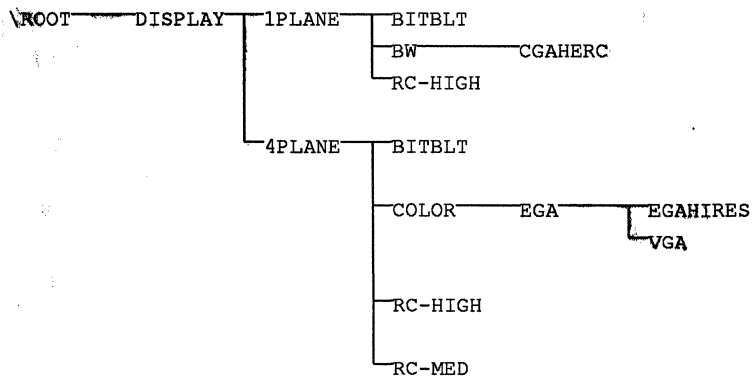
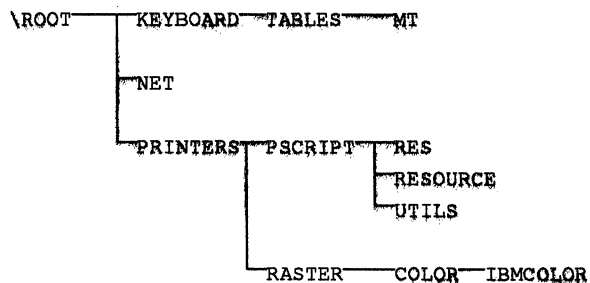


DDK

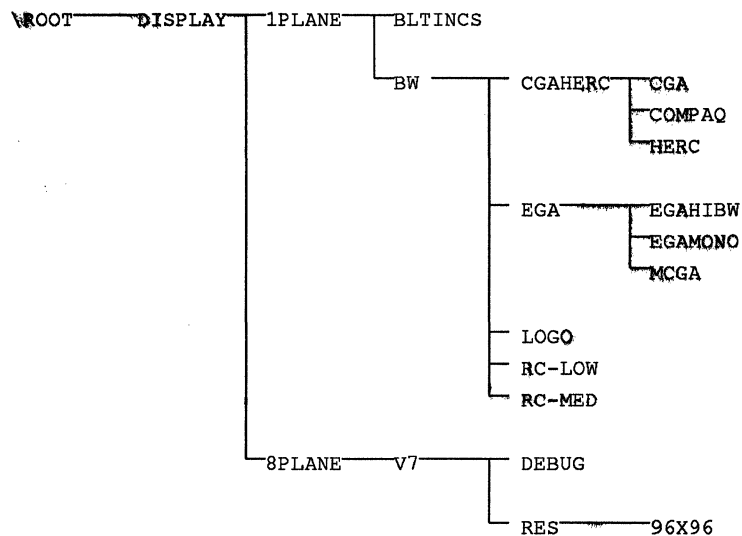
1/8



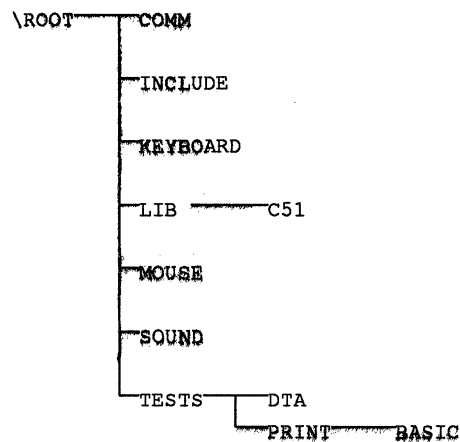
5



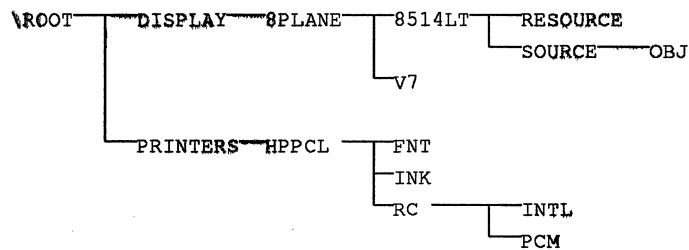
2



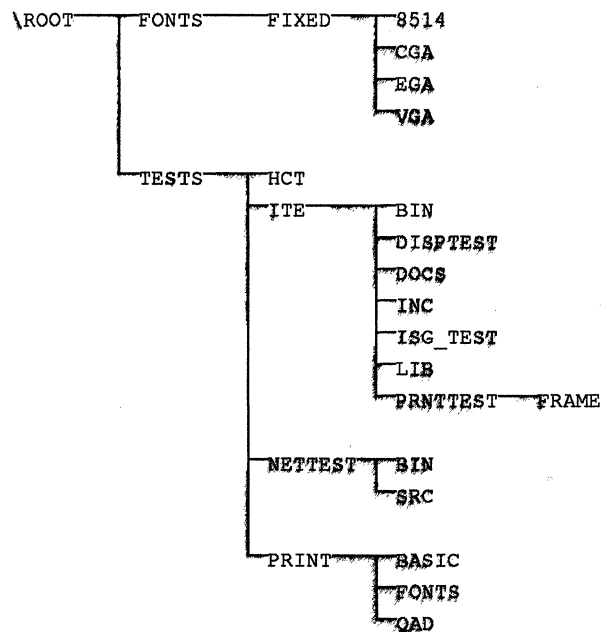
6



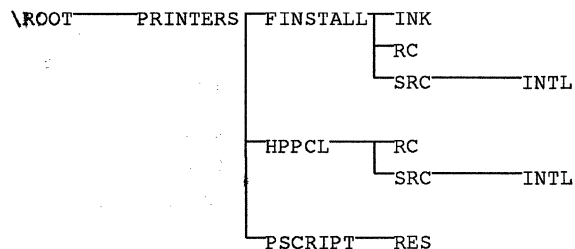
3



7



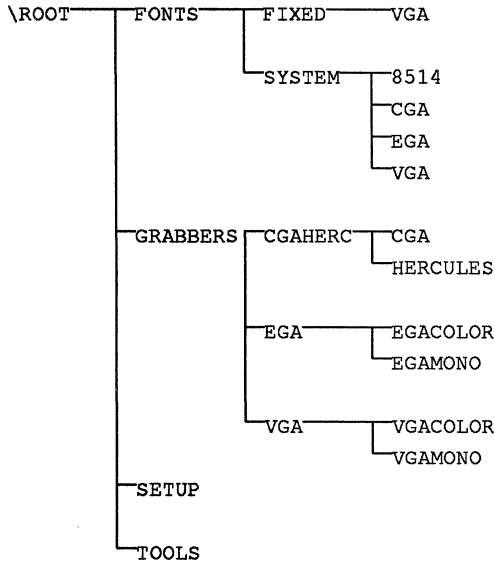
4



DEVICE DRIVER

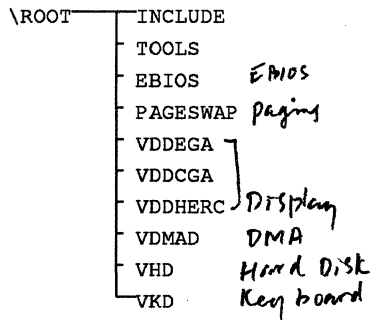
DDK

8/8

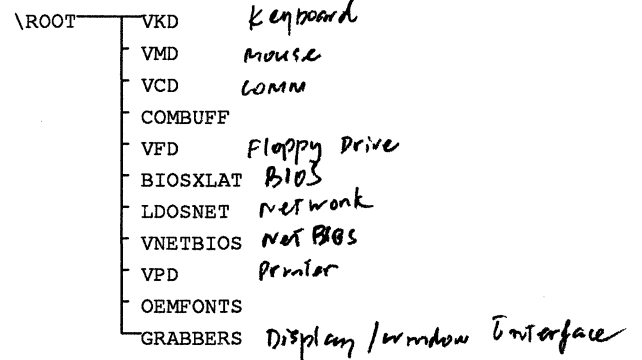


VIRTUAL DEVICE

1/1



2/2



**Microsoft Windows 3.0 Device Driver Kit (DDK)
Final Pre-Release Version
Release Notes**

Important Note:

As you may have noticed the DDK is now comprised of only 10 5.25" diskettes. This is because we are now providing the files in a compressed format to conserve disk space. You will need to expand the files to their normal size before using them. To do so, copy the four utilities (TREEEX2.BAT, TREEEX3.BAT, TREEEXP.BAT, WALK.EXE) that are located in the root directory on the first diskette over to your hard disk. Use the DOS command XCOPY to copy over the desired directories on the diskettes to your hard disk. In the top level directory on the hard disk where the driver sources were copied, type the command "TREEEXP". This utility will use the other utilities and decompress all of the files in the current directory as well as all subdirectories below it. Ensure that you will have sufficient free disk space on your hard disk prior to decompressing the files.

**May 1, 1990
Microsoft Confidential**

Table of Contents

Section 1 -- Overview of the DDK Pre-Release	4
1.1 Purpose of This Release	4
1.2 Changes Since the 1.63 DDK Pre-Release	4
1.3 DDK Pre-Release Contents	5
1.4 Status of the Software and Documentation	6
1.5 Using Microsoft OnLine with This Pre-Release	6
1.6 Role of the Windows 3.0 SDK to the Windows 3.0 DDK	6
1.7 Integrating Final Driver Changes into Your 3.0 Driver	7
Section 2 -- Installation of the DDK Pre-Release Software	7
2.1 What Else is Required?	8
2.2 Notes About the Win3/C5.1 Compatible Libraries	8
2.3 Notes About LINK4 for Windows Device Drivers	9
2.4 Libraries for Windows Drivers and 386 Enhanced Mode Grabbers	9
2.5 Updated Include Files	9
2.6 Special Notes on Building 386 Virtual Devices	9
2.6.1 386 Virtual Devices Special Tools	10
Section 3 -- Preparing Windows 2.x Drivers to Run Under Windows 3.0	10
3.1 What You Need To Do	10
3.2 Where to Look in the DDK Pre-Release	11
Section 4 -- Preparing Windows/386 2.x Virtual Devices to Run Under Windows 3.0 386 Enhanced Mode	12
Section 5 -- Debugging in this Pre-release	13
Section 6 -- Description of Windows Driver sources	13
6.1 Display Driver Sources	13
6.1.1 1 Plane Drivers - CGA, EGA monochrome, EGA high resolution B&W, Hercules, Plasma, and MCGA	14
6.1.2 4 Plane Drivers - EGA and VGA	14
6.1.3 8 Plane Drivers - Video Seven VRAM VGA and 8514/a	15
6.1.4 8514/a Resources	15
6.1.5 Windows Real and Standard Mode Grabbers	15
6.1.6 System Font Sources	16
6.2 Display Driver Test Application	17
6.3 Printer Driver Sources	18
6.3.1 PCL/HP LaserJet Printer driver version 3.3	18
6.3.2 PostScript Printer driver version 3.3	18
6.3.3 Raster Driver Sample Sources	19
6.4 Printer Driver Test Application	20
6.5 Keyboard Driver Sources	20
6.6 Mouse Driver Sources	20
6.7 COMM Driver Sources	20

6.8 Sound Driver Sources	21
6.9 Network Driver Sources and Test Application	21
Section 7 -- Description of the Virtual Device sources	21
7.1 The Display Devices (CGA, EGA, VGA, 8514, HERC)	22
7.2 The Display/Windows Interface (GRABBER)	23
7.3 The DMA Device (VDMAD)	23
7.4 The EBIOS Device (EBIOS)	24
7.5 The Hard Disk Device (VHD)	24
7.6 The Keyboard Device (VKD)	24
7.7 The Mouse Device (VMD)	24
7.8 The Netbios Device (VNETBIOS)	25
7.9 The Network Device (DOSNET)	25
7.10 The BIOS Device (BIOSXLAT)	25
7.11 The Paging Device (PAGESWAP)	25
7.12 The Printer Device (VPD)	25
7.13 The COMM Device (VCD)	26
7.14 The Floppy Drive Device (VFD)	26

Section 1 -- Overview of the DDK Pre-Release

This kit provides the final pre-release of the Windows 3.0 DDK documentation and sample sources. With this information, you can complete development for the features you may want to support in your Windows drivers and/or 386 Virtual Devices under Windows 3.0

Along with this DDK release, you should have received the Windows 3.0 SDK Final Pre-Release. We recommend you review the SDK Pre-Release notes, and install the software in that kit before installing the software in this kit.

The SDK Pre-Release kit contains the retail software for Windows 3.0, including support for Real, Standard, and 386 Enhanced modes. It is important that you read the SDK release notes on how to install the retail software. The SDK also contains some of the Windows tools you will need to build your driver. See the SDK release notes for more information on its contents and for installing the SDK software.

This Pre-Release is provided to you under a Non-Disclosure Agreement. All software and release note information provided in all DDK Pre-Releases is Microsoft confidential.

1.1 Purpose of This Release

There are two objectives of this Pre-Release:

- Provide you the tools and updated software to test, debug and modify existing 2.x drivers to work properly under Windows 3.0.
- Provide you the updated tools and sources to begin development of a 3.0 driver, that takes advantage of the new features provided only in Windows 3.0.

The software in the SDK and DDK pre-release kits is much more stable than versions release previously. We feel you should be able to develop, compile, link, test and debug a device driver or 386 virtual device for Windows 3.0.

1.2 Changes Since the 1.63 (RC2) DDK Pre-Release

This section will help you to better understand the differences between the previous DDK Pre-Releases and this final Pre-Release. If you were working with a prior release, you should read this section. Also, you should review this same topic in the SDK Pre-Release notes since it will contain additional information you will need.

Some of the specific differences include:

- Both updated software diskettes and documentation have been provided in this release.
- Sources for Windows device drivers and 386 Virtual Devices have been updated to reflect the changes since the 1.63 release. See the next section "DDK Pre-Release Contents" for a complete list.
- A Hardware Compatibility Test has been added.

- The Windows version of the C5.1 run-time libraries. With the 1.63 release, the SDK now provides the new Windows version of the C run-time libraries which are compatible with both the C6.0 and C5.1 compilers. Device drivers created with these new libraries will not binary compare with those that are available in the Windows 3.0 retail product. As a result, the DDK includes the Windows version of the C5.1 run-time libraries that is only compatible with the C5.1 compiler and which were used in creating the drivers in the Windows 3.0 retail product.

1.3 DDK Pre-Release Contents

In summary, you should have received the following:

- The Windows 3.0 SDK Pre-Release kit. It contains the Windows 3.0 retail software and SDK tools you need in addition to the DDK pre-release. See the SDK Pre-Release Notes for details on its contents and steps you need to take to install the necessary Windows 3.0 software.
- These Pre-Release Notes, which point out how to use the documentation and software available in this kit.

