

# *A BEGINNERS GUIDE TO DOS PROGRAMMING*

Book 2

Windows 95 DOS Development

Axle

30 January 2024



Microsoft, flag logo designed by Jonathan D. Cowles and Jeff Boettcher, Public domain, via Wikimedia Commons

## Table of Contents

Licence .....	5
Rewvisions.....	5
Preface.....	7
Installing Windows 95 .....	10
Setting up a VirtualBox guest .....	10
File transfers.....	20
Setting up Windows 95 .....	22
Base install .....	22
Quick usability tweaks .....	82
Windows 95 Display drivers fix.....	92
Windows 95 Upgrades.....	106
Multiple Floppy drive issue .....	112
Update Internet Explorer .....	116
Runtime Libraries .....	134
DirectX.....	134
Microsoft Visual C(++) 6 Runtime Files .....	139
VisualBasic Runtime libraries .....	140
Essential Applications .....	140
Archives.....	140
Restore tools .....	149
System Clean-up.....	162
System tools .....	175
Applications.....	185
Install sumary .....	196
HFS over IP .....	197
DOS – Windows.....	214
Booting to MS-DOS real mode.....	225
A. Restart in MS-DOS mode .....	226
Audio Drivers.....	230
Mouse Drivers .....	231
CD-ROM Drivers.....	233
B. Use F8 at boot and select “6. Command prompt only”.....	235
C. Change BootGUI=1 to BootGUI=0. Use Win to start Windows.....	236
DOS navigation.....	239

DOS Directories .....	239
Essential navigation .....	240
Application launch BAT .....	244
Installing Development tools .....	244
MS QuickBASIC .....	251
QBASIC V1.1 .....	252
QuickBASIC V4.5 .....	256
QuickBASIC Extended 7.1 (QBX) .....	271
Visual BASIC V1.0 for DOS.....	272
DJGPP (C/C++) .....	284
FreeDOS Bonus CD install.....	284
Setting up from the DJGPP website .....	287
Setting up DJGPP IDEs.....	291
Open Watcom .....	313
Turbo C (1.0, 1.5, 2.0) .....	313
Other C [BCC, CC386].....	338
FreeBASIC DOS 32-bit V1.07.1a .....	338
FreeDOS Bonus CD V1.07.1a .....	339
Use FDNPKG package manager from the repository.....	341
Manual Install – Not recommended .....	343
Long File Names.....	344
Start-up batch file .....	344
FED IDE setup .....	345
FreeBASIC additional .....	360
DOjS (+ Jsh) .....	361
Win 32 Development environments .....	367
GCC, MinGW Dev-C++(Bloodshed V401, Orwell V 511(x86)-GCC4.9.2) .....	367
Windows Scripting Host (WSH) V 5.6 .....	368
AutoIt (Old Versions 3.2.12.x ) .....	368
Microsoft Visual C++ 6.0 Standard Edition.....	369
Microsoft Visual Basic 6.0 (Enterprise) .....	369
Debuggers & Utilities.....	369
FreeDOS Debug - Debug is an extended MS Debug clone.....	369
Insight Debugger V1.24 (DOS) .....	369
FreeDOS base install .....	369
UPX .....	370

STRINGS.COM.....	370
DOS HexED .....	371
UHEX .....	371
HexCompare.....	371
Video Test.....	371
MONITORS .....	371
NSSI .....	371
Info Plus.....	372
HWInfo 16-bit 32-bit.....	372
Hexit.....	372
HIEW .....	372
Executing Inspector DBG.....	372
Mem (FreeDOS \BIN) .....	372
Mode (FreeDOS \BIN) .....	373
XMSTest (FreeDOS \BIN).....	373
XMSStat (FreeDOS \BIN) .....	373
ListVESA.....	373
FoxCalc.....	373
RCal .....	373
p7Zip .....	373
Windows 95 utilities: .....	373
System Monitors:.....	373
Hex Editors: .....	374
Debuggers: .....	374
PE-Info:.....	375
Resource edit:.....	376
Image tools:.....	376
Office tools: .....	377
Archive: .....	378
Libraries .....	378
FreeBASIC Library list (DOS, Win9x, Win x96-64) .....	379
Some Audio libraries.....	379
PDCurses (Curses).....	380
GRX graphics library.....	381
MGRX (GRX) 2023 .....	382
Allegro.....	383

## A Beginners Guide To DOS Programming

GfxLib_FB (libgfx2) .....	384
LibGD.....	385
Nuklear (Could be ported to DOS) (JS Capable) See Microwindows .....	385
FLTK (DOS, C++) see Nano-x, Microwindows .....	385
Microwindows (DOS, Win32, Linux) Nano-X.....	385
JMGUI (GfxLib, MyGL, YAGL) .....	386
TCL/Tk .....	387
DUGL.....	387
Library Summary.....	388
DOS Development Summary .....	389
Supplemental .....	390
Additional Information .....	390
DOS and DOS development links and guides (Unsorted) .....	390

### Licence



This work is licensed under a Creative Commons Attribution 4.0 International License.

<https://creativecommons.org/licenses/by-nc-nd/4.0/>

©Ozz-i-sofT ® 2022-2024

Written by Alexander Maddern (aka Axle)

~~With many thanks to Daniel Moore for proof reading and corrections as well as the many ideas, examples and suggestions that have been provided.~~

Many thanks to all the application developers past and present whose work is included in this guide.

This document is provided in the hope that it will provide a useful overview of the concepts that are covered and that many concepts will only be covered in part or brief. The author does not accept liability for the accuracy of the content provided within this document. The reader should seek to obtain documentation for the specific programming languages and platforms used within this document. It is recommended to use a sandboxed virtual environment to test all of the examples that have been provided.

Revisions		
Version	Date	Notes
Draft V 0.1	26/07/2023	Basic outline
Draft V 0.2	06/01/2024	Windows base install.
Draft V 0.3	07/01/2024	System upgrades to W95C +
Draft V 0.4	10/01/2024	Essential applications installed.
Draft V 0.5	15/01/2024	Added DOS mode configs and DOS drivers.
Draft V 0.6	19/01/2024	Added explanations on DOS Mode and Windowed Mode.
Draft V 0.7	21/01/2024	Added DJGPP and TurboC
Draft V 0.8	24/01/2024	Added FreeBASIC compiler.
Draft V 0.9	25/01/2024	Completed all chapters.
Draft V 0.10	28/01/2024	Proof read and many corrections. This is a “ <b>Preview Draft</b> ”.
Draft V 0.11	30/01/2024	Corrections. Add source code archive. Add Strings25.zip. This is a “ <b>Preview Draft</b> ”.

TODO:

Proof read. [Partial]

## Preface

This guide is secondary companion to “A BEGINNERS GUIDE TO DOS PROGRAMMING” - Book 1 “DOS Development”.

In these two guides I am going to explain the basics of setting up some DOS development environments. These development environments are applicable to DOS 16-bit and DOS 32-bit with the significant focus being upon 32-bit protected mode software development. I am not going to cover any specifics about DOS programming itself, just setting up the different programming environments and some libraries. Having a functional programming environment will facilitate the learning of the different programming techniques for DOS, so I view this as a first step for anyone interested in DOS programming.

Although the content of two guides can be used individually or complimentary to each other, the latter is the more common practice. We can use the Windows 95 implementation for speed and convenience and the FreeDOS implementation as our target and test platform. The FreeDOS implementation is released under GPL and similar and has less licencing restrictions compared to the proprietary MS-DOS so this is the primary platform that I will focus upon for DOS software development. Windows 95 (OSRx.x) is a secondary DOS development platform that offers some additional convenience with GUI based IDEs and editors as well as the ability to multitask. The underlying MS-DOS 7.0 in Windows 95 is mostly compatible with FreeDOS 1.3 (Approximate to MS-DOS 5.0 to 6.2) so most applications can be prototyped in the Windows 95 DOS-7.0 environment and will run successfully on FreeDOS 1.3. Please note that the Windows 95 development environments have a primary focus upon DOS 32-bit protected mode development.

There is a 3<sup>rd</sup> option called “Cross Compiling” where we make use of a modern operating system and development environment. The output code is compiled for the target system and will not run on the host system where it is created. In this case we need to have our target system available at all times to test our output code. I will be showing the Windows 95 option as we can test most of our code directly from within the same platform.

Windows 95 and its associated updates are covered by proprietary copyright licences so I can't offer downloads for the required files so you will need to source them yourself based upon the file names and other guides. If you have an original legal copy of Windows 95 then it is OK to use it as long as you adhere to the Licence agreement. That being said, Microsoft tends to be somewhat tolerant of the personal use of Windows 95 in an educational or historical context.

Please be sure to check the section “Additional information” at the end of the guide for references and links to other guides on setting up and optimising the operating system setups. There are a great deal of well written guides that go into depth about setting up DOS and Legacy Windows environments.

I will be using VirtualBox 7.0 for the guides. You can also use the legacy VirtualBox version 6.x. Note that VirtualBox Additions will not work for DOS or Windows 95. VirtualBox does not officially support legacy operating systems such as DOS or Window 9x. I will show some workarounds in the later sections.

This guide will take you through the essentials of setting up Windows 95(C upgraded) as well as some of the essential applications and some drivers. In this guide I will use the Windows 95B (OSR 2.0/2.1) install which will be upgraded to 95C (OSR 2.5). I do not recommend starting at Windows 95

Chicago, or Windows 95A (SvcPck 1, OSR 1). You can begin with a Windows 95C if you wish and start the updates at Internet Explorer 4.1.

I will also offer some convenience customisations for using Windows 95 by adding some additional runtime updates as well as a number of convenience applications and finally the guides for setting up a number of software development environments. Although I have attempted to keep the sections above as separate as possible they will sometimes blend together in practice to form a complete setup.

The order of install of some of the service releases and runtime updates are important. Runtime environments (DLLs) are known to be problematic in the Windows 9x series with many conflicting updates and versions. There is no simple fix to this problem other than sourcing the correct library components that match with the application you are attempting to use. Where you encounter what may be a dll version conflict in the system, place the required library version components in the directory alongside of the application executable instead of in the system path. The application will always search its own directory before looking to the system for the required files avoiding the need to install conflicting versions into the Windows system. This is the simple workaround for what was dubbed “DLL Hell” back in the day.

Pay particular attention during the stage that upgrades Internet explorer 4.0. This update package brings 2 significant changes. The first is that the old file explorer application is removed and replaced by a new file Explorer interface that runs on top of the Internet Explorer engine. IE 4.x + and the file manager is inseparable from IE. This update also offers the opportunity to include the Default Windows 98 Active Desktop. I would recommend installing when asked as you cannot install active desktop at a later stage. You only get that one chance to install it.

Setting up the DOS development environments and associated tools are very similar to the “A BEGINNERS GUIDE TO DOS PROGRAMMING” - Book 1 “DOS Development “ with some Windows 32-bit specific settings. You may need to refer to Book – 1 for some DOS specific guides that are not duplicated in this Book – 2.

If you are unfamiliar with Windows 95 or DOS then use this guide as a practice run to test different ideas, take some notes along the way and then create your own install at a later time once you are more familiar. If you are already familiar with Windows 95 or DOS you may want to skip to specific customisations or application installs.

I recommend keeping a log of installed applications and changes that you make to the system. Custom installs particularly relating to DOS can get messy very quickly and you will lose track of and forget what changes you have made. Batch files, autoexec.bat and config.sys are extremely fluid in DOS environments. They will be changed and customised by the user as well as application installers making incremental changes to the autoexec.bat (FDAUTO.BAT in FreeDOS). Keep a backup folder with timestamped copies of your autoexec and config files for safety. I also recommend keeping a template of all custom batch files created for later reference. You can copy paste the batch file code instead of writing it from scratch. If you make an error in a batch file you always have a correct reference to check against. Windows 95 will also create shortcuts (PIF links) to your DOS applications which allow you to set specific environments that you would otherwise have to set in a batch file.

In Windows 95 DOS can be accessed by booting the system in real mode, or via the command (DOS) emulator when running in the standard desktop mode. For this guide I will be using the standard 32-bit desktop mode.

There are many helpful guides available on the internet that go deeper into the specifics of some of the setup tasks as well at targeting specific programming environment tasks. You will need to search out and look through some of these guides for more focused help. I have created a list of URLs at the end of this guide with links to other documents available at the time of writing.

I provide a condensed summary of customisation and installed applications at the end of the guide.

**Note:** If you copy and paste source code from this document you will likely introduce Unicode characters into the ANSI text documents used in DOS. Sometimes this can appear as white space and other anomalies. Please make sure you check the code in an ANSI capable text editor such as Notepad++ before copying codes into any DOS source files.

### NOTE!

Windows and DOS 7.1 are a hybrid dual boot system. The Windows OS always has some degree of control over the underlying MS-DOS. This means that there is no “pure” stand alone DOS in Windows 9x systems. When we boot into DOS mode the windows system configurations still exist as Windows controls the boot process for both operating systems. We have to keep this in mind when running DOS applications under “DOS Mode” (pseudo real mode) and Windows “emulated windowed mode” (32-bit protected mode).

Any time we set DOS configuration and environment variables we are also making them available under Windows protected mode which can often lead to configuration conflicts between DOS settings and Windows DOS emulated settings. Always treat DOS 7.1 as being an inherent part of Windows.

The safe way is to leave the CONFIG.SYS and AUTOEXEC.BAT files empty and use the Windows Program Information File (PIF) and set the individual DOS application environment from the Windows PIF. Windows will then “Exit To DOS” with the correct settings for that application.

If you want a “Pure” DOS system then it is better to use MS-DOS 6.22, or the more liberally licenced FreeDOS.

If the CONFIG.SYS and AUTOEXEC.BAT files have any entries set they will be processed at boot time (Longer boot time) and used as part of the default (Global) configuration and environment when using the console emulator while running Windows. You will need to take this into consideration when setting additional local environment settings in batch files when running DOS applications. Windows typically creates a PIF with the memory and environment variables set which can lead to unintended configuration conflicts with the CONFIG and AUTOEXEC settings.

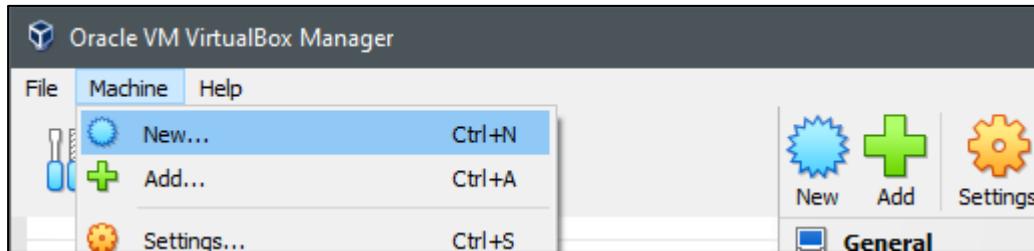
## Installing Windows 95

128MB, V32MB VBoxSVGA (No 3D), 4GBHDD x 2

Note that I accidentally used VBoxVGA instead of SVGA in this guide. It won't make any difference to the setup tasks other than some minor differences in the driver used by the SciTech Device Driver for video. You can use either adaptor successfully, but I would recommend the SVGA.

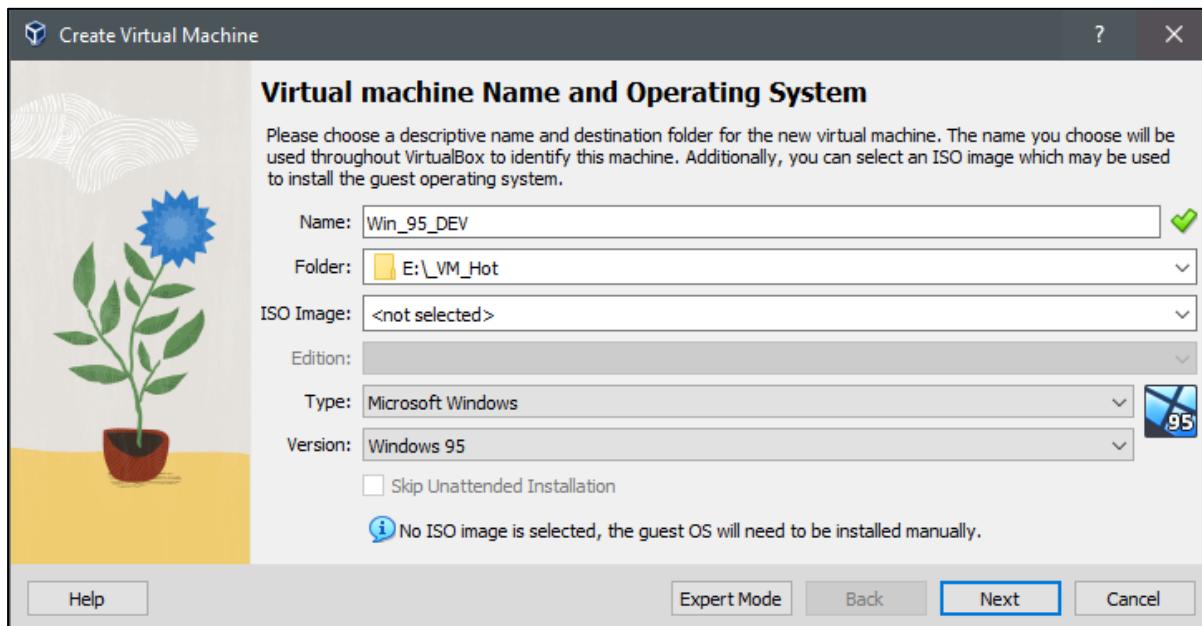
### Setting up a VirtualBox guest

In the VirtualBox Manager select "Machine -> New..."

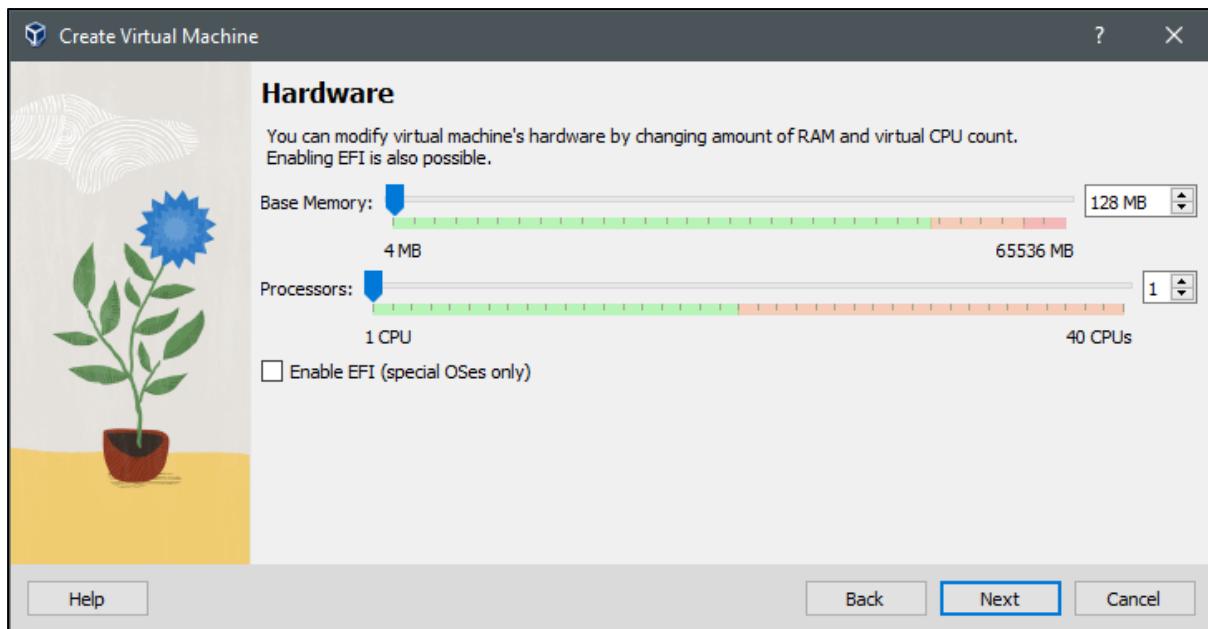


Give the guest machine a recognisable name and select the location where you store your guest operating systems.

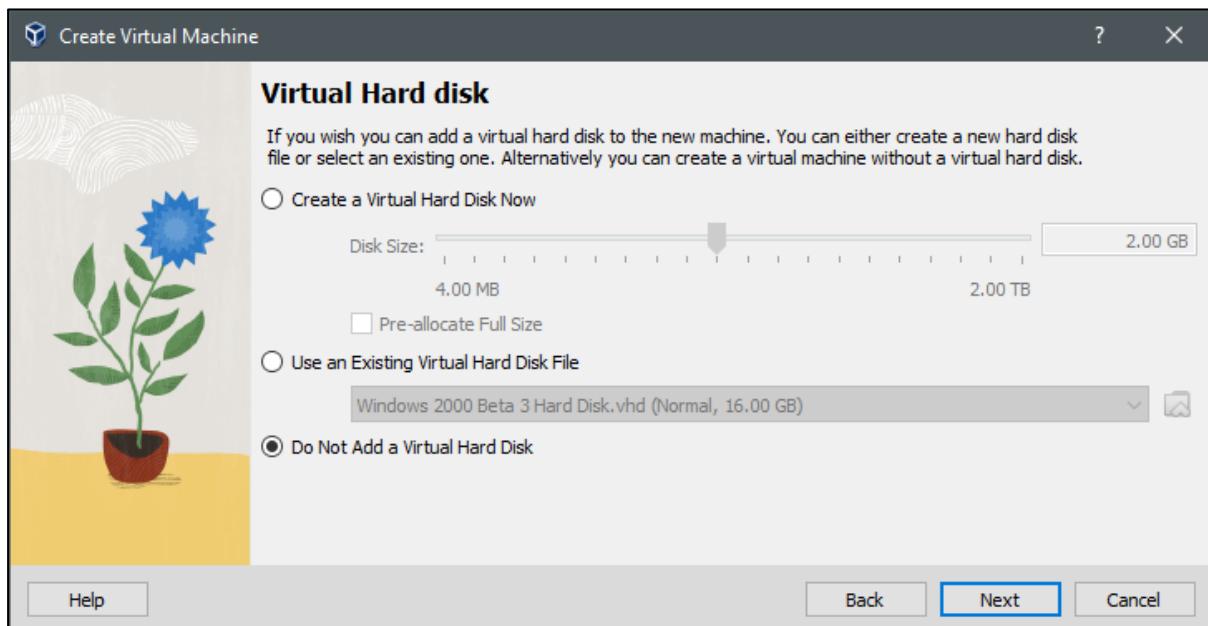
For "Type" select "Microsoft Windows" and then select Windows 95 in "Version" and then "Next".



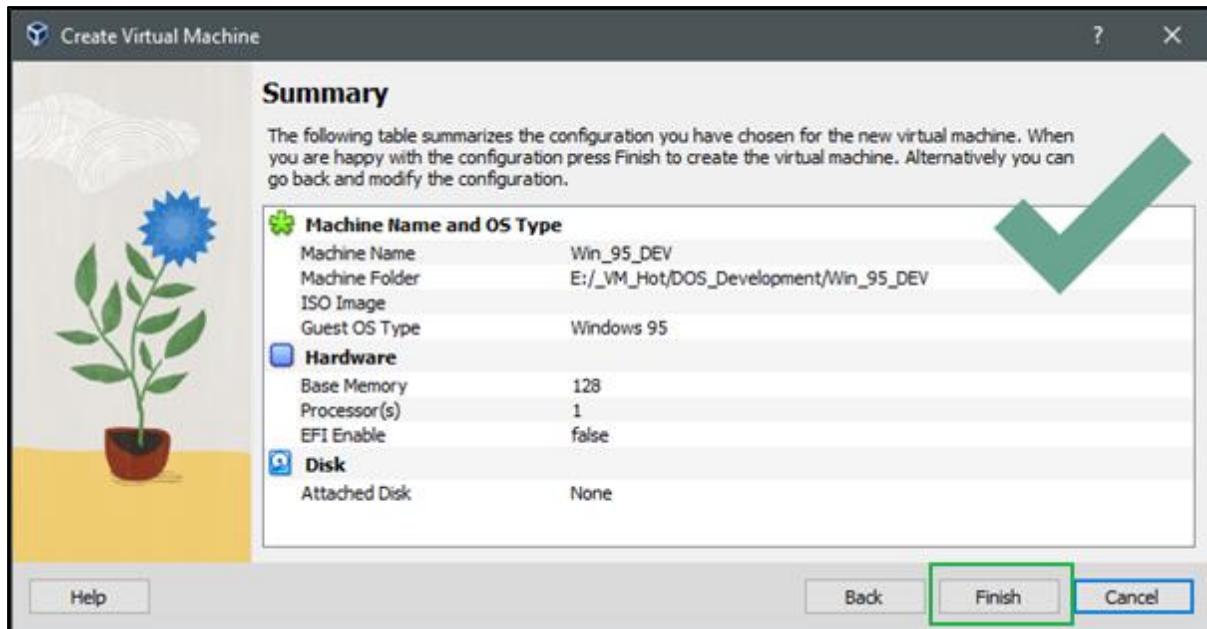
Change the default hardware settings to 128 MB and 1 CPU and select "Next". 128MB of RAM will allow for some of the upgrades. You can always alter it latter to emulate low memory machines if needed. Never select more than a single CPU as Windows 95 does not have the native ability to recognise more than One CPU. However, some non OEM upgrades can make this possible but I would not recommend it.



Do not create a virtual hard disk at this stage. VirtualBox V 6.x does allow you to create a Windows .VHD drive at this stage, but for VirtualBox V 7.0 you will need to select expert mode at the beginning to be able to select a .VHD as the drive type. This is OK to leave for now as we will just create the virtual hard drive later.

