jBASE Configuration

Contents

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 *Italic*	In syntax, italic indicates information that you supply. In text, italic also indicates UNIX commands and options, filenames, and pathnames.
Courier	Courier indicates examples of source code and system output.
Courier Bold	**Courier Bold** In examples, courier bold 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-

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.

CONFIGURATION

This manual describes how to configure and tune the various jBASE and system parameters. Some of this information can also be found in the jBASE release advisory along with installation instructions for the specific jBASE software release.

Platform Considerations

On most SVR4, systems the tunable parameters are pre-set according to available system resources and only one or two UNIX kernel parameters may require modification for jBASE.

Large File Support

Solaris 8

Solaris 8 UFS file system supports large files. When using the mount_ufs(1M) command to mount the file system, make sure to add the 'largefiles' option.
Sample output from 'mount' showing file systems with large file support

```
# mount

/home on /dev/dsk/c0t8d0s0
read/write/setuid/intr/largefiles/logging/onerror=panic/dev=800038
on Fri Mar 21 16:48:41 2003

/data on /dev/dsk/c0t9d0s0
read/write/setuid/intr/largefiles/onerror=panic/dev=800040 on Fri
Mar 21 16:48:41 2003
```

AIX 5.2

AIX has two journal file systems, jfs and jfs2. Both can support large files. To see which you currently have, run the command '**lsjfs**' as root user.

Sample output

```
#MountPoint:Device:Vfs:Nodename:Type:Size:Options:AutoMount:Acct:Oth
erOptions:LvSize:FsSize:FragSize:Nbpi:Compress:Bf:AgSize:
```

```
/data:/dev/lv00:jfs:::9961472:rw:yes:no::9961472:9961472:4096:4096:n
o:false:8:

/huge:/dev/lv01:jfs:::54263808:rw:yes:no::54263808:54263808:4096:409
6:no:true:64:
```

The column Bf means 'Big files allowed' and in this example is set to true to allow the use and creation of large files. Change this via **SMIT**.
JFS2 is large file enabled by default and is the best choice for file system on AIX.

HP-UX 11

By using the vxfs file system and SAM (system Administration tool) you can create volume which are large file enabled.

Sample output from /etc/fstab showing the volumes *largefiles* setting

```
/dev/data/large /large vxfs rw,suid,largefiles,delaylog,datainlog 0
2
```

Modifications to the UNIX Kernel

UNIX requires two parameter changes to run jBASE efficiently.

ULIMIT: This parameter affects the maximum size of files, which a user can create and should always be set to the maximum.

NOTE: one user may create a file and later another user who writes to it may unknowingly increase the size of the file when the data in the file exceeds the group size. The ULIMIT parameter should therefore be set to the maximum size to which you expect any file to grow.

Memory, mapped limit: used for memory-mapped files. For most UNIX installations, you can use the idtune (or a similar) command to change the memory map limit.

NOTE: that some systems do not support memory-mapped files, or the parameter might already be configured correctly on your system - refer to the UNIX man pages or the manuals for your system for more information.

jBASE Configuration

Daemon Configuration

Name	Description	Configuration File
jRLA	Record Locking Arbitrator	Config_jRLA
jRFS	Remote File Services	jnet/jrfs_config

Note: The daemons jBTP and jPML are no longer used in jBASE 4.1

User Configuration

UNIX	WINDOWS
IJU - Creates a "*. profile", which sets up user defaults.	The iju utility, which is available on windows prompts the user for setup information concerning the account, and generates a REMOTE.cmd file, which is read when the user logs in via telnet (akin to UNIX .profile). The following main environment variables, will be configured:
JBCLOGNAME	Defaults to login name
JEDIFILENAME_MD	Defaults to %HOME%\MD
JEDIFILENAME_SYSTEM	Defaults to the parent of %HOME%
JEDIFILEPATH	Defaults to "."
TERM	No default

It will give information concerning other important jBASE environment variables (such as JBCOBJECTLIST) and will run UpdateMd to update the MD if it is present, or creates one.

