Chapter
10 *Printing*

This chapter examines technical issues related to printing in Microsoft Windows 3.1. This chapter provides a technical supplement to the basic Windows documentation.

Also see the PRINTERS.WRI and NETWORKS.WRI files in your WINDOWS directory.

· *Windows Resource Kit:* Chapter 2, "The Windows Setup Information Files"; Chapter 3, "The Windows Files"

· *Glossary terms:* BIOS, buffers, interrupts, spooler

Contents of this chapter

Flowchart Series 4
Printing Problems

To print a document or other data in a Microsoft Windows application, you must have a printer properly connected to a serial or parallel port (for a stand-alone PC) or properly connected to the network. You must also have a Windows printer driver file (.DRV) installed and configured for your printer.

About Windows Printing

Most printer drivers are installed and configured automatically during Setup based on your choices about the type of printer in your system. If the Printer Setup list does not contain your printer, use the following decision tree to find how to set up unsupported printers.

Unsupported Printer Compatibility Decision Tree

·   Is it a laser printer?

HPPCL (LaserJet) compatible?
®**Set up as a Hewlett-Packard LaserJet Plus.**

PostScript compatible?
    Color PostScript?
    ®**Set up as a QMS-ColorScript.**
    35-font Plus font set or superset thereof?
    ®**Set up as an Apple LaserWriter Plus.**

·   Is it a dot matrix printer?

9-pin dot matrix?
    IBM compatible?
    ®**Set up as IBM Proprinter.**
    Epson compatible?
    ®**Set up as Epson FX-80 for narrow,
        or FX-100 for wide carriage.**

24-pin dot matrix?
    Compatible with Epson or IBM 24-pin printers?
            IBM 24-pin compatible?
            ®**Set up as IBM Proprinter X24.**
            Epson LQ compatible?
            ®**Set up as Epson LQ-1500.**

·   Is it a plotter?

HPGL compatible?

®**Set up as Hewlett-Packard 7475A.**

·   Other?

®**Use Generic/Text Only/Daisywheel driver.**

Contact the manufacturer to see if custom drivers are available.

Figure 10.1

Printers dialog box

To display this dialog,
in Control Panel click:

Figure 10.2

Printer Setup dialog box
for HP LaserJet Series II and compatible printers

To display this dialog,
click Setup in the
Printer Setup dialog

For details about installing and configuring a printer, see
Chapter 5, "Control Panel," in the *Windows User's Guide.* Or
choose the Help button in the Printer Setup dialog box.

Printer Ports

Printers in stand-alone PC systems are usually connected either to parallel or serial ports. This section examines technical details related to LPT and COM ports for stand-alone systems.

A special case, the EPT port, is described in "WIN.INI [Ports] Settings and the LPT*x*.DOS Method" later in this chapter.

Printing from Parallel Ports

Standard IBM-compatible systems can have up to three parallel printer ports, commonly known as LPT ports. As the name suggests, parallel ports transmit data a full 8-bit character at a time. Data is transmitted to LPT ports through system I/O ports, which are individual serial data lines to the processor. Because LPT ports transmit a byte at a time, they use eight I/O ports in parallel.

The standard for the pin connections on parallel ports was developed by Centronics. Parallel port handshaking is done with hardware signals on the Centronics standard system. These signals include:

·   **Select In**, which means that the printer is online.
·   **Busy**, which means the printer's buffer is full or the printer is occupied.
·   **Acknowledge**, which is pulsed when the printer accepts a character.
·   **Paper Out**, which means the printer is out of paper or jammed.

An application can access LPT ports in three ways:

1. Non-Windows applications usually use the MS-DOS device names **LPT1:**, **LPT2:**, and **LPT3:** with standard MS-DOS Interrupt 21 file-access services. Printing with MS-DOS device names is both the slowest and most compatible printing method.

2. Applications can access the printer through the ROM BIOS

Interrupt 17 printer-port services, which sends one character to the printer at a time (an inefficient process). When you access a printer at MS-DOS level, MS-DOS eventually performs LPT port output through Interrupt 17.

3. LPT ports can be addressed directly, which requires 100 percent hardware compatibility, but is the fastest by far of the three methods.

The following table lists the default IBM-compatible I/O port starting locations and hardware interrupts for parallel ports.

