JRCONFIG, STATLIN, COMSPEC FROM NEWCOMB

JRCONFIG Version 2.14

This device driver will make the configuration during the boot process quick and easy for an IBM PCjr. If you use this program, please send a contribution of $10 to

Larry Newcomb
9210 Adelaide Dr.
Bethesda, Md. 20817

This contribution will encourage me to continue to enhance this program, and to write other programs for the PCjr. If you have any suggestions for future enhancements or other utility programs, send me a letter and I will consider them.

New to version 2.14:

1 Fixed bug in JRCONFIG that caused the system to lock up if the -v option was used with a value less than -v16.

2 Fixed bug in STATLIN that caused STATLIN to disable itself if the -s option in JRCONFIG was less than the default value. Although it is not a bug, STATLIN can cause problems if the -v option is used in JRCONFIG with a value less than -v6.

New to version 2.13:

1 -a & -b will move the screen to the left or right.

2 The -x option will allow the step rate to be selectable. Useful if you have a 3.5 inch drive.

3 The -p and -q options have been removed. The print screen function was causing problems, so I removed it. I may make it a separate program and put it in a future release of JRCONFIG.

4 -f will set the boot screen color.

New to version 2.12:

Note: This version was never intended to be released. It contains several bugs, one of which resulted in the removal of the print screen function.

1 A device driver to access the ramdisk allocated by JRCONFIG.NRD.

2 Both versions (.DSK and .NRD) install INT 60. If you use a program that also installs INT 60, it is likely to cause a problem. I do not know of any programs that use INT 60. If anyone knows of a program that uses INT 60, please let me know.

3 The selective print screen will now save to a disk file.

4 -q will enable a help line for the selective print screen function.

5 -n will disable color for monochrome monitors.

6 STATLIN.COM will display information on line 26.

New to version 2.11:

1 I have included the patch to INT 9 that is supposed to fix the incompatibility with MS QuickBasic. I do not have QuickBasic, so I have not tested its effectiveness.

2 -j will fix the incompatibility problem that some communications programs have with the jr's internal COM port.

3 -m will turn off the internal speaker.

4 -g will read a battery backed clock and set the time in DOS.

5 Install 2 different Version 1.4 of the jrSynchro cartridge.

6 -p the selective printscreen function has been changed slightly. It now checks to make sure it is in a compatible mode. The screen attributes are saved in a buffer in the second text page, and restored after printing.

New to version 2.10:

1 I have changed the names of the two versions to clear up some of the confusion over which version does what. Also the defaults for all options now work the same for both versions.

```
    JRCONFIG.DSK - the .DSK signifies a ramDiSK and replaces the
                   old .SYS extension
    JRCONFIG.NRD - the .NRD signifies  No  RamDisk  and replaces
                   the old CONFIGJR.SYS
```

2 There are some programs that needed to have MODE CO80 run before they would display boarder colors properly. I believe I have fixed the -t3 option so this will no longer be necessary.

3 -x option to fix the step rate in DOS 3.x; this will make the diskette drives run quieter.

4 -e option to change the number of entries in the directory of the ramdisk.

5 -y option to install the jrSynchro cartridge if you have it plugged in.

6 -p adds a printer online check for shift-print screen, and will allow selection of lines to print.

7 -h will force the top of DOS user RAM to a given amount.

8 The DEBUG session for RAM.SYS has instructions for changing the number of entries you can have in the ramdisk directory.

9 The check for the PCjr ROM BIOS has been removed.

You MUST have at least 256K of memory to use this program.

To use this driver put the file JRCONFIG.DSK on a bootable DOS 2.1 disk. On the same disk put the following line as the FIRST line of your CONFIG.SYS file. Then re-boot.

```
    device=jrconfig.dsk -t2
```

This will:
Install your expansion memory (256K to 640K).
Install a 91K ramdisk in your internal memory (first 128K). This ramdisk will take the first available letter after your physical drives (starting with C:). If you only have one drive, drive B: will still be a phantom drive. By filling up your first 128K, DOS will be forced to load your programs into your expansion memory, which will cause your programs to run faster.
Put your screen in 80 column, low intensity, black & white.

Options:
(-a -b -c -d -e -f -g -h -j -k -l -m -n -s -t -v -w -x -y)

-a will move the screen n positions to the left, use -an where n is from 1 to 7.