Windows Driver and Font sources for:

1 plane (2 color) display drivers for CGA, Hercules, Compaq Plasma, IBM MCGA, EGA high resolution black and white, and EGA monochrome displays.

4 plane mode (16 colors) display drivers for EGA high resolution and VGA displays.

8 plane mode (256 colors) display driver for the Video Seven VRAM/FastWrite VGA display. and 8514/a.

Printer drivers for PCL / HP LaserJet driver version 3.3 (HPPCL), PostScript driver version 3.3 (PSCRIPT), and Epson and IBM Color Printer driver sample sources.

IBM AT style keyboard driver and international keyboard tables.

Fonts for CGA, EGA, VGA, and 8514/a resolution Windows 3.0 proportional system and Windows 2.x fixed pitch system fonts.

- 386 Enhanced Mode Virtual Device sources for:

CGA, Hercules, EGA, VGA, and 8514/a Display (VDD)

CGA, Hercules, EGA, VGA, and 8514/a display/Windows Interface (GRABBER)

Mouse pointing device (VMD)

Keyboard device (VKD)

Hard disk device (VHD)

COMM port device (VCD)

DMA device (VDMAD)

Printer port device (VPD)

Floppy drive device (VFD)

MS-NET network NETBIOS transport layer interface (VNETBIOS)

PS/2 EBIOS Page (EBIOS)

Memory Page Swapping (PAGESWAP)

ROM BIOS device (BIOSXLAT)

- Utilities and Building Tools:

PFM Editor for PFM and PCM fonts (PFMEDIT.EXE)
PostScript resource tool (MKPRN.EXE)
A special version of the C Linker (LINK4.EXE)
Masm 5.10B (flat model) for 386 virtual devices (MASM5.EXE)
Link386 for 386 virtual devices (LINK386.EXE)
Special Add Header utility for 386 virtual devices (ADDHDR.EXE)
Special Mapsym for 386 virtual devices (MAPSYM32.EXE)

There are a total of 10 enclosed software diskettes in 5.25" 1.2 MB format.

More detailed information on the Tools mentioned above can be found in Sections 2, 6 and 7. Detailed descriptions of the software sources mentioned above are documented in Sections 6 and 7 of these release notes.

1.4 Status of the Software and Documentation

The sources provided in this final pre-release are the same as those used in the final Windows 3.0 retail release. The documentation provided has been reviewed for technical accuracy and any differences from the final product will be only minor editing changes.

For information on reporting bugs in the software or documentation, please see the SDK Pre-Release notes section 1.6 "Correspondence Via Microsoft OnLine".

1.5 Using Microsoft OnLine with This Pre-Release

In the SDK Pre-Release notes, there are steps for connecting to the Microsoft OnLine system to get additional information throughout the pre-release program and for reporting bugs you may find in the software. See Section 1.6 "Correspondence Via Microsoft OnLine" in the SDK Pre-Release notes for more information.

Note, you should review your Microsoft OnLine documentation first if you are using it for the first time.

1.6 Role of the Windows 3.0 SDK to the Windows 3.0 DDK

With Windows 3.0, tools and documentation in the Windows 3.0 SDK are required for developing drivers with the Windows 3.0 DDK. There are several references to the SDK documentation throughout the DDK. You should plan to review the documentation provided in the SDK Pre-Release, to get a better understanding of the new feature support available in Windows 3.0. The 1.63 DDK documentation contains additional information on these subjects, which is necessary for writing Windows device drivers.

Information on writing 386 Virtual Devices is provided in the Windows 3.0 DDK. You will no longer need to order a separate kit. In writing 386 Virtual Devices, you will need to use the tools available in both the Windows 3.0 SDK and the Windows 3.0 DDK.

Also, you should review the SDK Pre-Release for special notes on the tools provided in the SDK Pre-Release kit.

1.7 Integrating Final Driver Changes into Your 3.0 Driver

Since we may have begun development on beta level sources, you should use a mechanism for integrating our final optimizations and bug fixes into your 3.0 driver or virtual device.

By using an intelligent file comparison utility, you should be able to compare the source files provided in previous pre-releases with those we are shipping in this final 3.0 DDK. All changes in the final driver sources should be integrated into your final driver, if applicable. This will ensure you have the most current level of support in your driver.

We suggest you maintain an original copy of the driver sources you build your driver from specifically for this purpose.

We recommend you compare the differences in the driver sources you received in the 1.63 release to the sources provided in this release. This will help to outline the bug fixes and other changes we have made since that release.

Section 2 -- Installation of the DDK Pre-Release Software

The SDK Pre-Release notes contain directions for installing the Windows 3.0 software and for installing the basic tools you will need for building device drivers.

Before installing the DDK software, it is necessary for you to install the SDK software. See Chapter 2 "Installing the Pre-Release Software" in the SDK release notes for more information.

The DDK Pre-Release kit is mainly comprised of driver and virtual device sources. There is no formal installation process for these files, since they can simply be copied from the diskettes to your hard disk. Many of the driver source directories are organized into source trees containing files and sub-directories. We recommend you use the DOS XCOPY command in most cases to ensure you are copying the directory images just as they are on the diskettes. This is necessary if you plan to use the Make and Batch files which have been set up to build the drivers. If you change the organization of the files on your hard disk, be sure to make the appropriate changes to the Make or Batch file for building the driver.

Once you have installed the SDK tools, you should determine which sources you need and copy these to the appropriate directory on your hard disk. The sections below discuss the additional tools you may need, some of which are provided in this kit. You should also review the detailed information in Sections 6 and 7 for information on building the sources.

Note, it may be possible a source directory tree will span across several of the DDK diskettes. Be sure to review all diskettes to ensure you have copied all of the appropriate files.

2.1 What Else is Required?

You will need to provide the programming languages you prefer. We recommend using Microsoft C 5.1 and/or Microsoft Masm 5.1 for developing your Windows drivers and Masm 5.10B for developing 386 virtual devices. Note, Masm 5.10B is included in this release since it is not available as a retail language product today. The driver and virtual device binaries in the Windows 3.0 software (provided with the SDK Pre-Release kit) were built with these tools. You should follow the installation instructions provided with these products before you install the SDK or DDK software.

You may also need the tool, IMPLIB.EXE, provided with Microsoft C when building the drivers provided in this kit. This tool is NOT installed automatically when you install your Microsoft C software. You will need to do this manually.

Included in this Pre-Release are special tools, libraries and include files modified for Windows 3.0 device driver and 386 3.0 virtual device development. These files are not provided in the SDK, only the DDK. They are necessary for building the sources in this kit.

The SDK Pre-Release notes discuss the status of the tools provided in that kit. See Chapter 5 "Using This Pre-Release of the Windows 3.0 SDK" in the SDK release notes.

If you have directories for the 2.0 version of tools on your hard disk, be sure to set your DOS PATH and SET commands to include the 3.0 tools, include, and library directories instead of or before the 2.0 directories.

Also, you need to determine whether or not you need the 386 virtual device sources. For example, if you are writing a printer driver, you do not need to write a 386 virtual device, (because access to these devices are already virtualized at the printer port level in the Virtual Printer Device); but, if you are writing a display driver you probably will. If your device can be shared among several 386 Virtual Machines while running under 386 enhanced mode (e.g. DOS standard applications can call your device, as well as Windows), then you should provide a virtual device. Some examples of devices we will provide virtual device support for in Windows include: displays, the printer or parallel ports, serial ports, keyboards, pointing devices, NetBIOS, special hard disks, the math co-processor, etc.

