignment status for print jobs

READNEXT in a program after SELECT/GET-LIST

With jBASE 4.1 all programs execute in the same process unless explicitly executed via the chars(255) *.k construct.

UNIX

jBASE initialization on UNIX is usually performed in the .profile.

Windows

jBASE initialization on Windows usually occurs when the first jBASE program executes.

Initial Environment Variables

Variables cannot be imported again after jBASE initialization.

PATH	Pathnames of executables
HOME	Pathname of user home directory
LD_LIBRARY_PATH	Pathnames of system libraries (UNIX only)
LIBPATH	Pathnames of system libraries (AIX only)
SHLIB_PATH	Pathnames of system libraries (HPUX only)
LANG	Language type (UNIX only)
LC_ALL	Locale (Unix only)
TZ	Timezone (Unix only)
TERM	Specifies terminal type as per terminfo database
TERMINFO	Specifies directory for terminal settings
JBCPORTNO	Forced Port number for use by user. (Usually allocated)
JBCLOGNAME	User name to use in-place of login id
JBCGLOBALDIR	Pathname of jBASE global configuration directory
JBCRELEASEDIR	Pathname of jBASE release directory
JBCEMULATE	Emulation to be used for this user
JEDIFILEPATH	Directory Paths of application files location
JEDIFILENAME_MD	Pathname of file to be used for MD entries
JEDIFILENAME_SYSTEM	Pathname of file to be used for SYSTEM entries

Additional jBASE Environment Variables

EXECUTION - Environment Variables

JBCBACKGROUND	Set to 1 to run “PORT” as background task
JBCRUNDIR	Set to executable directory for RUN command
JEDIENABLEQ2Q	Set to 1 to force detection of Qptr to Qptr
JEDI_DISTRIB_DEFOPEN	Set to 1 to defer OPENs of distributed file parts
JEDI_SECURE_LEVEL	Set security level for flushable files (such as J3s or jPLUS files) Level 1. No, flush Level 2. Flush on link modification Level 3. Flush after update, default. (Network failure)
JEDI_INDEX_MMAP_ON	Set to force use of memory mapping on indexes when updating memory mapped files
JEDI_AIX_FILE_MMAP_ON	Set to force use of memory mapping of j4 files on AIX multi-processor machines
JEDI_AIX_OBJECT_MMAP_ON	Set to force use of memory mapping of .el files on AIX multi-processor machines
JEDI_OBJECT_MMAP_OFF	Turn off memory mapping of .el files
JBC_TCLRESTART	Set to command to execute instead of shell
JBC_ENDRESTART	Set to command to execute after end from debugger
JBCRESTARTPROG	Set to command to execute after off
JBCOBJECTLIST	Set to alternate path(s) for user subroutine libraries Windows - %home%\lib UNIX - \$HOME/lib
JBC_BLOCK_SYSTEM14	Set to 1 to force a 100-millisecond delay on SYSTEM (14) calls.
JBASE_ERRMSG_NON_NUMERIC	Controls behaviour when a numeric operation on a non-numeric value is encountered
JBASE_ERRMSG_ZERO_USED	Controls behaviour when a zero used condition is encountered
JBASE_WIN_TERM_SVR	This should be set on servers running Windows Terminal Server before starting the License Server, and for all sessions wishing to access jBASE licences.

DEVELOPMENT - Environment Variables

JBCDEV_BIN	Set to alternate path to catalog executables. Windows - %home%\bin UNIX - \$HOME/bin
JBCDEV_LIB	Set to alternate path to catalog libraries. Windows - %home%\lib UNIX - \$HOME/bin
JBCDEBUGGER	Set to 1 to force entry into debugger.
JBCTTYNAME	Specify alternate tty name for output. (UNIX only)
LIB	Specify additional paths for linking with libraries. (NT only)
INCLUDE	Specify additional paths for header files
JPQDEBUG	Set to 1 for PROC to enter "DEBUG" at IP input

MISCELLANEOUS - Environment Variables

JBCERRFILE	Specify alternate error message file
JBCPOOLERDIR	Specify alternate spooler directory
JBC_DESPOOLSLEEP	Specify the interval for despoolers to check for queued jobs
JBC_CRREQ	Controls linefeeds, formfeed and carriage returns in print jobs
JBC_OLD_SP_EDIT	Specifies the alternative SP-EDIT format
JBCLISTFILE	Specify alternate select list file

TERMINAL - Environment Variables

JBCSCREEN_DEPTH	Specify alternate terminal depth
JBCSCREEN_WIDTH	Specify alternate terminal width
JBCPRINTER_DEPTH	Specify alternate printer depth
JBCPRINTER_WIDTH	Specify alternate printer width
JBCPRISM	Set hard coded prism sequences (NT only)
JBC_STDERR	Set to 1 to redirect standard error to standard out. Useful for capturing output that would normally be sent to the screen.
JBCCREATEFLAGS	Set to 0, 1, and 2 for output redirection. (NT only) 0 Direct to current console (default) 1 Direct to new console 2 Detached for no console

EMBEDDED SQL - Environment Variables

Setting these environment variables overrides the jcompile built-ins when processing files containing Embedded SQL using the -Jq<flavour> option.

