



Chapter 11 Utilities

Introduction

The PPT 41xx Software Development Kit includes several utilities which can be used by the software developer or system administrator to make the PPT 41xx terminal more useful. Some of these utilities have been mentioned earlier in this manual. This chapter covers each of these utilities in detail.

RIGHTSW.COM

This utility allows a program or batch file to determine whether or not the right side switch on the PPT 41xx terminal is being depressed. XSYMBIOS must be loaded in order for correct replies to be returned.

Batch File Code example:

```
RIGHTSW
IF ERRORLEVEL 1 GOTO right_Switch_Pressed
IF NOT ERRORLEVEL 1 GOTO right_Switch_Not_Pressed
```

An exit code of zero (0) is returned if the switch is not currently depressed. An exit code of (1) is returned if the switch is currently depressed. The DOS Batch File syntax IF ERRORLEVEL and IF NOT ERRORLEVEL can be used to check the exit code of the program.

It is often useful to loop on the results of this program to wait for one of the switches to be pressed or released.

LEFTSW.COM

This utility allows a program or batch file to determine whether or not the left side switch on the PPT 41xx terminal is being depressed. XSYMBIOS must be loaded in order for correct replies to be returned.

Batch File Code example:

```
LEFTSW
IF ERRORLEVEL 1 GOTO left_Switch_Pressed
IF NOT ERRORLEVEL 1 GOTO left_Switch_Not_Pressed
```

An exit code of zero (0) is returned if the switch is not currently depressed. An exit code of (1) is returned if the switch is currently depressed. The DOS Batch File syntax IF ERRORLEVEL and IF NOT ERRORLEVEL can be used to check the exit code of the program.

It is often useful to loop on the results of this program to wait for one of the switches to be pressed or released.

DSKACHK.COM

This utility allows a program or batch file to determine whether or not an SRAM card is present in DOS logical drive A (socket 1):

Batch File Code example:

```
DSKACHK
IF NOT ERRORLEVEL 1 GOTO Disk_In_A
IF ERRORLEVEL 1 GOTO No_Disk_In_A
```

An exit code of zero (0) is returned if the SRAM card can be successfully accessed. An exit code of one (1) is returned if the SRAM card cannot be successfully accessed. The DOS Batch File syntax IF ERRORLEVEL and IF NOT ERRORLEVEL can be used to check the exit code of this program.

This program should be used to check for the presence of an SRAM card in the socket before any attempt to access files on the SRAM card. Failure to check first may result in an error from DOS and an “Abort, Retry, or Fail” message which will wait for a key press.

Once the SRAM card has been validated as present, the files may be accessed, possibly further verified by the use of the DOS Batch File syntax IF EXIST or IF NOT EXIST.

DSKBCHK.COM

This utility allows a program or batch file to determine whether or not an SRAM card is present in DOS logical drive B (socket 2):

Batch File Code example:

```
DSKBCHK
IF NOT ERRORLEVEL 1 GOTO Disk_In_B
IF ERRORLEVEL 1 GOTO No_Disk_In_B
```

An exit code of zero (0) is returned if the SRAM card can be successfully accessed. An exit code of one (1) is returned if the SRAM card cannot be successfully accessed. The DOS Batch File syntax IF ERRORLEVEL and IF NOT ERRORLEVEL can be used to check the exit code of this program.

This program should be used to check for the presence of an SRAM card in the drive before any attempt to access files on the SRAM card. Failure to check first may result in an error from DOS and an “Abort, Retry, or Fail” message which will wait for a key press.

Once the SRAM card has been validated as present, the files may be accessed, possibly further verified by the use of the DOS Batch File syntax IF EXIST or IF NOT EXIST.

COM1ON.COM

This program applies power to the serial port (COM1) and sets the suspend timer to -1. The program should be executed before running any third party software that uses the serial port.

Failure to switch the serial port on before using third party software will prevent communications from working.

SLEEPOFF.COM

This program stops XSYMBIOS from placing the CPU into sleep state after 5 seconds of inactivity. (It does this by disabling sleep state for the CPU.) The program should be executed before running any program that handles it's own interrupts. (This usually means all communications programs.) Failure to disable CPU sleep state can result in a severe performance degradation, since the interrupts will not be seen as “activity” for the purposes of resetting the activity timers.

SUSPOFF.COM

This program prevents XSYMBIOS from suspending the system when no I/O activity is detected for the suspend time-out period. It should be executed before running any program, during which the system should not time out.

XSYMRST.COM

This program resets all XSYMBIOS power management parameters to the default conditions, and therefore undoes the effects of SUSPOFF, COM1ON, and SLEEPOFF programs.