The diskettes in this release contain all the software modules and MAKE and batch files necessary to build the drivers. You only need to copy the appropriate files to your hard disk. Each set of sources are set up in individual directories. You should use the DOS XCOPY or COPY commands to copy all files and any sub-directories associated with those sources (see detailed descriptions in Sections 6 and 7 for the directories required for each set of sources).

2.2 Notes About the Win3/C5.1 Compatible Libraries

The Windows 3.0 SDK is now providing a new Windows version of the C run-time libraries that are compatible with *both* the C 5.1 compiler and the forthcoming C 6.0 compiler. Since the Windows 3.0 retail product was created with the older Windows version of the C run-time libraries that is compatible with only the C 5.1 compiler, we are providing these older libraries with the DDK. This is so that people can build a device driver with the sources we provide and then compare it with one that currently ships with the Windows 3.0 retail product to see if the driver that they create binary compares with the identical one in the retail product. This exercise provides a good acid test that your development environment is set up correctly. These libraries are in the \LIB\C51 directory on the Device Driver Samples and Tools Disk 6.

2.3 Notes About LINK4 for Windows Device Drivers

The LINK4.EXE linker has been added to the DDK, although we recommend you modify the MAKE files to use the linker provided with your program language software. LINK4.EXE is only provided to help you build the sources with the MAKE files included in this kit. If you choose to use LINK4, you should copy this utility to your tools directory on your hard disk. This should correspond to the tools directory you may have set up with your programming language and/or with the SDK.

You will find this utility on the Device Driver Samples and Tools disk 8 under the \TOOLS directory.

2.4 Libraries for Windows Drivers and 386 Enhanced Mode Grabbers

There are four libraries on the diskettes you need to copy to your Windows 3.0 Library directory. Note, this should correspond to the directory you just created and setup with the SDK Pre-Release Install program.

You will find these files on the Device Driver Samples and Tools disk 6 under the directory \LIB. Use the DOS COPY command to copy the necessary libraries to your hard disk. All other necessary Libraries were installed by the SDK installation process.

2.5 Updated Include Files

There are include and header files for building Windows device drivers and virtual devices. The files are on the Device Driver Samples and Tools disk 6 under \INCLUDE. The 386 virtual device include files are on the Virtual Device Samples and Tools disk 1 under \INCLUDE. See section 2.5 below for more information on the 386 virtual device include files.

You should add the appropriate Windows include and header files to your Windows 3.0 Include directory which you created during the SDK Pre-Release Install procedure. See the special notes on 386 include files below for more information on installing these files.

2.6 Special Notes on Building 386 Virtual Devices

In order to build the sample virtual devices correctly, you need to install all the files in the \INCLUDE and \TOOLS directories on the Virtual Device Samples and Tools disk 1:

- TOOLS contains utilities which are required for building virtual devices, but are not currently part of the standard Windows 3.0 SDK tool set and are not readily available with most Microsoft language products. See the following section for details on each tool.
- INCLUDE contains .INC files used by the sample virtual device sources. This directory must be placed at the same directory level as the source directories, because they are referenced in the MAKE files as "..\include\sample.inc". This requirement does not apply to include files provided with the Windows 3.0 SDK, or supplied with standard Microsoft language products.

The \INCLUDE directory can be moved if you modify the MAKE files appropriately.

2.6.1 386 Virtual Devices Special Tools

The files in the \TOOLS directory should be copied (using DOS COPY) into your Windows 3.0 Tools directory setup by the SDK Pre-Release Install program.

This directory contains the following tools:

- ADDHDR is a utility used to modify the headers of virtual device drivers. It takes only one parameter, the name of the virtual device file, and should be run as a standard part of every build procedure.
- LINK386 is a special version of the Microsoft Segmented-Executable Linker provided specifically for use in building virtual devices.
- MAPSYM32 is a version of the standard MAPSYM utility used to convert map files produced by the linker into symbol files used by some debuggers. This version has been modified to understand 32 bit offsets.
- MASM5 is a special version of the Microsoft Macro Assembler, version 5.10B, which supports the 32-bit address space required for virtual devices. It has been named masm5.exe so it will not conflict with your standard masm executables.
- WIN386.EXE is a debug version of Enhanced Mode Windows which is needed for debugging virtual devices.

Each of these tools are necessary to build the virtual device sample sources with the MAKE files provided.

Section 3 -- Preparing Windows 2.x Drivers to Run Under Windows 3.0

3.1 What You Need To Do

Because of the new protected mode memory model, it may be necessary for you to update your device driver to work properly under Windows 3.0 running in protected mode. Certain coding rules need to be followed in order to avoid General Protection errors (GP Violations) which stop the execution of Windows (see the SDK documentation for details on memory management and the rules of protected mode). Windows 3.0 will check each driver it loads to verify if it has been marked for running in protected mode. Note, the issue is mainly when running in protected mode. Users will have the option of running in real mode in order to run 2.x level Windows applications and drivers, but they will lose many of the advantages of version 3.0.

Windows 3.0 may refuse to load some 2.x drivers when running in protected mode (display, comm, mouse, sound, system, and keyboard drivers fall into this category). Windows will reject most drivers which are not compatible with protected mode. When it encounters one, it will print an error message and return to DOS. The only drivers which can be loaded in this way are printer drivers, but the user will receive a warning message every time such a driver is loaded (this could be several times during a work session under different applications).

With this in mind, it is important for you to evaluate your current level of support and determine which of the following options you choose to follow:

- 1) Test, debug, and mark your existing driver to run under Windows 3.0 (this is not an option for display or keyboard drivers). Redistribute this driver with your product today.

The driver should be built with 2.1 tools. Then, you will need to use the MARK utility supplied in the SDK Pre-Release to mark your driver for memory use. See Chapter 3 of the SDK release notes. Note, drivers built with the 3.0 tools will NOT run under Windows 2.x.

This is not the most desirable option, but at least gives the user the opportunity to run Windows 3.0 with full protected mode memory support.

- 2) You should develop a new 3.0 driver (based on your existing 2.x driver or from the driver samples in this kit) which is built with the 3.0 tools and supports 3.0 features. This driver should be made available at the same time we announce and make available the Windows 3.0 retail product.

This is the most desirable solution since your driver will be able to take full advantage of new 3.0 features and the user gets full benefits of the Windows product and your hardware device.

One of the main objectives for providing you this DDK Pre-Release is to provide you the tools necessary for you to achieve the most desirable option (option 2).

3.2 Where to Look in the DDK Pre-Release

The 3.0 Device Driver Adaptation Guides provide detailed information on the new 3.0 feature support you should consider implementing in your device driver. You should review each chapter which applies to your type of device. Below is an overview of what you will find in the DDK documentation.

Chapter 3 contains new information on writing Display drivers for 3.0 features such as:

- Palette Manager
- Device Independent Bitmaps (DIB)
- Greater than 64K font support
- Running in protected mode
- Support for new visuals (cursors, icons, and bitmaps)

Chapter 5 contains new information on writing Printer drivers. One of the most prominent discussions is on Device Initialization. Two new APIs have been added to Windows 3.0 to give Windows applications the ability to initialize printer settings (ExtDeviceMode() and DeviceCapabilities()). Your printer driver should support these new functions. New printer escapes have been added (mainly for high end devices) and some have been updated. You should evaluate if your driver needs to support the new escapes or if you need to modify existing ones.