JBC_SQLLIBS	Set alternate SQL libraries for embedded SQL
JBC_SQLPREPROC	Set alternate SQL pre-compiler command
JBC_SQLCOPTS	Set alternate SQL options for C compiler (also passed via nsqprep for MSSQL)

EXAMPLE

For Oracle Pro*C Embedded SQL pre-compiler, on Windows the following environment variables can be set (assuming e.g. **ORACLE_HOME=C:\Oracle\product\9.2.0.1.0\Client_1**):

```
JBC_SQLLIBS=%ORACLE_HOME%\oci\lib\msvc\oci.lib /libpath:%ORACLE_HOME%
%\precomp\lib /libpath:%ORACLE_HOME%\precomp\lib\msvc orasql9.LIB
JBC_SQLPREPROC=proc mode=oracle ltype=none ireclen=255 oreclen=255
iname=
```

Assuming that the PATH environment is also configured for Embedded SQL, the command **jcompile -Jqo SqlDemo.b** compiles the jBASE BASIC program, including passing it through the Oracle Pro*C pre-processor.

CREATE-FILE - Environment Variables

JEDI_PREFILEOP	Parameters take precedence before command line
JEDI_POSTFILEOP	Parameters take precedence after command line

EXAMPLE

To convert all files on “jbackup” tape to J4 files set the following environment variable is using jrestore.

Export JEDI_PREFILEOP=TYPE=J4 (UNIX) Can use quotes to surround multiple parameters
set JEDI_PREFILEOP=TYPE=J4 (NT)

Don't use quotes

jRFS – Environment Variables

JBCNETACCESS	Specify the location of the jRFS security access file
JBCNETDIR	Specify the location of the jRFS configuration files
JRFS_SERVERNAME	Allows the jRFS client to override the service port
JRFS_HOSTNAME	Allows the jRFS client to override the target host

PUTENV - Command

Used to set environment variables for the current process

COMMAND SYNTAX

PUTENV (expression)

SYNTAX ELEMENTS

Expression should evaluate to a string of the form:

EnvVarName=value

Where EnvVarName is the name of a valid environment variable and value is any string that makes sense to variable being set.

If PUTENV function succeeds it returns a Boolean TRUE value, if it fails it will return a Boolean FALSE value.

NOTES:

PUTENV only sets environment variables for the current process and processes spawned (say by EXECUTE) by this process. These variables are known as export only variables.

See also GETENV.

EXAMPLE

```
IF PUTENV ("JBCLOGNAME=":UserName) THEN
    CRT "Environment configured"
END
```

GETENV - Command

All processes have an environment associated with them that contains a number of variables indicating the state of various parameters. The GETENV function allows a jBASEBASIC program to determine the value of any of the environment variables associated with it.

COMMAND SYNTAX

GETENV (expression, variable)

SYNTAX ELEMENTS

The expression should evaluate to the name of the environment variable whose value is to be returned. The function will then assign the value of the environment variable to variable. The function itself returns a Boolean TRUE or FALSE value indicating the success or failure of the function.

NOTES:

See the UNIX documentation for the Bourne shell (sh) or the Windows on-line help for information on environment variables. [Click here](#) for information regarding environment variables unique to the jBASE system.

See also PUTENV

EXAMPLE

```
IF GETENV ("PATH", ExecPath) THEN
    CRT "Execution path is ":ExecPath
END ELSE
    CRT "Execution path is not set up"
END
```

PATH

DESCRIPTION

The PATH variable contains a list of all directories that contain executable programs. As a minimum, this should contain the shell default value plus the path /the shell sees usr/jbc/bin so that j JBASE commands. You will also wish to add the path of your application executable directory (such as \${HOME}/bin).

VALUES

for any directory, the user has privileges

DEFAULT

The default depends entirely upon your UNIX system and how it has been set up.

SETTINGS

Normal UNIX environment variable, so it can be set at any time by the commands:

UNIX	Windows
<code>PATH=\$PATH:/apps/bin</code> <code>export PATH</code>	<code>SET PATH=%PATH%;D:\apps\bin</code>

LD_LIBRARY_PATH

DESCRIPTION

This is a SVR4 UNIX only variable and should be set to /usr/jbc/lib.

VALUES

Colon separated library file paths.

DEFAULT

None

SETTINGS

Normal UNIX environment variable, so it can be set at any time by the commands:

```
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/jbc/lib
export LD_LIBRARY_PATH
```


TERM

DESCRIPTION

On **UNIX**, this variable should be set to your terminal type as defined by the UNIX terminfo database

On **Windows**, it should be set to a value in any of the directories under %JBCRELEASEDIR%\misc.

VALUES

On **UNIX**, any valid terminfo database entry

On **Windows**, any file name (up to the underscore) in the directories under %JBCRELEASEDIR%\misc. Additional terminal definitions can be created using the jtlic command.

DEFAULT

On **UNIX**, the default depends upon your system and how it has been set up.

SETTING

Normal environment variable, so it can be set at any time by the commands:

UNIX	Windows
<code>TERM=vt220</code> <code>export TERM</code>	<code>SET TERM=vt220</code>

TERMINFO

DESCRIPTION

The TERMINFO environment variable is used for terminal handling. The environment variable is supported only on platforms that provide full support for the terminfo libraries that System V and Solaris UNIX systems provide.

VALUES

The TERMINFO environment variable defines a directory where the terminal settings are read from.

DEFAULT

On UNIX, the default depends upon your system and how it has been set up.