Example

COM1ON	Enable power to serial port
SLEEPOFF	Prevent processor from sleeping
SUSPOFF	Prevent processor from timing out
PCPLUS	Execute third party program
XSYMRST	Restore power management parameters for XSYMBIOS

NOSYS.COM

NOSYS.COM is a utility that modifies the boot sector of a diskette (or SRAM card formatted as a diskette) to redirect the boot drive to drive C:. NOSYS.COM may be executed on a PC, or on a PPT 41xx terminal. The effects of NOSYS.COM can be undone by SYS.COM.

The NOSYS syntax, complete with command-line parameters, is as follows:

```
NOSYS <Drive:>
```

The following command-line parameter is required:

<Drive:> Drive to be redirected

If NOSYS.COM is executed on a PC, then <Drive:> can identify any diskette drive (or SRAM card emulating a diskette drive). If NOSYS is executed on a PPT 41xx terminal, then <Drive:> must be A: or B:.

The default boot sequence on a PPT 41xx terminal is A: C:, if diskette A is set to PCMCIA. If socket 1 is configured for DISKETTE A, and if drive A: contains an SRAM card that is not formatted as a system disk, then the terminal will not be able to boot. Using NOSYS.COM to modify the boot sector of the SRAM card allows the terminal to boot from drive C:, thus making it unnecessary to format the SRAM card as a system disk.

Note: Do not use NOSYS.COM to modify the boot sector of a diskette to be processed by DISKCFG, as this will prevent a boot from drive C: on the PPT 41xx terminal.

HIDOS.SYS XMS

HIDOS.SYS is an extended memory manager (XMM) that implements the LIM XMS 2.00 (Lotus-Intel-Microsoft Extended Memory Specification, Version 2.00). As such, HIDOS.SYS provides access to upper memory, high memory, and extended memory for programs that can run in those areas. It also ensures that no two programs can use the same part of high or extended memory or the same upper memory block (UMB) at the same time.

HIDOS.SYS must be installed on the terminal to provide access to memory above the 640K conventional memory threshold.

To install HIDOS.SYS, simply add a **device** command for HIDOS.SYS to the beginning of the CONFIG.SYS file. This **device** command must come *before* any **device** commands for programs or device drivers that use extended memory. If HIDOS.SYS is installed along with PCCEMM.SYS, the **device** command that loads it from CONFIG.SYS must come *before* the command that loads PCCEMM.SYS.

The following two lines added to the CONFIG.SYS file cause MS-DOS to be loaded into the high memory area (HMA) at boot time:

```
DEVICE=d:[path]\HIDOS.SYS
```

```
DOS=HIGH
```

where *d*: and *[path]* indicate the drive and directory in which the driver is located.

The application programming interface (API) functions associated with the extended memory manager are accessed through its control function. The address of the control function is determined via Interrupt 0x2F. An application program should first determine if the XMS driver has been installed. Next it should retrieve the address of the control function. Then it can invoke any of the XMS API functions that are available in the specification.

For a listing of the available functions and descriptions of what they do, refer to *eXtended Memory Specification (XMS)*, ver. 2.0, 1988, Microsoft Corporation, 16011 NE 36th Way, Box 97017, Redmond, WA 98073.

PCCEMM.SYS EMS

PCCEMM.SYS is the MS-DOS device driver which, in conjunction with the Chips and Technologies F8680 PC/CHIP, provides support for LIM EMS 4.0 (Lotus-Intel-Microsoft Expanded Memory Specification Version 4.0). This section describes the installation of PCCEMM.SYS, its **device** command line parameters, and associated error messages.

If PCCEMM.SYS is installed along with HIDOS.SYS, then the **device** command that loads it from CONFIG.SYS must come *after* the command that loads HIDOS.SYS.

PCCEMM.SYS is supplied as a device driver that can be loaded at boot time. To install it, the user must edit the CONFIG.SYS file so that it contains the following **device** command:

```
DEVICE=d:[path]\PCCEMM.SYS [options]
```

where *d:* and *[path]* indicate the drive and directory in which the driver is located.

PCCEMM.SYS has been designed so that, under normal circumstances, it does not need any command line options in order to provide satisfactory performance. However, command line options are provided so that the driver may be customized to the specific needs of the user.

The **device** command line parameters listed in Table 11-1 are available as *options* for users who want to tailor this driver to their own specifications.

Option names may be abbreviated to their first two characters. When multiple options are entered on the command line, they should be separated by one or more spaces.

Table 11-1. PCCEMM.SYS Parameter Options

Option	Description
HANGLES= <i>hhh</i>	Specifies the number of handles and names available. <i>hhh</i> must be in the range between 16 and 255. The default value is 64.
IOADDR= <i>aaa</i>	Overrides the default base input/output address at which the PC/CHIP should be configured. <i>aaa</i> can be 208, 218, 258, 268, 2A8, 2B8, or 2E8.
DIAGS= <i>xxx</i>	Specifies whether EMS memory diagnostics are performed during initialization of the driver. <i>xxx</i> must be ON or OFF. The default is OFF.