Chapters 10 and 12 contain information on Common Functions and Data Structures used by different type so device drivers. You should be sure to review the new enhancements to StretchBlt() if you are writing a display or printer driver.

Chapter 7 documents the new Windows Network interface.

Chapter 8 documents the new Keyboard structure supported under Windows 3.0. If you are writing a keyboard driver, you should review this information.

Chapter 9 discusses the Sound, Comm and mouse drivers. No new support has been added in the sound and mice drivers other than modifications to run correctly in protected mode. The COMM driver has been enhanced to support COM1 through COM4 and to run in protected mode.

The SDK documentation provides an overview of the new Help System, which will be provided under Windows 3.0. You should consider using the Help System for Windows printer drivers in place of readme type text files. . The PCL / HP LaserJet and Postscript driver sources included in this kit demonstrate the use of the Help manager (Help button is accessible via the printer dialog box).

Section 4 -- Preparing Windows/386 2.x Virtual Devices to Run Under Windows 3.0 386 Enhanced Mode

Under 386 enhanced mode in Windows 3.0, a new structure for Installable Virtual Device support has been added. The interface for Virtual Devices is very different from the 2.x versions of Windows/386. All new DDK documentation has been written to cover the new support available in Windows 3.0 386 enhanced mode.

2.x versions of Windows/386 virtual devices (VxD) are not fully compatible with Windows 3.0 386 enhanced mode because of this new interface. You should evaluate whether or not it is more efficient for you to convert your existing VxD to the new interface, or to start building a new VxD based on the sample sources provided in this kit. This decision will depend on your device and the extent to which you choose to support it.

Below is a brief overview of some key areas you should review in the DDK documentation:

Chapter 16 summarizes how Installable Virtual Devices work under 3.0, but you should read all chapters in Part 3 "Writing Virtual Devices" for complete details on this new support.

A complete reference to all new functions and services has been provided in Part 4. In addition, we have provided various sample virtual device sources demonstrating the use of the Virtual Machine Manager (VMM) services. See section 7 of these release notes for more information on the sample sources.

A few other important items which you need to be aware of:

- OFFSET32: This macro should be used in place of the standard Masm OFFSET directive, in all virtual device code. This will allow the code to adapt to future versions of the macro assembler.
- The V86MMGR (MMGR = memory manager) services: This feature will allow real-mode devices to tell 386 enhanced mode about the API they provide, and allow a generic 386 virtual device to map these calls between virtual and protect modes.

Additional information on "V86 Mode Memory Manager Device Services" is included in Chapter 40 in the DDK..

Section 5 -- Debugging in this Pre-release

There are several debugging tools provided with the SDK Pre-Release including Codeview Windows,

(CVW), SYMDEB, and WDEB386. If you have the appropriate hardware necessary for running CVW, you should try using this tool for your debugging. See the SDK Pre-release notes for more information on using CVW. Otherwise, you should use the SYMDEB debugger whenever you are debugging in the Real Mode memory model and the WDEB386 debugger when debugging in 386 enhanced mode. See the SDK documentation for more information on using SYMDEB and WDEB386.

Section 6 -- Description of Windows Driver sources

Below are detailed descriptions of the Windows 3.0 driver sources contained in this Pre-Release. The information includes where you can find the code on the diskettes, the procedure for building the driver and the features the code demonstrates.

The sources in this DDK Pre-Release are provided for your development use. You should use the sources provided as a code base or as examples on how you should structure your driver.

The sources are setup in sub-directories on the enclosed diskettes. You should use the DOS COPY or XCOPY command to copy all files and related sub-directories to your hard disk. It is important that you maintain the structure as provided on the diskettes for the accompanying MAKE and batch files to work properly. Remember to expand the files to their normal size by using TREEEXP. (see note on the cover page of these notes).

6.1 Display Driver Sources

We are providing 1, 4 and 8 plane display driver sources in the DDK. You will find all the 1 plane driver sources on the Device Driver Samples and Tools disks 1 and 2 under \DISPLAY\1PLANE. The 4 plane sources are on disk 1 under \DISPLAY\4PLANE. The 8 plane sources are on disks 2 and 3 under \DISPLAY\8PLANE.

6.1.1 1 Plane Drivers - CGA, EGA monochrome, EGA high resolution B&W, Hercules, Plasma, and MCGA

All of these drivers are 1 plane drivers. These drivers have been modified to run Bi-Modal (in Real and Protected memory modes), support Device Independent Bitmaps, and greater than 64K fonts. The sources to these drivers are expected to change in the future to accommodate bug fixes, but in general are relatively complete. They are located on the Device Driver Samples and Tools disks 1 and 2 under \DISPLAY\1PLANE.

These drivers share common code in many areas. Thus, the directory structures should be copied in their entirety to your hard disk. We suggest you use the DOS XCOPY command to do this.

You need to change directory to the appropriate sub-directory for the driver you wish to build. Each driver directory has a batch file, MAKEIT.BAT to build the driver. You will find a batch file for each driver under:

\DISPLAY\1PLANE\BWCGAHERC\CGA\MAKEIT.BAT	for CGA
\DISPLAY\1PLANE\BWCGAHERC\COMPAQ\MAKEIT.BAT	for Compaq Plasma
\DISPLAY\1PLANE\BWCGAHERC\HERC\MAKEIT.BAT	for Hercules monochrome
\DISPLAY\1PLANE\BWEGA\EGA\HIBW\MAKEIT.BAT	for EGA hi res black/white
\DISPLAY\1PLANE\BWEGA\EGA\MONO\MAKEIT.BAT	for EGA monochrome

`\DISPLAY\PLANE\B\WEGA\MCGA\MAKEIT.BAT`

for MCGA

If your display device is based on the 1 plane CGA, EGA, or Hercules standard, you should base your driver code on either of these drivers. Otherwise, these sources should provide you with the sample code necessary for you to design and plan for a new driver.

6.1.2 4 Plane Drivers - EGA and VGA

Both the EGA and VGA driver sources utilize all 4 planes to support 16 colors. These drivers have been modified to run Bi-Modal (in Real and Protected memory modes), support Device Independent Bitmaps, and greater than 64K fonts. The sources to these drivers are expected to change in the future to accommodate bug fixes, but in general are relatively complete.

These drivers share common code in many areas. Thus, all the files in the `\DISPLAY\4PLANE` directory structure found on the Device Driver Samples and Tools disk 1 should be copied in their entirety to your hard disk. We suggest you use the DOS `XCOPY` command to do this.

You need to change directory to the appropriate sub-directory for either the VGA or EGA driver and execute the batch file, `MAKEIT.BAT`. You will find a batch file for each driver under:

`\DISPLAY\4PLANE\COLOR\EGA\EGAHIRES\MAKEIT.BAT`
`\DISPLAY\4PLANE\COLOR\EGA\VGA\MAKEIT.BAT`

for EGA hi res color
for VGA

If your display device is based on the 4 plane EGA or VGA standard, you should base your driver code on either of these drivers. Otherwise, these sources should provide you with the sample code necessary for you to design and plan for a new driver.

6.1.3 8 Plane Drivers - Video Seven VRAM VGA and 8514/a

We have provided you sources to the Video Seven VRAM/FastWrite VGA and 8514/a driver. These drivers use all 8 planes available on the card for 256 color support. The code provides a good example of the use of the Palette Manager and Device Independent Bitmaps (DIB). The driver is adapted to run Bi-Modal and supports >64K fonts. This driver source can also be used as an example of how a packed pixel driver is implemented under Windows.