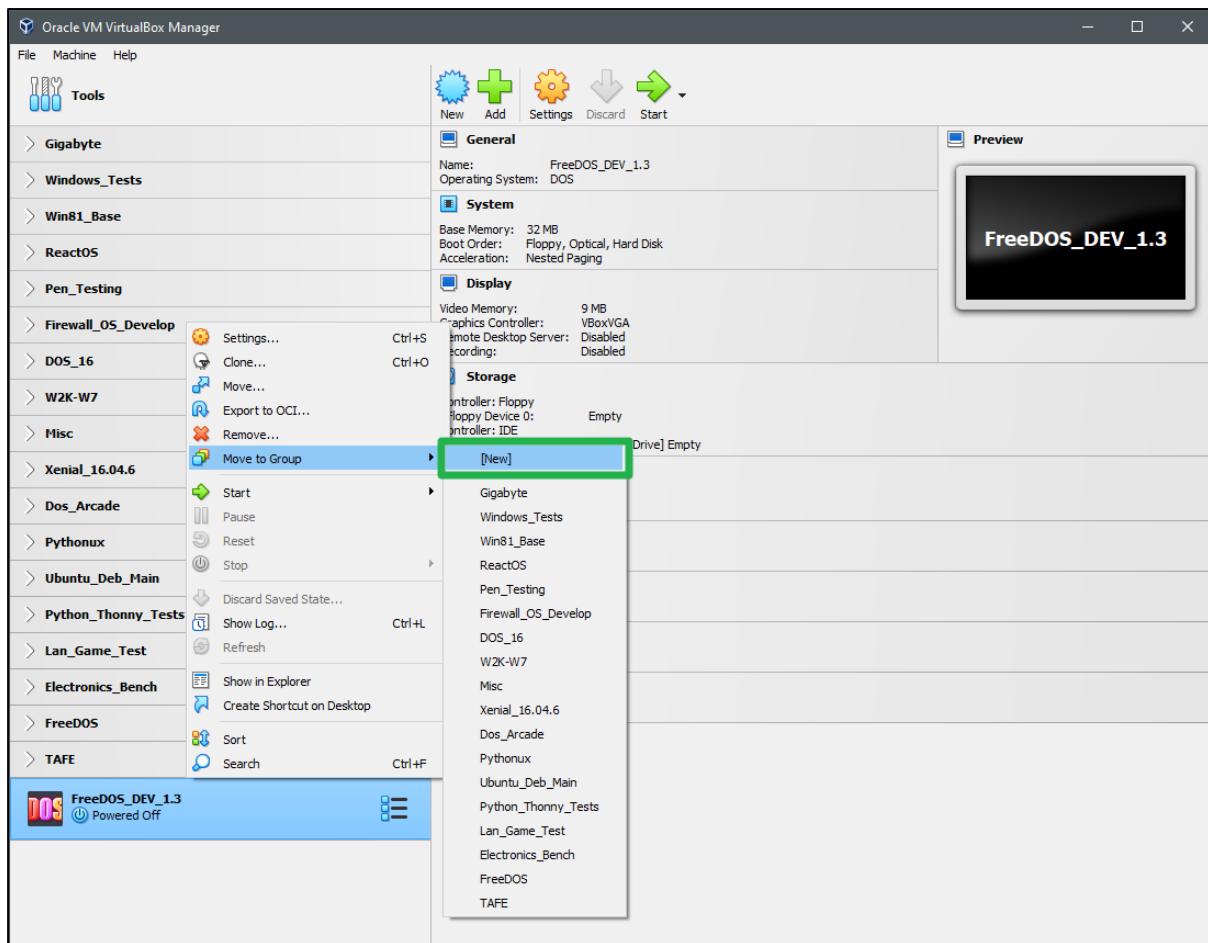


Check that the summary has the correct details and select “Finish” and then “Continue” on the warning dialog.



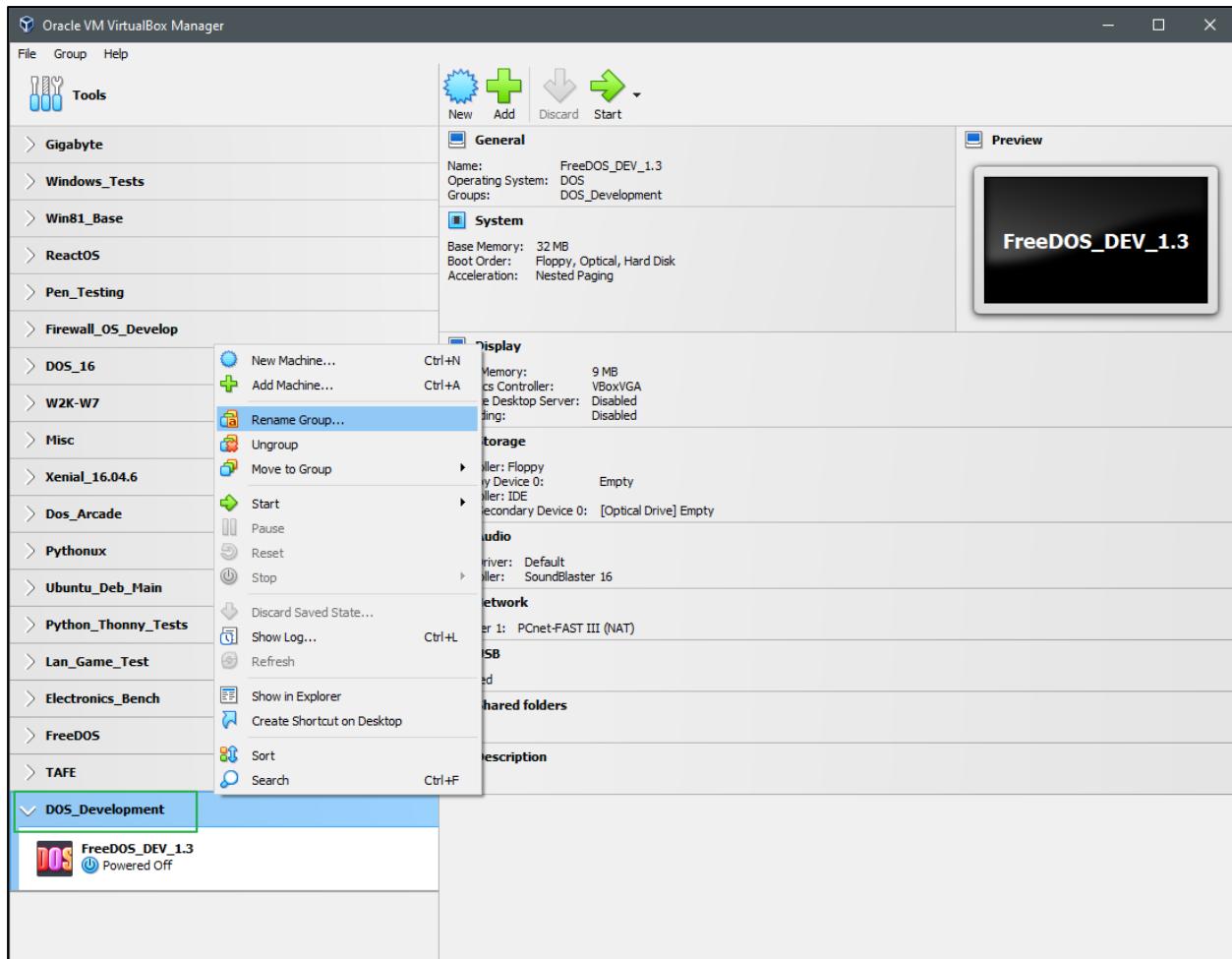
At this point you can place your new Windows 95 Virtual machine into a group for easier organisation if you want. You must have at least one premade virtual machine to create a group. The newly created group will not be named and you will need to give it an appropriate name after it has been created. I have used the guide from the FreeDOS, so your current VM will be Windows 95 instead. You can also drag and drop a virtual machine into an existing group.

## A Beginners Guide To DOS Programming



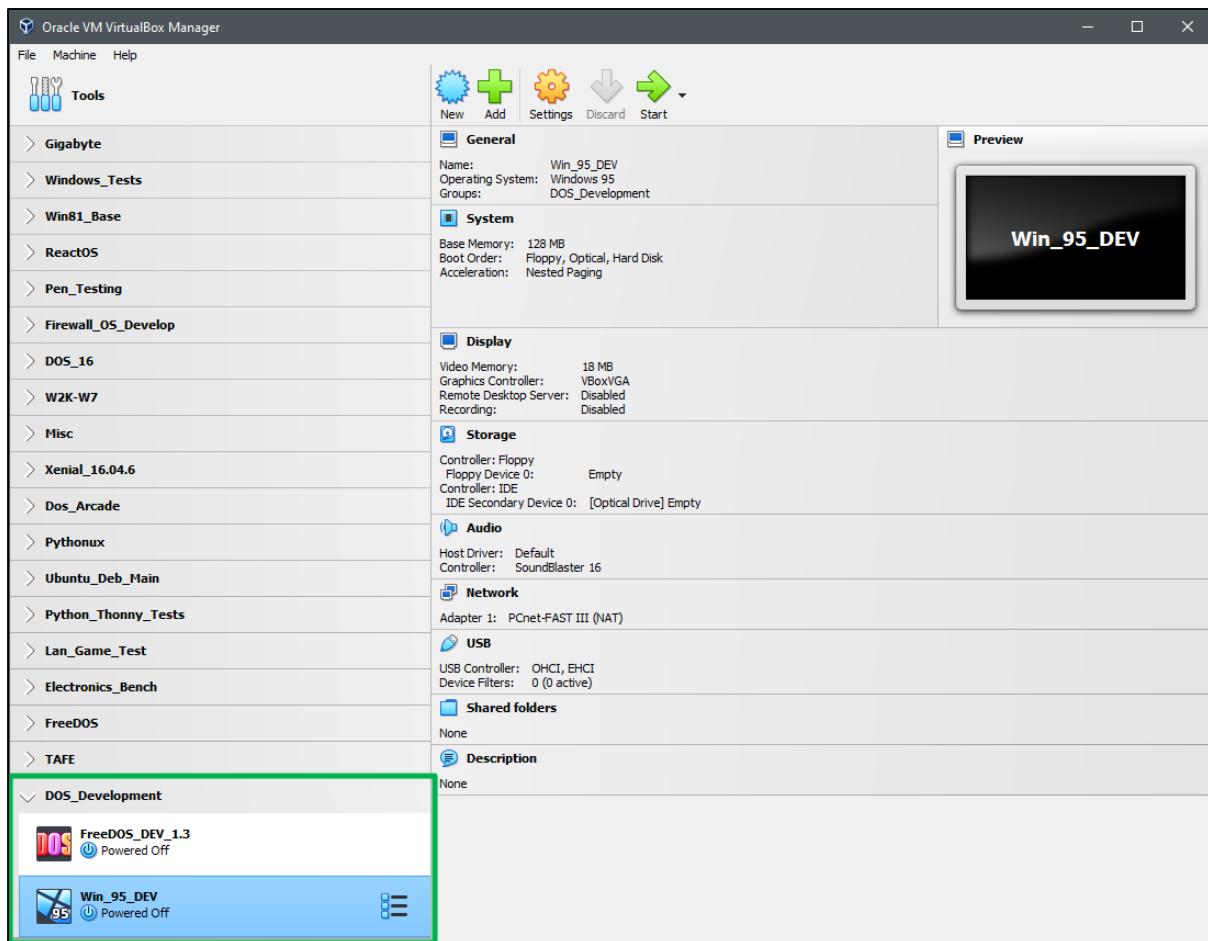
Next right click on the “New Group” and select “Rename Group” and [Enter] to save the new name.  
All Virtual machines and files in the group will be moved to a subdirectory of the group name.

## A Beginners Guide To DOS Programming



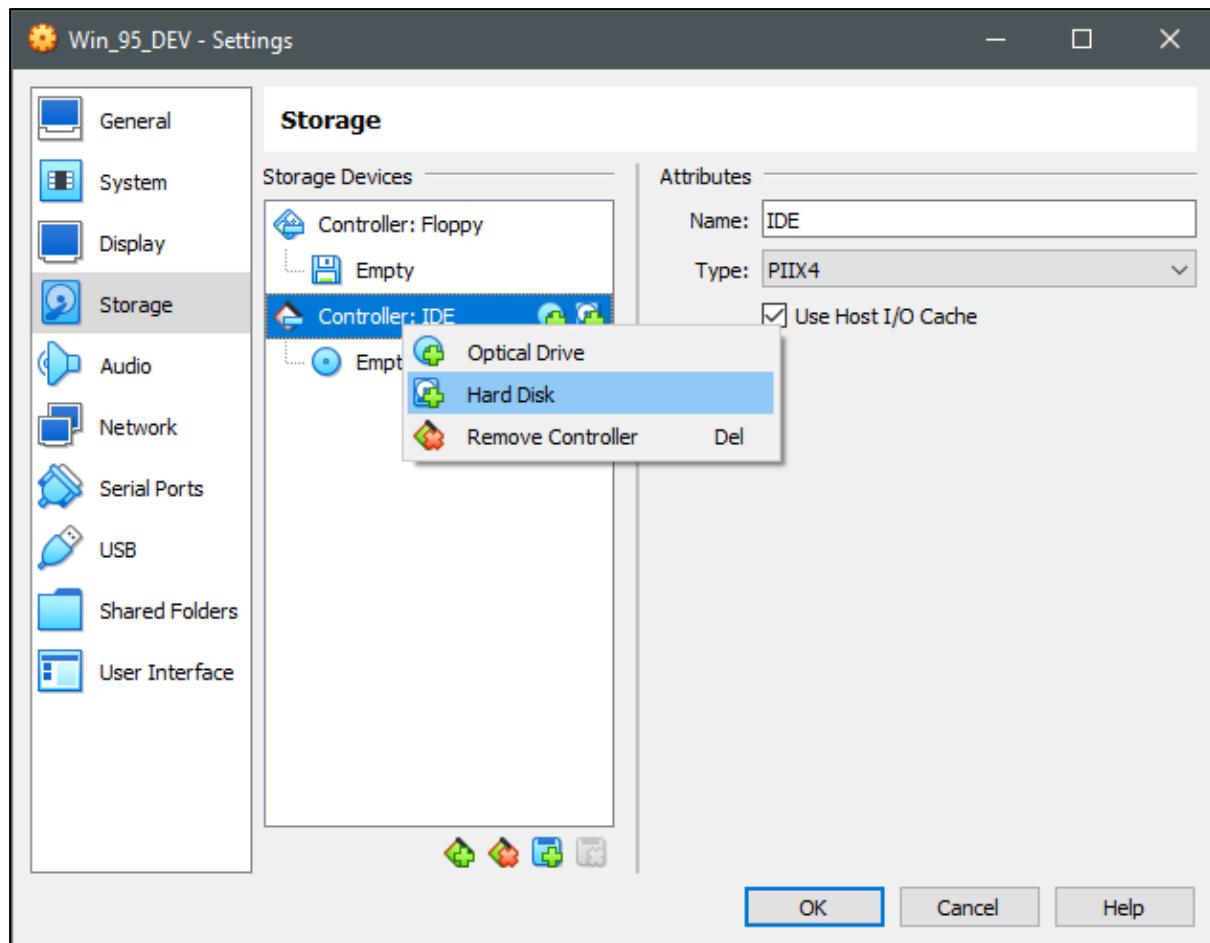
In the accompanying FreeDOS guide I will also be adding the FreeDOS DOS development environment to this group. In the following I have both FreeDOS and Windows 95 in the group "DOS Development".

## A Beginners Guide To DOS Programming

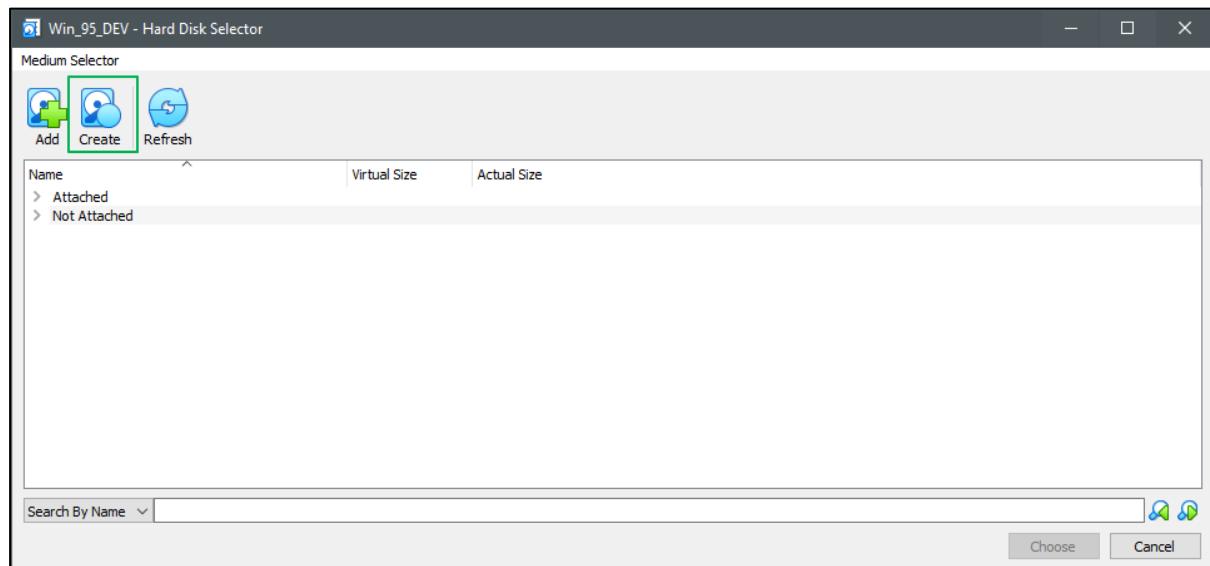


Next go to “Settings” for your virtual machine. We can now create the VHD based virtual drive. I typically use Windows as my host system for virtual machines. Mounting a Virtual Hard Drive (VHD) is a native operation in windows 8.1 onwards making read/write access to the guest OS hard drive easily accessible from the host system. If you are using Linux as your host OS for VirtualBox virtual machines don’t worry as a command line tool “libguestfs-tools” will allow you to mount the \*.VHD and internal file systems. Keeping in mind that DOS and early windows cannot install guest additions and as such we cannot easily set up shared folders through VirtualBox, but don’t worry there are many other easy ways to transfer files to your virtual OS.

Click on the “Controller: IDE” and Select Add Hard Disk.



In the next windows select “Create”.

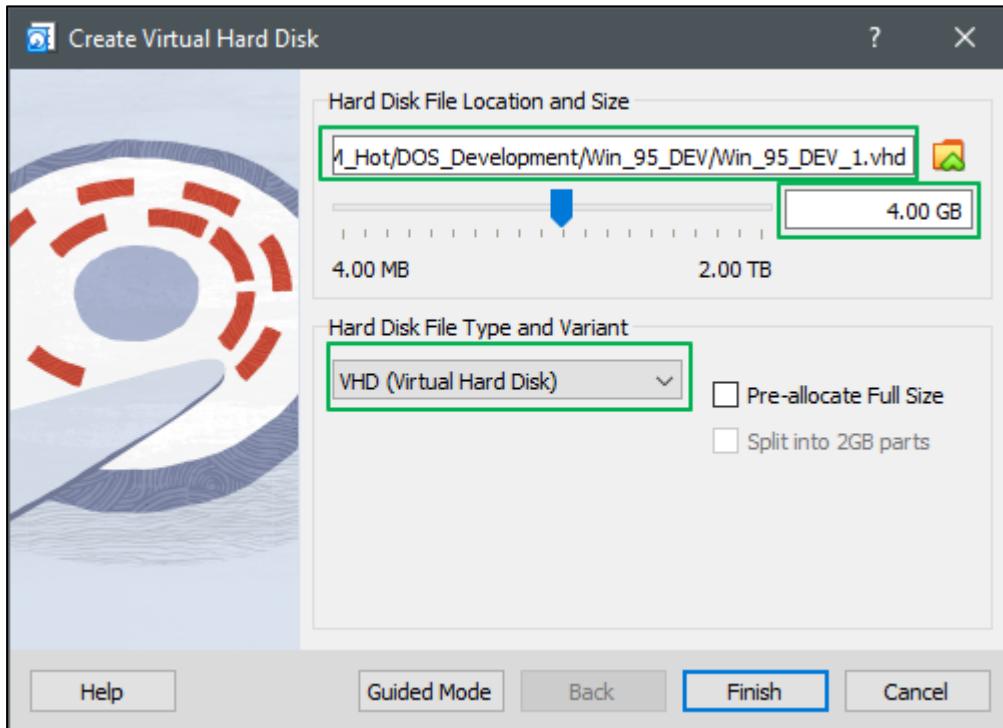


In the next window you can create a hard drive for your machine. Windows 95 does not require a large drive but will expand in size quickly if you are setting up development environments and resource files so I recommend using at least 4GB. Be sure to select the VHD Disk file type. Click

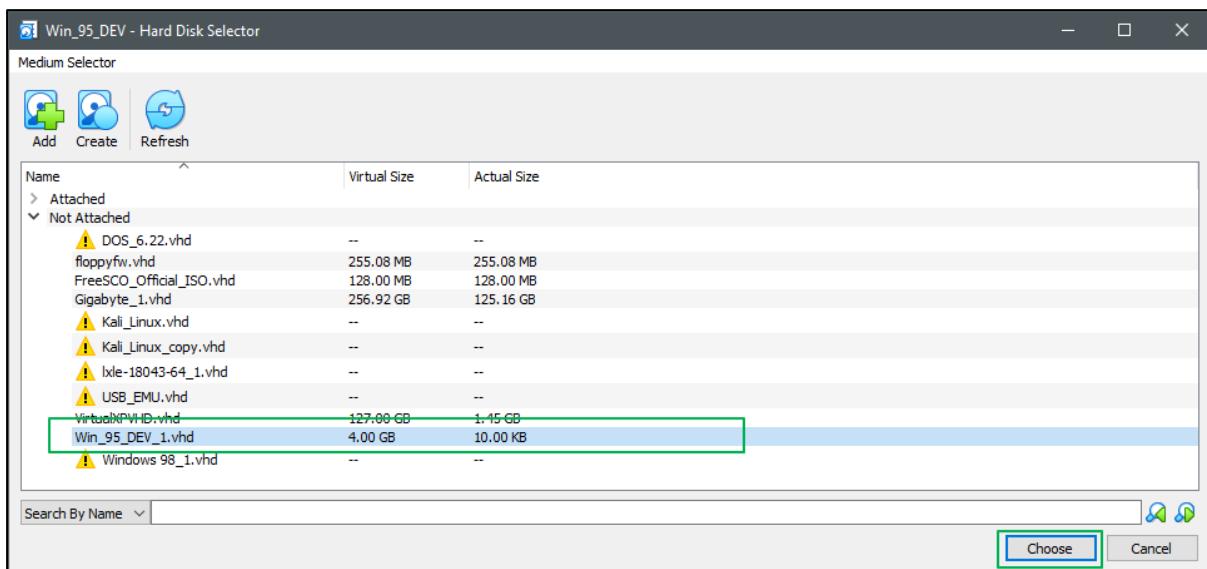
## A Beginners Guide To DOS Programming

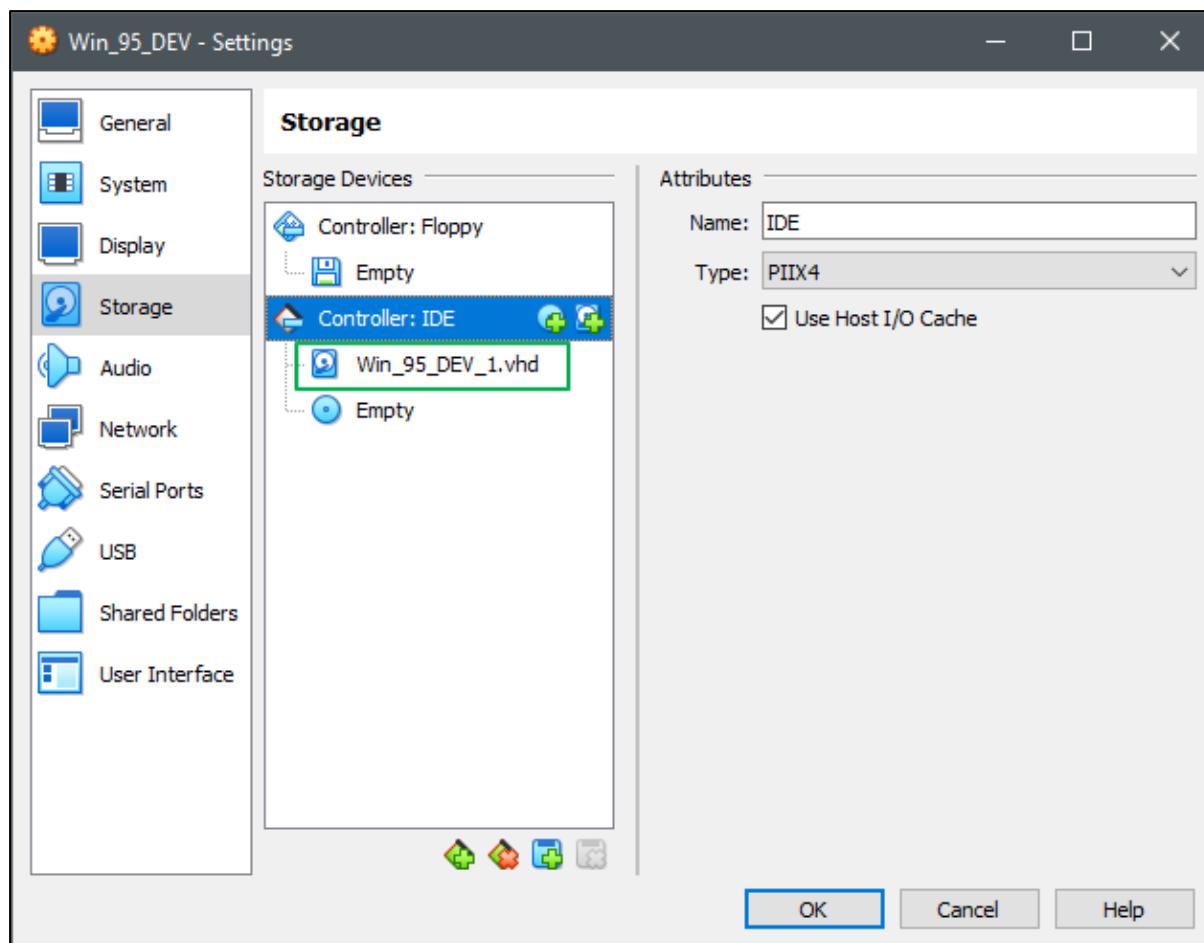
“Finish” to continue. Take note that the path for your group and virtual machine are correct. Take note of the virtual hard drive name.

In VirtualBox V 6.x select Dynamic drive and for V 7.0 leave [ ] pre-allocate full size unchecked. This allows the drive to expand in size only as needed and saving some drive space on the host OS.



Next scroll down to the “Not Attached” section in the virtual media manager window and select the drive name that was created above. Select “Choose” to add the drive to your virtual machine. Usually it will already be selected/highlighted.





I most often attach a second 2GB virtual hard drive for file storage, sorting and backups. I usually attempt to keep my main system drive as clean and uncluttered as possible. This makes it easier to create emergency backup and restore points for the virtual machine while keeping the backup size a little smaller.

To make use of an additional drive in Windows 95 you may need to “Initiate” the drive, and then create a partition and format to Fat32. The simple way of doing this in Windows is to open “Disk Management” and select “Action -> Attach VHD”. Navigate to the location of your guest virtual hard drive and select OK. The virtual hard drive will now be mounted and accessible in the same way as any other hard drive. Note that you cannot have the drive in use in a running VirtualBox machine and the Windows drive manager at the same time. Don’t forget to unmount the VHD from your host OS before starting your virtual machine guest.

After mounting the new VHD you will see the notice “Not Initialised”. Right click on the notice and select “Initialise Disk”. Be sure to select Master Boot Record (MBR) for the partition type when asked. Next create a New “Simple Volume” and partition the full 2GB of the drive. Select FAT32 as the file system.