-b will move the screen n positions to the right, use -bn where n is from 1 to 7.
Note: Both -a & -b MUST come after the -t option on the command line in CONFIG.SYS. If the -t option comes after either -a or -b, there effect will be cancelled.

-c will turn on the keyboard click.

-d will cause DOS to recognize extra floppy drives, use -dn option, where n is 1 to 4.

-f will set the screen color at boot time and when the mode is changed. Use -fnnn where nnn is from 0 to 255.

-e will set the number of directory entries allowed on the ramdisk. The default is 16 entries. Use -en where:

```
     -e1 will allow  32 entries
     -e2 will allow  64 entries
     -e3 will allow 128 entries
```

Each entry uses 32 bytes of the RAM reserved for the ramdisk.

-g will read the time and date from a battery backed clock and set the time and date in DOS. This was written for a JRCAPTAIN with the clock chip at address (37D hex - 893 decimal). If you have changed the jumper to set your clock chip at (27D hex - 637 decimal) then use -g637.
If you have another type of clock board, this option may or may not work for you. If your clock is at another address,

use -gnnn where nnn is the decimal address of the clock chip. If there is a problem reading the clock chip, JRCONFIG will beep and give the general error message about "the options list in CONFIG.SYS". If you are sure that nothing is wrong with your clock chip, then it is either at another address or it is incompatible with the code in JRCONFIG.

-h is not designed for the average user. I use it for testing programs I want to put outside the DOS user area. -hnnn will put nnn as the top of ram, regardless of how much ram you actually have; ie, -h512 will set the top of ram to 512K. If you don't have that much ram, your system will probably crash. You should NOT use this option and -l at the same time.

-j will put the address for the COM port on the motherboard into the BIOS data area for COM2. This will solve the problem that some communications programs have when using the COM port on the motherboard and an external modem. When you specify COM2, the program will now find it; and it will use the correct hardware interrupt.

-k will set Caps-Lock (Note: If you have a keyboard with an indicator light for Caps-Lock, the light will not be turned on.)

-l will look for extra RAM, starting at 640K and will stop at 736K, and give it to DOS. (Note: 640K is where the ROM stops testing RAM and 736K is where video ram "logically" starts.) This option will only perform a superficial check for RAM. You will need to use a ramtest program to check for bad chips above 640K.

-m will turn off the jr's internal speaker. If you run a program that turns the speaker back on, it will probably stay on.

-n will prevent color modes for monochrome monitors. JRCONFIG will intercept a call to change to CO40 or CO80, and make a call for BW40 or BW80. It will have no effect on graphics modes.

-s sets the size of the ramdisk, use -sn where n is the number of K bytes you want the ramdisk to be. The ramdisk size may be a little larger than the value you put in, because of video page boundaries. The default size is 91K for DOS 2.1 and 79K for DOS 3.1. (10K is the smallest ramdisk you can have, because that is how much room is left on the video page where IBMDOS.COM is loaded with DOS 2.1, and this space can not be used for anything else anyway. With DOS 3.1 you get about 15K, because IBMDOS.COM is larger and video ram must be moved to a higher video page). There is no upper limit on the size of the ramdisk (other than the amount of RAM you have in your computer).

-t will set the video display mode, use -tn where n is:

Alpha Modes
0 for 40 X 25 B/W (Default)
1 for 40 X 25 Color
2 for 80 X 25 B/W
3 for 80 X 25 Color
Graphics Modes
4 for 320 X 200 4 Color
5 for 320 X 200 B/W 4 Shades
6 for 640 X 200 B/W 2 Shades
7 is not a valid mode
Extended Graphics Modes
8 for 160 X 200 16 Color
9 for 320 X 200 16 Color
10 for 640 X 200 4 Color

You can get some funny looking screens when you boot up in a graphics mode because DOS expects to be in Alpha mode.

-v will reserve video ram, where n is 4 to 96. Normally 16K is set aside for video ram, you can either decrease or increase this amount with this option. Using -v4 will reserve 4K for video, giving you an extra 12K of user ram, but you must not use any graphics modes or a program that uses text pages other than the first test page. If you are concerned about speed, you may not want to use this option with a value less than 16. The extra RAM that you get this way will be in the video page, and that will be in the first 128K. Any other device drivers and COMMAND.COM will load into this area. Unless you have some large device drivers, the regular DOS programs you run will start in this low RAM. Don't be fooled by the size of the files on the disk. JRCONFIG is over half initialization code, the resident portion is less than 2K. Only about 3K of COMMAND.COM will load into low memory, the rest is the transient portion that is at the top of DOS memory. High values are used for reserving ram to be used for high-res graphics (32K) or extra video pages.