To demonstrate the use of DIBs and the Palette Manager in this driver, you can run the sample application, `SHOWDIB`, provided in the SDK Pre-Release.

You will find these sources on the Device Driver Samples and Tools disks 2 and 3 under the directory `\8PLANE`. You need to copy the entire directory structure to your hard disk. Use the DOS `XCOPY` command to do this.

Note, the `8PLANE` directory contains an empty sub-directory call `DEBUG`. You must have this directory available when you build the driver, otherwise the `MAKE` files will fail.

To build the Video Seven VRAM/FastWrite VGAdriver, you need to run the batch file:

`\DISPLAY\8PLANE\V7\MKP.BAT`

This builds the `V7VGA.DRV` file in the `DEGUG` directory.

To build the 8514/a driver, move to the `\8PLANE\8514\SOURCE` directory and type:

MAKE DRIVER

This build the 8514.DRV in the OBJ directory.

6.1.4 8514/a Resources

The 8514/a resource files have been provided on the Device Driver Samples and Tools disk 3 under `\DISPLAY\8514\8PLANERESOURCE`. These resources are for high resolution displays (1024 x 768 or greater) and should be used if applicable to your display device. This will ensure your cursors, icons, bitmaps, and color tables will be displayed properly.

For more information on the use of these resources, see Chapter 3 "Display Drivers: New Features" in the DDK Device Driver Adaptation Guide.

6.1.5 Windows Real and Standard Mode Grabbers

The Windows real and standard mode grabbers are on the Device Driver Samples and Tools disk 8 under `\GRABBERS`. Since much of the grabber code for different displays are common, you should copy the `GRABBERS` sub-directory in its entirety to your hard disk.

The individual grabbers are in the following directories:

```
\GRABBERS\CGAHERCCGA
\GRABBERS\CGAHERCHERCULES
\GRABBERS\EGAEGACOLOR
\GRABBERS\EGAEGAMONO
\GRABBERS\VGAVGACOLOR
\GRABBERS\VGAVGAMONO
```

You should run `MAKE` with the `MAKE` files provided in each of these directories to build the appropriate grabber.

6.1.6 System Font Sources

We have provided the new Windows 3.0 proportional system font and the Windows 2.0 fixed pitch system font. If your display driver supports resolutions other than the display resolutions provided in Windows, you may need to modify the system fonts for your display.

You will need to copy the `\FONTS` directory to your hard disk in its entirety. Use the `DOS XCOPY` command to do this. You will find this directory on the Device Driver Samples and Tools disk 8.

The 3.0 proportional system fonts are hand tuned for the screen and match the Adobe font widths for their Helvetica font. If you need to adapt the system font for a display with a different horizontal resolution than the ones supported by these fonts, you need to adjust the font widths accordingly. This will ensure the character will best "copy fit" the printed character with those on the screen.

You can modify these files with the Font Editor tool provided with the SDK Pre-Release.

Before you will be able to build any fonts, you must make the FONTS.OBJ file. You do this by running;

MASM FONTS;

from the \FONTS directory. You only need to do this once.

You can easily build the proportional system font files by executing the MAKE file from the \FONTS\SYSTEM directory:

MAKE SYSTEM

The font file names for the new proportional system fonts are:

CGASYS.FON	for (640 x 200) CGA resolution
EGASYS.FON	for (640 x 350) EGA high resolution
VGASYS.FON	for (640 x 480) VGA resolution
8514SYS.FON	for (1024 x 768) 8514/a resolution

In addition to the new 3.0 system font, the Windows 2.0 fixed pitch system font is necessary for older Windows applications running under Windows 3.0. All Windows applications which are not "marked" to run with the new proportional font, or which were not developed as a Windows 3.0 application, will be given the fixed pitch font for displaying text in dialog and message boxes, and the client area.

Windows 3.0 supports both of these files, so it may be necessary for you to also modify the appropriate fixed pitch font.

You will find these fonts in the \FONTS\FIXED directory. There is a MAKE file called FIXED, which can be executed from this directory by running:

MAKE FIXED

The font file names for the 2.0 fixed pitch system fonts are:

CGAFIX.FON	for (640 x 200) CGA resolution
EGAFIX.FON	for (640 x 350) EGA high resolution
VGAFIX.FON	for (640 x 480) VGA resolution (square pixels)
8514FIX.FON	for (1024 x 768) 8514/a resolution (square pixels)

If you want to build both the system and fixed fonts at once, you can run the make file FONTS from the \FONTS directory.

6.2 Display Driver Test Application

NOTE: The makefiles for all test applications assume use of the C6.0 libraries. If you want to build with C5.1, tell the linker to use slibw.lib in place of libw.lib.

The DDK now contains a display driver test application. You will find the complete documentation, program executable and source to this test under the \TESTS directory on the Device Driver Samples and Tools disks 16 and 17. The program is called DISPTTEST.EXE and is in the \TESTS\ITENBIN directory. The documentation for running the display test program is in ITENDOCSDISPTTEST.TXT. A more comprehensive display driver test, DTA.EXE is provided in the \TESTS\DTA directory. Documentation

on this test, DTA.DOC is also include in this directory.

The sources have been provided to aid in your debugging should you find a problem while running the display test program. If you need to build this program for some reason, you should follow the instructions in the README.TXT file under the \TESTSITE root directory.

6.3 Printer Driver Sources

We are providing driver sources for our PCL/HP LaserJet and Postscript printer drivers. In addition, sample sources for dot-matrix black/white and color raster printers are also included. All of these source samples provide full support for Windows 3.0 features.

6.3.1 PCL/HP LaserJet Printer Driver

The PCL/HP LaserJet driver sources are being provided as examples for the new Device Initialization APIs, use of the Printer Font Installer, the use of new Windows 3.0 controls in the printer dialog box, and the use of the new Help Manager. The two new APIs for Device Initialization are documented in the DDK Pre-Release documentation in Chapter 5 "Printer Drivers" (ExtDeviceMode() and DeviceCapabilities()).

The device mode dialog box should be used as an example for designing a 3.0 only driver dialog box interface.

Included in the 1.33 version of the DDK pre-release were the "Printer Fonts Kit" and the "Technical Notes on the PCL Driver". You should review these notes for information on font support under the PCL/HP LaserJet driver and for new features which have been incorporated into the 3.3 version of this driver. For example, the Soft Font Installer has been removed from the core driver and placed into a separate Dynamic Link Library (DLL). It will install font information for both soft and cartridge fonts. A new source tree was created for the installer (now called Printer Font Installer).

The PFM Editor (PFMEDIT.EXE) documented in the Printer Fonts Kit is provided on the Device Driver Samples and Tools disk 8 under \TOOLS.

The PCL/HP LaserJet core driver sources are on the Device Driver Samples and Tools disk 3 under \PRINTERS\HPPCL. The HPPCL directory contains subdirectories necessary for building the driver. You can build the driver by running the MAKEIT.BAT batch file from the ..\HPPCL directory.

The Printer Font Installer source is on the Device Driver Samples and Tools disk 4 under \PRINTERS\FINSTALL. You can build the font installer by running the MAKEIT.BAT file from the ..\FINSTALL directory. This module is called by selecting the "Fonts..." button in the HPPCL driver dialog box.

The HPPCL driver and the Printer Font Installer are being provided to serve as a base for your Windows 3.0 printer driver development.