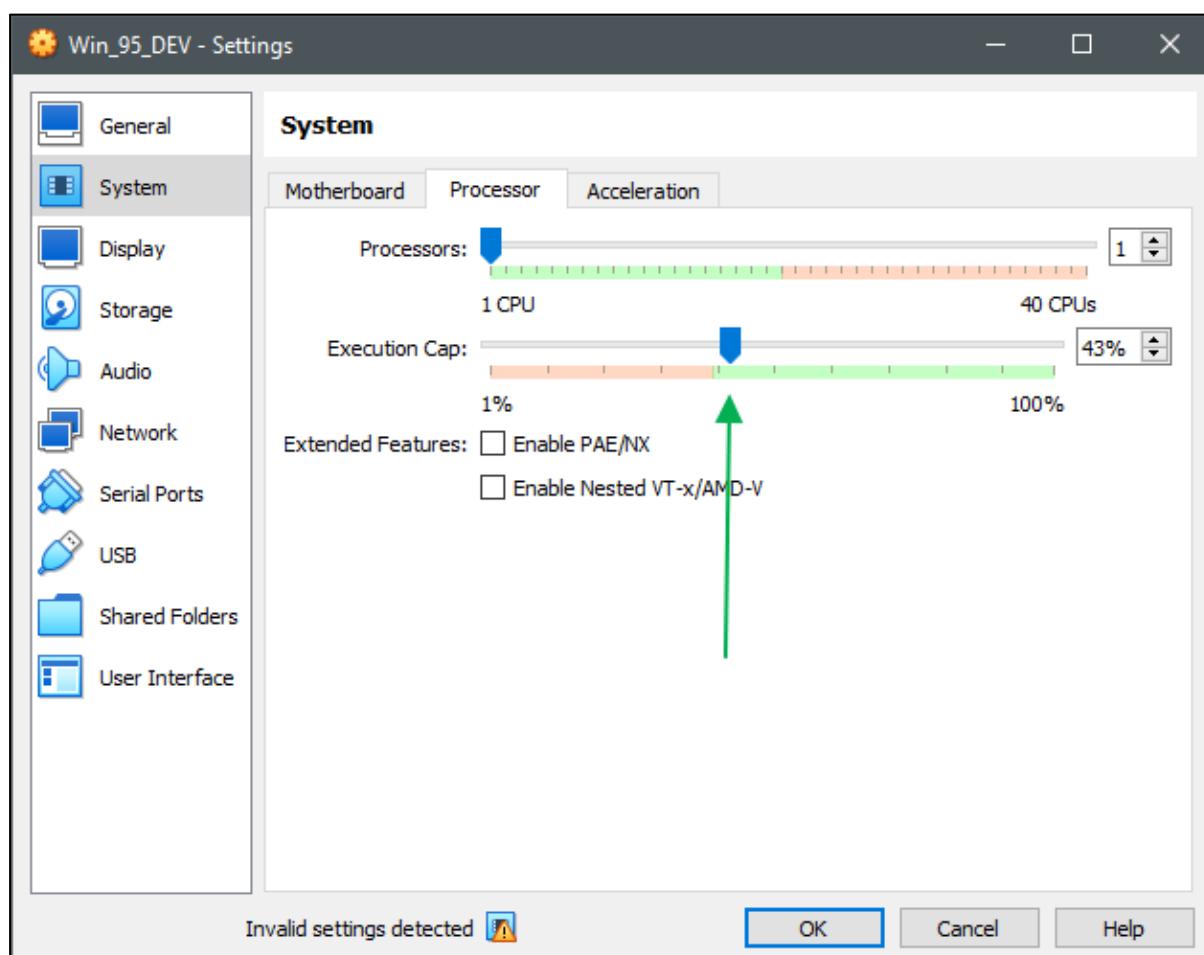
**Note:** You can also initialize and create a DOS partition by using the FDISK tool from a DOS boot disk when the Guest OS (VM) is running. Or by booting the guest VM with a boot disk such as GParted.

Note that FAT32 is only supported under FreeDOS and MS-DOS 7.0 (Windows 95b) onwards. If you wish to use other DOS operating systems you will need to format the drives as FAT16. Although FAT32 supports drives up to 124GB FAT32 cannot recognise files larger than 2 GB. Another problem is legacy issues in applications between DOS16 and DOS/Win32. Some system applications can have problems recognising drives above 2GB. I have found 2GB drives a reasonably safe size for FreeDOS and Windows32 systems. The maximum drive size for FAT16 is 2GB so this is a safe drive size for backward compatibility. The 4GB system drive should not be an issue while running DOS application under Windows 95 but be aware just the same.

If you are using an operating system that does not support the mounting of Virtual Hard Drives, you can easily boot the guest machine with a live CD (ISO) such as GParted to initialise and partition the drives, or use a Windows 98 boot disk image and FDISK followed by FORMAT. The FORMAT tool is not available on Win9x boot disks and you will need to mount a Windows 9x install CD to access it.

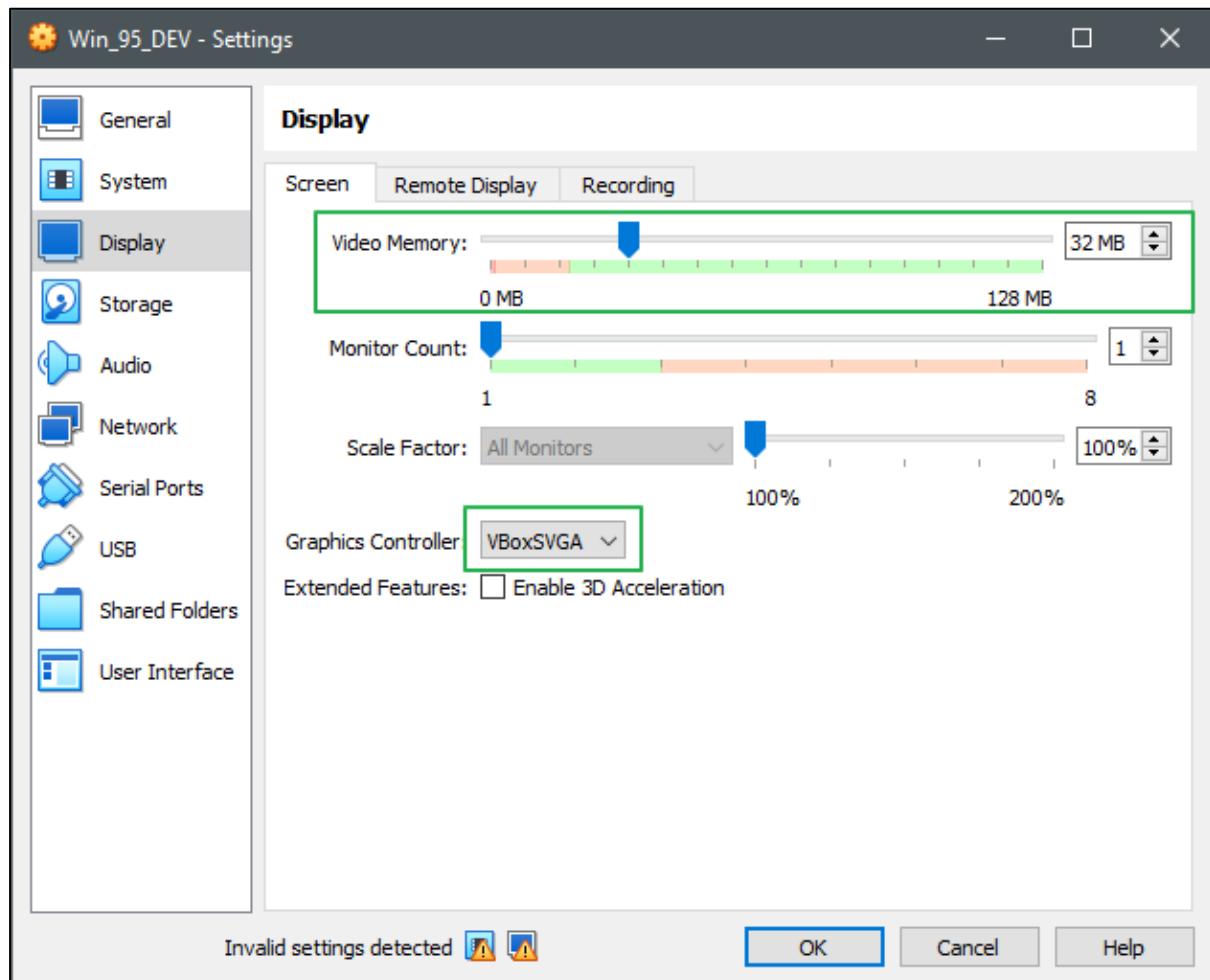
Windows 95 has known issues with hardware acceleration and fast CPUs in VirtualBox. Under CPU execution cap reduce the speed to the lowest percentage in the green. If this does not work on your host then you may need to apply one of the 3<sup>rd</sup> party CPU speed fixes for Win95 before it can boot. I have personally never had a problem booting Windows 95 in VirtualBox or needed the patches.

You can gradually test higher CPU settings after the full install has completed.



Next select the “Display” tab. select 32MB for the video memory. You can go up to 64 but 32 is typical for high end cards of the period.

Also change the Graphics controller from VBoxVGA to VBoxSVGA.



At this point you are ready to install the Windows 95 Operating system.

### File transfers

There are many ways to transfer files to and from a VirtualBox guest. I have already shown the Windows 8.1 + drive manager solution by mounting the VHD file as a Windows drive.

It is worth becoming familiar with some of the floppy drive image types as well as the tools for mounting these images. Most DOS software was distributed on floppy drives and this is reflected in the large assortment of “Floppy drive images” that are available. A bare bones DOS OS typically expects software from a floppy drive, and many of the applications available expect the install to be done from a floppy drive. Take the time to view the contents of a DOS application install disk as you will often find Batch install files and other install configuration files for the application. Many application install scripts and executables will often contain an uninstall switch “install.bat”

/uninstall". Read the documentation using an image mounting tool to become familiar with the documents and options.

Also note that Windows 9x often expects Windows application installers to be run from the C: (OS System) drive. If you attempt to run a Windows application install from a drive other than A: or C: and it fails, then move the installer to the C: drive and try again.

Here are some other quick methods to transfer files into a running DOS guest.

### 7-Zip

<https://www.7-zip.org/>

7-Zip can extract files from many compressed archive types including VHD and VDI files. Remember that it can only view and extract and cannot create or update virtual hard drives.

### Folder2Iso

<https://www.trustfm.net/software/utilities/Folder2Iso.php>

Folder2Iso is a portable Window and Linux application that creates an ISO from any folder. The root folder can contain sub-folders.

It's a GUI of mkisofs.

Works under Windows XP, Vista, 7, 8, 10 and Linux.

It is a simple tool to create an ISO from a folder of your selected files. The ISO can then be selected and mounted as an optical drive in your guest OS. This is one of the simplest methods to transfer files to a virtual machine. You can delete the ISO once the transfers are done.

### ImDisk Toolkit

<https://sourceforge.net/projects/imdisk-toolkit/>

Windows XP, Vista, 7, 8, 8.1, 10 or 11 (32 or 64-bit).

This tool will allow you to mount many of the common disk image types. It is also very useful for viewing floppy disk drive images which are a common medium for DOS applications. You can also create optical and floppy images with this tool.

Some removable image types require specialised file systems and can be a little bit tricky to work out. I find the most simple is to use a renamed copy of the FreeDOS 1.44MB boot disk and use it as a floppy net for transferring files between a host computer and a virtual machine guest. In most instances using the folder2iso tool is far more convenient for larger file systems.

There are many other image manipulations tools available. The above are just a few of the most useful.

### HFS

Windows 95 has TCP/IP networking and Web browser capabilities. We can use tools such as Http File Server (HFS) on our host machine and then use the Windows 95 web browser in our virtual guest OS to transfer files over http in the browser. I will cover a section on using HFS in a later chapter. The Common browsers in Windows 95 will be Internet Explorer or SeaMonkey.

rejetto <https://www.rejetto.com/hfs/>

See the section “HFS over IP”.

---

## Setting up Windows 95

### Base install

[95C (OSR2.5) is the same install as below. Updates for 95C begin at IE 4.1]

Note: Windows 95 is renowned for freezing at the shutdown/reboot BIOS screen stage. You may need to use the VirtualBox “File -> Close” “Power off the machine” on a regular basis.

Installing Windows 95 is very dependent upon the version and upgrades required for that version as well as being depending upon the hardware requirements of the individual PCs. As such there is no “One size fits all” approach to setting up Windows 95. The following is focused upon a VirtualBox virtual machine install.

The essential flow is from a base install of 95B, CPU patch (If required), Dirextx 8.x update, Scitech Video, and other essential drivers and fixes. Windows 95B ships with IE 3.x and this is important as from IE4.0 onwards as IE and the Windows UI desktop become an integrated system component. Many of the Windows 95 system and desktop updates are part of the IE install up to IE 5.5. In short updating IE updates Windows 95. Note that Windows 95C base install is essentially the combined upgrades from W95b up to IE 4.0 as well as some other system fixes. You can begin with 95C and just apply the IE updates from IE 4.1 to Version 5.5. The Win95C installer is almost identical to the Win95B installer with some additional setup options for IE 4.0 and the active desktop environment at the first logon.

There is a very loose order of updates from W95B to the latest end of support. Some installs and updates may require the update of another component before proceeding. To make this even more complicated some components such as IE for example may require other updates so you never really know for sure until you encounter a xx component missing or required. The VirtualBox install shown here flows quite well without any issues but this does not guarantee problems will not be encountered on other Host operating systems and hardware.

Another common issue is naming and versions as MS updates from the period can often be labelled with generic names. For example a DirectX installer can have the same name, but be from a number of different version, or C runtime libraries that can cover multiple versions but have the same package and installer name. In many instances such as Internet explorer some versions are required to be installed in order, but some updates can be skipped.

Be careful with the differences between Win 3.1, Win95, Win98 and Win-NT as the update names and patch names can be the same across all 4 systems. A Win-NT update will fail on Win95 and vice versa.

There are many sites with good information as to what fixes apply to what problems as well as a lot of information and guidance around the large number of system updates that are available. I will list some of the common sites below, or if they are no longer available be sure to check archive.org (Wayback Machine) for copies of the original web sites.

Many applications labelled as being Windows 95 compatible will not run on earlier editions of Windows 95 including 95B OSR2 until the Internet Explorer and other system components have been updated. The UI and Windows system APIs introduced by IE are a necessity for these applications to work.

If you have not already done so download the “Windows 95B (OSR2) install ISO” or use your own purchased CD ROM as well as the floppy boot disk image. You may find it more convenient to use the correct ISO version download and your legitimate install key from your CD ROM. Place the ISO and boot disk image in a directory that is easy to find when using the virtual guest. I typically have a dedicated directory of ISO files, or place them in the virtual machine file directory. You will need to extract the ISO file from the downloaded archive if it is a zip file.

You can often find these at winworldpc.com or archive.org. Keys are easy to find or you can also use the MS master key.

The original OEM CD rom is typically about 597MB (611,594KB), but can vary a little depending upon the compression used for the ISO image. Be aware of potential non OEM ISOs. Do some research on the most trusted sources before choosing an ISO image. Winworldpc is generally accepted as safe.

“windows95b.iso”

Note: You can obtain a copy of the Windows 95C (OSR2.5) from winworldpc if you wish to try that pathway and reduce some of the upgrades required for Win95B.

Most Windows 95 CD-ROMs and ISOs are not bootable, so you will need to also download the floppy boot disk image. I recommend using the Windows 98SE boot image as it has superior CD-ROM drivers for the ISO file as well as better HDD partitioning tools.

You can get this from winworldpc and many other locations.

<https://winworldpc.com/product/microsoft-windows-boot-disk/98-se>

“Microsoft Windows 98 Second Edition - Boot Disk (3.5-1.44mb).7z”

“Windows 98 Second Edition Boot.img”

You should now have the Install ISO, Boot image and install key available for your virtual machine.

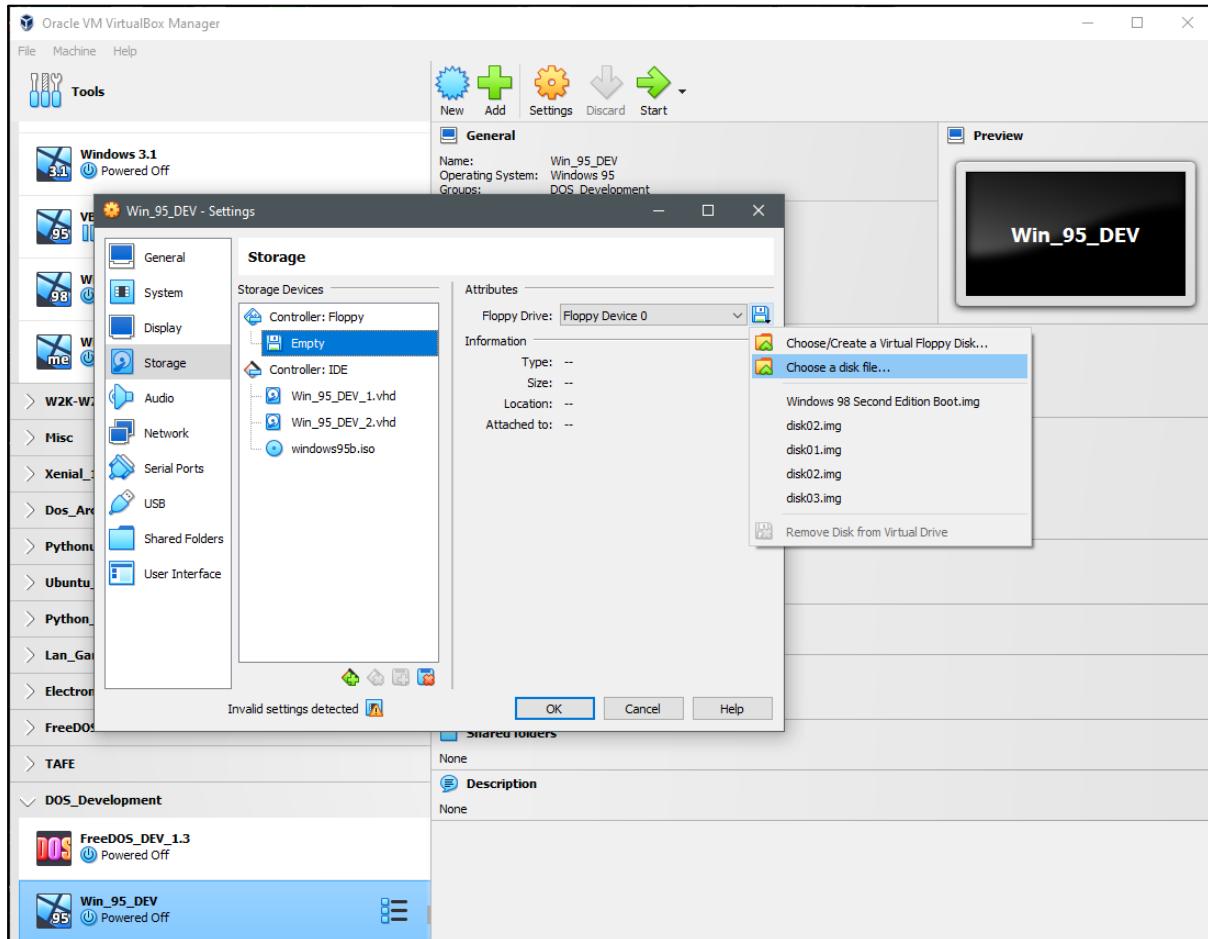
“windows95b.iso”

“Windows 98 Second Edition Boot.img”

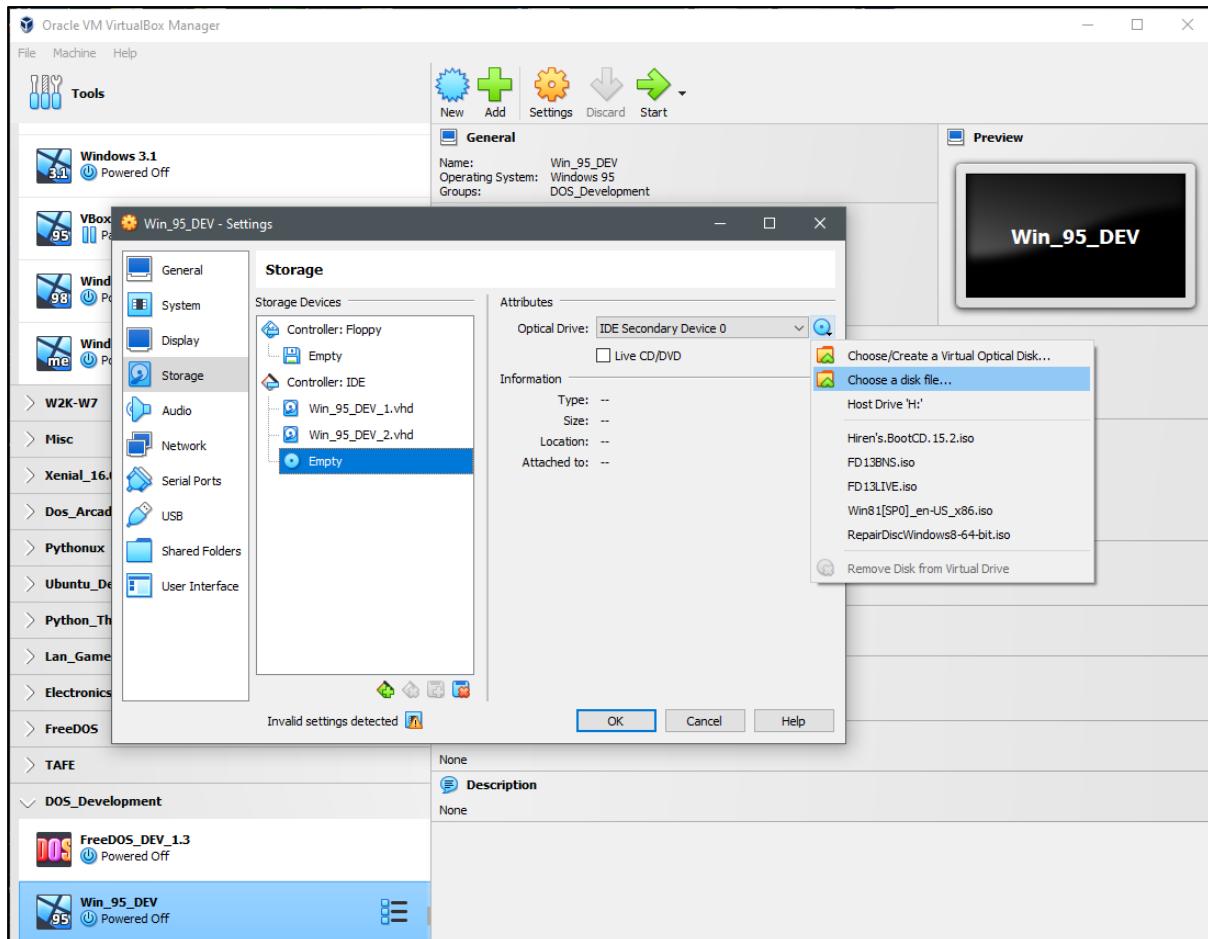
"nnnnn-OEM-nnnnn-nnnnn"

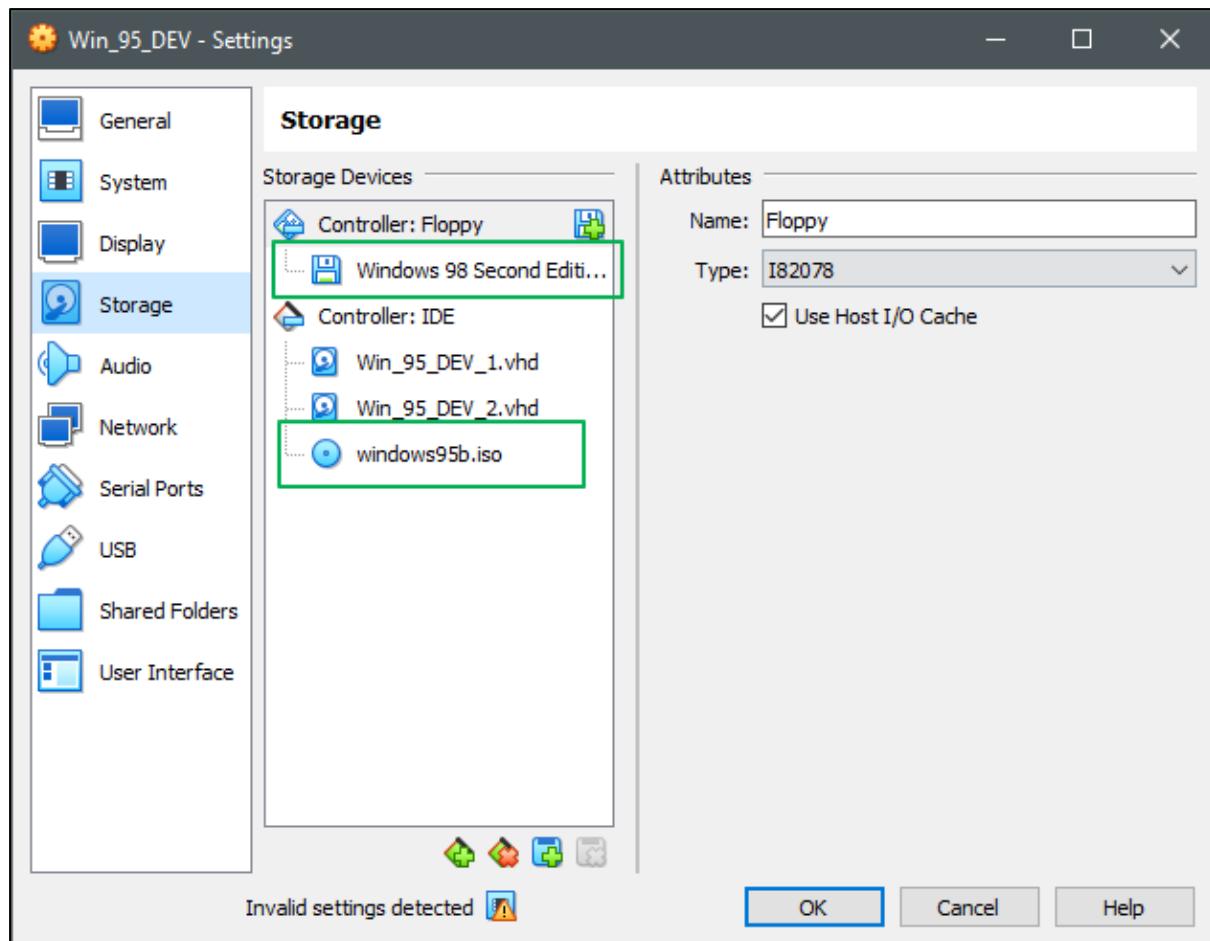
There are many different ways of installing Windows 95 depending upon the version, and how the ISO was created. My personal ISO is bootable and I can install directly from the bootable ISO. The ISO that I am using is not bootable so I will show the most typical install method for windows 95. There can be many variations on this method.

Open the setting panel for the Windows 95 guest and mount both the floppy drive image as well as the Windows 95 install ISO.