-w will put a clockwise spinning character in on the top row of your screen, use -wnn where nn is the column you want it to appear in.

-x will set the proper step motor rate in DOS 3.x, this will make the disk quieter. Note: this will not take effect until after you see the initialization message.
The default step rate for the PCjr is 13; which is the best rate for most 5.25 inch drives. If you want to make it some other value, use -xnn where nn is from 1 to 15. If you have a 3.5 inch drive that is noisy, using n = 14 may quiet it down; although it may make the 5.25 drive slightly noisier. You may need to experiment with different values for the best value for your combination of drives. If the step rate is set wrong for your drive it can become noisy and you may have read/write errors, so use this options carefully. Note, if you do not specify a step rate, the default value is used.

-y will install the jrSynchro cartridge if you have it in a slot. If you are not familiar with this cartridge. It replaces the keyboard and floppy disk interrupts, and allows the keyboard to be used while the disk drive is being accessed. It seems to work fine with DOS 2.1, but there are some problems with DOS 3.1.

It also will give you a 47 character buffer (DOS gives you 15 characters) if you install the buffer in the first 64K of RAM. Unfortunately, this is very restrictive if you want to force your programs into expansion RAM. This is the primary reason I install it in JRCONFIG. I allocate the buffer inside JRCONFIG, which is always in the first 64K. If you want the 47 character buffer, you can put JRSYN47 in your AUTOEXEC.BAT. I have used three versions of the cartridge, 1.2 and two marked 1.4.

Use -yn where n is:

```
    0       for version 1.2
    1 or 3 for version 1.4 with parity check
    2 or 4 for version 1.4 with no parity check
```

There are at least 2 versions of 1.4. The version I have used for n=1&2 came from PC Enterprises. The one for n=2&4 came from Dave Cox and was combined in the same cartridge with jrVideo.

If you have another version, try using one of the available options anyway. If you see JRSYNC in the initialization message that JRCONFIG prints, then JRCONFIG found the cartridge. If your computer locks up, you have an incompatible version.

Cartridge Basic compatibility:

Cartridge Basic is written to run in the low RAM of an un-expanded PCjr. If you have a copy of JBASIC.COM, you can run it to enter Cartridge Basic when using expansion RAM. It is a short program that sets up an environment for Cartridge Basic to run in. One of the things it does, is to get information from the ramdisk device driver about low memory and the video page. JRCONFIG will give it the information it needs to set up the proper environment. If you use JRCONFIG.NRD, then you can use LOWRD.SYS to get the necessary information (see below). It also checks for a hardware register that is on a MS Booster. If it can not find this register, it will abort. Below is a printout of a DEBUG session that will remove the check for the MS Booster, and change the drive that it expects the ramdisk to be. If JRCONFIG is drive C:, you can ignore the 2 lines that change the drive from C: to D:. If JRCONFIG is E: then change the 03 to 05. The forth to last line changes the name that DEBUG uses to DBASIC.COM, so that DEBUG will write a new file. I chose the name DBASIC.COM because this new modified program will work when JRCONFIG is D:. The 6797 at the beginning of most lines will probably be a different number for your DEBUG session.

```
A>debug jbasic.com                          start debug session
-u108b                                      un-assemble code
6797:108B BA3F02      MOV   DX,023F   hardware address on booster
6797:108E B091        MOV   AL,91
6797:1090 EE          OUT   DX,AL
6797:1091 BA3D02      MOV   DX,023D   hardware address on booster
6797:1094 B05A        MOV   AL,5A     write a byte to the register
6797:1096 EE          OUT   DX,AL
6797:1097 EC          IN    AL,DX     read a byte from the register
6797:1098 3C5A        CMP   AL,5A     compare read to written byte
6797:109A 750E        JNZ   10AA      jump if they are different
6797:109C BA070B      MOV   DX,0B07
6797:109F BB0300      MOV   BX,0003   drive C:
6797:10A2 B004        MOV   AL,04     read function
6797:10A4 B444        MOV   AH,44     I/O control
6797:10A6 CD21        INT   21        get information from ramdisk
6797:10A8 7306        JNB   10B0      jump if read is successful
-e1097
6797:1097  EC.90              replace read instruction with NOP
-e10a0                        examine ramdisk drive letter
6797:10A0  03.04              change from  C: to D:
-ndbasic.com                  change name of file
-w                            write new file
Writing 15AF bytes
-q                            quit and return to DOS
```