Port	PC-AT default	PS/2 default	Interrupt	
LPT1	378h		3BCh	7
LPT2	278h		378h	5
LPT3	3BCh		278h	7

Specify a parallel port connection for your printer by choosing the Connect button in the Printer Setup dialog box.

Figure 10.3

Connect dialog box

Printing from Serial Ports

Serial ports, also known as COM ports, transmit data in a 1-bit-wide stream. Windows 3.1 accesses COM ports directly. The following table lists the default IBM-compatible I/O port locations and hardware interrupts for serial ports.

Port	Default	Interrupt
COM1	3F8h	4
COM2	2F8h	3
COM3	2E8h	4
COM4	2E0h	3

W

For 386 enhanced mode, you can set nonstandard values by choosing the Ports icon in the Control Panel, and clicking the Advanced button in the Settings dialog box for the selected COM port. (This also changes the entries for **COM*x*BASE=** and **COM*x*IRQ=** in the **[386enh]** section of SYSTEM.INI.) However, we strongly advise you to change the hardware to fit the Windows defaults rather than changing the Windows defaults to fit the hardware.

Figure 10.4

COM Port Settings
dialog box

To display this dialog,
in Control Panel, click

then click Settings

Click the Advanced
button to display this
Advanced COM Port Settings dialog box

In 386 enhanced mode, COM ports are buffered for non-Windows applications, but buffering is not necessary for Windows itself. Buffer size can be set only by changing the **COM*x*Buffer=** setting in the **[386enh]** section in SYSTEM.INI.

The key to proper serial printing is correct communications settings. Serial communication requires handshaking, which defines a method for a printer to tell Windows that its buffer is full. Serial ports can be configured for No Handshaking, XON/XOFF (software) Handshaking, or Hardware Handshaking by choosing the Ports icon in the Windows Control Panel. The settings are

saved in the **[ports]** section of WIN.INI, as described in the next section.

Most serial print devices use hardware handshaking, although some can be configured for XON/XOFF. We recommend that you do not use the No Handshaking setting for a serial printer. Your printer's documentation should tell you what communications settings to use. Usually, the setting is 9600 baud, No parity, 8 bits, 1 stop bit, and Hardware Handshaking.

WIN.INI [Ports] Settings and the LPTx.DOS Method

W The values that can be listed in the **[ports]** section of WIN.INI are described in Chapter 4, "The Windows Initialization Files." The default entries are:

[ports]

```
LPT1:=
LPT2:=
LPT3:=
COM1:=9600,n,8,1
COM2:=9600,n,8,1
COM3:=9600,n,8,1
COM4:=9600,n,8,1
EPT:=
FILE:=
```

Windows only recognizes the entries **LPT1:=** through **LPT3:=**, and **COM1:=** through **COM4:=** as hardware port entries. Windows performs tests to determine how to print to the LPT ports (see "Determining Print Method" later in this chapter). The COM ports are always accessed directly. The **FILE:=** entry is specially recognized by Windows. When you connect a printer using **FILE:=**, Windows will prompt you for a filename for the print job.

Any valid MS-DOS device name or filename can be placed in the **[ports]** section. Windows prints to any entry in the **[ports]** section (other than the **LPT***x***:=** and **COM***x***:=** entries) using standard MS-DOS services. The **EPT:=** entry is an example of this. EPT ports require a special card and special software and are only used by certain printers, such as the IBM Personal Pageprinter. Because the **EPT:=** entry has no special meaning to Windows, Windows simply outputs to EPT as a standard file or device through MS-DOS. If you have printer drivers that are recognized at the MS-DOS level by a device name, you can place them in the **[ports]** section and print to the device.

The **LPT***x***:=** entries are also valid MS-DOS device names. But if they have entries in the **[ports]** section in the standard format, Windows does not treat them as MS-DOS device names. However, you can force Windows to print to LPT ports through MS-DOS by changing the format of their entries in the **[ports]** section.

To make Windows stop recognizing a port as a hardware port, give

it a three-letter filename extension instead of a colon in the related LPT entry in the **[ports]** section of WIN.INI. Windows will print to the port as an MS-DOS device instead. For example, you can change the entry **LPT1:=** to **LPT1.DOS=** to accomplish this. Any three-letter extension will do, but the letters "DOS" are used as an example in the *Windows Resource Kit*. If you upgraded from Windows 3.0, this entry might appear in your WIN.INI file as "LPT*x*.OS2."