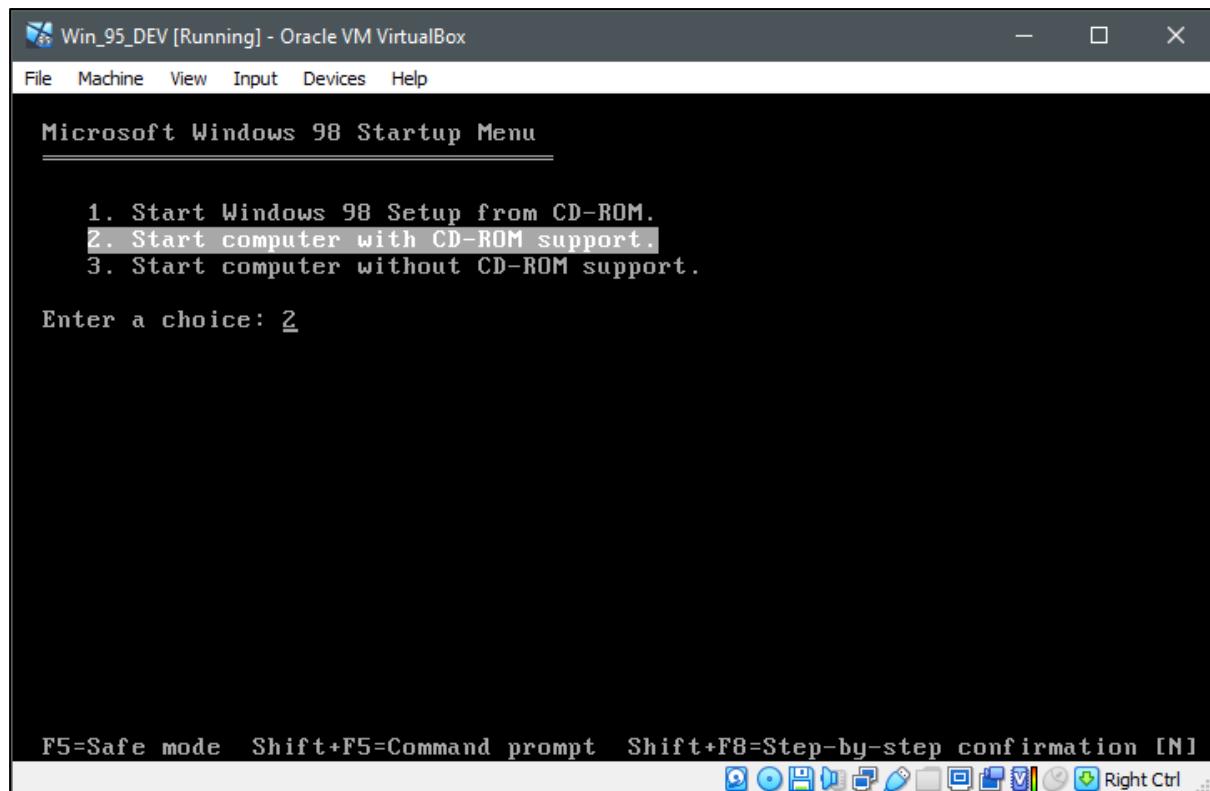
## A Beginners Guide To DOS Programming



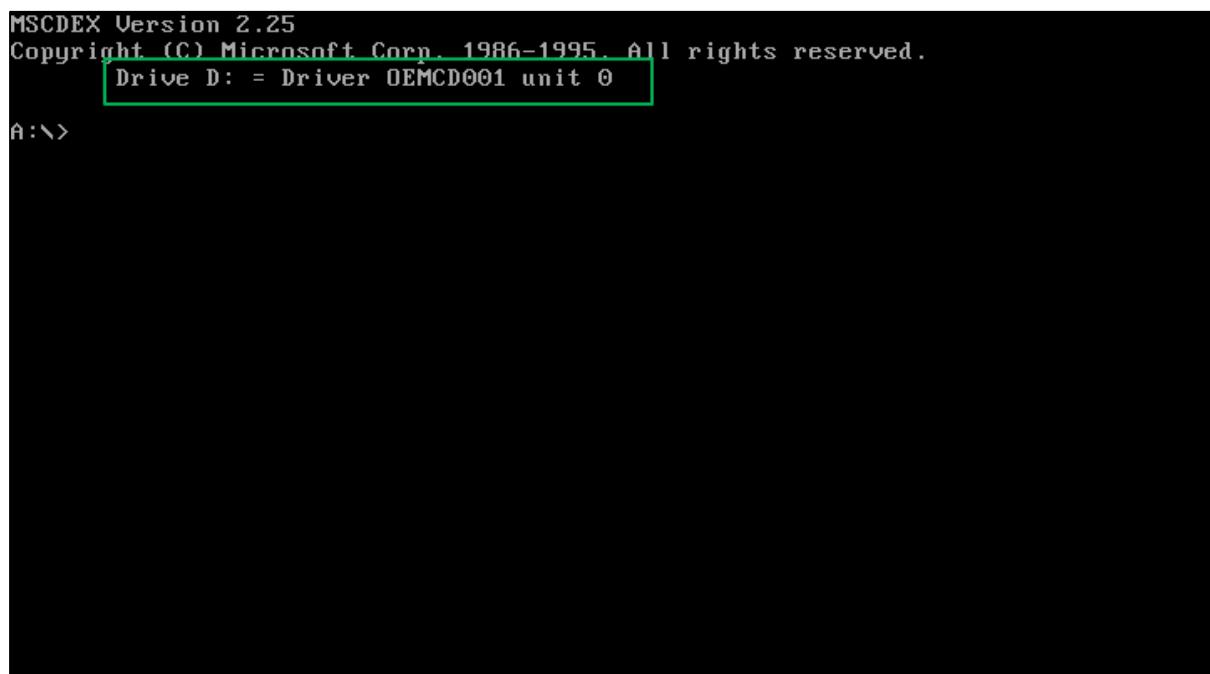


Start the Windows 95 virtual machine then select:

2. Start computer with CD-ROM support.



You will notice that drive D: has been designated as the CD-ROM drive. I have not yet initialize, partitioned or formatted the 2 attached hard drives at this point. C: is always “reserved” as an OS System partition even though it has not yet been allocated.

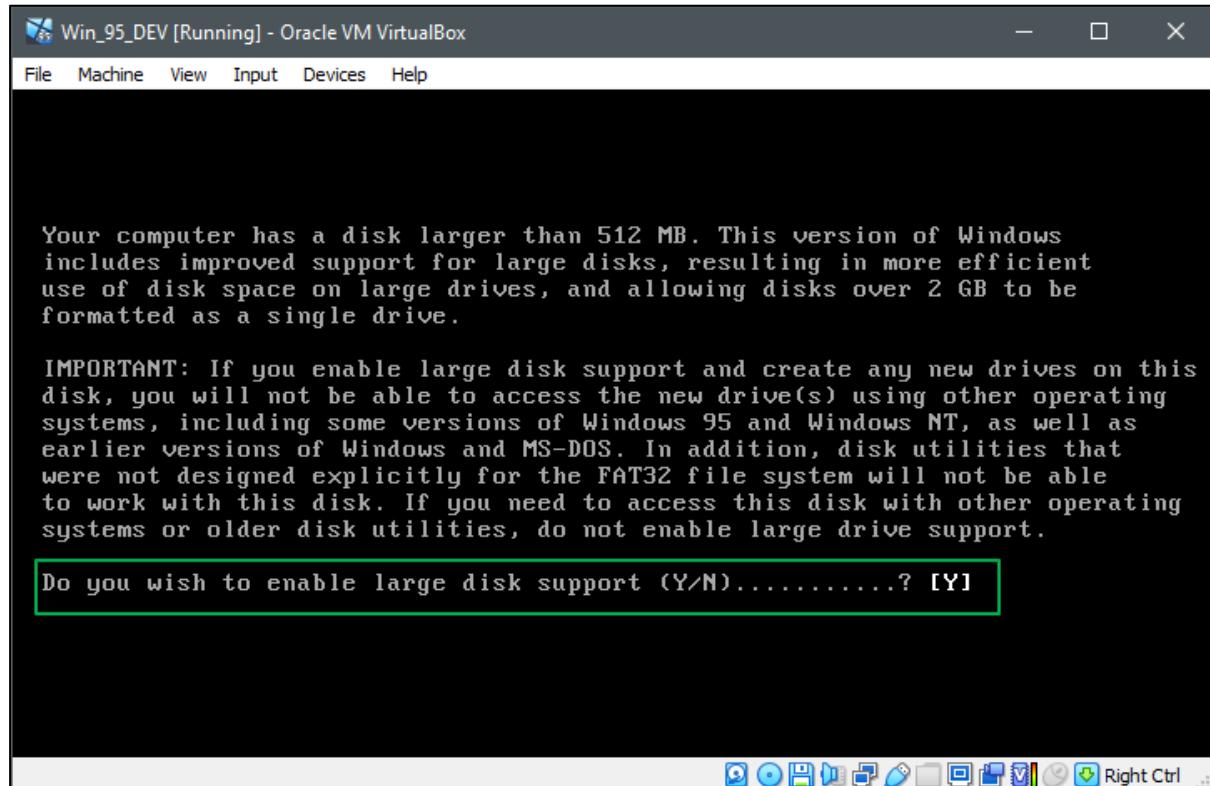


Next we will set up the hard drive partition ready for formatting.

Next type the FDISK command into the command prompt followed by enter.

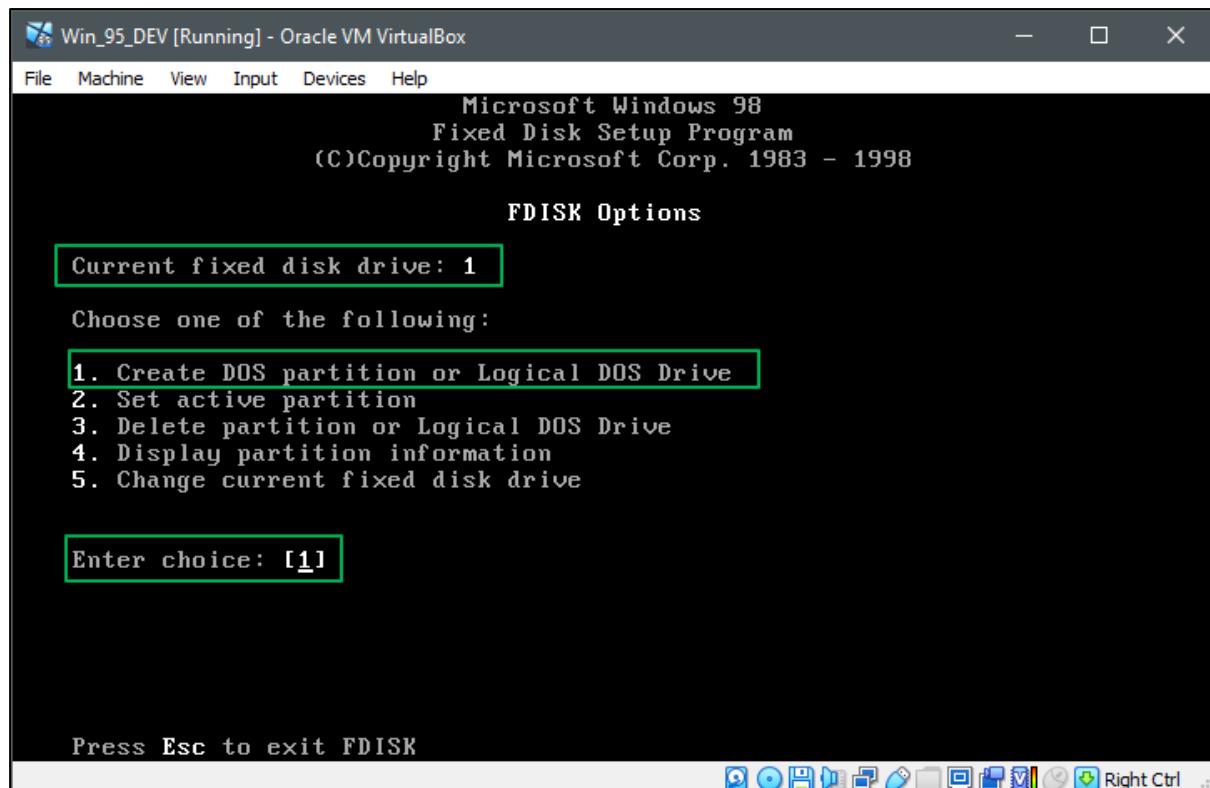
Select [Y] yes to enable large disk support (Fat32 partition tables).

This relates to my notes with regard to early 16-bit DOS implementations that could not recognise large hard drives. Windows 95B (OSR2) or later and FreeDOS can use large hard drives (Fat32).

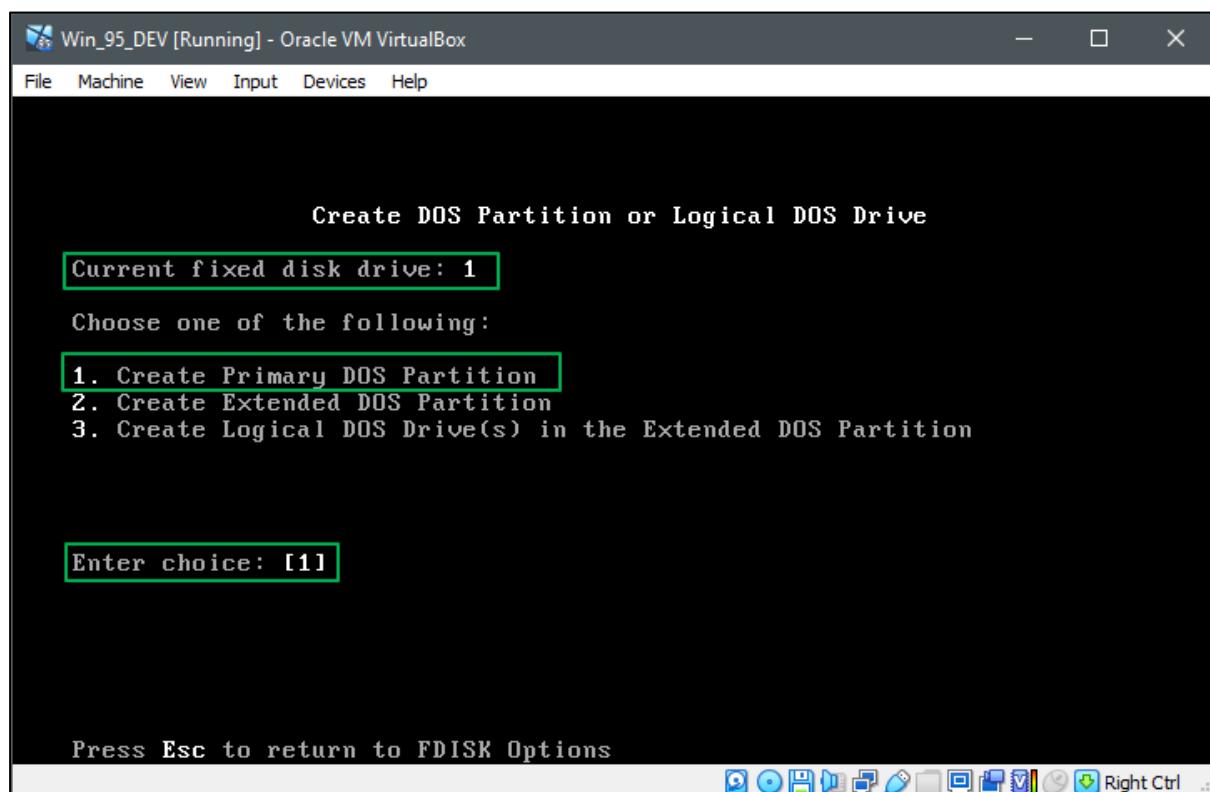


Next we will create a partition on the fixed disk drive 1. If you have created an additional storage drive for your virtual machine you can also set it up at this stage. The additional storage drive will be fixed disk drive 2. We are going to use the entire HDD space for a single partition.

Disk 1 is already selected so choose option 1. to create the partition and select [Enter] to continue.

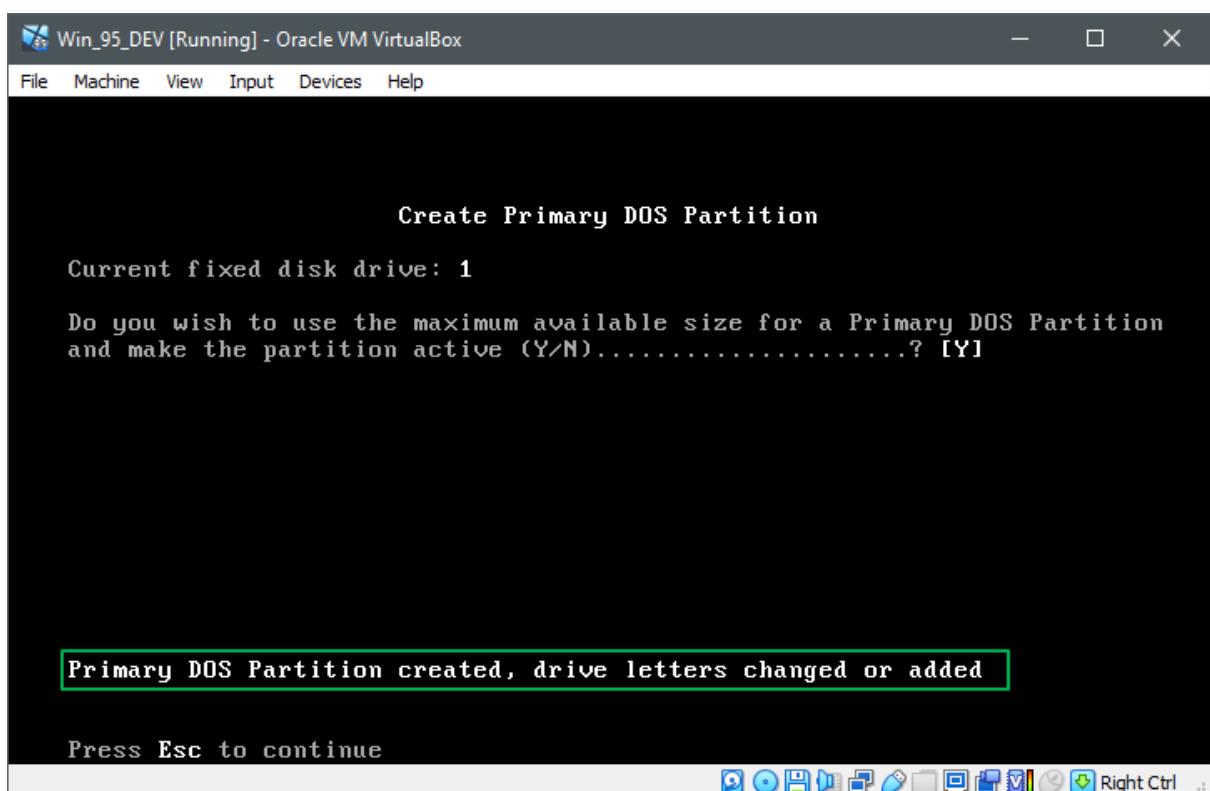
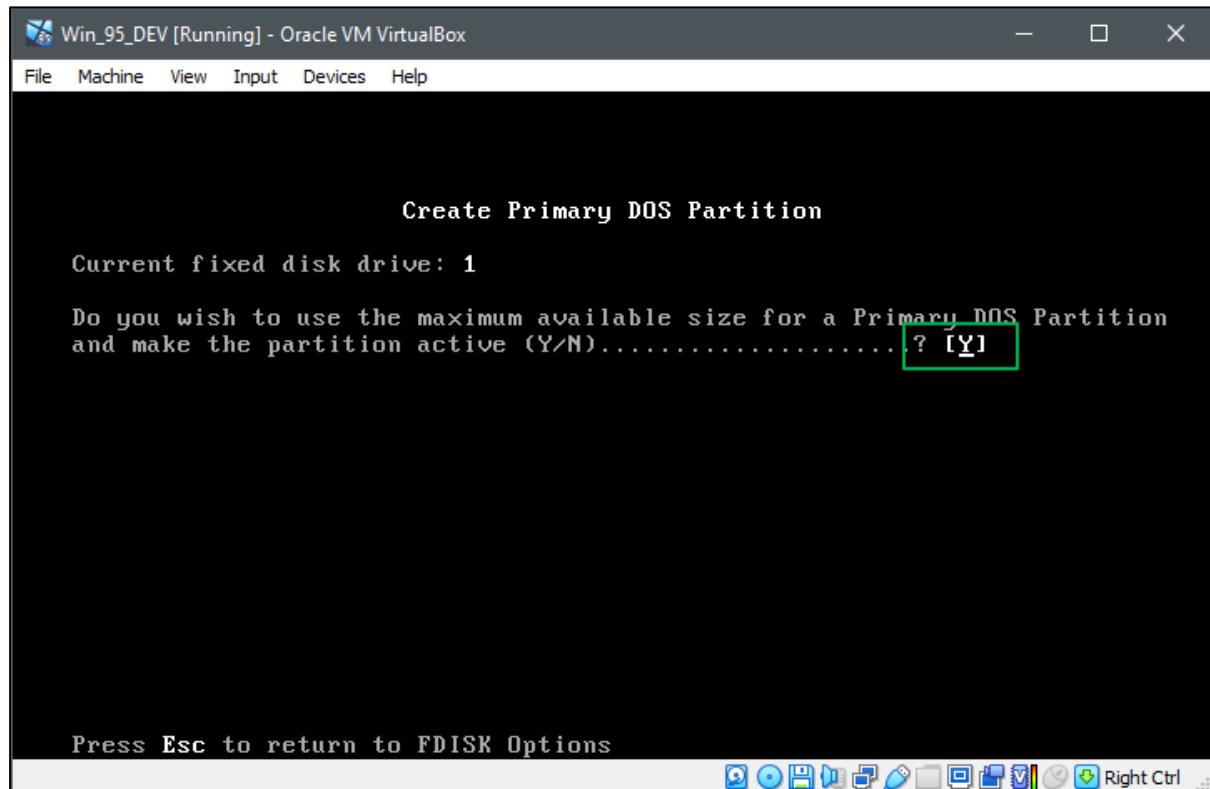


Select 1. Create Primary DOS Partition, and then select [Enter] to continue.



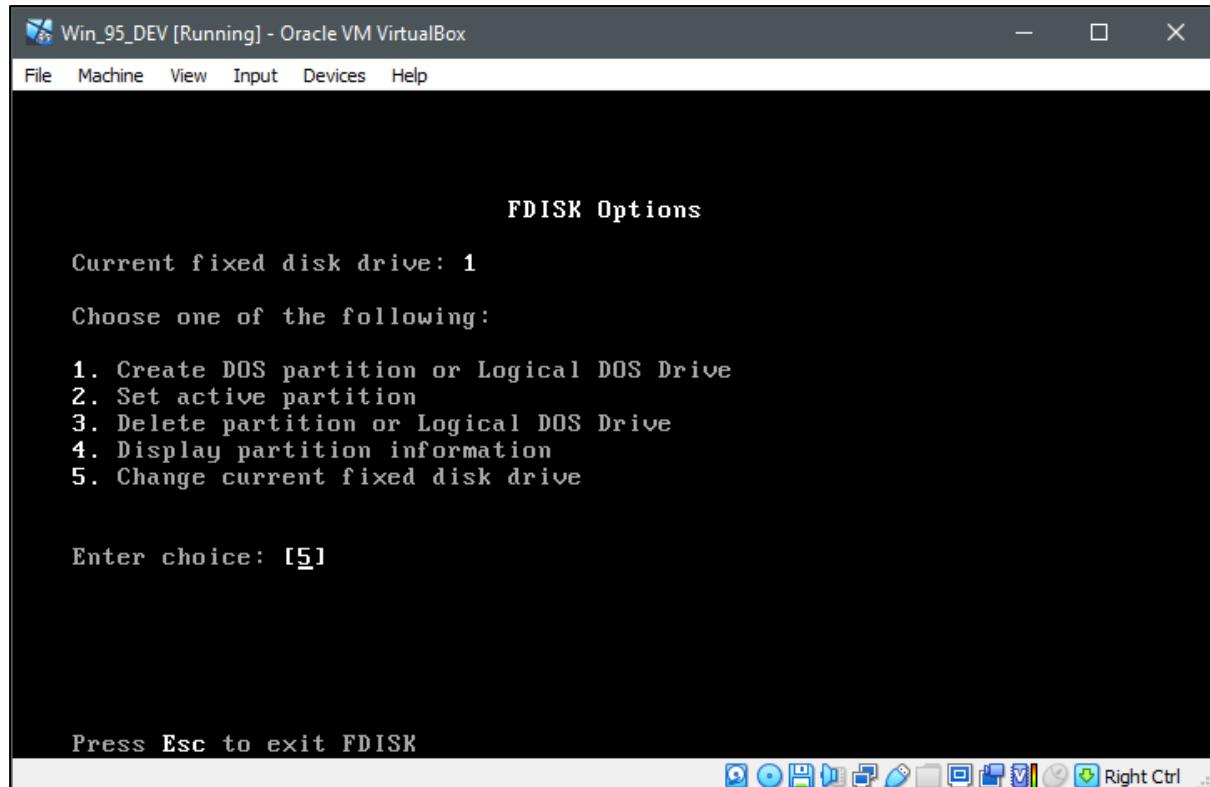
## A Beginners Guide To DOS Programming

Select [Y] yes to using the maximum available space and setting the partition as "Active". The OS install partition must be set as Active to be bootable. Only ever set a single partition as Active. Press [Enter] to continue.

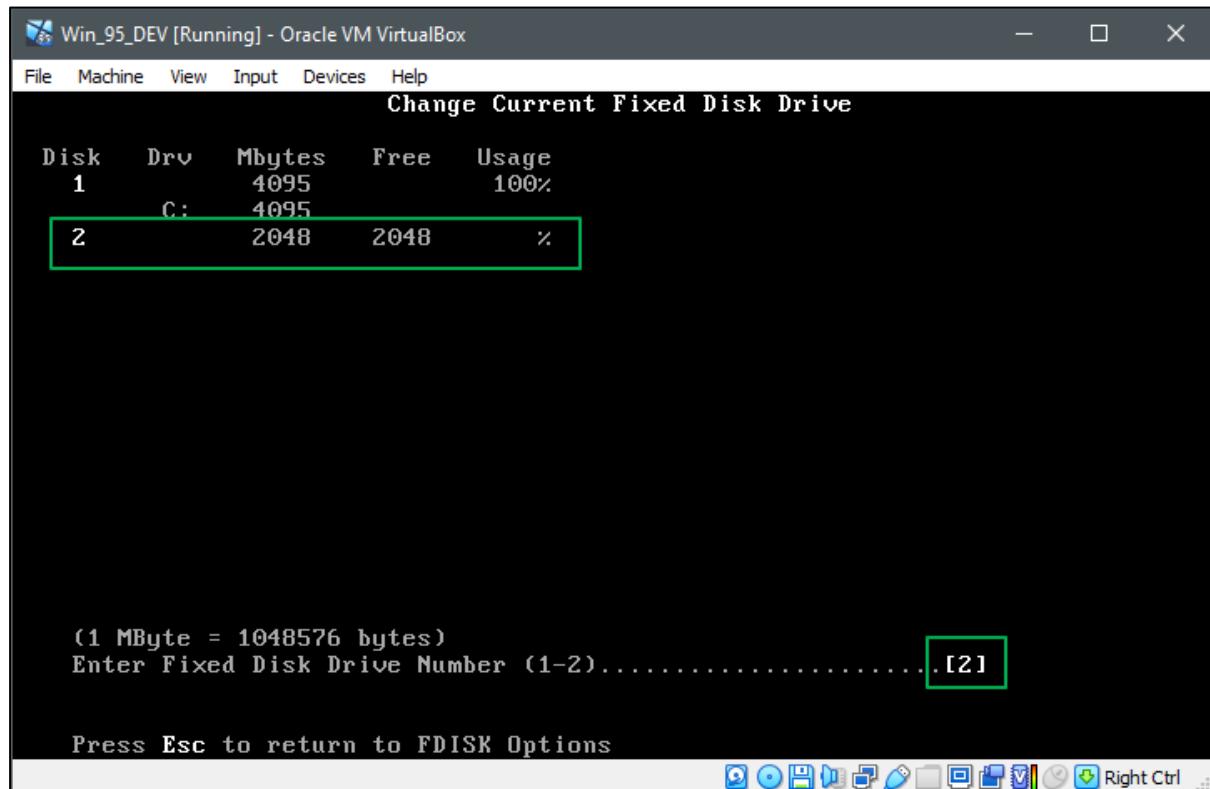


Press [Esc] to finish this task.