JRCONFIG.NRD (formerly called CONFIGJR.SYS) is the version that will not install a ramdisk. This version will not cause problems with hard disk drivers that insist on being drive C:. All the defaults are the same as they are in JRCONFIG.DSK. If you want your hard disk device driver to load into external memory, do not use the -s option. This will waste some of your internal RAM, but your hard disk driver will run faster. If you DO NOT want to waste any of your RAM, use -s1 to make all RAM usable. If you have one of the hard disks that have a ROM on the controller, you may still experience some increase in speed by sacrificing your internal memory, because all device drivers and COMMAND.COM will load into external memory. You should experiment to see if it makes a difference for your configuration.

LOWRD.SYS (LOW RamDisk) is a device driver that will access the ramdisk allocated by JRCONFIG.NRD. When JRCONFIG.NRD is installed, it installs the code to operate a ramdisk and allocates memory for it. But since it is a character device driver, it can not be operated as a ramdisk (a block device). LOWRD.SYS will access the ramdisk allocated in JRCONFIG.NRD, but the drive letter will be assigned to LOWRD.SYS.
I use INT 60 to pass information between the two device drivers. I am not aware of any other programs that use INT 60, but if you have one it could cause problems. LOWRD.SYS only uses INT 60 during initialization. If another program

installs INT 60 after LOWRD.SYS is installed, it should not cause a problem.
Just add the line

```
device=lowrd.sys
```

to your CONFIG.SYS. It can be anywhere in CONFIG.SYS except the first line, and will be assigned the next free drive letter by DOS.

EXAMPLE:

```
device=jrconfig.dsk -d2 -t3 -c
```

will:
- Tell DOS you have two floppy disk drives.
- Boot up in 80 column color mode.
- Turn on the keyboard click.

STATLIN.COM is a program that will display a status line on line 26 of your display. This program uses RAM that is in the second test page; therefore, you must not use the -v option in JRCONFIG with a value less than -v6.
On the left of this line will be the status of your Caps Lock, Num Lock, and Scroll Lock keys. A hyphen will be displayed after the key name if in the unlock position and an up arrow for the lock position. (I also put the status of the Insert key in STATLIN, but I removed it because some programs did not use the keyboard flag to get the status of the Insert key. COMMAND.COM is one of these programs.)
On the right of the screen will be date and time.
In the center will be the status for the ramdisk in JRCONFIG and the ramdisk RAM.SYS. When STATLIN is installed, it uses INT 60 to tell both ramdisks to display there spinning characters in the status line. If you have not activated this function in JRCONFIG with the -w option, it will not be displayed in the status line.

STATLIN will also blank the screen after a period of inactivity. There are 18.2 ticks of the timer every second (except during disk reads and writes). The default number of ticks is 4096 or slightly less than 4 minutes. Whenever the keyboard interrupt (INT 9) or the video interrupt (INT 10) is executed, the tick counter will be reset. If you run a program that writes directly to the screen, the number of ticks will not be reset. This can cause the screen to go blank even though something is being written to it. If the screen has been blanked, I suggest you use one of the shift keys to re-display it, because they are "safe" keys to touch.

CAUTION: There is an elusive bug in the screen blanker.

Sometimes, when the screen is blank and I press a key to reactivate it, the screen becomes all green (I have a green phosphorous monitor). The system is OK, except that the screen is solid green. If I wait until the screen blanks again, it will appear when I press a key. This happens infrequently and I have given up trying to fix it. I believe it may be a peculiarity of the jr's Video system, and may not be fixable. If I continue to type when the screen is green, everything will work, I just can't see what I'm typing.

options (-b -c -d -e -f)

-b will change the number of ticks to count before blanking the screen. Use -bnnn where nnn is from 33 to 65535.

-c the default color gives a shade of green that I like on my green phosphorous monochrome monitor. If you don't like this color, use -cnnn where nnn is the color you want. COLOR.COM will show you the color options available. If you make an error in the -c parameter, you will get the default color.

-d will disable the spinning character in RAM.SYS.

-e will disable the spinning character in JRCONFIG.