SETTING

Normal environment variable, so it can be set at any time by the commands:

UNIX	Windows
<code>TERMINFO=/usr/term</code> <code>export TERMINFO</code>	<code>SET TERMINFO=c:\term</code>

JBCPORTNO

DESCRIPTION

This defines your current port number and is useful when a particular user wishes to keep the same port number whenever they log on. On UNIX, it takes a sensible default, but this default may change if the UNIX configuration is changed. On Windows, port numbers are allocated sequentially from zero.

VALUES

Decimal port number

On UNIX, lists and ranges of port numbers can be specified.

DEFAULT

None

SETTING

UNIX	WINDOWS
Set in the .profile prior to execution of initial jBASE program	set before invoking the jSHELL

NOTES

UNIX	Windows
JBCPORTNO=300,304,310,350,360-390 export JBCPORTNO	SET JBCPORTNO=300
On UNIX platforms, jBASE will assign the lowest available port number from the list or range specified. If all port numbers specified by JBCPORTNO are already in use then the user is denied access. Port number is already logged on and in use	On Windows , if the specified port number is in use then the connecting process is given the next highest port number available. jBASE OBJEX processes are automatically assigned port numbers from 5,000. Processes run in the background (see jstart -b) are assigned port numbers from 10,000 but a GETENV () on JBCPORTNO will always return -1.
From Port 45, tty /dev/pts/1, JBASE pid 16754	

JBCLOGNAME

DESCRIPTION

The account name as perceived by commands such as “WHO” or conversions such as U50BB will normally be returned as the login name of the UNIX user (LOGNAME variable). However if you wish your users to login with personal ids but execute as if they were all on the same account you may set this variable to override the default. The account name will be returned to whatever this environment variable is set.

VALUES

any character string

DEFAULT

None

SETTING

As per normal environment variable

UNIX	WINDOWS
setup in the . profile	set before running any jBASE program

NOTES

UNIX	Windows
JBCLOGNAME=PAYROLL export JBCLOGNAME	SET JBCLOGNAME=HARRY

JBCGLOBALDIR

DESCRIPTION

Defines the directory for jBASE global files

VALUES

Valid file path

DEFAULT

The default value is the same as JBCRELEASEDIR

UNIX	Windows
/usr/jbc	C:\JBASE4\4.1

SETTING

UNIX	Windows
As per normal environment variable, should be setup in the *.profile JBCGLOBALDIR=/usr/jbc export JBCGLOBALDIR	This is set in the registry when jBASE is installed. See HKEY_LOCAL_MACHINE/SOFTWARE/JA C/jBASE/3.0/CURRENT_CONFIG This value can be overridden by setting JBCGLOBALDIR as an environment variable.

JBCRELEASEDIR

DESCRIPTION

Defines the release directory for the jBASE system executables and libraries

VALUES

Valid file path

UNIX	Windows
/usr/jbc (default)	C:\JBASE4\4.1 (default)

SETTING

UNIX	Windows
On UNIX, as per normal environment variable, should be Set in the .profile prior to execution of initial jBASE program JBCRELEASEDIR=/usr/jbc3.1 export JBCRELEASEDIR	SET JBCRELEASEDIR = c:\jbase4\4.1

JBCDATADIR

DESCRIPTION

Defines the location for jBASE to determine any configured databases. Overrides the default setting for the spooler directory.

NOTES

When the JBCSPOOLERDIR is not defined, the default setting for the jBASE spooler directory is \$JBCDATADIR/jbase_data. When JBCDATADIR is not set, the default setting is \$JBCGLOBALDIR/jbase_data.

If the JBCGLOBALDIR is not set, it defaults to \$JBCRELEASEDIR.

If the JBCRELEASEDIR is not set, the default settings are /opt/jbase4/4.1 (Unix) or c:\jbase4\4.1 (Windows)

VALUES

Valid file path

UNIX	Windows
/opt/jbase4/4/1/jbase_data	C:\JBASE4\4.1\jbase_data

SETTING

UNIX	Windows
On UNIX, as per normal environment variable, should be set in the .profile prior to execution of initial jBASE program JBCDATADIR=/usr/jbc/data	SET JBCDATADIR = c:\mydata

JBCDEFDICTS

DESCRIPTION

Specifies one or more files that are used to hold dictionary items for use by jQL. When JBCDEFDICTS is set, jQL will scan each specified file for dictionary items that cannot be located in the dictionary of the queried file. When JBCDEFDICTS is not set, jQL will scan just the dictionary of the queried file and then the MD / VOC.

VALUES

Colon separated file paths (Unix)

Semicolon separated file paths (Windows)

DEFAULT

None

SETTING

UNIX	Windows
JBCDEFDICTS=/mydicts	SET JBCDEFDICTS=c:\mydicts
export JBCDEFDICTS	

JBCEMULATE

DESCRIPTION

When importing legacy applications, this variable tells the jBASE what system it originally ran on. NOTE: that programs and subroutines imported from different systems may be freely mixed.

VALUES

jBASE, adds, ape, fuj, prime, ros, r83, r91, ultimate, universe
.The values reference labels in the Config_EMULATE file.

DEFAULT

The default is jBASE, which will suit most imported applications.

SETTING

Normal environment variable, so it can be set at any time by the commands:

UNIX	Windows
<code>JBCEMULATE=Value</code>	<code>SET JBCEMULATE=Value</code>
<code>export JBCEMULATE</code>	

JDIAG

DESCRIPTION

This environment variable provides a variable amount of jBASE trace information depending on which options are specified.