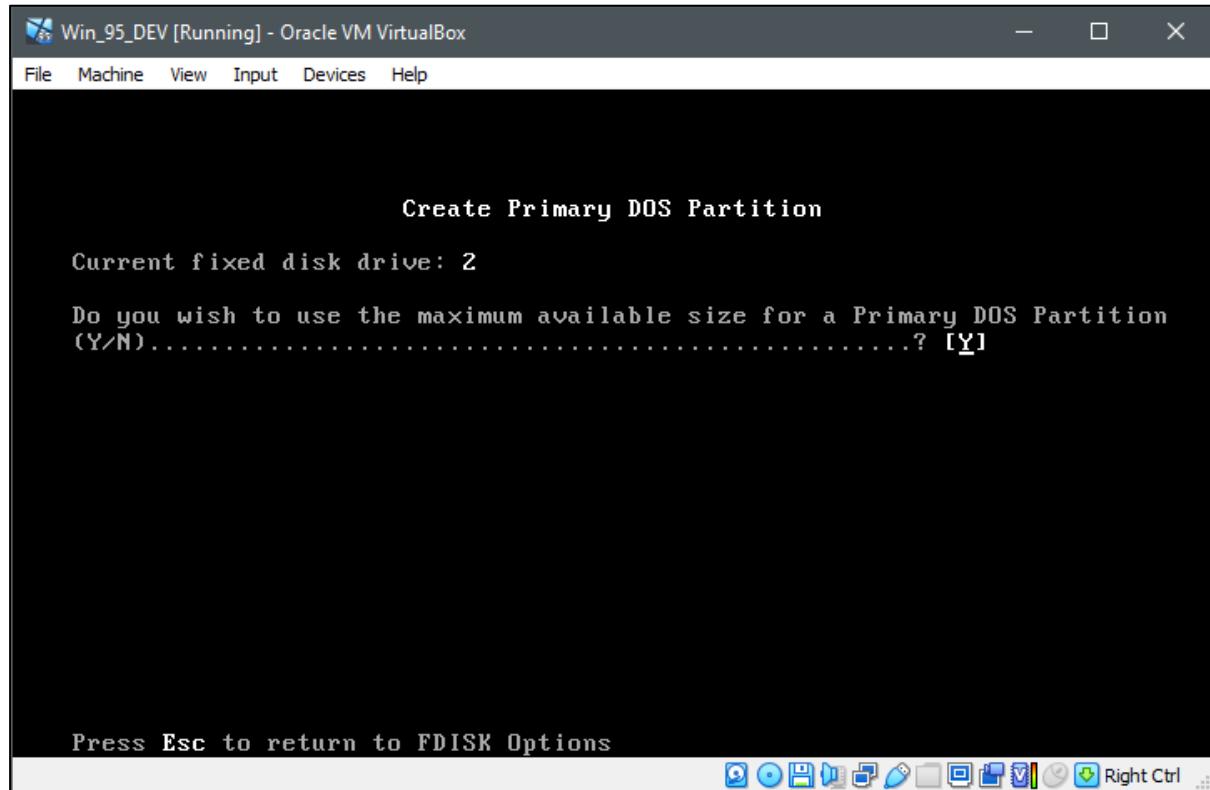
To set up a second drive select 5. Change current fixed drive and then [Enter] to continue.



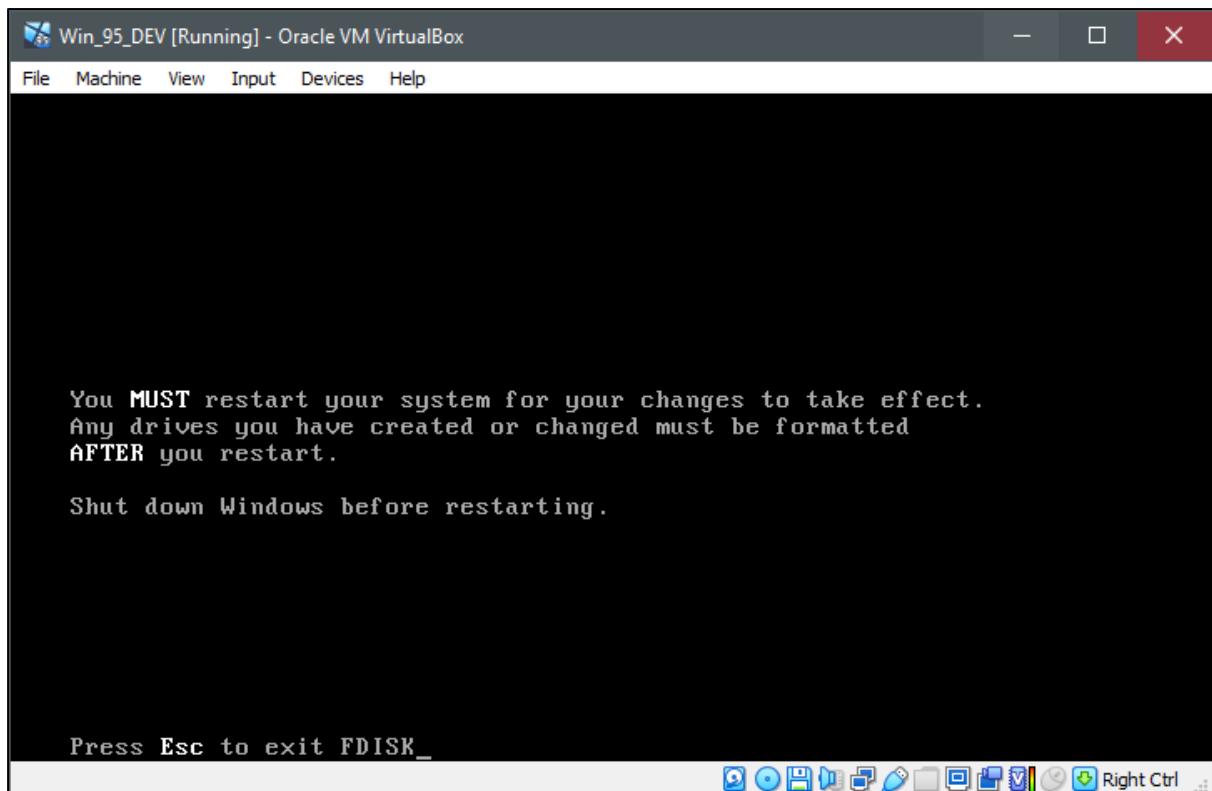
Select Disk [2] then press [Enter] to continue.



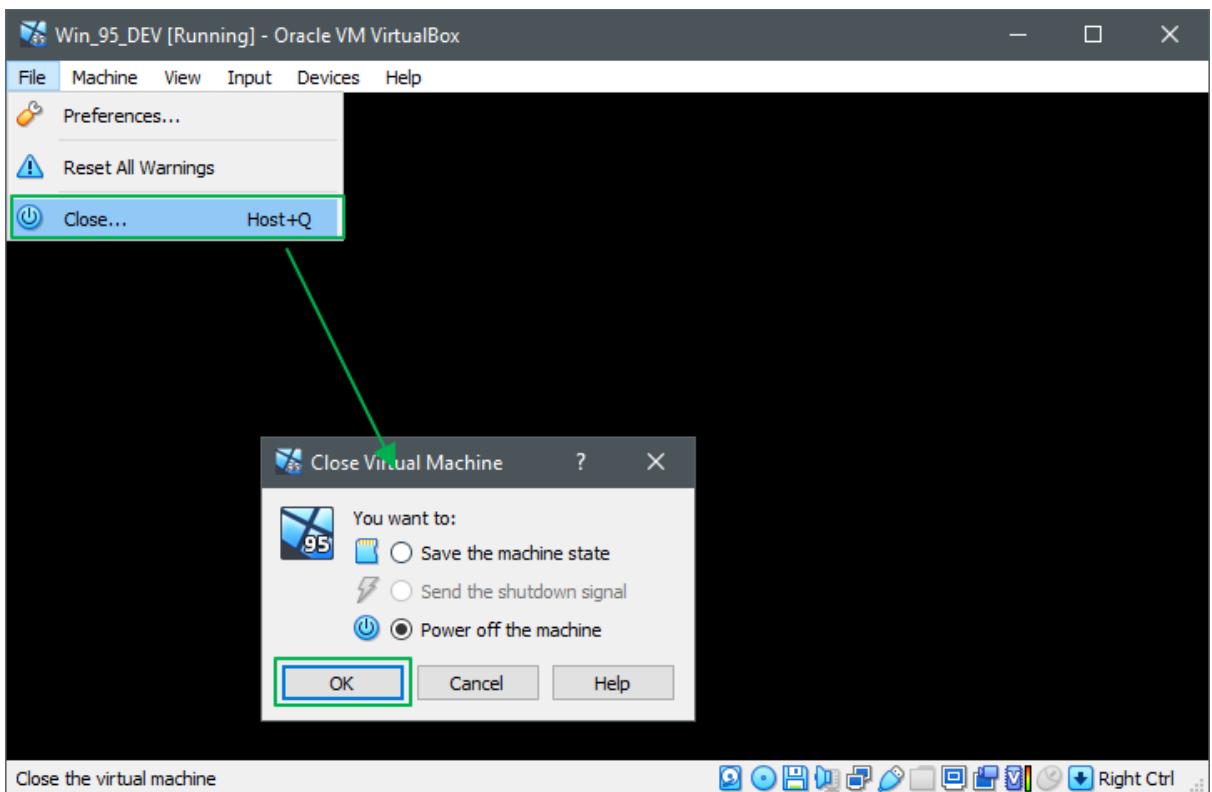
Repeat the previous steps to create a partition on drive 2. You will notice that it won't ask to enable an Active partition this time as Disk 1 is already set as Active.



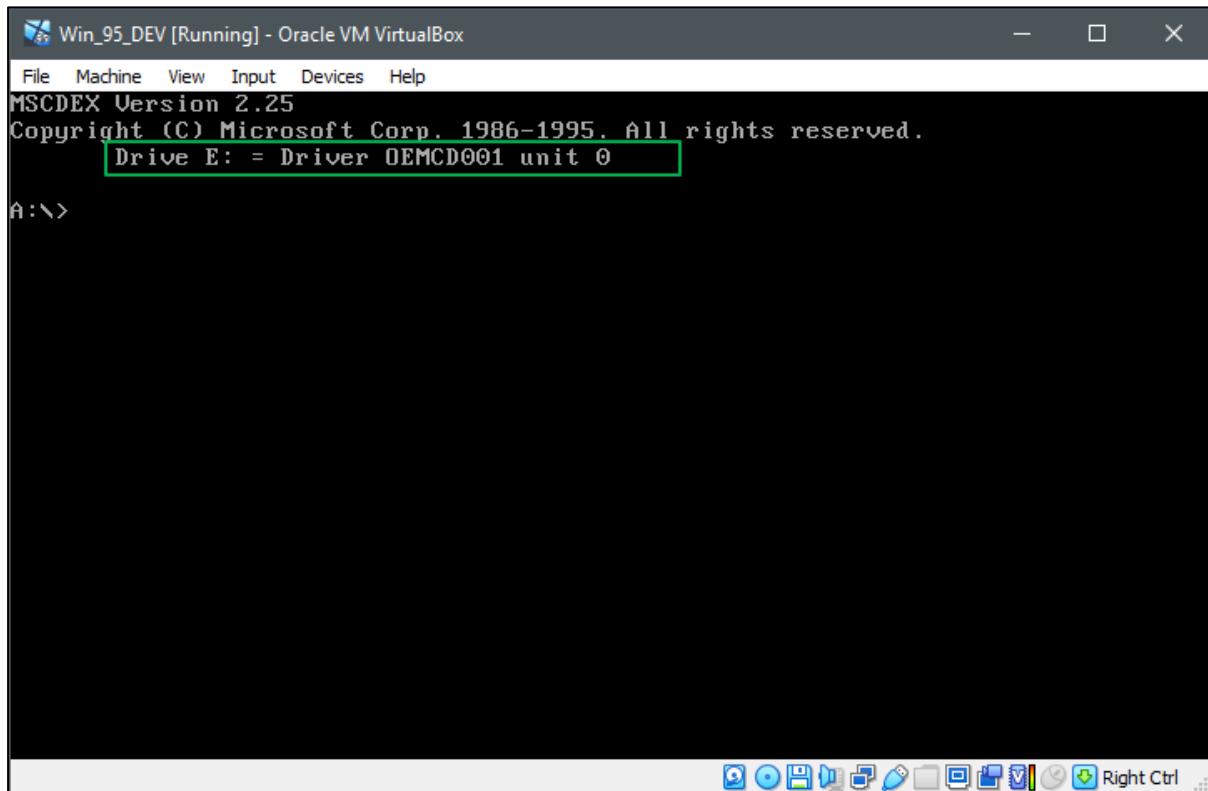
Once both fixed hard drives have been partitioned select [Esc] to exit FDISK. You will need to reboot the virtual machine and restart using the floppy boot drive.



MS-DOS has no default shutdown or reboot commands so just turn off the virtual machine.



Remember to start the floppy boot with CD-ROM support. This time you may notice that your CD-ROM has moved after partitioning both HDDs. Always take notice of the CD-ROM drive when you boot from the floppy drive.



Next we will need to format both drives as FAT32. The DOS boot disk does not contain FORMAT.COM so we will have to navigate to our Windows 95 install drive to locate it. Type in E: followed by [Enter] to change to the CD-ROM (Assuming E: was the CD-ROM drive shown at boot time).

Navigate to the WIN95 install directory **CD WIN95**

If you do a DIR /a /w command you should see format.com in the list.

## A Beginners Guide To DOS Programming

Win\_95\_DEV [Running] - Oracle VM VirtualBox

```
File Machine View Input Devices Help
Bad command or file name

A:>E:

E:\WIN95>dir /a /w

Volume in drive E is WINDOWS95B
Directory of E:\WIN95

[.] [ ] CS3KIT.EXE      DELTEMP.COM      DOSSETUP.BIN
EXTRACT.EXE   FORMAT.COM      LAYOUT.INF      MINI.CAB
MSINFO.IMG     OEMSETUP.BIN    OEMSETUP.EXE  PRECOPY1.CAB
README.TXT    SAVE32.COM      SCANDISK.EXE  PRECOPY2.CAB
SETUP.EXE      SETUP.TXT      SETUP25I.EXE  SCANPROG.EXE
SUHELPER.BIN   SWINST4.EXE    WB16OFF.EXE  SMARTDRV.EXE
WIN95_04.CAB   WIN95_05.CAB    WIN95_06.CAB  WIN95_07.CAB
WIN95_09.CAB   WIN95_10.CAB    WIN95_11.CAB  WIN95_12.CAB
WIN95_14.CAB   WIN95_15.CAB    WIN95_16.CAB  WIN95_17.CAB
WIN95_19.CAB   WIN95_20.CAB    WIN95_21.CAB  WIN95_18.CAB
WIN95_24.CAB   WIN95_25.CAB    WIN95_26.CAB  WIN95_23.CAB
WINSETUP.BIN   WOWKIT.EXE    XMSMMGR.EXE  WIN95_28.CAB

56 file(s)      82,847,308 bytes
2 dir(s)           0 bytes free

E:\WIN95>_
```

The screenshot shows a DOS terminal window titled "Win\_95\_DEV [Running] - Oracle VM VirtualBox". The command "dir /a /w" is run, listing files in the current directory. The file "FORMAT.COM" is highlighted with a green border. The output shows various system files like CS3KIT.EXE, OEMSETUP.EXE, and various WIN95\_XX.CAB files. The total disk space is 82,847,308 bytes.

The format command help.

Win\_95\_DEV [Running] - Oracle VM VirtualBox

```
File Machine View Input Devices Help
56 file(s)      82,847,308 bytes
2 dir(s)           0 bytes free

E:\WIN95>format /?
Formats a disk for use with MS-DOS.

FORMAT drive: [/V[:label]] [/Q] [/F:size] [/B : /S] [/C]
FORMAT drive: [/V[:label]] [/Q] [/T:tracks /N:sectors] [/B : /S] [/C]
FORMAT drive: [/V[:label]] [/Q] [/1] [/4] [/B : /S] [/C]
FORMAT drive: [/Q] [/1] [/4] [/8] [/B : /S] [/C]

/V[:label] Specifies the volume label.
/Q      Performs a quick format.
/F:size Specifies the size of the floppy disk to format (such
        as 160, 180, 320, 360, 720, 1.2, 1.44, 2.88).
/B      Allocates space on the formatted disk for system files.
/S      Copies system files to the formatted disk.
/T:tracks Specifies the number of tracks per disk side.
/N:sectors Specifies the number of sectors per track.
/1      Formats a single side of a floppy disk.
/4      Formats a 5.25-inch 360K floppy disk in a high-density drive.
/8      Formats eight sectors per track.
/C      Tests clusters that are currently marked "bad."

E:\WIN95>_
```

The screenshot shows a DOS terminal window titled "Win\_95\_DEV [Running] - Oracle VM VirtualBox". The command "format /?" is run, displaying the help documentation for the FORMAT command. It details the syntax for formatting drives, including options for volume label, quick format, disk size, tracks, sectors, and cluster testing.

Type in the following to format the C: OS drive (The /s switch is optional).

```
FORMAT C: /S [Enter]
```

/s will copy the system boot files to the hard drive making it bootable to DOS.

**NOTE:**

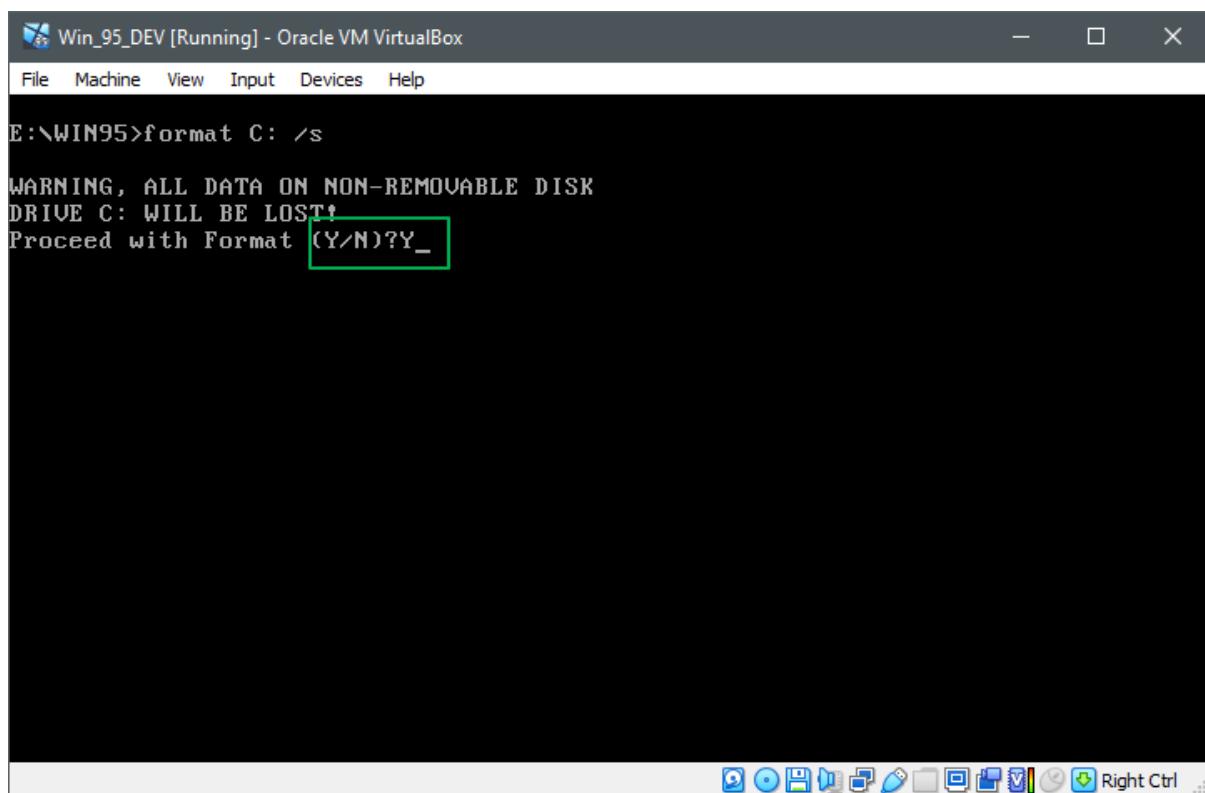
Only IO.SYS is not required and an alternative Windows specific file will be copied by the Windows install anyway.

MSDOS.SYS Does not contain any entry and will be overwritten by the Windows installer using Windows specific boot options.

Command.com is not required as it will exist in the C:\Window\command.com directory and can be deleted later.

I typically only place them on the C: drive temporarily so that I can boot from the C: drive and launch the installer form C:\WIN95\setup.exe without needing the floppy disk.

Select Y to proceed.



You can enter a short Name for this partition if you want, or you can change it after installing windows. I am going to leave it blank for now. Just press [Enter] to continue.

```
E:\>WIN95>format C: /s
WARNING, ALL DATA ON NON-REMOVABLE DISK
DRIVE C: WILL BE LOST!
Proceed with Format (Y/N)?Y

Checking existing disk format.
Formatting 4,094.66M
Format complete.
Writing out file allocation table
Complete.
Calculating free space (this may take several minutes)...
Complete.
System transferred

Volume label (11 characters, ENTER for none)? _
```

Next format your second hard drive if you partitioned it. It should be drive D:

FORMAT D: [Enter]

Both drives are now formatted and ready copy the windows install files. It is possible at this stage to run the installer from the windows 95 install CD, but the first boot can have issues finding some of the driver files from the CD. Most usually copy at minimum the E:\WIN95 directory to the C: drive and run the install from that location. The later Windows setup will find any required driver and setup files without needing the CD-ROM mounted.

Navigate to your C: drive.

C: [Enter]

Create a new directory named WIN95

MD WIN95

## A Beginners Guide To DOS Programming

The screenshot shows a DOS terminal window with a blue header bar containing the title "Win\_95\_DEV [Running] - Oracle VM VirtualBox". The menu bar includes "File", "Machine", "View", "Input", "Devices", and "Help". The main window displays a command-line interface. The user has entered "MD WIN95" followed by "dir /a /w". The output shows the volume information (no label, serial number 2A5E-130C), the directory of C:, and a file listing. The file "COMMAND.COM" is listed with size 10.SYS, and "MSDOS.SYS" is listed with size 316,289 bytes. A box highlights the text "[WIN95]" next to "MSDOS.SYS". The bottom of the window shows a toolbar with various icons and the text "Right Ctrl".

Then enter the following to copy the install files from the CD-ROM

```
COPY E:\WIN95\*.* C:\WIN95
```

The screenshot shows a DOS terminal window with a blue header bar containing the title "Win\_95\_DEV [Running] - Oracle VM VirtualBox". The menu bar includes "File", "Machine", "View", "Input", "Devices", and "Help". The main window displays a command-line interface. The user has entered "COPY E:\WIN95\\*.\* C:\WIN95". The output shows the names of 56 files being copied from the E:\WIN95 directory to the C:\WIN95 directory, including files like WIN95\_10.CAB through WIN95\_28.CAB, WINSETUP.BIN, and XMSMMGR.EXE. The bottom of the window shows a toolbar with various icons and the text "Right Ctrl".

All of the Windows 95 install files and driver files now exist in the C :\WIN95 directory. This is also convenient if you wish to install additional Windows components or drivers at a later time without the need for the Install CD-ROM (ISO).

The screenshot shows a DOS window titled "Win\_95\_DEV [Running] - Oracle VM VirtualBox". The window contains the following text:

```
File Machine View Input Devices Help
E:\WIN95\WIN95_20.CAB
E:\WIN95\WIN95_21.CAB
E:\WIN95\WIN95_22.CAB
E:\WIN95\WIN95_23.CAB
E:\WIN95\WIN95_24.CAB
E:\WIN95\WIN95_25.CAB
E:\WIN95\WIN95_26.CAB
E:\WIN95\WIN95_27.CAB
E:\WIN95\WIN95_28.CAB
E:\WIN95\WINSETUP.BIN
E:\WIN95\WOWKIT.EXE
E:\WIN95\XMSMMGR.EXE
      56 file(s) copied

C:\>dir /a /w

Volume in drive C has no label
Volume Serial Number is 2A5E-130C
Directory of C:\

COMMAND.COM      IO.SYS          MSDOS.SYS
                3 file(s)    316,289 bytes
                1 dir(s)   4,201,910,272 bytes free

C:\>
```

A green rectangular box highlights the text "[WIN95]" located to the right of the file statistics.

Unmount the “windows95b.iso” from the VirtualBox “Devices -> Optical drives” menu with remove disk from virtual drive.

Navigate to the Windows install directory CD WIN95

The Windows 95 install directory has 2 setup files; SETUP.EXE and SETUP32.EXE. The SETUP32.EXE is for running the install from a windows environment. We are going to use the SETUP.EXE for DOS.

Run the “SETUP.EXE” to begin the install.

## A Beginners Guide To DOS Programming

Win\_95\_DEV [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Volume in drive C has no label  
Volume Serial Number is 2A5E-130C  
Directory of C:\WIN95

[.]	[..]	CS3KIT.EXE	DELTEMP.COM	DOSSETUP.BIN
EXTRACT.EXE	FORMAT.COM	LAYOUT.INF	LAYOUT1.INF	MINI.CAB
MSINFO.INF	OEMSETUP.BIN	OEMSETUP.EXE	PRECOPY1.CAB	PRECOPY2.CAB
README.TXT	SAVE32.COM	SCANDISK.EXE	SCANDISK.PIF	SCANPROG.EXE
SETUP.EXE	SETUP.TXT	SETUP251.EXE	SETUP32.EXE	SMARTDRV.EXE
SUHELPER.BIN	SWINST4.EXE	WB16OFF.EXE	WIN95_02.CAB	WIN95_03.CAB
WIN95_04.CAB	WIN95_05.CAB	WIN95_06.CAB	WIN95_07.CAB	WIN95_08.CAB
WIN95_09.CAB	WIN95_10.CAB	WIN95_11.CAB	WIN95_12.CAB	WIN95_13.CAB
WIN95_14.CAB	WIN95_15.CAB	WIN95_16.CAB	WIN95_17.CAB	WIN95_18.CAB
WIN95_19.CAB	WIN95_20.CAB	WIN95_21.CAB	WIN95_22.CAB	WIN95_23.CAB
WIN95_24.CAB	WIN95_25.CAB	WIN95_26.CAB	WIN95_27.CAB	WIN95_28.CAB
WINSETUP.BIN	WOWKIT.EXE	XMSMMGR.EXE		

56 file(s) 82,847,308 bytes  
2 dir(s) 4,201,910,272 bytes free

C:\WIN95>setup.exe

Please wait while Setup initializes.