6.3.2 PostScript Printer Driver

The PostScript driver is also a good example of the new Device Initialization APIs, the new Windows 3.0 controls, and the use of the Help Manager. Note, the device mode interface between the PCL/HP LaserJet driver and the PostScript driver are consistent where possible. These sources also serve as an example of the new and updated escapes documented in Chapter 5 "Printer Drivers".

This version of the driver has RGB color PostScript support.

Device Independent color Bitmaps (DIBs) are supported directly in the driver. This is a good example of how you can support this 3.0 feature on color printer devices. For more info on DIBs see Chapter 4 "Display Drivers" in the DDK and the SDK documentation.

Several optimizations have been made to the header file, which has reduced the size from previous versions of the driver (3.0 and 3.1) by one third.

The option to change the Graphics Resolution has been removed from the device mode dialog interface. Under Windows 3.0, StretchBlt is implemented at the driver level, thus, the need to change the device resolution to speed up bitmap output is no longer needed. See Chapters 10 and 12 "Common Functions" and "Data Structures and File Formats" in the DDK documentation for more information on the StretchBlt() enhancements. An example of its use is in the PostScript driver source file STRCHBLT.C.

Percent scaling has been added to the driver. The user can specify the scale of the output to be printed in the printable area of the page.

The driver sources are on the Device Driver Samples and Tools disk 4 under \PRINTERS\SCRIPT. A batch file is provided to build the driver. It is called MAKEPS33.BAT.

The PostScript driver has been modified to make it easier to add new device support, without modifying the source files. Provided in the DDK is a tool called MKPRN.EXE. You use this tool to compile PostScript Printer Description (PPD) files into Windows Printer Description (WPD) files. You can then install .WPD files for the PostScript driver by choosing the "Add Printer" option in the driver's dialog box. Once installed, the device will be selectable from the list of printers in the driver.

In the 1.33 version of the DDK Pre-Release kit, we provided "Technical Notes on the Postscript Driver", which discussed the MKPRN utility. The PostScript driver and MKPRN has been modified since that release. Instead of generating 3 printer resource files, MKPRN will only generate one .WPD file. The 1.52 and later PostScript driver will only install .WPD files. You MUST re-compile your PPD file with the 1.52 or later version of MKPRN to successfully install and test your printer with the 1.52 or later version of the PostScript driver.

The MKPRN.EXE tool is on Device Driver Samples and Tools disk 8 under \TOOLS.

6.3.3 Raster Driver Sample Sources

Also included in this kit are sample sources for an Epson 9 Pin printer and an IBM Color Printer driver. These samples have been updated from the original 2.x level samples to reflect new 3.0 feature support for device initialization and an enhanced printer dialog box interface.

The IBM Color Printer driver sources provide an example of supporting 3 plane color bitmaps with a color library, although, you may choose to implement a different method for supporting color on your device.

The files are on the Devices Driver Samples and Tools disk 5 under \PRINTERS\RASTER. To build the Epson driver you need to run the MAKE file under \PRINTERS\RASTER\EPSON:

```
MAKE EPSON
```

To build the IBM Color Printer driver, you need to run the BATCH file under \PRINTERS\RASTER\COLOR\IBMCOLOR:

```
MAKEIT.BAT
```

These raster driver sources should provide you a good example for implementing dot-matrix, ink-jet, and other raster device driver support.

6.4 Printer Driver Test Application

Updated in this release of the DDK is the printer driver test application. You will find the complete documentation, program executable and source to this test under the \TESTS directory on the Device Driver Samples and Tools disk 7. The program is called PRNTTEST.EXE and is in the \TESTS\ITE\BIN directory. The documentation for running the printer test program is in \TESTS\ITE\DOCS\PRNTTEST.TXT.

The sources have been provided to aid in your debugging should you find a problem while running the display test program. If you need to build this program for some reason, you should follow the instructions in the README.TXT file under the \TESTS\ITE root directory.

6.5 Keyboard Driver Sources

The new keyboard sources are on the Device Driver Samples and Tools disk 5 under \KEYBOARD. There is a batch file provided to build the IBM style keyboard driver and all the international tables for foreign keyboards. You will also need to copy and expand the INT31.INC include file which can be found on the \INCLUDE directory on the Virtual Device Samples and Tools disk 1.

The keyboard driver structure has been changed dramatically since Windows 2.x. If you currently have a 2.x level driver, you will need to rewrite the driver to the new structure. We suggest you start your development from the sources provided in this kit, adding support for your keyboard where appropriate.

To build the drivers, you need to run the MAKEKEYB.BAT batch file in the \KEYBOARD directory.

6.6 Mouse Driver Sources

The Microsoft Mouse driver sources are on the Device Driver Samples and Tools disk 6. Copy all the files from the \MOUSE directory to your hard disk with the DOS XCOPY or COPY command. To build the driver, run:

```
MAKE MOUSE
```

The mouse driver has few changes from the version shipped with the Windows 2.1 DDK. The driver was only modified to run correctly in the protected mode memory model. No further changes are expected, except bug fixes.

6.7 COMM Driver Sources

The COMM driver sources are on the Device Driver Samples and Tools disk 6. Copy all the files from the \COMM directory to your hard disk with the DOS XCOPY or COPY command. To build the driver, run:

```
MAKE COMM
```

This driver requires the include file INT31.INC. You will need to copy this file from the Virtual Devices and Tools disk 1 from the \INCLUDE directory. Make sure this file is in your include path.

This driver has been enhanced to support COM1 through COM4. New API and changes to support dual mode execution with Windows 3.0 have also been added.

6.8 Sound Driver Sources

The sound driver sources are on the Device Driver Samples and Tools disk 6. Copy all the files from the \SOUND directory to your hard disk with the DOS XCOPY or COPY command. To build the driver, run:

```
MAKE SOUND
```

The sound driver has few changes from the version shipped with the Windows 2.1 DDK. The driver was only modified to run correctly in the protected mode memory model.

6.9 Network Driver Sources

The MS-Net network driver sources are on the Device Driver Samples and Tools disk 5. Copy all the files from the \NET directory to your hard disk with the DOS XCOPY or COPY command. To build the driver, run:

```
MAKE MSNET
```

6.9.1 Network Driver Test Application and Hardware Compatibility Test

Included in this release of the DDK is the network driver test application. You will find the complete documentation, program executable and source to this test under the \TEST directory on the Device Driver Samples and Tools disk 7. The program is called NETTEST.EXE and is in the \TESTS\NETTEST\BIN directory. The documentation for running the network test program is in \TESTS\NETTEST\README.TXT.

Also included in this release of the DDK is the network driver test application. You will find the complete documentation, program executable and source to this test under the \TEST directory on the Device Driver Samples and Tools disk 7. This test consists of several programs and can be found in the \TESTS\HCT directory. The documentation for running the network test program is in \TESTS\HCT\README.DOC.

These test programs are still under development and will be further enhanced for the final DDK product.

Section 7 -- Description of the Virtual Device sources

Each source directory contains one or more .ASM source files, a .DEF file, and two MAKE files. The first is usually called MAKEFILE. (or MAKEVGA. for the VGA VDD) and is for use with Microsoft NMAKE or with other Unix-style MAKE utilities. The second, called by the name of the file being produced (e.g. VPD.), is for use with the standard Microsoft MAKE utility. Either MAKE file can be used to produce virtual devices. Note, NMAKE is not provided in this release, but has been shipped with some of Microsoft language products (for example, Fortran 4.5).

These MAKE files offer examples of the proper uses of the tools, options, and general build procedures that should be followed in developing virtual devices.

As mentioned previously, these sources are provided to you to begin development for Windows 3.0 386 enhanced mode virtual device support.