VALUES

Colon separated name and value pairs from the following options;

profile={off|short|long|user|jcover|all}

filename={stdout|stderr|tmp|pathname,refresh_mins} %p can be used for process ID

memory={off|on|verify}

branch={off|on|verbose}

trace=env_name{,env_name ...}

DEFAULT

Not set.

SETTING

UNIX	Windows
JDIAG=profile=on:branch=on export JDIAG	SET JDIAG=profile=on:trace=on

JEDIFILEPATH

DESCRIPTION

This environment variable provides a list of directories in which to search for jBASE data files. If an MD or VOC file is configured with F / Q pointers, these take precedence over the directories in the JEDIFILEPATH.

VALUES

Colon separated file paths (**UNIX**)

Semicolon separated file paths (**Windows**).

DEFAULT

The current directory

SETTING

As per normal environment variable, so it can be set at any time. The use of relative file paths (such as ".") should be avoided as it can result in unintended file access.

UNIX	Windows
JEDIFILEPATH=/appvol/WB1:/appvol /WB2:/appvol/WB3 export JEDIFILEPATH	SET JEDIFILEPATH=F:\apps\WB1;F:\apps \WB2;F:\apps\WB3

JEDIFILENAME_MD

DESCRIPTION

This variable should be used if you require the use of the MD/VOC file to hold Q pointers, jCL programs, paragraphs or entries for the jQL language. If you have loaded an account-save into your home directory then you might wish to set this variable. This will then allow you to:

Execute jCL programs and paragraphs directly from the MD/VOC (using jsh or EXECUTE/CHAIN etc.) On systems with 14 char filename limits, create cross-reference items for executables from the original name to the new name. F pointers and Q pointers in an MD / VOC take precedence over paths in the JEDIFILEPATH.

VALUES

Valid file path; while it is not required, it is strongly advised that this value be set to the complete path of the MD and not a relative path (as an example, “/home/bob/MDJD” should be used instead of “./MDJD”).

DEFAULT

None

SETTING

As per normal environment variable, so it can be set at any time by the commands:

UNIX	Windows
JEDIFILENAME_MD=/usr/GLOBAL/MD	SET
export JEDIFILENAME_MD	JEDIFILENAME_MD=D:\GLOBAL\VOC

JEDIFILENAME_SYSTEM

DESCRIPTION

If you are using Q pointers in a defined MD/VOC file then you may well need to create a SYSTEM file to define the home directories of other accounts. By default Q-pointers are resolved by using the \$JBCRELEASEDIR/src/SYSTEM file. Setting the JEDIFILENAME_SYSTEM variable to an alternative hash file or directory can change this.

While it is not required, it is strongly advised that this value be set to the complete path of the system file and not a relative path (as an example, “/home/islandjim/SYSTEMJD” should be used instead of “./SYSTEMJD”).

VALUES

Valid file path

DEFAULT

None

SETTING

As per normal environment variable, so it can be set at any time by the commands:

UNIX	Windows
<code>export</code>	<code>SET</code>
<code>JEDIFILENAME_SYSTEM=/home/alternative/SYSTEM</code>	<code>JEDIFILENAME_SYSTEM=D:\home\alternative\SYSTEM</code>

JEDIENABLEQ2Q

DESCRIPTION

Resolve this environment variable by setting to Q-pointer-to-Q-pointer chains. The maximum chain length allowed is 11. NOTE:: that this environment variable enables Q-pointer-to-Q-pointer resolution. Q-pointer to F-pointer resolution is not supported.

VALUES

1

DEFAULT

Not set

SETTING

As per normal environment variable

UNIX	Windows
JEDIENABLEQ2Q=1 export JEDIENABLEQ2Q	SET JEDIENABLEQ2Q=1

JEDI_DISTRIB_DEFOPEN

DESCRIPTION

Setting this environment variable will defer the OPENing of component or part files in a distributed file set until the component file is required to be opened by the application program.

VALUES

1

DEFAULT

Not set.

SETTING

As per normal environment variable

It should be setup before the main file is accessed.

UNIX	Windows
<code>JEDI_DISTRIB_DEFOPEN=1</code>	<code>SET JEDI_DISTRIB_DEFOPEN=1</code>
<code>export JEDI_DISTRIB_DEFOPEN</code>	

JEDI_SECURE_LEVEL

DESCRIPTION

Defines the security level for files which support configurable flushing.

VALUES

- 1 Switches off secure mode.
- 2 When certain changes occur that could corrupt the file in the event of a failure, the file data is flushed from memory to disk. Normal updates will not be flushed.
- 3 All file updates will cause the file data to be flushed from memory to the disk.

DEFAULT

3

SETTING

As per normal environment variable

UNIX	Windows
JEDI_SECURE_LEVEL=2	SET JEDI_SECURE_LEVEL=2
export JEDI_SECURE_LEVEL	

Performance Implications

There is a performance penalty to pay for running in secure mode levels 2 and 3.

Level 2 will protect against file corruption by flushing the file from memory to disk when certain operations occur. However, it will not protect against loss of data. Most operating systems will periodically flush this data, usually a tuneable system and often with a default of every 60 seconds. Therefore, if you can withstand a loss of up to 60 seconds of data, and your primary concern is that the files are not corrupted in the event of a system failure, then this is the level for you. Minimal impact on performance is seen so long as your files are properly sized. Even if they go out of the main group, performance is only impacted if the extended group size keeps varying considerably.