Setup is now going to perform a routine check on your system.

To continue, press ENTER. To quit Setup, press ESC.

Right Ctrl

Setup will do a disk scan to check for damaged drive sectors.

Win\_95\_DEV [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

**Microsoft ScanDisk**

ScanDisk is now checking the following areas of drive C:

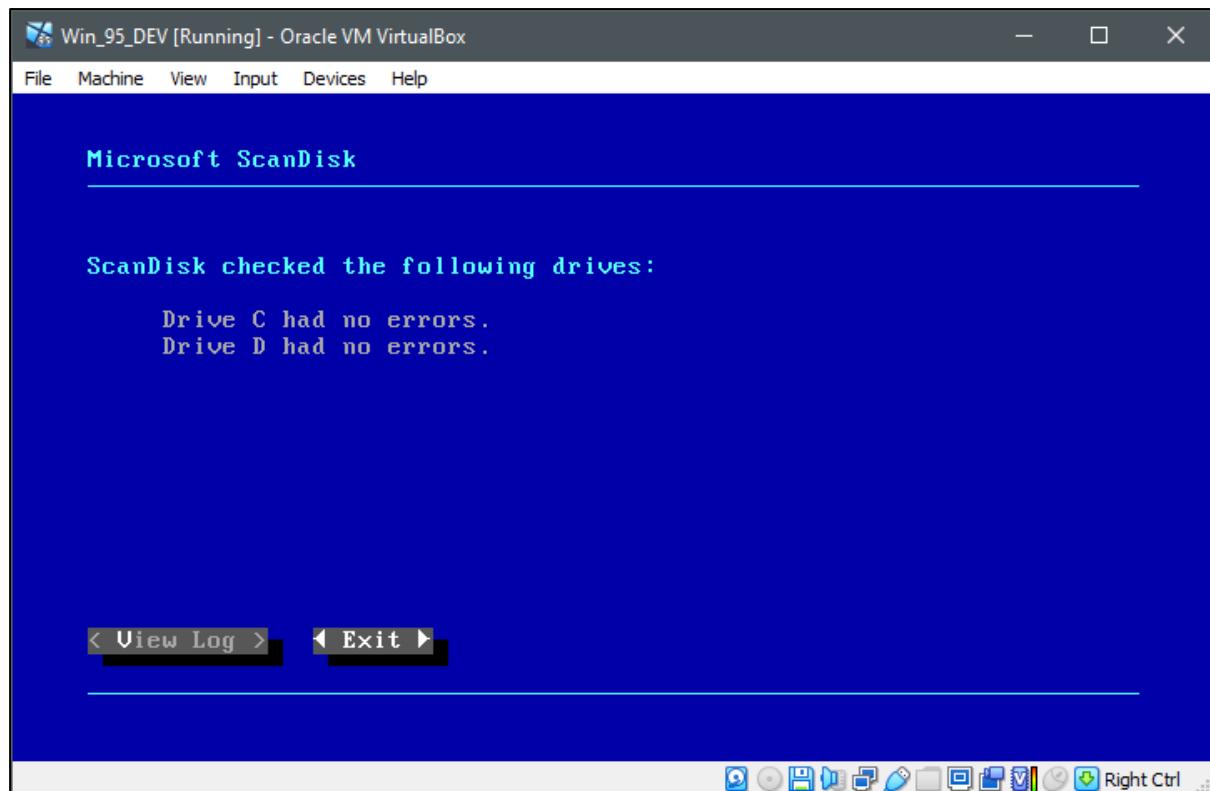
- ↳ Media descriptor
- ↳ File allocation tables
- ↳ Directory structure
- » File system
- Free space
- Surface scan

< Pause > < More Info > < Exit >

95% complete

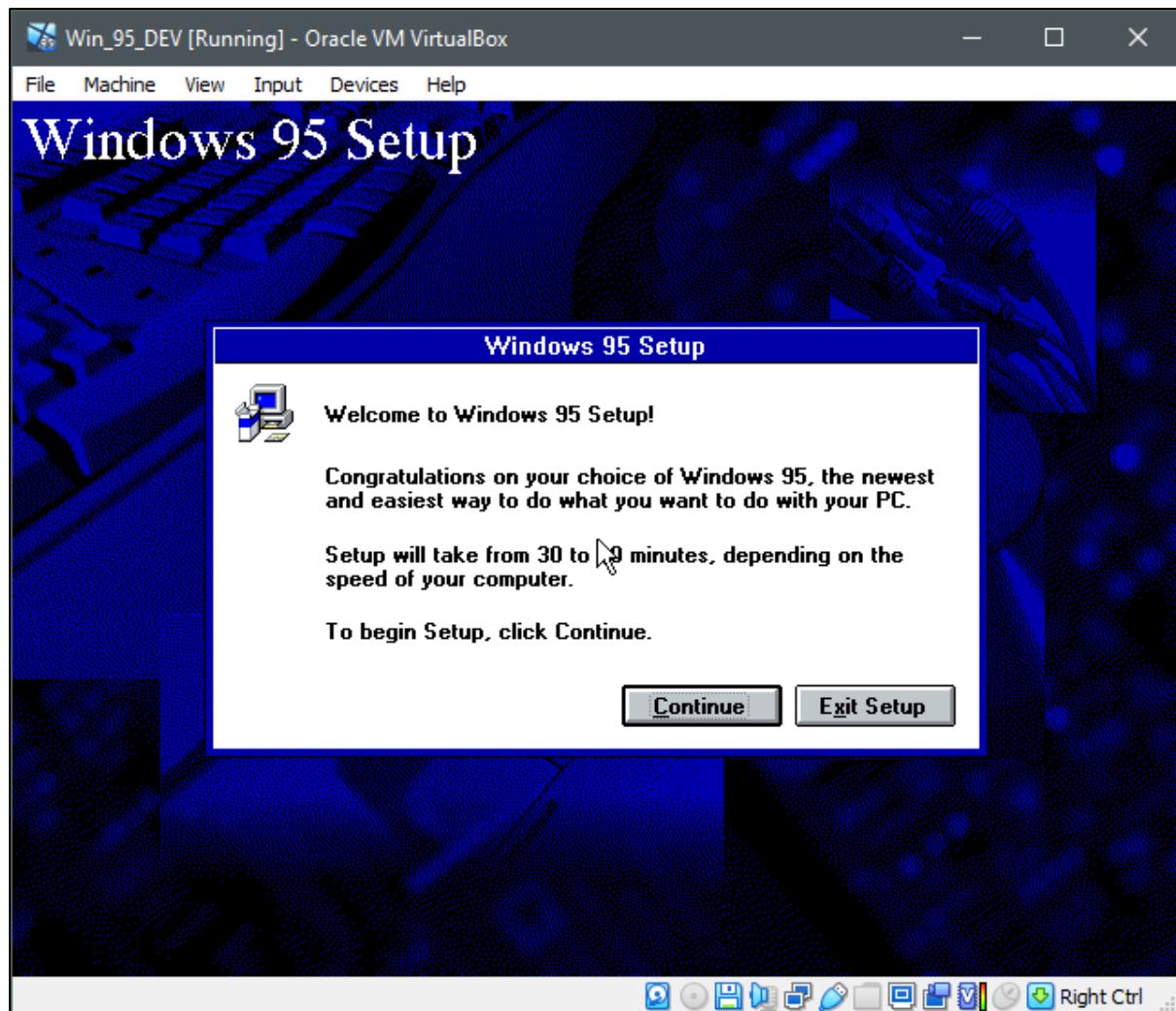
Right Ctrl

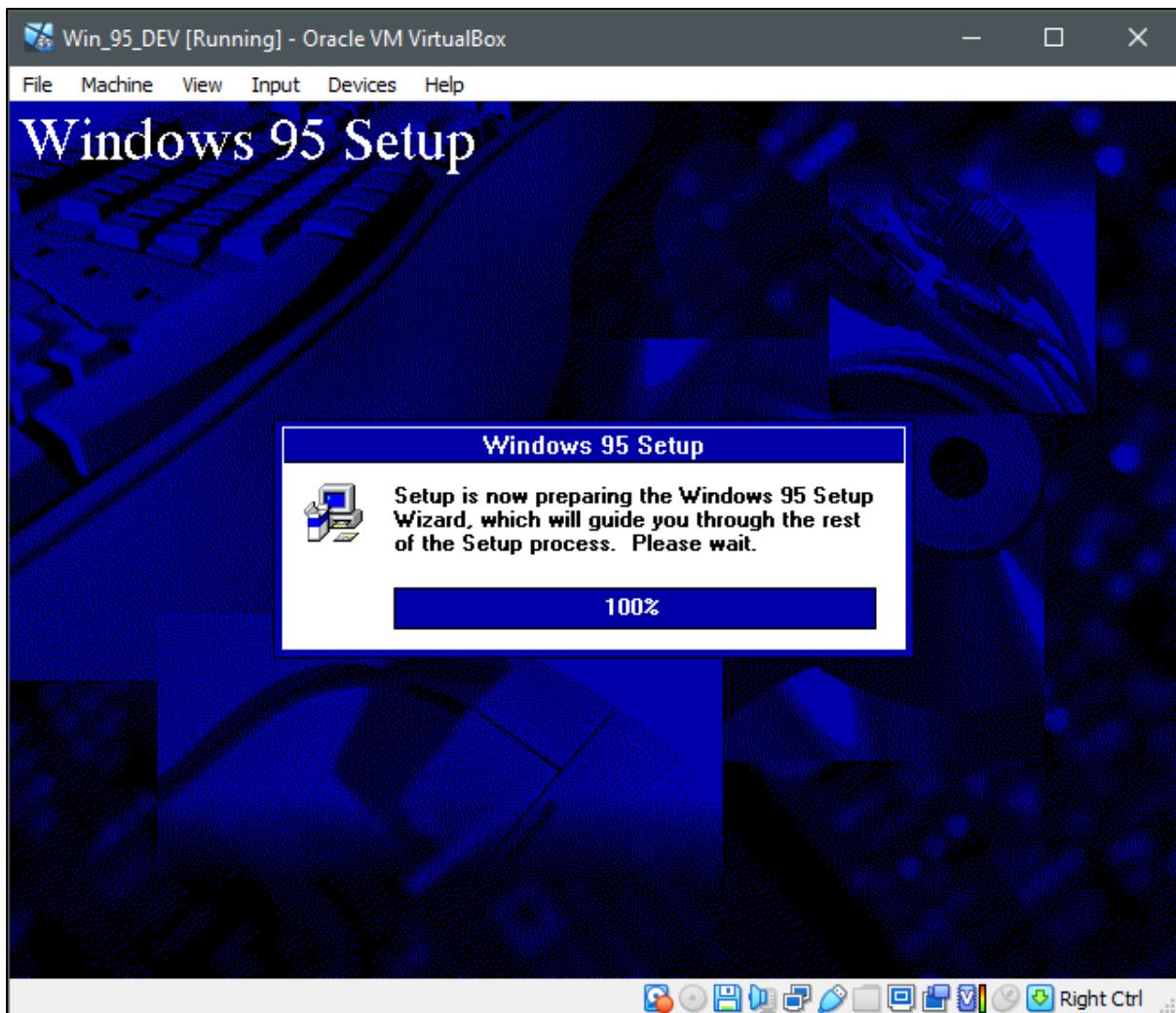
Select Exit to continue when the scan has finished..



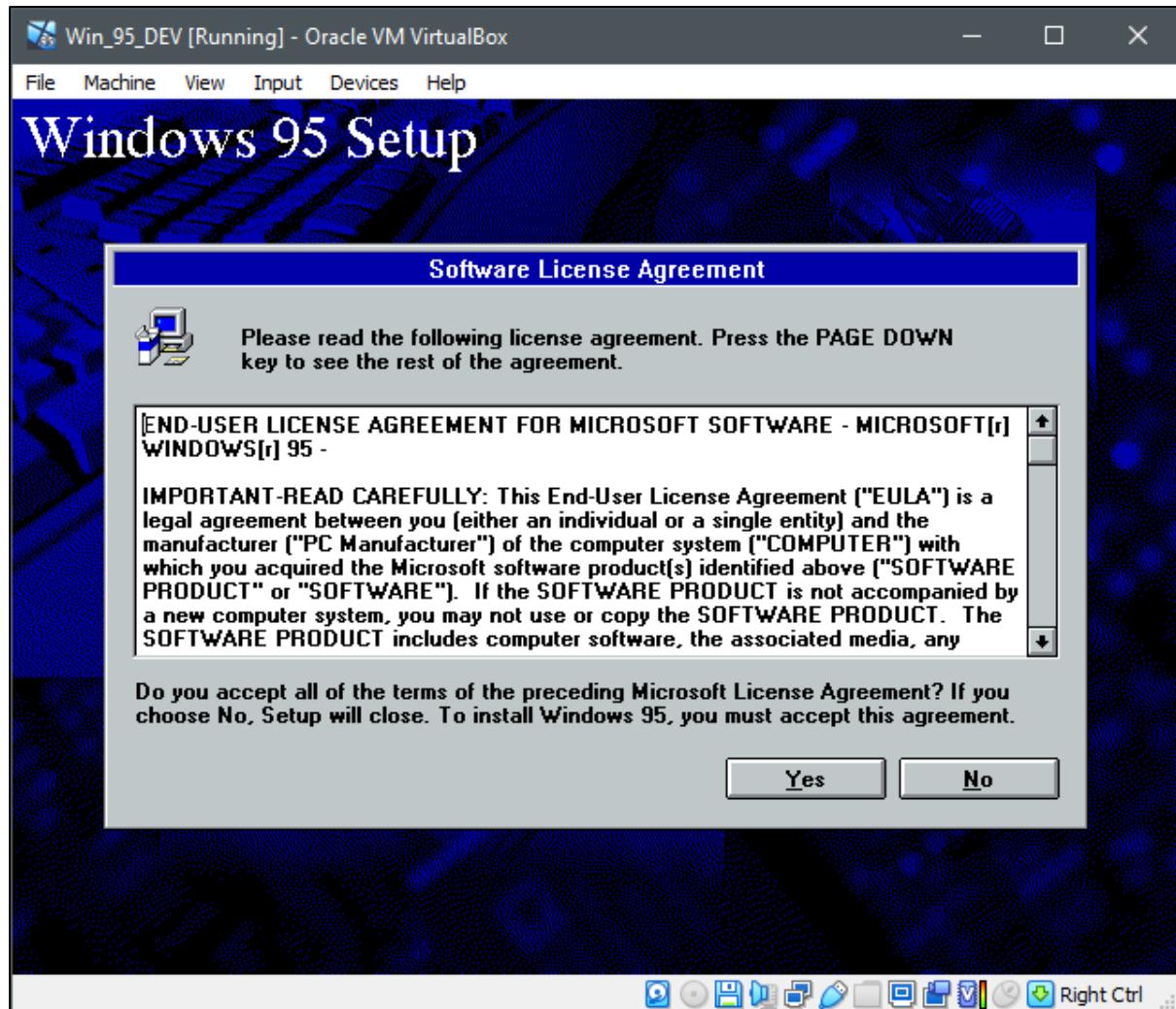
Next you will be met with the Windows 95 setup options. Select “Continue” to begin the setup.

Note that you should be able to use your mouse at this stage.

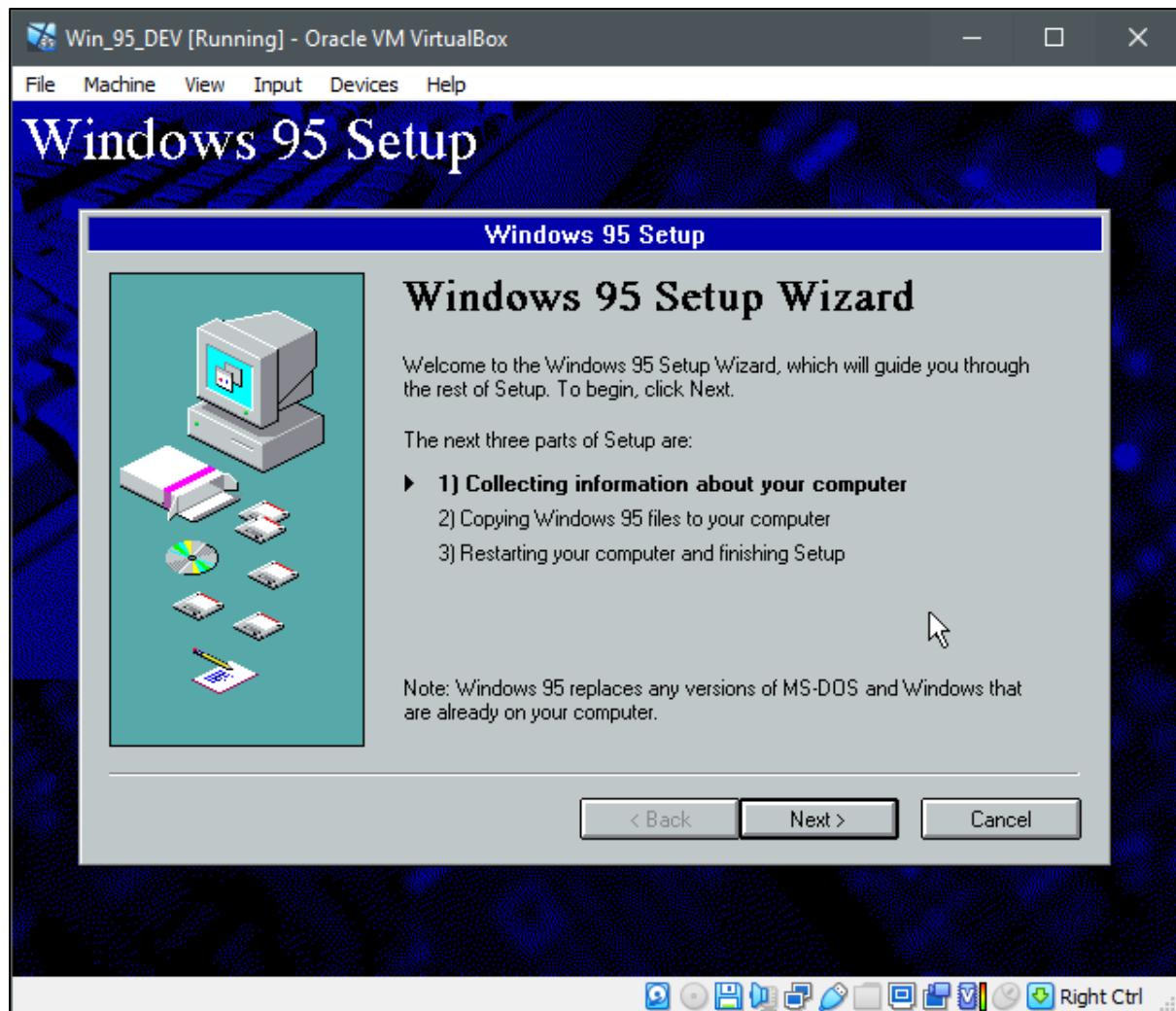




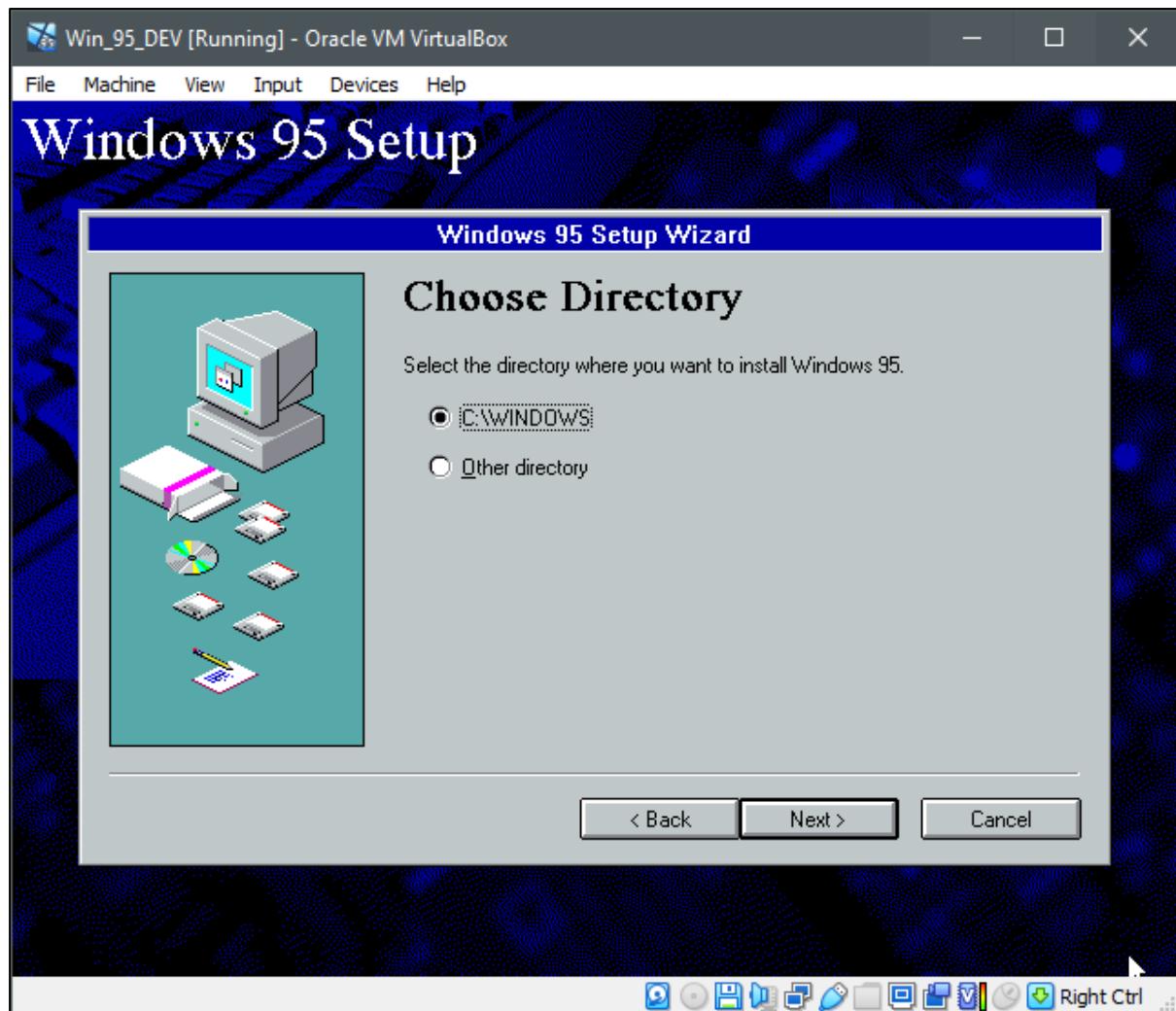
Accept the license agreement.

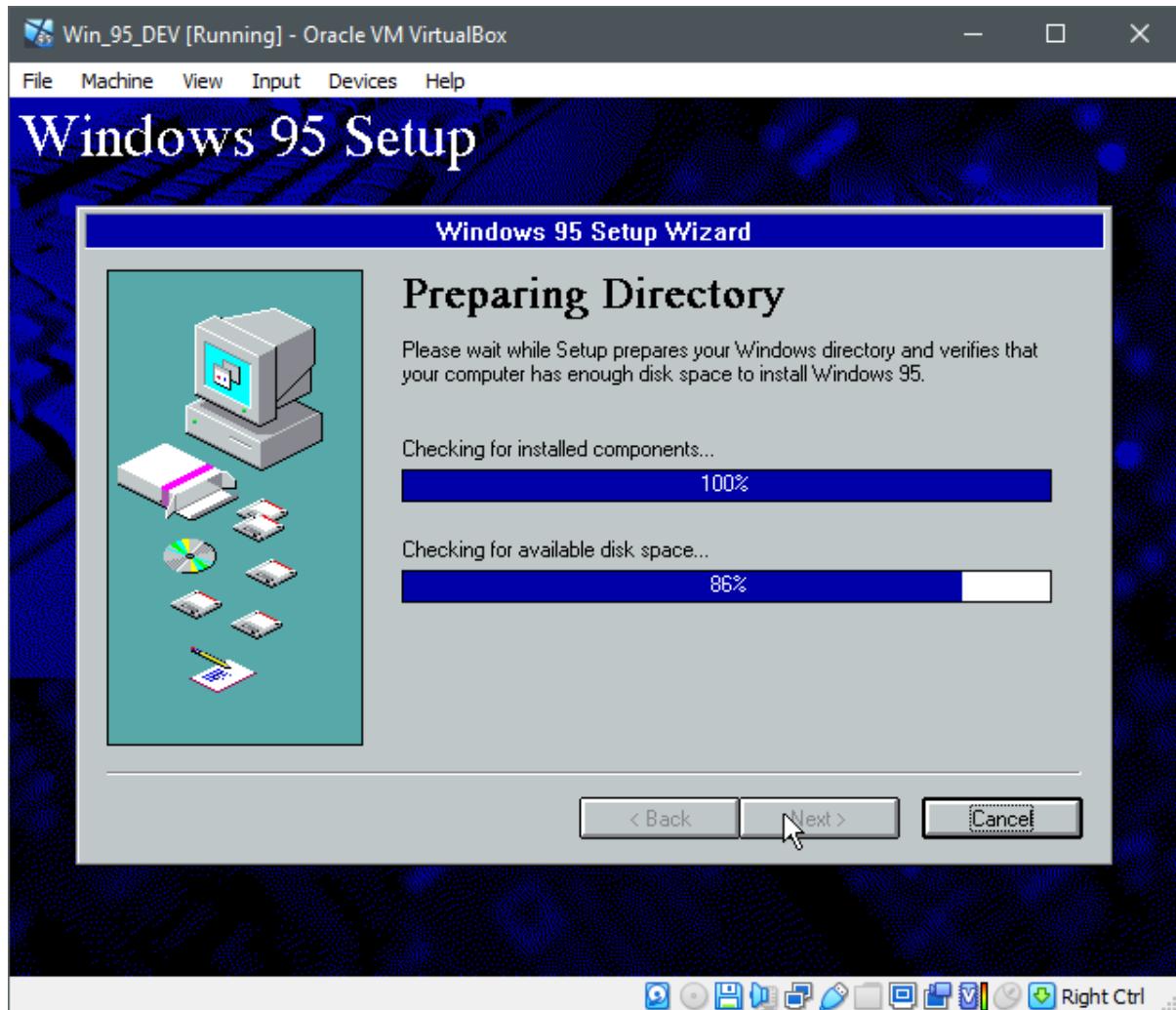


Next the setup wizard will check what hardware your computer has and begin copying files to the C: drive. Select next to continue.