Note, if you encounter random results or problems when building these virtual devices, it may be due to a low memory situation. This can be resolved by creating more memory to your system (removing TSRs, Network software, etc.).

7.1 The Display Devices (CGA, EGA, VGA, 8514, HERC)

This device virtualizes the video display and is the most complex of the devices supplied in the DDK.

The various VDD sources are on the Virtual Device Samples and Tools disk 1 under the directories \VDDCGA, \VDDEGA and \VDDHERC.

The \VDDCGA source tree contains files to only build the CGA VDD. This VDD also provides support for the Compaq Plasma display (also called IDC).

The \VDDEGA source tree provided uses conditional assembly to build two separate virtual devices to handle standard EGA, VGA and 8514/a video adapters.

The \VDDHERC source tree contains files to only build the Hercules VDD.

You will find different MAKE files for each of the VDDs in the appropriate directories mentioned above. Listed below are the MAKE file names (for NMAKE and DOS MAKE) for each VDD:

<u>Adapter</u>	<u>NMAKE</u>	<u>DOS MAKE</u>
CGA/IDC	MAKEFILE.	VDDCGA.
EGA	MAKEEGA.	VDDEGA.
VGA	MAKEVGA.	VDDVGA.
8514	MAKE8514.	VDD8514.
Hercules	MAKEFILE.	VDDHERC.

Note, if you are using NMAKE, you may need to rename the appropriate NMAKE file to "MAKEFILE". Some versions of NMAKE will always use the file name "MAKEFILE".

7.2 The Display/Windows Interface (GRABBER)

The grabbers for 386 enhanced mode are different than the grabbers run in real or standard modes. They are responsible for rendering a virtual machine's display context within a window; therefore, they are closely bound with the virtual display device (VDD). Each VDD needs to be accompanied by its own grabber as a linked pair.

The sources are on the Virtual Device Samples and Tools disk 2 under the \GRABBERS directory structure.

This release contains source for CGA, EGA, VGA, 8514/a, Compaq Plasma and Hercules grabbers. Since they are all built in the same directory, different MAKE files are provided for each of them. Below are the MAKE file names (for NMAKE and DOS MAKE) for each grabber:

<u>Adapter</u>	<u>NMAKE</u>	<u>DOS MAKE</u>
CGA	MAKECGA.	CGAGRB
EGA	MAKEFILE.	EGAGRB.
VGA	MAKEVGA.	VGAGRB.
8514	MAKE8514.	8514GRB.
Hercules	MAKEHERC.	HERCGRB.
ATT/Compaq	MAKEPLSM.	PLSMGRB.

Note, if you are using NMAKE, you may need to rename the appropriate NMAKE file to "MAKEFILE". Some versions of NMAKE will always use the file name "MAKEFILE".

The font files used by the grabbers when running in a window are provided in this kit. You will find them on disk 2 under \OEMFONTS.

Before building the OEM fonts, you need create the FONTS.OBJ file by running;

```
MASM FONTS;
```

from the \OEMFONTS directory. Once this is completed you run the MAKE file MAKEFONT:

```
MAKE FONTS
```

You need to make sure you have WINSTUB.EXE (provided in the SDK Pre-Release) in your DOS path before building the grabbers and/or OEM Font files. Otherwise, these files will not build properly.

7.3 The DMA Device (VDMAD)

This device handles direct memory access devices.

Virtual devices can support hardware cards which use DMA by calling the DMA Services provided by the VDMAD. These services were documented in the 1.33 DDK documentation.

See section 7.15, the Floppy Drive Device, for an example of a device which uses the DMA Services.

The VDMAD itself should not normally need to be customized; it would only need to be altered to support DMA on machines with non-standard architectures.

The sources are on the Virtual Device Samples and Tools disk 1 under the \VDMAD directory structure.

7.4 The EBIOS Device (EBIOS)

This device detects the EBIOS page on machines (such as the PS/2) where it is used, identifies it, and ensures that it is reserved as global memory.

It will probably not need to be modified, but is provided here as an example of a relatively simple device which passively accommodates something in the environment.

The sources are on the Virtual Device Samples and Tools disk 1 under the \EBIOS directory structure.

7.5 The Hard Disk Device (VHD)

This device virtualizes access to the hard disk. It also provides direct access to the disk for demand paging on compatible hardware.

This should be modified to support incompatible hard disks.

The sources are on the Virtual Device Samples and Tools disk 2 under the \VHD directory structure.

7.6 The Keyboard Device (VKD)

This is one of the more complicated devices, as it not only virtualizes the keyboard but also interacts with the Windows shell to handle hotkeys and other special functions.

It should be modified to support other, nonstandard keyboards.

The sources are on the Virtual Device Samples and Tools disk 2 under the \VKD directory structure.

7.7 The Mouse Device (VMD)

This device virtualizes the mouse and maps the int33 API between protect and virtual modes.

The sources are on the Virtual Device Samples and Tools disk 2 under the \VMD directory structure.

7.8 The Netbios Device (VNETBIOS)

This device maps the Netbios API between protect and virtual modes, allowing Windows applications to access the network. It also handles asynchronous network transactions by mapping the application's buffer into global memory, so the network software can access it when the asynchronous event occurs (even if another virtual machine is running at the time).

This should be modified by network vendors who extend the standard NetBIOS interface and also serve as a guide to writing virtual devices for other types of network software. The sources are on the Virtual Device Samples and Tools disk 2 under the \VNETBIOS directory structure. Also, see the virtual device sources for DOSNET (below).

7.9 The Local DOS Network Device (LDOSNET)

In general, this device manages network connections and assures network integrity across all virtual machines.

This source was included in previous pre-releases of the DDK under the name DOSNET. However, it has been replaced by a new DOSNET device which makes network connections global across all virtual machines. The version included here keeps connections local, but allows inheritance. It has therefore been renamed LDOSNET to avoid confusion.

The sources are located on the Virtual Device Samples and Tools disk 2.

The device should be modified when your software does not use standard DOS redirector calls for handling network connections.

7.10 The BIOS Device (BIOSXLAT)

This device maps the ROM BIOS API between protected mode and virtual 8086 mode, allowing Windows applications and device drivers access to ROM BIOS services.

This should be modified when there are non-standard ROM BIOS calls that pass pointers to memory and which are used by Windows applications or device drives.

You will find the sources on the Virtual Device Samples and Tools disk 2 under the directory BIOSXLAT.

7.11 The Paging Device (PAGESWAP)

This device is used by 386 enhanced mode for demand paging at either the int21 or int13 level.

The sources are on the Virtual Device Samples and Tools disk 1 under the \PAGESWAP directory structure.

7.12 The Printer Device (VPD)

The Printer Device virtualizes access to the parallel ports. If a second virtual machine tries to access one of the ports while it is being used by another application, a contention dialog is presented to the user allowing them to resolve the dispute of ownership.

The sources are on the Virtual Device Samples and Tools disk 2 under the \VPD directory structure.

7.13 The COMM Device (VCD)

This device virtualizes the standard serial ports on ISA architecture machines. It supports COM1 through COM4.

It should be modified to add support for different chip sets or for additional COM ports.

You will find the sources on the Virtual Device Samples and Tools disk 2.

7.14 The Floppy Drive Device (VFD)

This device is responsible for two things:

- It removes special timer port trappings to ensure copy protection schemes work properly.
- It communicates with the VDMAD to synchronize DMA channel usage.

You will find the sources on the Virtual Device Samples and Tools disk 2.