Level 3 will protect against almost everything including loss of data. This impacts most on the system. The actual level of performance impact depends greatly on your application. For example, most of your updates may be to very large files in a pseudo-random manner (e.g. updating stock records, customer details etc.). In this situation, all this does is move the overhead from the operating system flush daemon that runs about every 60 seconds (see Level 2 above) to the process doing the update. Therefore, it may be a case of “What you lose on the roundabouts

you gain on the swings!” On the other hand, you may have a small file regularly being updated with things like current days orders. In this case the impact will be substantial as you will be causing a disk update for each application WRITE, whereas without this you might do many of these WRITES before the operating system daemon does a single write.

Another way to control your flushing of data to disk is to use transaction boundaries. For example, the following jBASE BASIC code will cause all data to be flushed to disk for all files regardless of the file type or file status

```
TRANSTART ELSE PRINT "Error" ; STOP
```

```
WRITE record1 TO filevar1
```

```
WRITE record2 TO filevar2
```

```
TRANSEND SYNC ELSE PRINT "Error" ; STOP
```

This mechanism guards against data loss but is less effective in protecting against file corruption should the server fail while the TRANSEND is actually in progress.

You should set JEDI_SECURE_LEVEL=1 when doing batch updates on a secure mode file. For example, when copying a number of records from a temporary file to a secure mode file called CUSTOMERS:

```
jsh -->export JEDI_SECURE_LEVEL=1
jsh -->COPY TEMPFILE * (O
TO: (CUSTOMERS
1217 Record(s) copied
jsh -->unset JEDI_SECURE_LEVEL
```

In the above example the secure mode is disabled during the COPY, command and so will perform much quicker. When the COPY is completed, it is normal operating system practice to flush all modified file data to disk anyway.

See Also: CREATE-FILE, Transaction Boundaries, Transaction Journaling

JBC_TCLRESTART

DESCRIPTION

Enables the Command Level Restart feature

VALUES

Restart Program

DEFAULT

Command Level Restart feature disabled

SETTING

UNIX	Windows
Create the JBC_TCLRESTART environment variable in the .profile prior to execution of initial jBASE program	Set before any jBASE program is invoked. The environment variable should contain the command string to execute when the user would otherwise go to a shell Prompt.

To later enable the feature, use the BITSET(-2); to later disable the feature, use the BITRESET(-2)

JBC_ENDRESTART

DESCRIPTION

Enables the Break/End Restart feature

VALUES

Restart program name

DEFAULT

Break/End Restart feature disabled

SETTING

UNIX	Windows
Create the JBC_ENDRESTART environment in the .profile prior to execution of the initial jBASE program	Set before any jBASE program is run. The environment variable should contain the command string to execute when the debugger is entered/exited.

To later enable the feature, use the BITSET (-3); to later disable the feature, use the BITRESET (-3)

JBCOBJECTLIST

DESCRIPTION

Defines the directories to find user shared object libraries where user subroutines are located.

VALUES

File paths Colon separated on **UNIX**. Semi-colon separated on Windows.

DEFAULT

%HOME%\lib (**Windows**)

\$HOME/lib (**UNIX**)

NOTE: it is good practice to set JBCOBJECTLIST explicitly rather than relying on the default setting. This is because the value of the HOME environment variable may change (for example after a LOGTO).

SETTING

UNIX	Windows
Set in the .profile before execution of the initial jBASE Program.	Set before the jSHELL is invoked.
JBCOBJECTLIST=\$HOME/lib:/home/TESTJBASE/lib	SET JBCOBJECTLIST=%HOME%\lib;C:\dev\TESTJBASE\lib
export JBCOBJECTLIST	

JBC_BLOCK_SYSTEM14

DESCRIPTION

When this environment variable is set, it calls SYSTEM(14) which results in a 100-millisecond delay.

VALUES

1

DEFAULT

not set.

SETTING

As per normal environment variable, the environment variable can be set dynamically with PUTENV

UNIX	Windows
JBC_BLOCK_SYSTEM14=1 export JBC_BLOCK_SYSTEM14	SET JBC_BLOCK_SYSTEM14=1

NOTE: Looking for type ahead data using SYSTEM(14) in a tight loop can have a detrimental impact on system performance because left unchecked, such loops can consume all available system resources. With JBC_BLOCK_SYSTEM14 set, each call to SYSTEM(14) incurs a 100-millisecond delay, so a loop with SYSTEM(14) doesn't waste system resources by looping too quickly.

It should be noted that the accuracy of the pause is dependent on the granularity of the system clock and the load on the system. Most operating systems and hardware will provide a granularity of 10 milliseconds.

JBASEUNIQUE

DESCRIPTION

When this environment variable is set to a directory, jBASE dynamically creates and deletes workfiles jBASEWORK_nn where nn is the port number. This can be used in place of JBASETMP.

VALUES

Any valid directory.

DEFAULT

not set.

SETTING

UNIX	Windows
<pre>JBASEUNIQUE=\$HOME/jBASEWORK export JBASEUNIQUE</pre>	<pre>SET JBASEUNIQUE=%HOME%\jBASEWORK</pre>

Setting this environment variable is recommended in a high user environment as a single workfile for all ports can result in a bottleneck.

JBASE_ERRMSG_NON_NUMERIC

DESCRIPTION

Defines behaviour when a BASIC program encounters a numeric operation being attempted on a non-numeric value. The default behaviour is to raise an error and drop into the debugger.