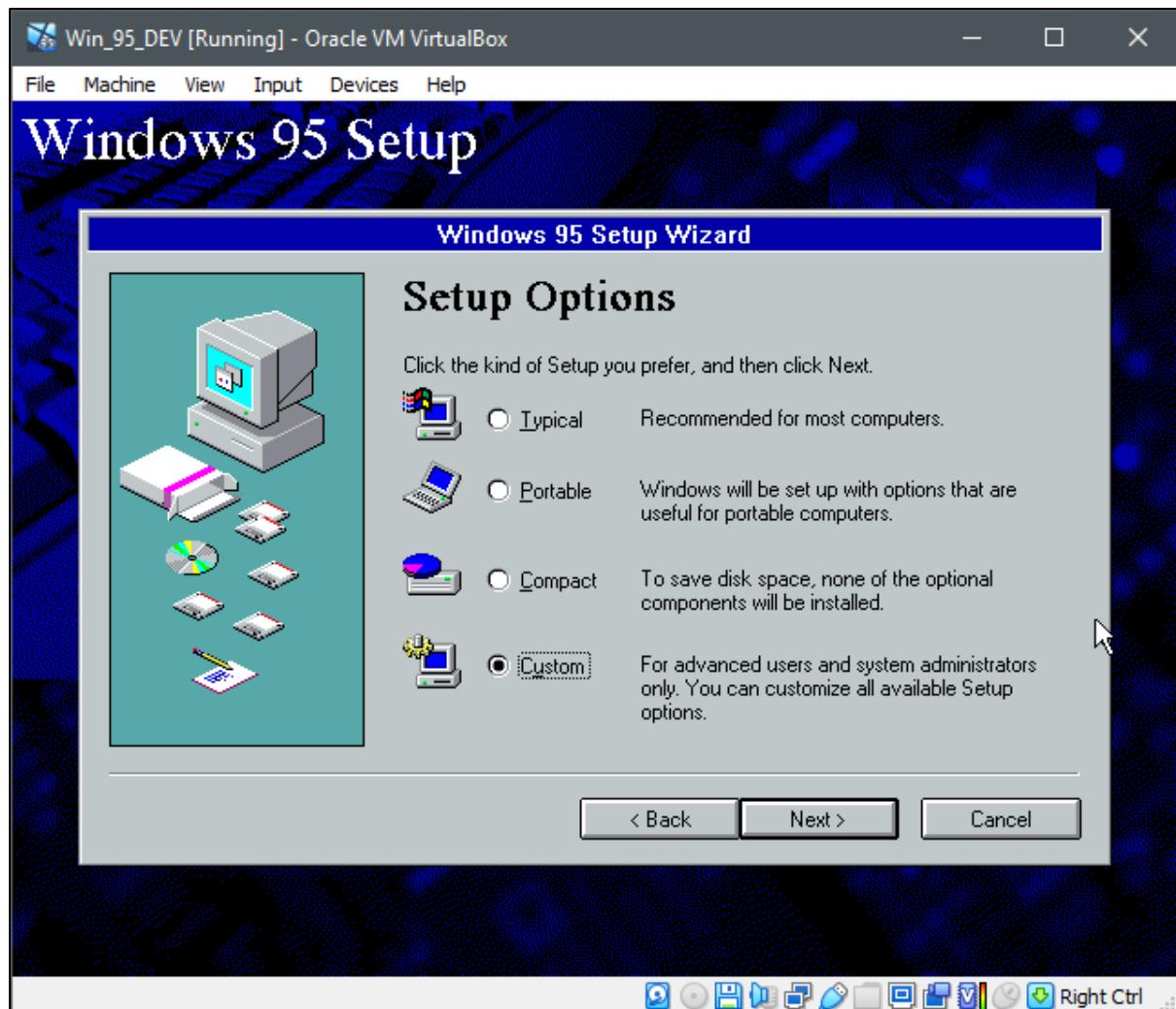


Use the default C:\WINDOWS directory and select Next.





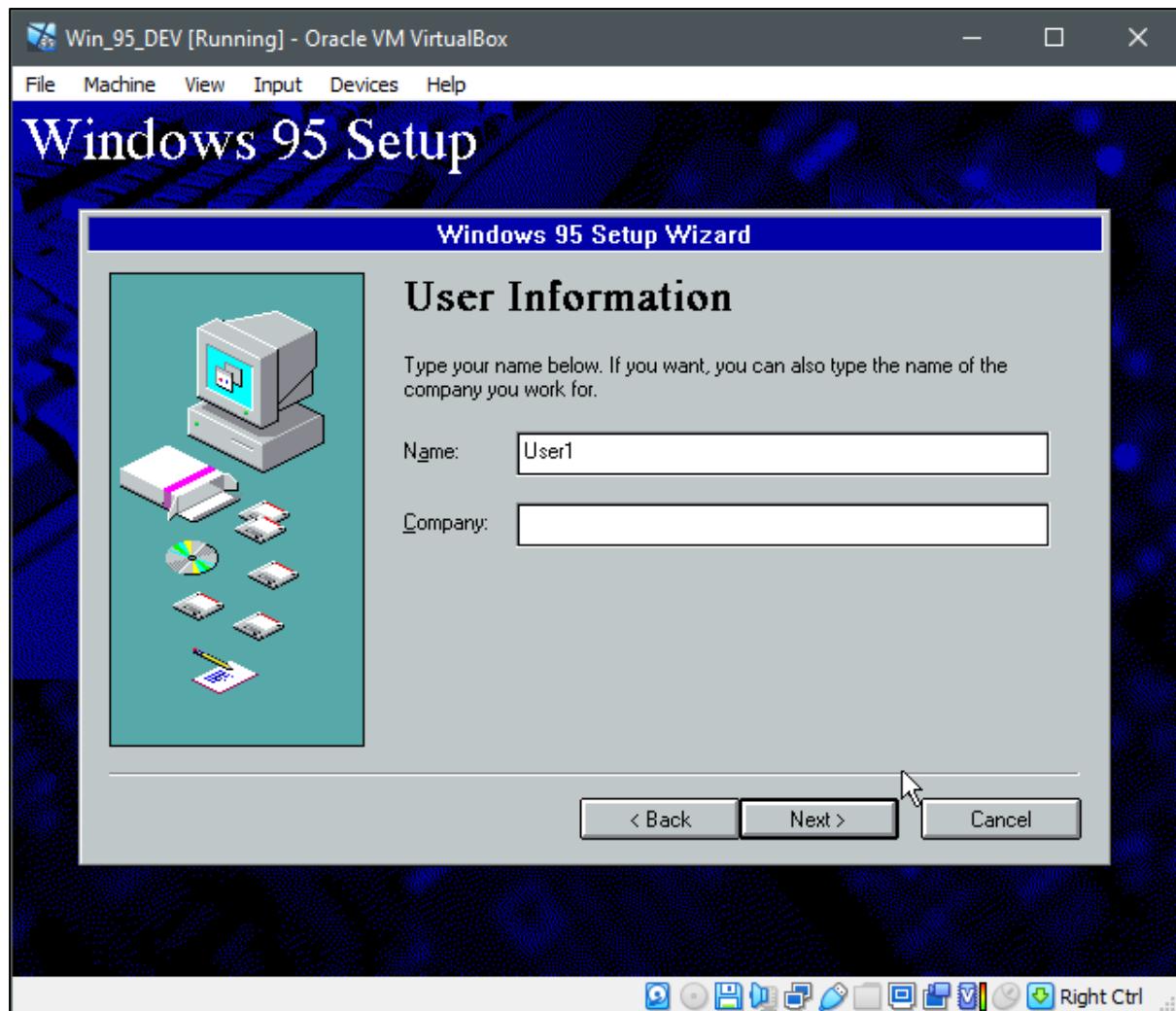
At the next screen I would recommend using the custom options. If you choose a “Typical” install you can always use the “Add windows features” option at a later time.



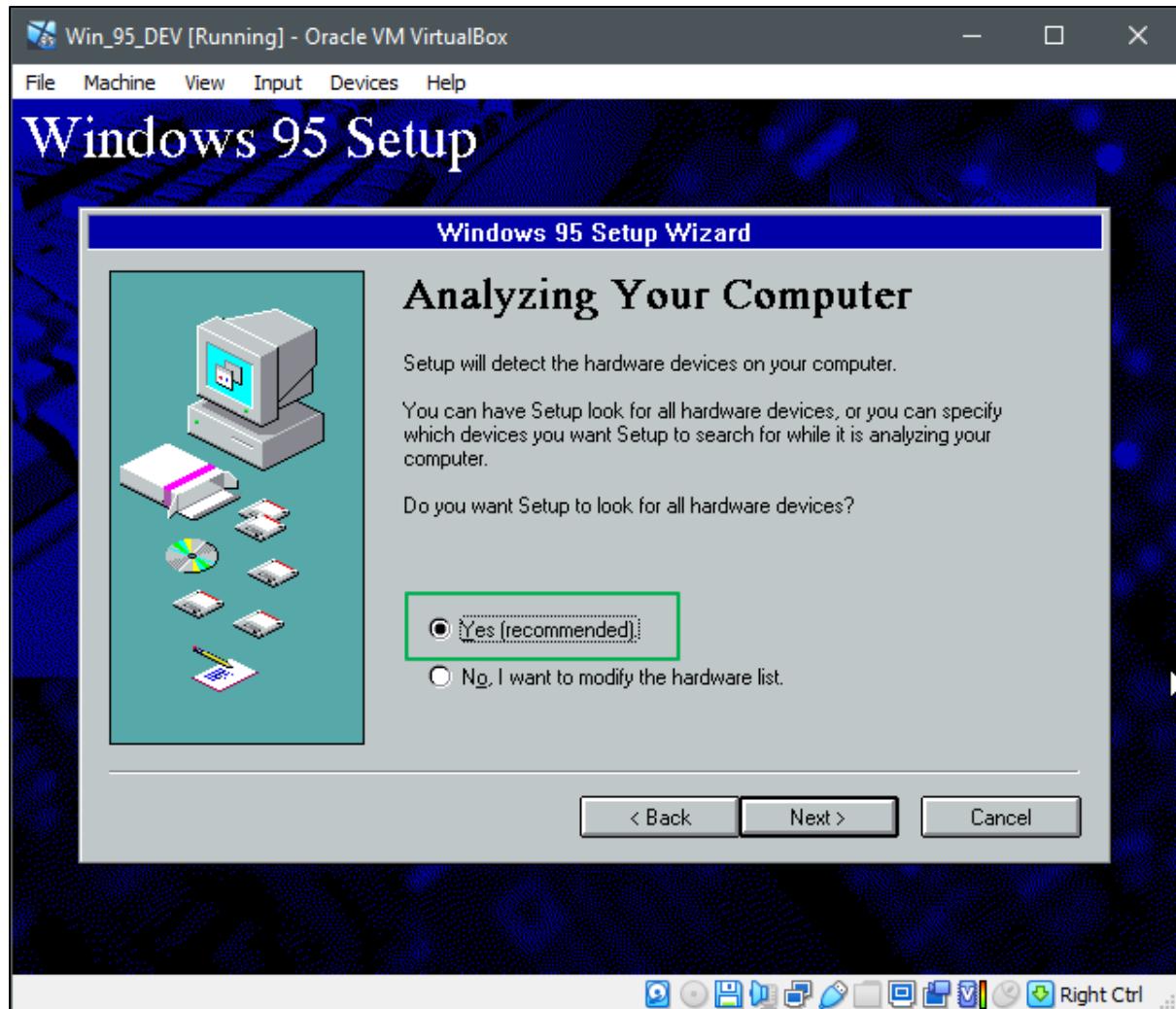
Enter the product Key then click next to continue.



Enter your primary user name. The company is optional.

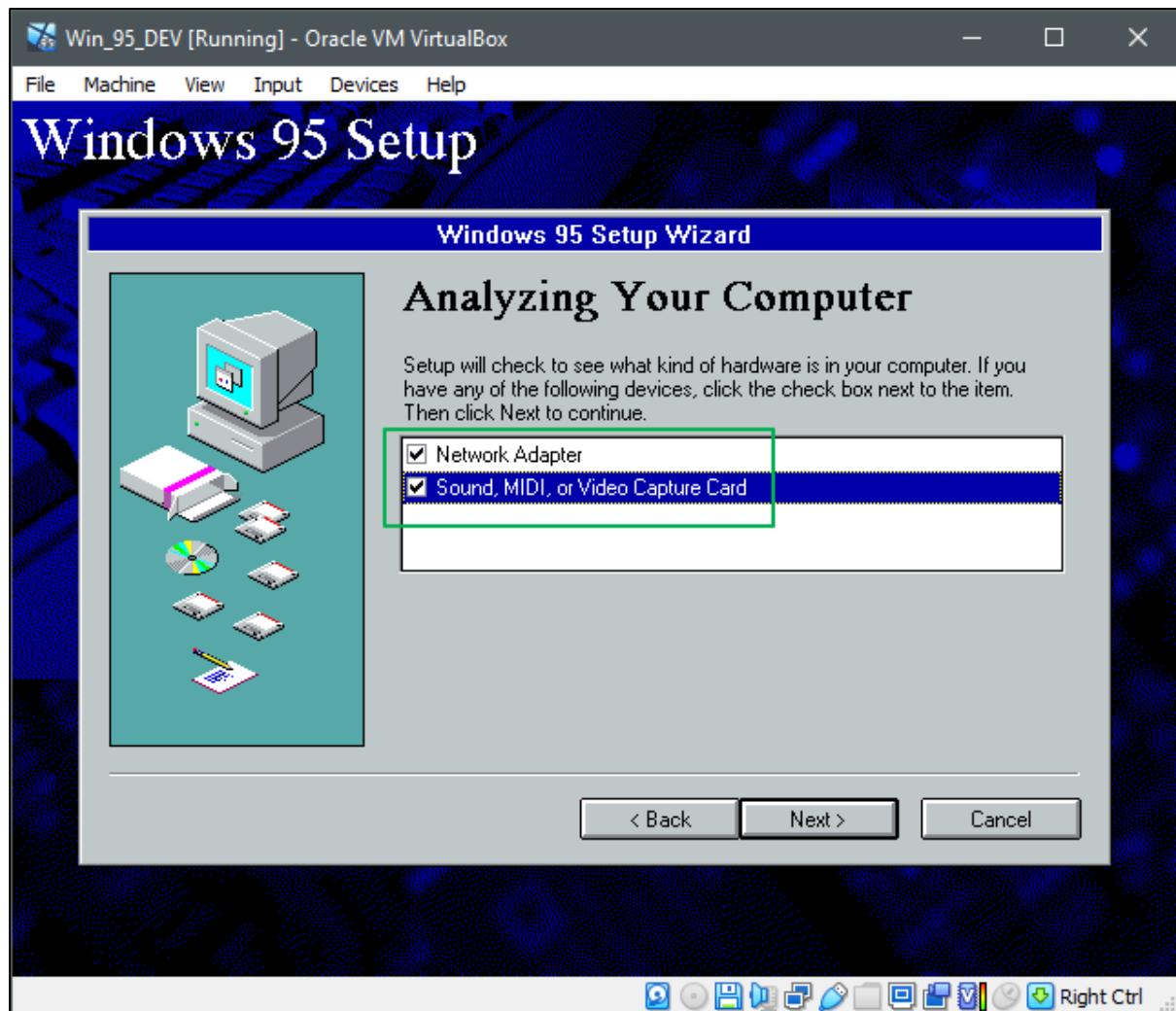


Next allow the setup program to look for hardware and matching drivers.



Select the Network and sound adaptors check boxes. This stage may take a little while to complete.

Note that the VirtualBox Sound Card has no MIDI controller capability. Also the network card will be incorrectly recognized which we will correct later.





In the next screen select any of the additional Windows 95 components that you may find useful.

Under accessories I would recommend adding to the defaults:

Character Map

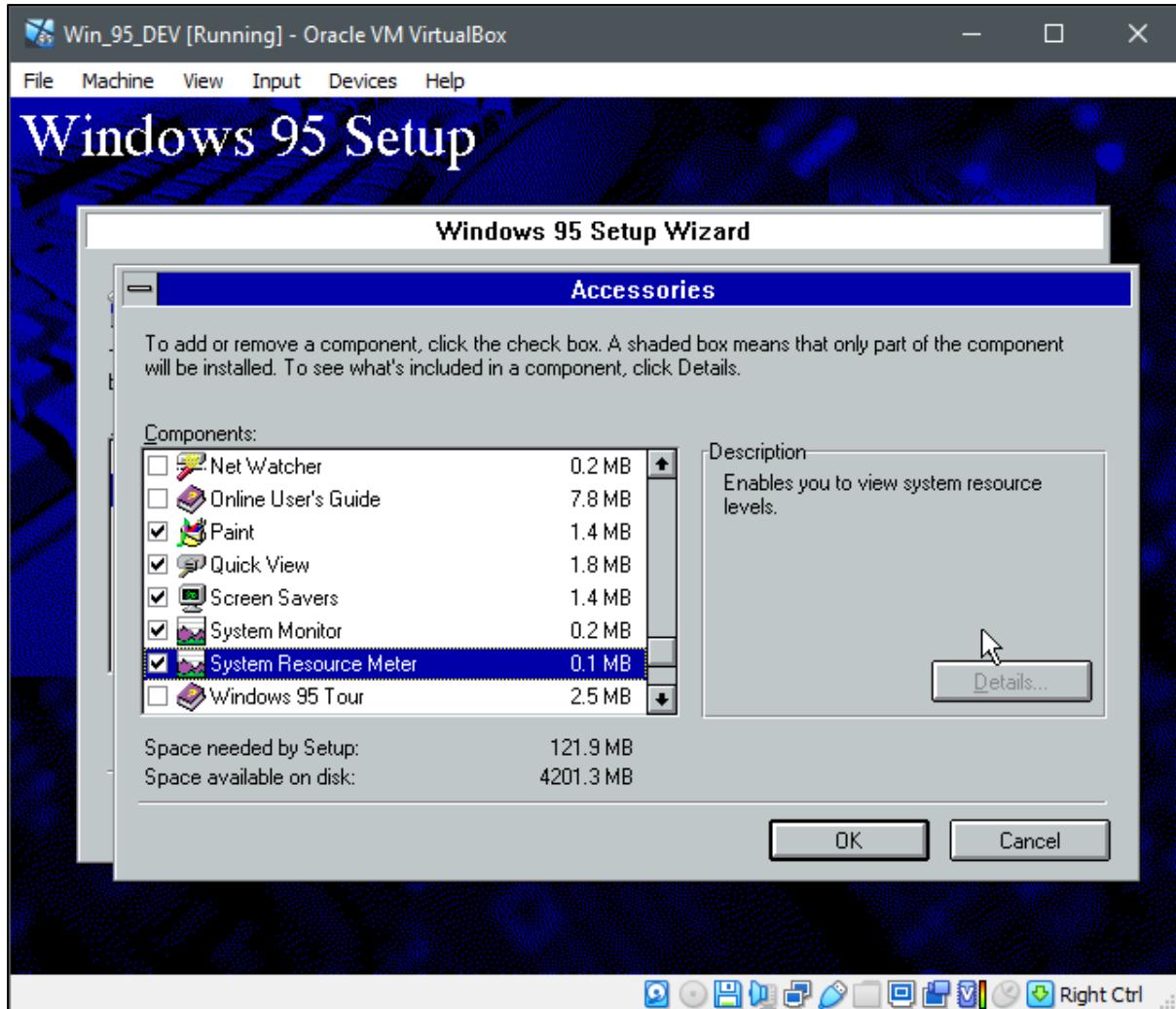
Desktop Management

Desktop Wallpaper

Quick View

System Monitor

System Resource Meter

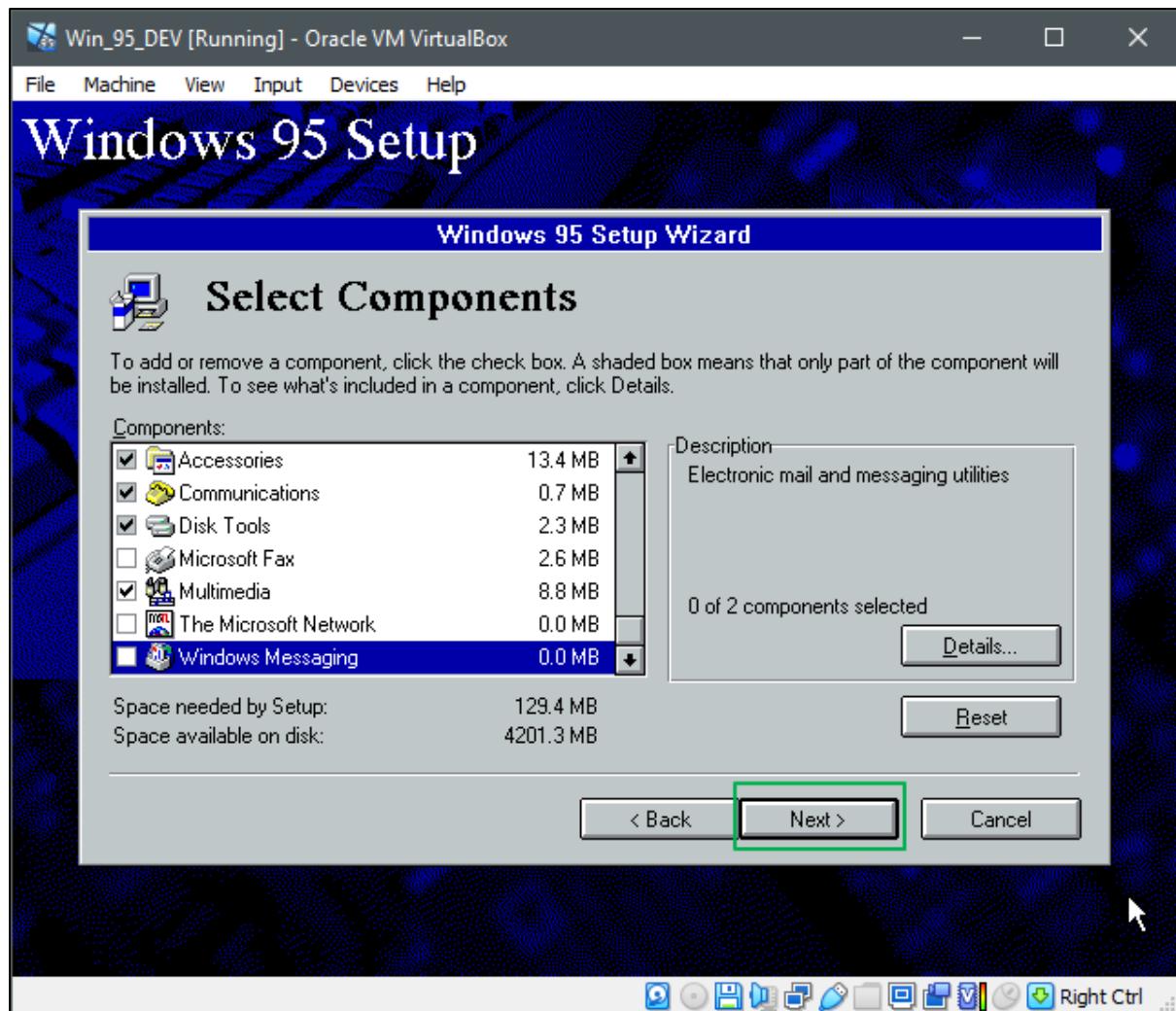


Most of the communications section are for dial up modems and are not really required, but you will need to leave them enabled as the Internet setup expects a modem to be available even though we won't use it.

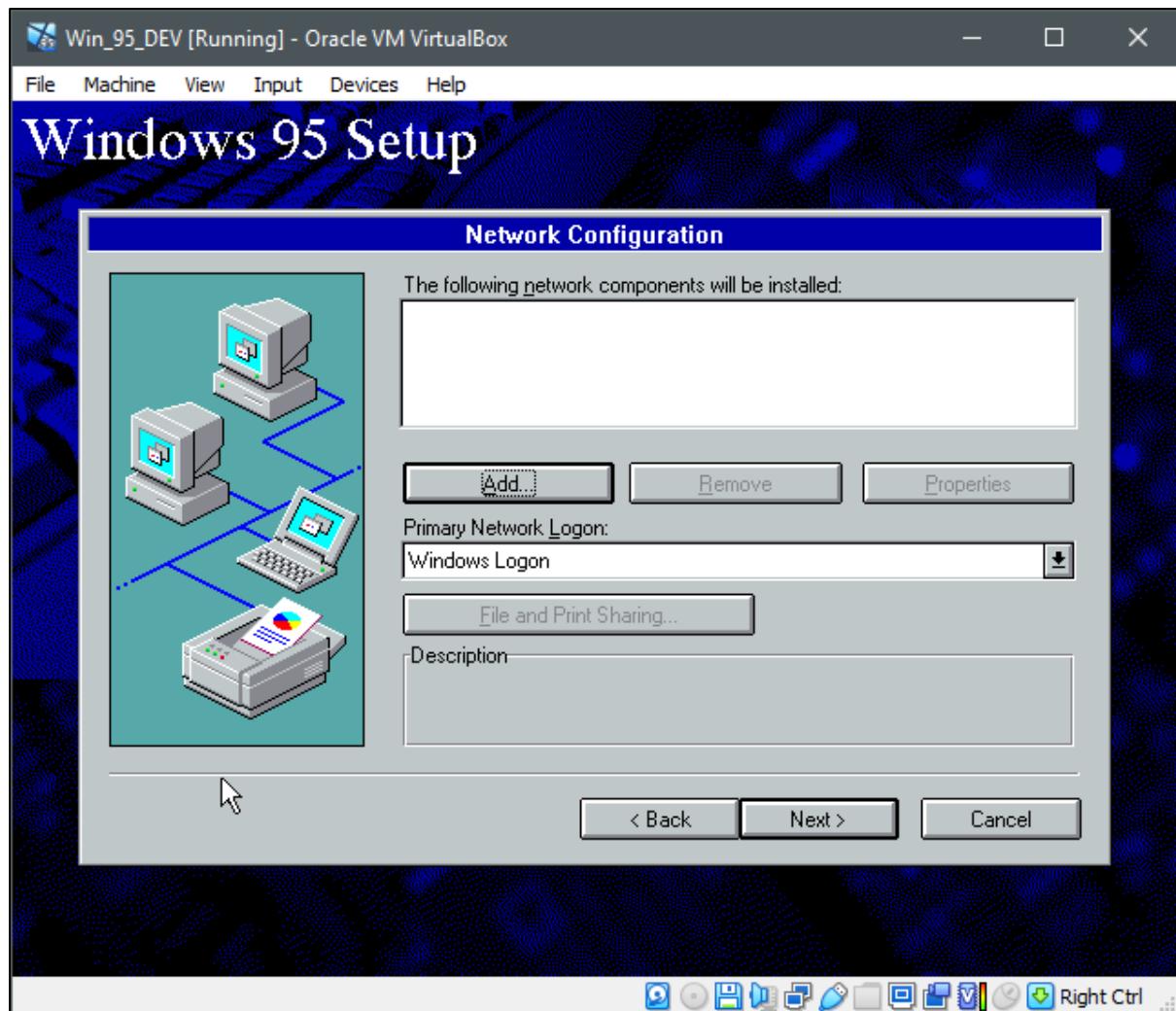
In the Multimedia section, select all.