-f will speed up the repeat key function.
-f2 decreases the default delay between repeat keys by about one third.
-f1 decreases the default delay between repeat keys by about two thirds.
The status line will be disabled whenever a program is using a graphics mode or any text page other than page one. This is not true for the spinning characters displayed by the ramdisks. The -d and -e options will disable the spinning characters in the ramdisk regardless of how they were originally installed, because the device drivers are altered by INT 60. Although it is very clumsy, you can use TRS management programs such as INSTALL and REMOVE to remove STATLIN from memory and then re-install it with different options. If you do remove STATLIN, the number of display lines will not be reset to 25. However, the DOS command CLS will erase what is displayed on line 26.

I hope this is as useful to others as it has been to me. If anyone has any questions about this device driver, or if you think there is a bug, you can write me at the address at the beginning of this documentation, or you can leave me a message on the jr BBS 301-468-0984. This bulletin board is operated by the METRO PCjr USER GROUP.

I have included another program that I find useful. I get very annoyed at the DOS message to insert a disk with COMMAND.COM and press a key. I think that putting COMMAND.COM on all my disks a poor answer. I got this program from a bulletin board, and it has made my life much easier.

COMSPEC.COM - If you put the following 3 lines in your AUTOEXEC.BAT, you will always have COMMAND.COM on your ramdisk that is created with JRCONFIG.DSK.

1. set comspec=c:\command.com - this line uses the DOS 'set' command to tell DOS where to find COMMAND.COM; however, there is a bug in DOS and this doesn't work by itself.

2. comspec c:\command.com - this line executes the program COMSPEC.COM, which causes DOS to recognize the 'set' command

3. copy command.com c: - this line copies COMMAND.COM to the ramdisk

NOTE: The numbers at the beginning of the 3 lines do not go in AUTOEXEC.BAT they are only there for easier reading.If you have 3 or 4 floppy drives, then the ramdisk will be D: or E:

I modified RAM.SYS to have a spinning character in the upper right corner of the screen. It is 2 spaces to the right of the one for JRCONFIG.DSK and spins counter clockwise. For those of you who don't already have it, add the following line to your CONFIG.SYS

```
  device=ram.sys
```

This will install the device driver. You then run the program SETRAM.EXE.

```
  setram d n
```

Where d is the drive specifier and n is the number of K bytes to use for the ramdisk. The ramdisk will take the next available letter; therefore, if you have three floppy drives (A: B: C:), JRCONFIG.DSK will be drive D:, and RAM.SYS will be E:. You may want to add this line to your AUTOEXEC.BAT. You can have RAM.SYS installed more than once in CONFIG.SYS, each occurrence will take the next drive letter.

You can remove the ramdisk at any time by running

```
  setram d
```

This will free up the ram used by the ramdisk. NOTE: you will loose any files that were on it. You can then recreate the ramdisk at any time, because the device driver is still installed.

If you want to move the spinning character to the left, you can change the position using DEBUG.COM. You can also change the number of directory entries. The following is what a DEBUG session will look like. The hyphens are the DEBUG prompt and are not typed in.

```
  debug ram.sys        This starts the program.
  -e29b                If you want to move the spinning character.
  XXXX:029B   9E.nn    DEBUG responds  with everything  on this line
                       except the  nn. This will  be the  new offset
                       of the  spinning character. Subtract two from
                       hexidecimal 9E to move one space to the left;
                       ie, nn can be any even number from 0 to 9E.

  -e11f                If you want to change the number of directory
                       entries
  XXXX:011F   40.nn    Replace 40 (64 decimal)  with nn,  the number
                       of entries  you want.  I suggest you only use
                       even multiples of 10  (16  decimal).  80 (128
                       decimal) is probably as high as you will want
                       to use. Every 10  (16  decimal)  entries will
                       use 512 bytes of memory.
  -w                   This  will  cause  the modified program to be
                       written to the disk.
  Writing 01EA bytes   The message from DEBUG.
  -q                   Ends the debug session.
```

Several people have asked me why JRCONFIG takes up so much RAM. The answer is that they have not used the options that will give them the most free RAM. Both the .DSK and .NRD versions fill up the first 128K of RAM, unless the -s option is used. If you use -s1, no user RAM will be used. Even though a small ramdisk is allocated; it uses RAM that is below the display video RAM, and that RAM can not be used by DOS. If you are really tight for RAM, you can use the -v option. By using -v4 you will get 12K of RAM that is usually allocated to the video buffer. There is a potential danger when you use the -v option with anything less than 16. If you run a program that uses video RAM above the first text page, your system will probably lock up; because that is where COMMAND.COM will load.