Table 11-1. PCCEMM.SYS Parameter Options (Continued)

Option	Description
SIZE=ssss	Specifies how much of the RAM above 1024K should be used for expanded (EMS) memory. If there is <i>no</i> extended memory (XMS) driver in the system, then the default for the EMS driver is to use as much memory as is available. If there is an XMS driver in the system, then the EMS driver defaults to using 256K for expanded memory. These defaults may be overridden by using this command line option, where <i>ssss</i> is specified in Kbytes and must be a multiple of 64K.

With the possible exception of **SIZE**, it is recommended that PCCEMM.SYS be installed *without* any options, i.e., with the defaults specified above. A typical configuration of the HIDOS.SYS and PCCEMM.SYS device drivers in the CONFIG.SYS file of a system with a total of 2 MBytes of memory might be:

DEVICE=HIDOS.SYS

DEVICE=PCCEMM.SYS SIZE=1024

DOS=HIGH

Table 11-2 lists the error messages that display when the system detects a problem while it is loading the PCCEMM driver.

Table 11-2. PCCEMM.SYS Loading Error Messages

Error Message	Explanation
"Initialization Error. Memory Manager not Installed"	This is a generic error message that is accompanied by another error message. It indicates that for some reason (given by the other error message) the EMM driver could not be installed correctly.
"Cannot detect mapping hardware"	The EMM driver could not find a F8680 PC/CHIP in the system. This error may indicate that the wrong driver has been installed for your system.
"Problems encountered with memory initialization"	The EMM driver could not set up the page frame correctly. Make sure that there is a contiguous 64K segment of unused memory space at D000.
"Interrupt 67 Vector already allocated"	Some other driver has already allocated the EMS interrupt vector. Remove all other EMS drivers from your CONFIG.SYS file.

Table 11-2. PCCEMM.SYS Loading Error Messages (Continued)

Error Message	Explanation
"I/O address specified is not one of the permitted values"	The EMS mapper Input/Output address you indicated using the IOADDR command line option is not a valid address. Make sure that it is 208.
"An equals (=) sign was expected"	One of your PCCEMM.SYS command line options was not immediately followed by an equals '=' sign. Check your CONFIG.SYS file to make sure the device command line installing PCCEMM.SYS has been formatted correctly.
"Number of handles invalid or out of range"	The value you specified for the HANDLES command line option is not valid. Make sure it is a decimal numeral between 16 and 255.
"Unrecognized command/option in command line"	The string displayed after this error message is not a valid command/option name. Edit the CONFIG.SYS file to make that all option names used on the device command line are valid (refer to the list of options given above).
"The command line is incomplete"	One or more of the options specified on the PCCEMM.SYS device command line is incomplete. Edit the CONFIG.SYS file and verify that all command line options are formatted correctly.
"Problems encountered in setting default page frame address"	The default page frame address (D000) is not valid. Make sure that no other driver is using this area of memory.
"Decimal number input overflow error"	The number of handles specified is not a valid number. Edit the CONFIG.SYS file and make sure that the parameter for the HANDLES option is formatted correctly.
"The EMS hardware is disabled or not functional"	This is a generic error message that is accompanied by another error message. It indicates that for some reason (given by the other error message) the EMS hardware is not operating properly.

The application programming interface (API) supported by EMS is accessed through Interrupt 0x67. An application should first determine that the EMS driver (i.e., PCCEMM.SYS) has been installed. Next, it should execute Interrupt 0x67. Then it can invoke the desired EMS function by specifying the appropriate function and subfunction (if required) codes in registers AH and AL, respectively.

For a listing of EMS API functions and descriptions of what they do, refer to *Expanded Memory Specification, ver. 4.0*, 1987, Microsoft Corporation, 16011 NE 36th Way, Box 97017, Redmond, WA 98073.

Installation Alternatives and Limitations

PCCEMM.SYS can be installed with or without HIDOS.SYS. Certain operating conditions apply depending on which alternative you choose. These conditions are described in the following:

1. When PCCEMM.SYS is installed *without* HIDOS.SYS:
 - all memory above 1024K is allocated as EMS memory by default.
 - the **SIZE** option can be used to reduce the EMS memory allocated but the remaining memory cannot be allocated to other purposes.
 - specification of a **SIZE** greater than the maximum available memory simply allocates the maximum available memory as EMS memory; it does *not* cause an error message to be displayed.
2. When PCCEMM.SYS is installed *with* HIDOS.SYS:
 - the default **SIZE** of EMS memory is 256K
 - the **SIZE=xxxx** option parameter can be used to allocate additional EMS memory up to all available memory above 1024K minus the 64K of high memory.
 - specification of a **SIZE** greater than the maximum available memory simply allocates the maximum available memory as EMS memory; it does *not* cause an error message to be displayed.
 - if the **SIZE** option is used to specify less EMS than the maximum available, the remaining XMS memory is *not* available for other purposes.
3. DOS=HIGH can be installed with HIDOS.SYS to load MS-DOS 5.0 in high memory.