You might think this type of entry would print to a file called LPT1.DOS, which is exactly what happens as far as Windows is concerned. Windows opens the file/device LPT1.DOS at MS-DOS level and sends printer output to it. At the MS-DOS level, any name beginning with the letters "LPT1" is a synonym for the LPT1 device. So MS-DOS redirects the output to the printer port. Filenames and device names are dealt with synonymously in MS-DOS. The LPT*x*.DOS method is a helpful solution when you are using borderline-compatible network software or parallel port hardware.

Port Entries in the ROM BIOS Data Segment

Problems can occur if your machine does not recognize a specified COM port. This generates the message "The COM*x* port is currently assigned to a DOS application. Do you want to reassign the port to Windows?"
Information on the I/O-port starting address for each of the installed serial and parallel ports is contained in the ROM BIOS data segment at memory location 0040:0000. Each entry is a 16-bit port address stored in two hexadecimal bytes. As with all 16-bit addresses, each pair of bytes should be reversed to read correctly. For example, the entry **F8 03** yields an address of 3F8 hexadecimal (the standard starting address for COM1).

To ensure that your machine recognizes a COM port, you can examine specific ranges of the ROM BIOS data segment with the MS-DOS Debug command to find correct values for the COM port. Then you can write a batch file script or define SYSTEM.INI entries to correct the problem.

Examining the ROM BIOS Data Segment

The following steps show how to determine if the machine is recognizing the existence of the COM port.

To use the MS-DOS Debug command to examine the BIOS table:

1. At the command prompt (outside of Windows), type **debug** and press ENTER.

    This will return a hyphen prompt (–).

2. At the hyphen prompt, type **d40:0** and press ENTER.

    A listing of the BIOS data appears, and the hyphen prompt returns, where you can type **Q** to quit Debug.

Look at the table of data on the screen. The first row is the important one. It will show a line like the following:

0040:0000   F8 03 F8 02 E8 03 00 00–BC 03 78 03 00 00 00 00

Everything to the left of the center hyphen are COM ports; everything to the right are LPT ports. This example shows that the computer has COM1 at address 03F8 (listed in reverse byte order), COM2 at 02F8, and COM3 at 03E8. COM4 is not found, displaying 0000 to the left of the hyphen.

If the machine recognizes the desired COM port (identified by the 0000 entries), skip to "SYSTEM.INI Modifications for Port Entries in ROM BIOS" later in this chapter.

Batch File for Port Entries in ROM BIOS

If the machine does not recognize the COM port (identified by the 0000 entries), make a script for Debug to execute in AUTOEXEC.BAT. The following examples for COM ports show this. The second line of the examples should be edited to match the hardware used in your computer. Although the examples given are the most popular, there are exceptions to these.

To create a script for debugging a COM port:

1. Create a batch file, replacing the values shown in the second line of the following table with the values that appear on your screen:

COM1	COM2	COM3	COM4
E40:0	E40:2	E40:4	E40:6
F8 03	F8 02	E8 03	E8 02
q (ENTER)	q (ENTER)	q (ENTER)	q (ENTER)

2. Give the file a name such as C:\FIXCOM3.DEB or C:\FIXCOM4.DEB, depending on which port the file is for.

3. Add a line to the end of AUTOEXEC.BAT such as the following, for COM3:

   debug < fixcom3.deb > nul

4. Save the AUTOEXEC.BAT and reboot the machine.

The **>** **nul** ending added to the line in AUTOEXEC.BAT keeps the script from being echoed back to the screen. You can use Debug again to examine d40:0 to see if the change has taken effect.

SYSTEM.INI Modifications for Port Entries in ROM BIOS

If the port is recognized by the machine, note the address listed when using Debug. Then edit the SYSTEM.INI **COM*x*Base=** lines to match the base I/O address used by the hardware. The examples shown above for COM1 and COM2 will not require any changes to the SYSTEM.INI file. However, the examples below for COM3 and COM4 require the following entries to the **[386enh]** section of the SYSTEM.INI file.