Spooler Configuration

UNIX	Windows
SP-NEWTAB and REST-SPOOLERS	SP-NEWTAB and REST-SPOOLERS as administrator from the jSHELL jspprint despools each queue Windows must use PROG lp -d device name or NT for default printers. If using PROG then spooled via jBASE despooler, else direct to device.

jBASE Daemons

jRFS (Remote File Services)

The jRFS daemon enables proprietary access to files from jBASE applications located remote systems.

NOTE: The jRFS daemon can be configured to start automatically at boot time by executing a jBASE script (/etc/rc2.d/S99jRFS), during the UNIX initialization routine for UNIX systems and by enabling as a service on NT.

jRLA (Record Lock Arbiter)

The lock arbiter is responsible for resolving all record locking conflicts for jBASE processes and runs in the background of your system. It is commonly referred to as the lock daemon.
If jRLA is not loaded, jBASE will use the normal operating system locks. This is acceptable for small user populations, but the operating system locking mechanism has limits on the number of locks available, and on performance.
Only enable the jRLA-locking daemon when no other users are executing jBASE programs, otherwise some applications will use operating system locks and others will use the jRLA-locking daemon.
In most situations, a process will take and release locks autonomously and disregard both the lock daemon and other processes. However, in certain situations it must inform the daemon of events or ask the daemon for permission. These situations only occur when there are clashes in locking such as when a process finds that a record is already locked or that another process requires the same lock.

jRLA Command

```
Called as:
jRLA -a {-b}
jRLA -c {-o}
jRLA -d {-v} {-filename ...} {-ppid ...} {-L}
jRLA -i {-bmuEPT} {-sr,b,g} {-tnn}
jRLA -k {E}
jRLA -S
```

Option	Meaning
-a	Attach to existing IPC resources
-b	Run in the Background (default)
-c	Clear the Lock Table
-d	Display the Lock Table
-ffile	Restrict the display option to file "file"
-i	Initialize shared memory and become the Record Lock Arbiter
-k	Kill the Lock Table and the Record Lock Arbiter
-m	Force jRLA to behave as a Multi-processor arbiter
-o	Override the "confirm you want to continue" message
-pnn	Restrict the display option to process id nn
-sr,b,n	Set lock table size to "r" record, "b" binary and "n" locks per group
-tnn	Set tidy up period to nn minutes
-u	Force jRLA to behave as a single-processor arbiter
-v	Verbose mode
-E	Errors caused by jRLA conflicts will be reported but ignored
-L	Display record locks
-P	Port jBASEd locks used (default is process id jBASEd locks)
-S	Force a tidy up now
-T	Time stamps all the locks

The jRLA daemon uses one of three possible locking implementations to control access to the record lock and binary lock areas, depending on platform and number of processors.
By default, the number of record or item locks configured is 3000 and the number of binary or group locks configured is 601.

jspprint: jBASE despooler process

Each jspprint despooler background process is responsible for despooling a specific spooler formqueue. The number of jspprint despooler processes will depend on the number of formqueues required to be active at any time. The jspprint processes can be started/stopped or suspended/resumed either directly by the spooler commands or by the spooler menu options.

jspprint

The jBASE despooler processes provides despooling facilities for the jBASE spooler. Each despooler process, jspprint, is responsible for a specific formqueue or formqueue number. Formqueue numbers are allocated and associated with a formqueue name, etc., by the SP-STATUS spooler menu. Either the SP-JOBS spooler job menu starts the despooler processes, or the: REST-SPOOLER restarts despoolers script, which can be incorporated into the system boot sequence.

jspprint Configuration

When the despooler process is initiated, the spooler tables are interrogated to find the related device configuration record in the jspool_log file, using the default spooler directory or the jBASE environment variable $JBCSPOOLERDIR. The configuration record key is defined as DEVCONFIG*nn, where nn is the associated formqueue number.

The configuration record contains various information for specifying device type, device name, form type, translation type and print job list. The jspprint process uses this information when despooling each print job. The form type, (default jspform_deflt), and translation type (default jspxl_deflt) configuration files are located in the default /usr/jbc/config or $JBCRELEASEDIR/config directory.

Defaults for the formqueue, such as banners, width, etc. are located in the jspform_deflt configuration file and modifying the jspxl_deflt configuration file can specify default character translations. The configuration files contain notes on their usage. You can either edit the default configurations, or change the form queue to use different configurations. Option 3 from SP-STATUS changes the form queue definition from the jspform_deflt to jspform_xxx where, xxx is the new formtype name to be associated with the formqueue.