VALUES

- 1 – Don't display an error message
- 2 – Don't enter the debugger
- 16 – Caller to place "0" in the target variable after operation
- 32 – Caller to place "" in the target variable after operation
- 64 – Caller to leave target variable alone after operation
- 128 – Caller to place source variable in the target variable after operation

DEFAULT

0 - Raise an error and drop into the debugger.

SETTING

The value stored in a bit mask so different behaviours can be combined by adding them together. For example, to suppress the error message and avoid going into the debugger – set the variable to 3. As per normal environment variable, the environment variable can be set dynamically with PUTENV

UNIX

```
JBASE_ERRMSG_NON_NUMERIC=3  
export JBASE_ERRMSG_NON_NUMERIC
```

Windows

```
SET JBASE_ERRMSG_NON_NUMERIC=3
```

NOTE: Supersedes JBASE_WARNLEVEL from previous versions of jBASE..

JBASE_ERRMSG_TRACE

DESCRIPTION

Defines behaviour when a BASIC program encounters an error

VALUES

1 – Log an error message to \$JBCRELEASEDIR/tmp/jbase_error_trace

DEFAULT

0 – Do not log the error.

SETTING

The only valid values for this variable are 1 or 0. Setting this variable will not interfere with the behaviour set by other JBASE_ERMSG environment variables. As per normal environment variables, it can be set dynamically using PUTENV

UNIX

```
JBASE_ERRMSG_TRACE=1  
export JBASE_ERRMSG_TRACE
```

Windows

```
SET JBASE_ERRMSG_TRACE=1
```

NOTE: Supersedes JBASE_WARNLEVEL from previous versions of jBASE..

JBASE_ERRMSG_ZERO_USED

DESCRIPTION

Defines behaviour when a BASIC program encounters a null variable. The default behaviour is to raise an error and drop into the debugger.

VALUES

- 1 – Don't display an error message
- 2 – Don't enter the debugger
- 16 – Caller to place "0" in the target variable after operation
- 32 – Caller to place "" in the target variable after operation
- 64 – Caller to leave target variable alone after operation
- 128 – Caller to place source variable in the target variable after operation

DEFAULT

0 - Raise an error and drop into the debugger.

SETTING

The value stored in a bit mask so different behaviours can be combined by adding them together. For example, to suppress the error message and avoid going into the debugger – set the variable to 3. As per normal environment variable, the environment variable can be set dynamically with PUTENV

UNIX

```
JBASE_ERRMSG_ZERO_USED=3  
export JBASE_ERRMSG_ZERO_USED
```

Windows

```
SET JBASE_ERRMSG_ZERO_USED=3
```

NOTE: Supersedes JBASE_WARNLEVEL from previous versions of jBASE.

JBASE_WIN_TERM_SVR

DESCRIPTION

This should be set on servers running Windows Terminal Server before starting the License Server, and for all sessions wishing to access jBASE licences. It enables global access to shared memory to enable MTS sessions to obtain a jBASE licence.

VALUES

Set or unset.

DEFAULT

Unset.

SETTING

UNIX	Windows
N/A	SET JBASE_WIN_TERM_SVR=1

JBASE_SVR_SESSION

DESCRIPTION

On a machine with mixed enterprise and server licenses available, indicates that a server license is required.

VALUES

Set or unset.

DEFAULT

Unset.

SETTING

On sites with both server and enterprise licenses installed, an enterprise license will be assumed unless JBASE_SVR_SESSION is set to 1. With server only licenses installed, JBASE_SVR_SESSION must be set in order to obtain a license. Failure to do so will result in a licensing error. With enterprise only licenses installed, setting this environment variable will not allow a license to be allocated and a license error will be produced.

UNIX	Windows
<pre>JBASE_SVR_SESSION=1 export JBASE_SVR_SESSION</pre>	<pre>SET JBASE_SVR_SESSION=1</pre>

JBC_INVERT_ALPHA_CHARS

DESCRIPTION

Provided to emulate input on UniVerse systems. If this environment variable is set, all INPUT, KEYIN() and IN statements will receive input values in the opposite case. In other words, lower case characters will become upper case and vice-versa. Characters within cursor control sequences are also inverted, consequently up, down, left and right arrows will no longer work as required with this variable set.

VALUES

Set or unset.

DEFAULT

Unset..

UNIX	Windows
JBC_INVERT_ALPHA_CHARS=1 export JBC_INVERT_ALPHA_CHARS	SET JBC_INVERT_ALPHA_CHARS=1

Internationalisation

JBASE_I18N

DESCRIPTION

Setting this environment variable switches on UTF8 processing in jBASE.

VALUES

Set or unset.

DEFAULT

not set.

SETTING

UNIX	Windows
<pre>JBASEI18N=1 export JBASEI18N</pre>	<pre>SET JBASEI18N=1</pre>

More information on Internationalisation can be found in the jBASE Internationalisation manual..

JBASE_CODEPAGE

DESCRIPTION

Setting this environment variable sets the codepage to use for UTF8 conversion. This will have no effect unless internationalisation is switched on using JBASEI18N..

VALUES

Any valid code page. Use jcodepages utility for a list of supported code pages.

DEFAULT

not set.

SETTING

UNIX	Windows
<pre>JBASE_CODEPAGE=latin1 export JBASE_CODEPAGE</pre>	<pre>SET JBASE_CODEPAGE=latin1</pre>

More information on Internationalisation can be found in the jBASE Internationalisation manual..