Default	*Value for COMxBase=*
3F8h	**COM1Base=3F8h**
2F8h	**COM2Base=2F8h**
2E8h	**COM3Base=3E8h**
2E0h	**COM4Base=2E8h**

If you are making adjustments in order to use COM4, you might need to add both the **COM3Base=** and **COM4Base=** lines to SYSTEM.INI; otherwise, **COM4Base=** will be set to the same as the **COM3Base=** default.

If you are using a serial mouse on COM1 and you have an internal modem on COM3, this configuration won't work, because the mouse is tying up the interrupt that you want to be shared by COM3. This behavior also occurs if you use a serial mouse on COM2 and want to use an internal modem on COM4.

Programs that Change the ROM BIOS Data Segment

Port entries in the BIOS data segment can be changed by network redirector software or by the MS-DOS **mode** command. The **mode** command does not change any existing nonzero entries. It places a **1** in port entries that didn't previously exist but that have been created by **mode** redirection. For example, if you have a physical LPT1, but not an LPT3, and you execute the **mode LPT3=LPT1** command, then output sent to LPT3 will be redirected to LPT1. Because LPT3 does not exist in hardware, the BIOS data segment area contains zeroes in the LPT3 section. When **mode** creates a virtual LPT3 through its redirection, it places a **1** in the LPT3 section of the data segment.

MS-Net and LAN Manager redirector-based networks operate in a similar manner. The redirector does not modify any nonzero entries in the data segment when it performs redirection. When you redirect an existing port, the port entry is not changed. This is done so that when the redirection is removed, you can again print to the physical port. However, when you redirect a nonexistent port, the zero entry in its section is changed to a **2**. Many applications check the BIOS data segment to determine whether a port exists. Both **mode** and the MS-Net redirector place nonzero entries in this field as necessary, so the application will function correctly.

Novell networks place a 3BCh entry in the data segment locations for LPT2 and LPT3 when the NetWare shell is loaded. If you do not have a physical LPT1 in the machine, you will have a zero in the port entry for LPT1. This will not cause a problem unless you are running on a Novell network without the Windows Novell Network driver loaded.

Determining Print Method

Windows uses different methods to print depending on the types of parallel ports it detects in your machine. Windows will either access the port directly, or will print by opening the port as an MS-DOS device (with MS-DOS Interrupt 21 services). Windows 3.1 never uses the ROM BIOS Interrupt 17 services for printing.

The following decision tree shows how Windows determines which method to use for printing. This sequence applies only to ports LPT1 through LPT3. Windows always directly accesses the COM ports.

·   If a Windows network driver is present:

Query network driver to see if printer port is network redirected.
>   If it is a network printer:
>   ®**Tag as Network, go to end.**

·   Examine ROM BIOS data segment at 0040:0008:

If port entry is zero:
®**Do not allow printing through this port.**

If port entry is not zero:
>   If port being examined is LPT2 or LPT3:
>   >   If the port entry is the same as any previous entries:
>   >   ®**Assume port is redirected and access through**

MS-DOS.

>   If port entry is greater than 100h, port exists:
>   ®**Directly access this port.**

>   If port entry is less than 100h, port exists but is nonstandard:
>   ®**Access this port through MS-DOS.**

·   Repeat for additional ports.

<table>
<tr><td>

Flowchart 4.2
Connect Print on
on Novell Network

</td><td>

Windows itself does not identify network printers. Instead, you load Windows proper with its network driver. To make this determination... Windows Setup, then Control Panel, look at redirected printers with whatever internal function. "Managing Network Printing" in the *Windows User's Guide*. For example, the MS-Net driver uses the MS-DOS Interrupt 21 function 5F to detect network printer redirection.

</td></tr>
</table>

Printing with a Network Driver

If the network does not place a nonzero value in a previously nonexistent port entry when it is redirected, Windows will not recognize it as a valid port. Instead, redirect an existing port, and also use the LPT*x*.DOS method as described in "WIN.INI [Ports] Settings and the LPT*x*.DOS Method" earlier in this chapter. The long-term solution in this case is for the network redirector to properly place a nonzero value in the port entry upon redirection.

Also, if the port entry for a redirected port is greater than 100h, Windows will attempt to directly access the port. Directly accessing a redirected port will not work. The method to use in this case is the LPT*x*.DOS method discussed above.

For up-to-date details about printing with a network driver, see the NETWORKS.WRI file in the WINDOWS directory.