Option 8 from SP-STATUS allows you to change the required character translations from the default jspxl_deflt to jspxl_xxx, where xxx is the new translation name to be associated with the formqueue.

Using the TERM command for individual print jobs can change the defaults for printer width and depth.

Once started, the jspprint process interrogates its allocated formqueue for print jobs. If no print jobs are found, the jspprint processes goes into a wait state and rechecks the formqueue

approximately every thirty seconds. If a print job is queued by the spooler process, the relevant despooler process is woken to handle the print job.

jspprint Command

The jspprint despooler can be started manually but is usually started by using the SP-RESUME spooler command or SP-JOBS menu option. The jspprint command expects a formqueue number, which it uses to lookup the formqueue configuration details. There is a one to one relationship of formqueue name to formqueue number.

COMMAND SYNTAX

```
Jspprint <options> FormQueueNumber
```

Spooler Initialisation

The spooler must be initialized before any of the spooler commands can be executed. This is usually done at installation time, but can be done whenever you need to reinitialize the spooler. When reinitializing the spooler, all the spooler tables and data are completely reset.
To reinitialize the spooler, use the following commands as root user:

```
# PATH=$PATH:/usr/jbc/bin
# LD_LIBRARY_PATH=/usr/jbc/lib
# export PATH LD_LIBRARY_PATH
# SP-NEWTAB
```

The path of your jBASE release directory should be substituted for /usr/jbc.
NOTE: AIX users may need to use LIBPATH instead of LD_LIBRARY_PATH.
You will be prompted to change four defaults, the group names to use, the directory to put the spooler in and so on. For most installations, the defaults will provide the optimum configuration, where at the prompt you can simply enter C.
The procedure will create the spooler tables and data in any directory you require - the default being /usr/jspooler.
NOTE: that if you choose a directory other than /usr/jspooler, you will need to change the JBCSPOOLERDIR variable in your ".profile" to reflect the new directory, or you can set a link from /usr/jspooler to the alternate directory.
The procedure will create a default formqueue named STANDARD, whose device name is /dev/lp. It will also initialize the queue to be active, which enables immediate printing. If you already have a device on /dev/lp immediately after spooler installation use option 4 from the SP-STATUS menu to configure the spooler device name to another device
If you already use the supplied UNIX spooler, the two can live in harmony by sending all the jBASE spooler output to the UNIX spooler by using option 4 from the SP-STATUS menu

Change the device type to PROG and the device name to LP. Any data in sent to the STANDARD queue in a jBASE program, will be re-printed via the lp UNIX spooler program. The jBASE spooler formqueues and print jobs can be manipulated by using the jBASE spooler commands directly or by using the SP-STATUS and SP-JOBS menus.

Restarting Despooler Processes

Individual jBASE despooler processes can be restarted by either using the SP-RESUME command directly, or from the SP-JOBS menu. For example:

SP-RESUME STANDARD

Alternatively, all defined formqueue despooler processes can be started at once by using the: REST-SPOOLER command. For example:

REST-SPOOLER

This command requires root privileges and pre-assigned environment variables such as the SP-NEWTAB command.
If the restart of despooler processes is required at system boot time, a script (for example, jSpoolInit) should be created to restart the required formqueue despoolers. Place the script in the /etc/init.d (or similar) directory and then set a link in the /etc/rc2.d (or similar) directory back to the script in the /etc/init.d directory. The script should use multiple invocations of SP-RESUME to restart selected formqueue despoolers or alternatively, it can use the REST-SPOOLER command to resume all defined printer queues.

EXAMPLE

```
jSpoolInit
JBCRELEASEDIR=/usr/jbc
export JBCRELEASEDIR
PATH=/etc:/bin:/usr/bin:$JBCRELEASEDIR/bin
LD_LIBRARY_PATH=$JBCRELEASEDIR/lib
export PATH LD_LIBRARY_PATH
case "$1" in
'start')
  echo "Starting jBASE despoolers"
  SP-RESUME STANDARD or :REST-SPOOLER
  ;;
'stop') ;;
esac
```

NOTE: AIX users may need to use LIBPATH instead of LD_LIBRARY_PATH.