JBASE_LOCALE

DESCRIPTION

Setting this environment variable sets the locale to use for UTF8 collation, sorting and date settings. This will have no effect unless internationalisation is switched on using JBASEI18N..

VALUES

Any valid locale. Use jlocales utility for a list of supported locales.

DEFAULT

not set.

SETTING

UNIX	Windows
<pre>JBASE_LOCALE=en_GB export JBASE_LOCALE</pre>	<pre>SET JBASE_LOCALE=en_GB</pre>

More information on Internationalisation can be found in the jBASE Internationalisation manual..

JBASE_TIMEZONE

DESCRIPTION

Setting this environment variable sets the timezone to use for UTF8 timestamp conversion into local time for display. This will have no effect unless internationalisation is switched on using JBASEI18N..

VALUES

Any valid timezone. Use jcodepages utility for a list of supported code pages.

DEFAULT

not set.

SETTING

UNIX	Windows
<pre>JBASE_TIMEZONE=Europe/London export JBASE_TIMEZONE</pre>	<pre>SET JBASE_TIMEZONE=Europe/London</pre>

More information on Internationalisation can be found in the jBASE Internationalisation manual..

Development

JBCDEV_BIN

DESCRIPTION

Defines the directory where user executables will be built when programs are CATALOGed.

VALUES

Valid file path

DEFAULT

%HOME%\bin (Windows)

\$HOME/bin (UNIX)

NOTE: it is good practice to set JBCDEV_BIN explicitly rather than relying on the default setting. This is because the value of the HOME environment variable may change (for example after a LOGTO

The value of JBCDEV_BIN can be overridden with the CATALOG -o option.

See also: JBCDEV_LIB, PATH.

SETTING

As per normal environment variable

UNIX	Windows
JBCDEV_BIN=/home/TESTJBASE/bin	SET
export JBCDEV_BIN	JBCDEV_BIN=C:\dev\TESTJBASE\bin

JBCDEV_LIB

DESCRIPTION

Defines the directory where user shared object libraries will be built when subroutines are CATALOGed.

VALUES

Valid file path

DEFAULT

%HOME%\lib (Windows)

\$HOME/lib (UNIX)

NOTE: it is good practice to set JBCDEV_LIB explicitly rather than relying on the default setting. This is because the value of the HOME environment variable may change (for example after a LOGTO).

The value of JBCDEV_LIB can be overridden with the CATALOG -L option.

See also: JBCDEV_BIN, JBCOBJECTLIST.

SETTING

As per normal environment variable

UNIX	Windows
JBCDEV_LIB=/home/TESTJBASE/lib	SET
export JBCDEV_LIB	JBCDEV_LIB=C:\dev\TESTJBASE\lib

JBCTTYNAME

DESCRIPTION

This variable defines your UNIX tty name. If this variable is not defined then jBASE must use the UNIX system call `ttyname ()` to locate it. On some systems, this function call is very slow but the use of this variable will greatly improve execution start-up times.

VALUES

any character string

DEFAULT

None

SETTING

As per normal UNIX environment, variable should be setup in the `.profile` before executing the initial jBASE program.

```
JBCTTYNAME=Jterm
```

```
export JBCTTYNAME
```

JBCERRFILE

DESCRIPTION

Sets the location of the jBASE error message file

VALUES

Valid path to a hashed file

DEFAULT

`$JBCRELEASEDIR/jbcmessages (UNIX) "`
`%JBCRELEASEDIR%\jbcmessages (Windows)`

SETTING

As per normal environment variable must be set before jBASE is invoked.

UNIX	Windows
<code>JBCERRFILE=/usr/global/jBASEerrors</code> <code>export JBCERRFILE</code>	<code>SET</code> <code>JBCERRFILE=C:\jstuff\jbcmessages</code>

NOTES: Setting this environment variable allows you to have more than one version of the messages displayed by jBASE. This could be desirable if you want different messages or different behavior when an error is encountered, depending on the user. For example, to prevent a program entering the debugger when an uninitialised variable is encountered, remove the ^WARNING^ string from the ZERO_USED message.

JBCSPOOLERDIR

DESCRIPTION

This environment variable defines the directory where the jBASE spooler entries are located.

VALUES

Valid file path

DEFAULT

/usr/jspooler (UNIX)

C:\JBASE30\jspooler

SETTING

As per normal environment variable

UNIX	Windows
setup in the .profile before executing the initial jBASE program	set before the jSHELL is invoked. If using telnet it should be set before the first executable in REMOTE.cmd.
JBCSPOOLERDIR=/home/alternative/jspooler	SET JBCSPOOLERDIR=C:\home\alternative\jspooler

JBC_DESPOOLSLEEP

DESCRIPTION

By default, the jBASE despooler processes on Windows check for queued jobs every 30 seconds. This environment variable can be used to decrease or increase that interval. The environment variable is not required on UNIX because the despooler processes are sent a signal when a new job has been generated.

VALUES

Number of seconds

DEFAULT

30

SETTING

Windows only: As per normal environment variable it should be set before form queues are started.

```
SET JBC_DESPOOLSLEEP=12
```

JBC_CRREQ

DESCRIPTION

Controls whether line feeds and form feeds are followed by a carriage return when printing to the spooler.

