WHITEPAPER SERIES

Processing CONFIG.SYS Commands Within ROM-DOS

The CONFIG.SYS file is the first file loaded and interpreted by the kernel during system boot. If the ROM-DOS boot diagnostics are enabled, the first portion of this processing happens after diagnostic "6".

For several reasons it is not practical to process CONFIG.SYS instructions in the order that they appear within the CONFIG.SYS file. Among these are:

- Menu processing
- The ability to load DOS and certain device drivers into High memory
- The NEWFILE command

The alternative to "line by line" processing is a multiple-pass system; some of this is documented in "DOS Internals" by Geoff Chappell (pp. 145-155).

In a multiple-pass system the F5, F8, and the SHIFT key affect CONFIG.SYS processing. For example, pressing F8 causes step-by-step confirmation making it possible to see what ROM-DOS is calling during PASS 2, PASS 3, and PASS 4 through the CONFIG.SYS.

It is possible to subdivide the CONFIG.SYS file into configuration blocks. Each block begins with a unique heading and extends up to but not including the next heading. Menus can be created using a [MENU] heading that will allow for users to choose which block to run during the startup process. The example at the end of this paper shows a CONFIG.SYS that uses blocks but does not use a menu and details how the system will process this CONFIG.SYS file and what is run during each pass. For a complete list of menu and CONFIG.SYS commands please refer to the ROM-DOS User's Guide.

PASS Processing Overview

In PASS 1, CONFIG.SYS is scanned for menus and blocks. If MENUs (and SUBMENUs) are present, they are displayed to the user as indicated. Commands related to menus are processed at this time, including MENUDEFAULT, MENUCOLOR, and NUMLOCK.

In PASS 2, 3, and 4, the only CONFIG.SYS commands that will be processed are located:

- Before any block definitions in the file
- Within a block whose name matches the selected menu item
- Within a block whose name is [COMMON]
- Within any block that is specified after an INCLUDE=

In order to ease future processing and handle INCLUDE variances, ROM-DOS reprocesses these commands, in order, to a new buffer. It is at this time that NEWFILE commands will insert the newfile into this buffer; please see the example section to walk through this process.

Next CONFIG.SYS Processing continues with PASS 2. The following commands are recognized during this pass:
- DOS
- STACKS

ROM-DOS then allocates stacks and high memory as appropriate, and continues with PASS 3. The following commands are recognized during this pass:

- DEVICE
- DEVICEHIGH
- COUNTRY
- BREAK
- BUFFERS
- FCBS

- FILES
- LASTDRIVE
- NUMLOCK
- SHELL
- SET
- SEARCHES

The specified BUFFERS, FILES, and FCBS (File Control Blocks) are all allocated into high or low memory, and other initializations take place. If boot diagnostics are enabled, ROM-DOS emits a diagnostic "7".

Finally, PASS 4 of CONFIG.SYS processing happens, and the following commands are recognized during this pass:
- INSTALL
- INSTALLHIGH

The memory used by CONFIG.SYS and the new buffer are discarded, and the memory used by the processing code and warning messages is also discarded. If boot diagnostics are enabled, ROM-DOS emits an "8".

At this time, final processing is done and eventually the AUTOEXEC.BAT file is processed.

Example CONFIG.SYS, Blocks Processing without [MENU]:

CONFIG.SYS	C:\CONFIG2.SYS	INTERNAL PROCESSING
-----------------	----------------------	----------------------------------
DEVICE=A	DEVICE=Z	*PASS 1*
[ONE]		BLOCKS ESTABLISHED
DEVICE=B		*PASS 2*
NEWFILE=C:\CONFIG2.SYS		DOS=HIGH
DEVICE=C		STACKS=9,256
[TWO]		*PASS 3*
DEVICE=D		DEVICE=A
[COMMON]		FILES=64
DOS=HIGH		BUFFERS=15,0
STACKS=9,256		DEVICE=E
FILES=64		DEVICE=B
BUFFERS=15,0		NEWFILE=C:\CONFIG2.SYS
DEVICE=E		DEVICE=Z
INCLUDE=ONE		DEVICE=F
DEVICE=F		

*During pass 3 the following devices are loaded within this order:
- A - first in file, outside any blocks
- E - COMMON is the first block processed
- B - the ONE block is included, and processed next
- Z - bring in the newfile (note- ROM-DOS does NOT process the DEVICE=C)
- F - back to the COMMON block
- D - not processed because TWO is not within an INCLUDE statement

There are other issues related to potential looping INCLUDE= or NEWFILE= blocks, but those are not within the scope of this document.