MAINTENANCE

jBASE requires very little maintenance and housekeeping, apart from those tasks that would normally be required by the UNIX system. The main jBASE tasks are as follows.

- Backups and restores
- Importing and exporting application programs and files
- Cleaning the jBASE tmp directory
- jBASE file sizing

Backups and Restores

Regular, usually daily, backups are essential to the good housekeeping of any system. There are two mechanisms for performing backups.

You can use existing UNIX commands, such as ‘tar’ or ‘cpio’, which work well, but should not be run while a jBASE application is updating files. Both ‘tar’ and ‘cpio’ perform a binary dump of the file data, and do not obey any set locks, to indicate that an update is in progress. In addition, these saves can be limited because they cannot be restored correctly on a system, which has a different architecture to the original system.

The preferred mechanism is to use the jbackup and jrestore jBASE utilities. The jbackup program will back up normal UNIX data files and directories as well as jBASE data files, and will respect any locks set by jBASE applications.

NOTE: that if you choose to run jbackup concurrently with other active online jBASE applications, your saved files will not be corrupt, but there is no guarantee regarding the continuity of any data saved from an active system.

JBACKUP

JBACKUP provides fast on-line backup facilities, which can be used to check file integrity.

Jbackup -Option {Inputlist}

Where inputlist is a file containing a list of files, default stdin

Option	Explanation
-bn	Set number of write buffers to n
-c	Dump control files such as indexes as binary files
-e EncMode	Encryption mode. See below for details.
-f Device	Save to device file, default stdout

-k EncKey	Encryption key, between 8 and 32 characters.
-l	Link files to be saved as separate UNIX or hash files
-mn	Maximum data capacity of media in Mb, default 100 Mb
-pn	Set priority, nice value of parent process
-s	Save summary of statistics to UNIX/NT file
-v	Verbose mode
-L file	Save from List file
-B	Force blocksize to 128k. Default 16k
-Cn	Force blocksize to n bytes, rounded to nearest k
-F	Use fixed block device. Use for quarter inch cartridge (qic) tapes (NT only)
-N	Suppress compression if supported by device (NT only)
-S Statfile	Save statistics of all saved objects in jBASE, file Statfile. The dictionary for this file is JBCRELEASEDIR/jbackup] D.
-O	Override no backup file option, save all
-R	Suppress automatic rewind at end of backup
-P	Print and scan files only, no save
-V	Verbose dot mode, displays a "." For each file
-A Acc	Save from user name home directory (UNIX only)

EXAMPLES

find /home -print | jbackup -P

Reads all records, files and directories under the /home directory provided by the find selection and displays each file or directory name as it is encountered. This option can be used to verify the integrity of the selected files and directories.

jbackup FILELIST -f /dev/rmt/floppy -m1 -v

Reads all files and directories listed in the UNIX file FILELIST and writes the formatted data blocks to the floppy disk device, displaying each file or directory name as it is encountered. The jbackup utility will prompt for the next disk if the amount of data produced exceeds the specified media size of one Mbyte.

Jbackup -AjJBASE -S/opt/jbase4/tmp/jJBASE_stats>/dev/null

LIST /opt/jbase4/tmp/jJBASE_stats USING /opt/jbase4/jbackup NAME TOTAL SIZE ID-SUPP

Reads all files and directories in home directory of user-id "jBASE" Generates statistics information and outputs blocks to stdout, which is redirected to /dev/null. The statistics information is then listed using the jbackup dictionary definitions to calculate the file space used.

jfind C:\users\myhome -print | jbackup -P

Reads all records, files and directories under the C:\users\home directory provided by the find selection and displays each file or directory name as it is encountered. This option can be used to

verify the integrity of the selected files and directories. This command should be run with jshell type sh rather than jsh.

jbackup: Encryption

jbackup encryption modes (-e or –E) may be one of:

RC2
BASE64
DES
3DES
BLOWFISH

Using extended mode (-E) runs a second level of encryption over the encrypted data.

Encryption requires a key of between 8 and 32 characters which can be provided on the command line (-k), or entered from the keyboard (twice for confirmation). If using standard input to provide a file list the key will have to be passed in twice before the first filename.

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