VALUES

- 0 No translation is performed
linefeed -> linefeed (unchanged)
formfeed -> formfeed (unchanged)
- 1 Specifies that a carriage return is required after each and every line feed when printing to the spooler
linefeed -> linefeed + carriage return
formfeed -> formfeed (unchanged)
- 2 Specifies that a carriage return is required after each form feed when printing to the spooler.
linefeed -> linefeed (unchanged)
formfeed -> formfeed + carriage return
- 3 specifies that a carriage return is required after each line feed and form feed when printing to the spooler.
linefeed -> linefeed + carriage return
formfeed -> formfeed + carriage return

DEFAULT

zero

NOTE: When printing to a Printronix printer on UNIX (which converts 'linefeeds' to 'linefeed + carriage return' but does not append 'carriage return' to 'form feeds') you should set JBC_CRREQ=two.

When printing binary data to a laser (or similar printer) on Windows you should set JBC_CRREQ=3

In addition, the device definition for the appropriate form queue should specify the -l and -n options to 'jlp' e.g. fqfred PROG jlp -d \\prntername -l -n

Alternatively, use the default Windows printer e.g. STANDARDNT

SETTING

As per normal environment variable, it must be setup before connecting to jBASE.

UNIX

```
JBC_CRREQ=2  
export JBC_CRREQ
```

Windows

```
SET JBC_CRREQ=3
```


JBCLISTFILE

DESCRIPTION

This environment variable specifies the file where stored lists are kept.

VALUES

Any valid path to a directory or hashed file

DEFAULT

If not set, jBASE will attempt to store lists in POINTER-FILE. If it can't be opened then it will store lists in jBASEWORK.

SETTING

As per normal environment variable, See also List Storage.

UNIX	Windows
<code>JBCLISTFILE=/home/jim/mylists</code>	SET
<code>export JBCLISTFILE</code>	<code>JBCLISTFILE=C:\globals\SAVEDLIST</code> S

JBCSCREEN_WIDTH

DESCRIPTION

Specifies the page width for paged terminal output, and overrides the value specified by the TERM type.

VALUES

Decimal number

DEFAULT

None

SETTING

As per normal environment variable, it should be setup before the jSHELL is invoked.

UNIX	Windows
JBCSCREEN_WIDTH=132 export JBCSCREEN_WIDTH	SET JBCSCREEN_WIDTH=132

JBCPRINTER_DEPTH

Description

This environment variable specifies the page depth for paged spooler output, and overrides the value specified by the TERM type.

VALUES

Decimal number

DEFAULT

None

SETTING

As per normal environment variable

UNIX	Windows
Setup in the .profile before executing the initial jBASE program JBCPRINTER_DEPTH=112 export JBCPRINTER_DEPTH	SET JBCPRINTER_DEPTH=112 Set before any jBASE program is invoked.

JBCPRINTER_WIDTH

DESCRIPTION

Specifies the page width for paged spooler output, and overrides the value specified by the TERM type.

VALUES

Decimal number

DEFAULT

None

SETTING

As per normal environment variable

UNIX	Windows
setup in the . profile before the jbcconnect command.	set before any jBASE program is invoked.
JBCPRINTER_WIDTH=125	SET JBCPRINTER_WIDTH=125
export JBCPRINTER_WIDTH	

jBASE Remote File Server (jRFS)

JBCNETACCESS

DESCRIPTION

Defines the location of the jRFS security access file `jnet_access`

VALUES

Valid file path

DEFAULT

`/usr/jbc/config` (UNIX)

`%JBCRELEASEDIR%\config` (Windows)

SETTING

As per normal environment variable

UNIX	Windows
<code>JBCNETDIR=/usr/jJBASEsetup</code> <code>export JBCNETDIR</code>	<code>SET JBCNETDIR=C:\JBASESETUP</code>

JBCNETDIR

DESCRIPTION

Defines the location of the jRFS configuration files

VALUES

Valid file path

DEFAULT

/usr/jbc/config (UNIX)

%JBCRELEASEDIR%\config (Windows)

SETTING

As per normal environment variable

UNIX	Windows
JBCNETDIR=/usr/jJBASEsetup export JBCNETDIR	SET JBCNETDIR=C:\JJBASESETUP

JRFS_REMOTE_JQL

DESCRIPTION

Specifies that the jQL command will run on the remote server and send the data back rather than querying line by line over the network

VALUES

1

DEFAULT

Not set

SETTINGS

As per normal environment values

UNIX	Windows
JRFS_REMOTE_JQL=1 export JRFS_REMOTE_JQL	SET JRFS_REMOTE_JQL=1

JRFS_LOCALPATH_JQL

DESCRIPTION

Specifies that the jRFS Server process will use the file name as 'opened' on the remote system rather than using the file name specified in the original select statement.

VALUES

1

DEFAULT

Not set

SETTINGS

As per normal environment values

UNIX	Windows
JRFS_LOCALPATH_JQL=1 export JRFS_LOCALPATH_JQL	SET JRFS_LOCALPATH_JQL=1

JRFS_SERVERNAME

DESCRIPTION

Allows the jRFS client to override the service port.

VALUES

1

DEFAULT

Not set

SETTINGS

As per normal environment values

UNIX	Windows
JRFS_SERVERNAME=1 export JRFS_SERVERNAME	SET JRFS_SERVERNAME=1

Comment Sheet

Please give page number and description for any errors found:

Page	Error

Please use the box below to describe any material you think is missing; describe any material which is not easily understood; enter any suggestions for improvement; provide any specific examples of how you use your system which you think would be useful to readers of this manual. Continue on a separate sheet if necessary.

Copy and paste this page to a word document and include your name address and telephone number. Email to documentation@jbase.com