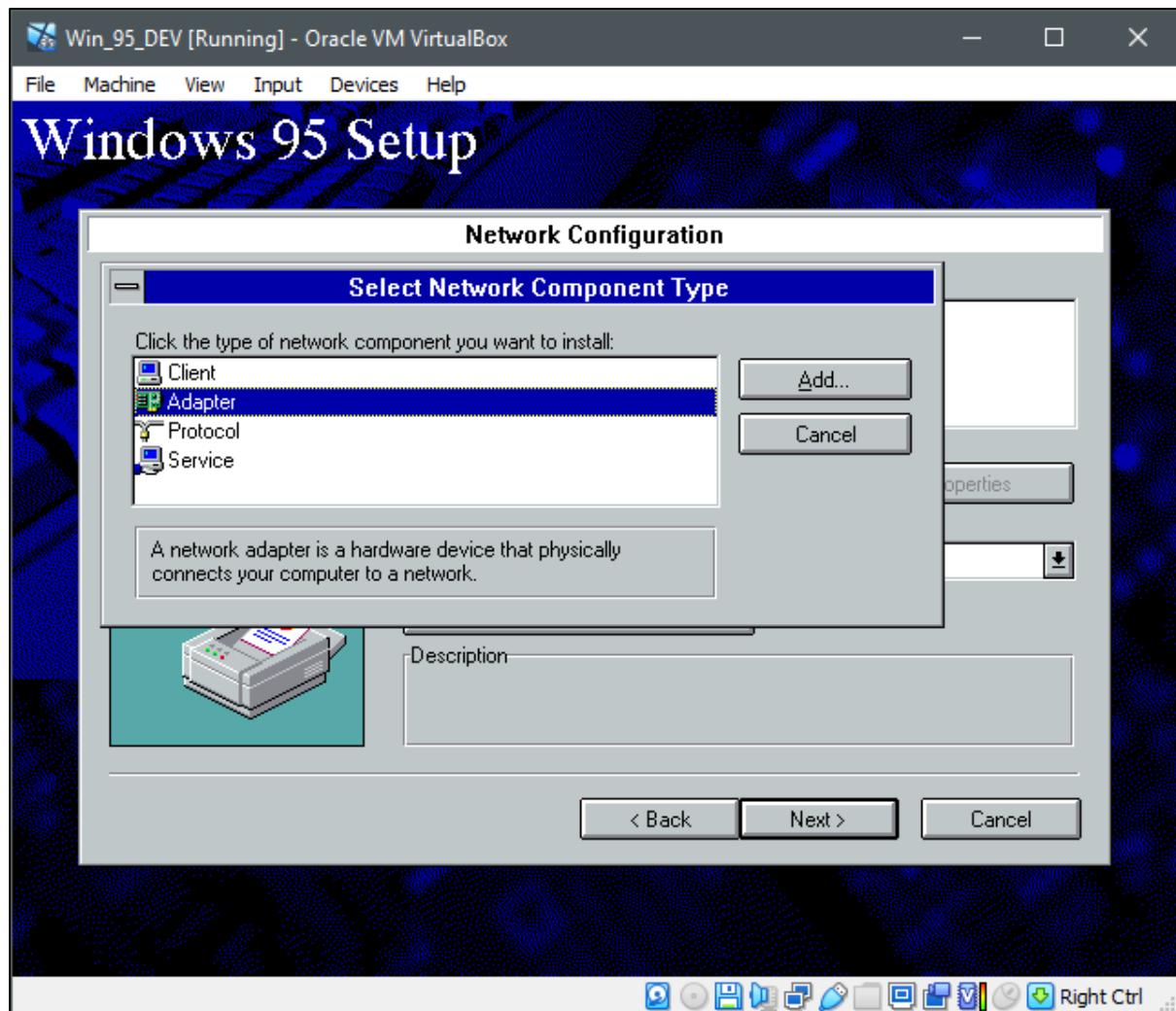
Select Next to continue the install.



In the next screen select Add to choose the network card.



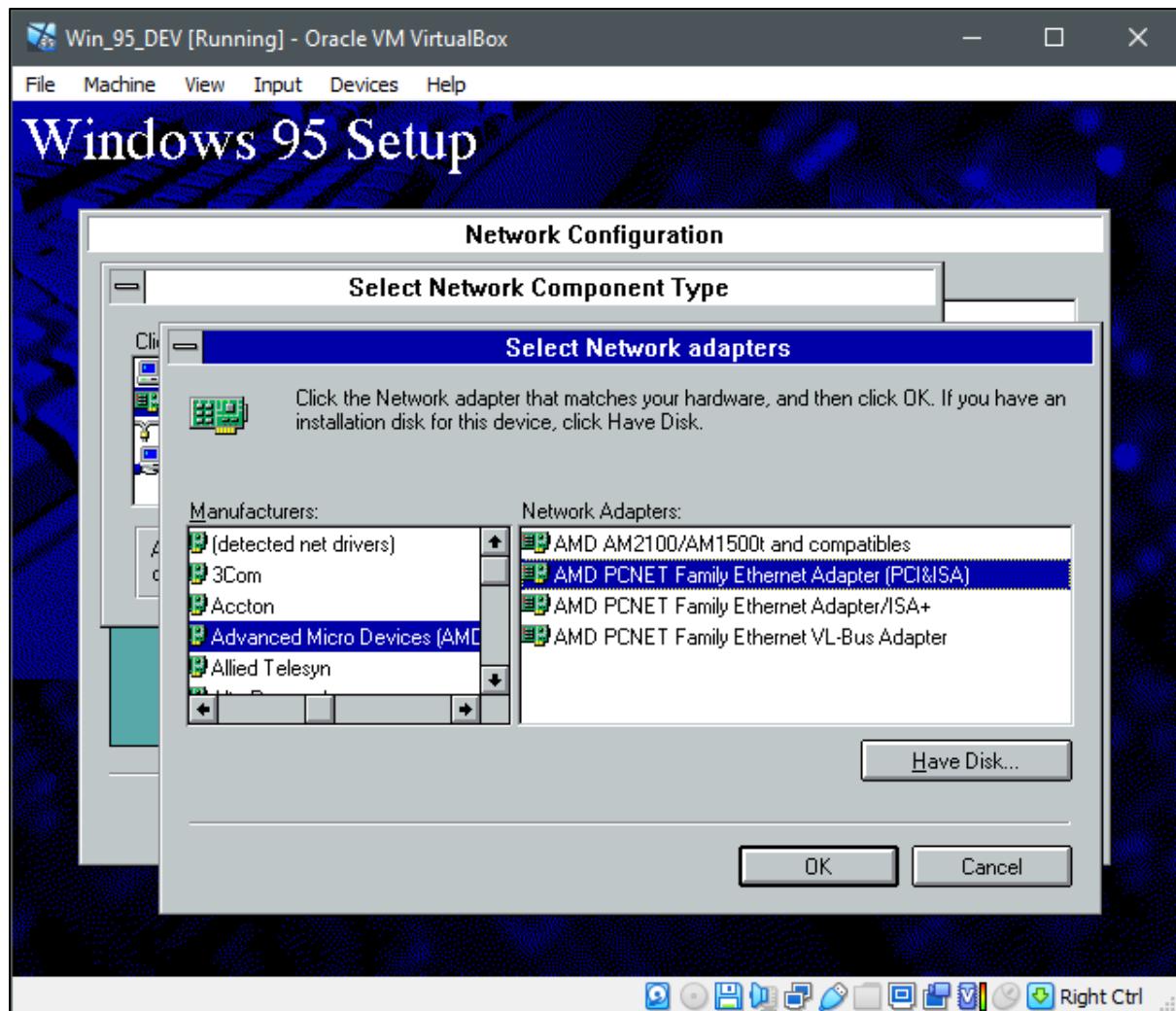
Next choose Adaptor and click Add.



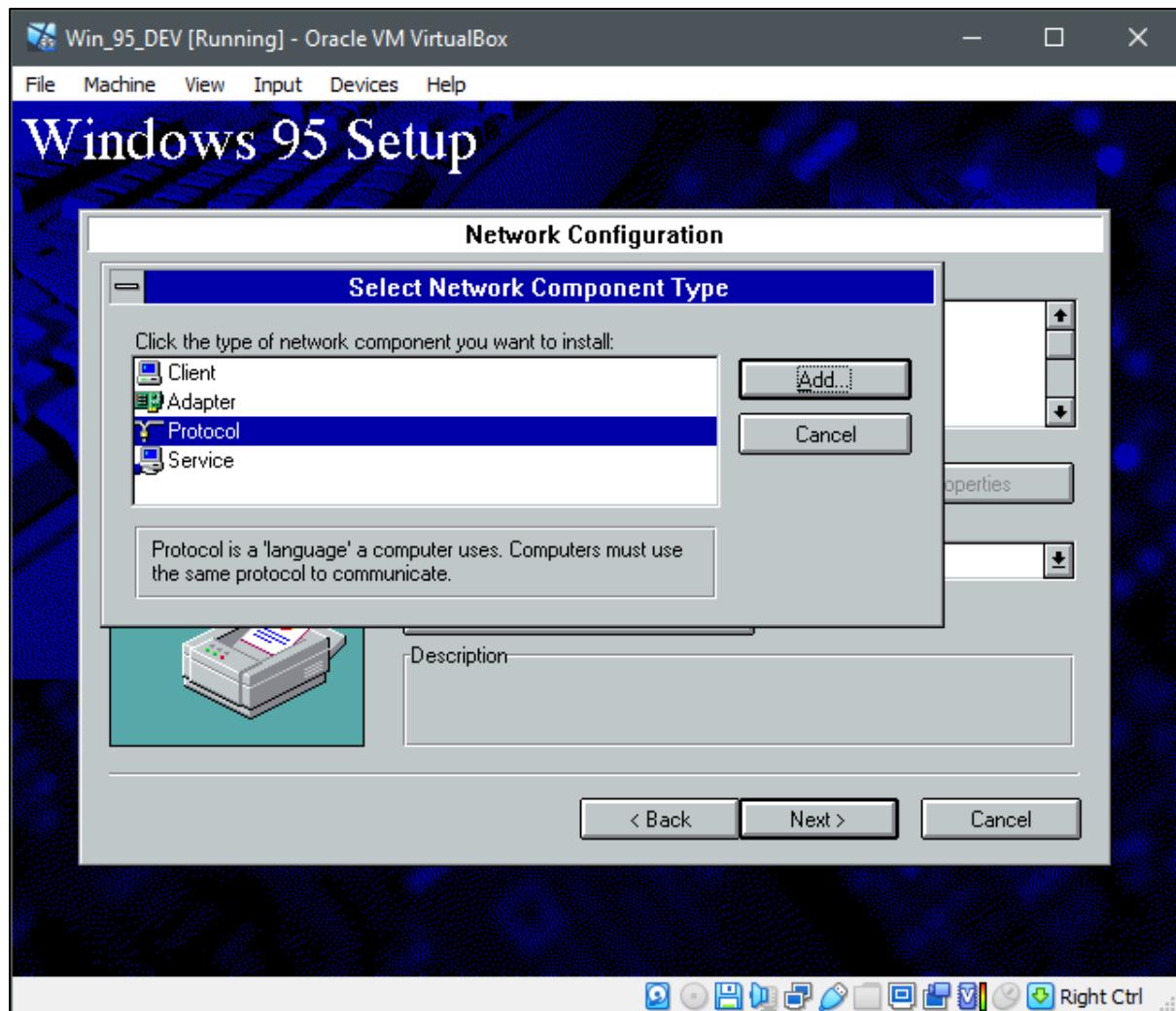
Scroll down to advanced Micro Devices (AMD) and then select the AMD PCNET Family Ethernet Adaptor (PCI&ISA), followed by OK.

This matches with the default VirtualBox card PCnet-FAST III (am79C973) card.

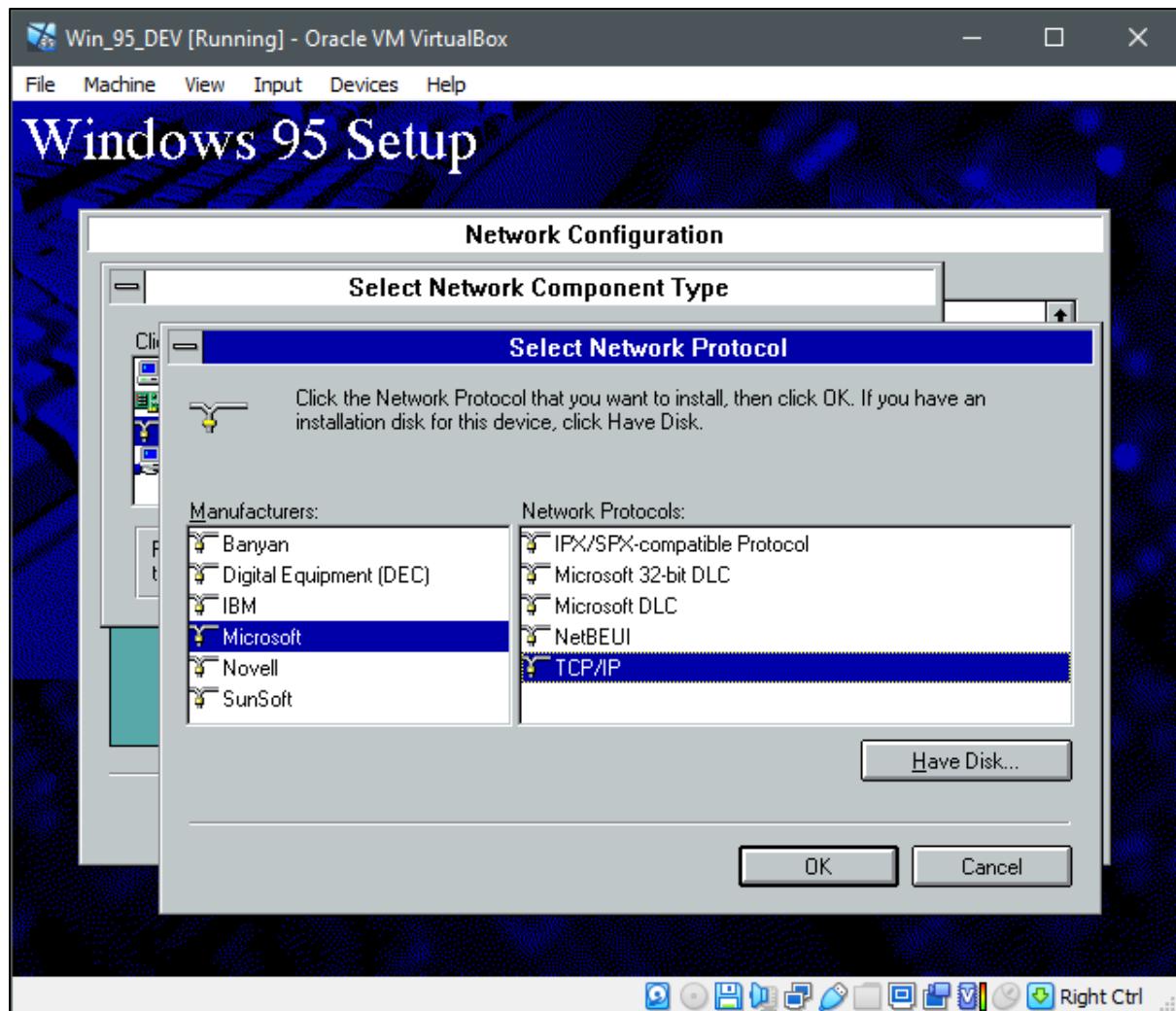
Note: Select the PCnet-FAST II card in VirtualBox if issues arise, but the version III card should be fine.



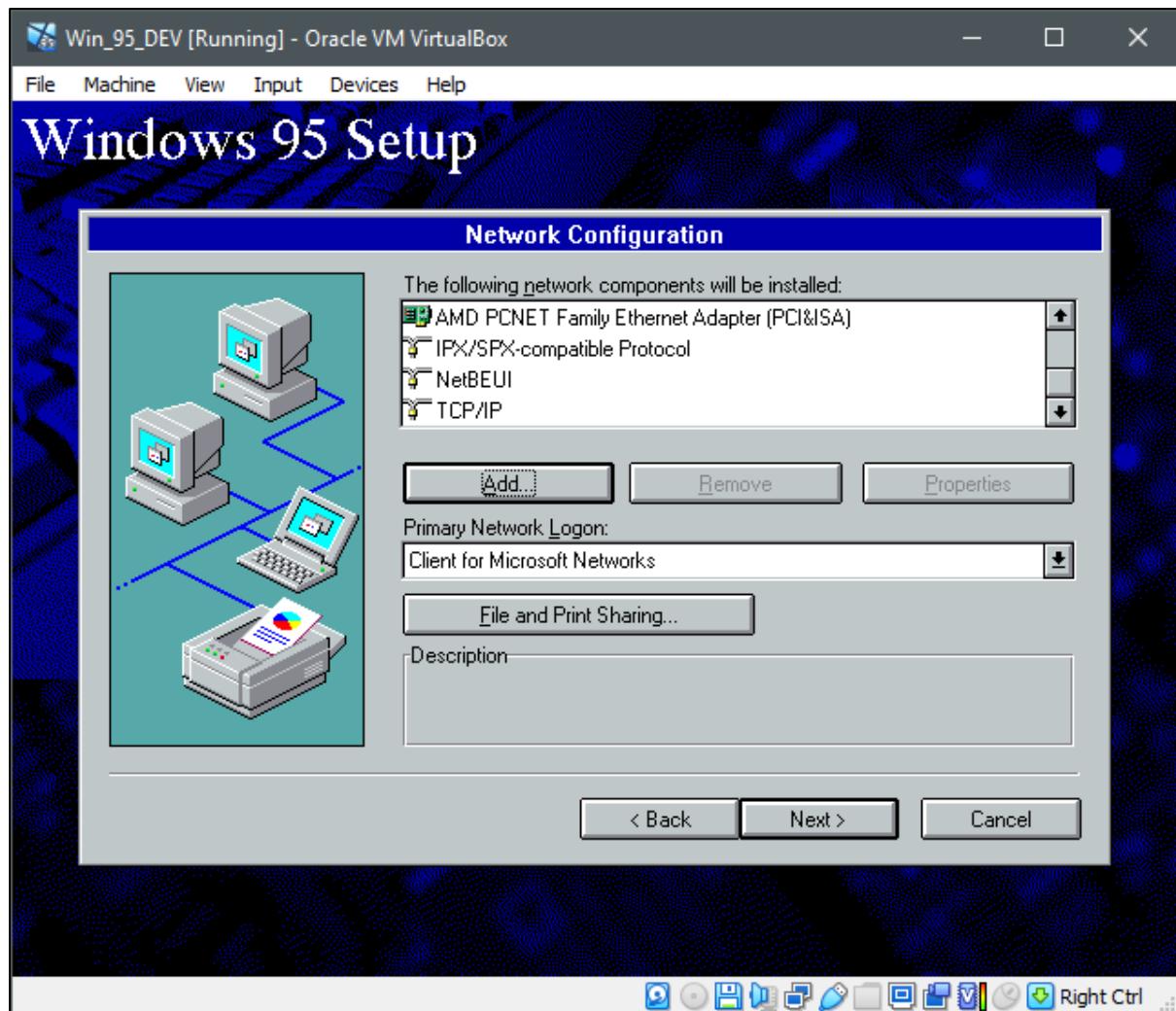
Next select Protocol and click Add.



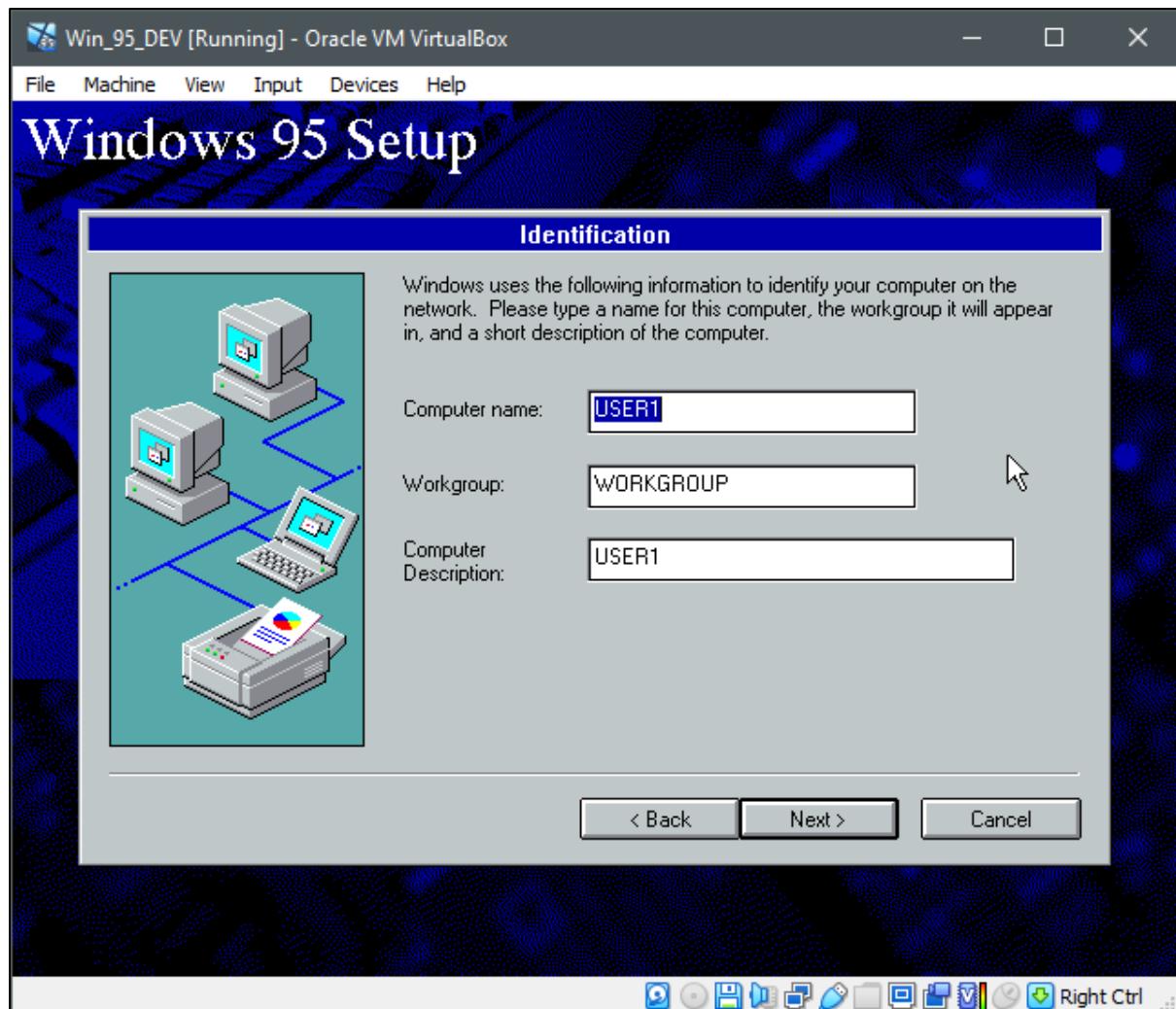
Scroll down to Microsoft and then select TCP/IP and then OK.

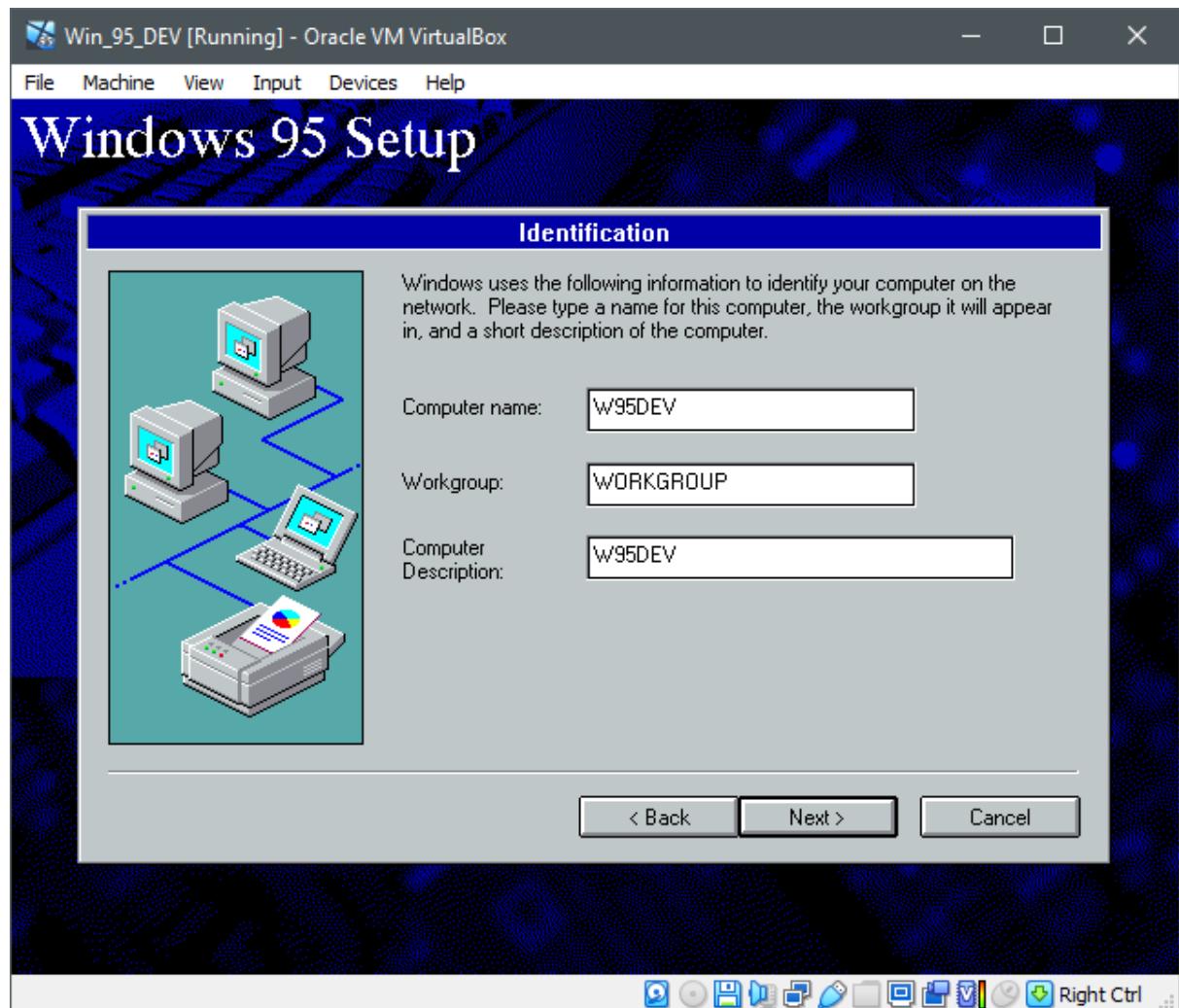


The required components for windows networking and internet will now be installed. Select next to continue.

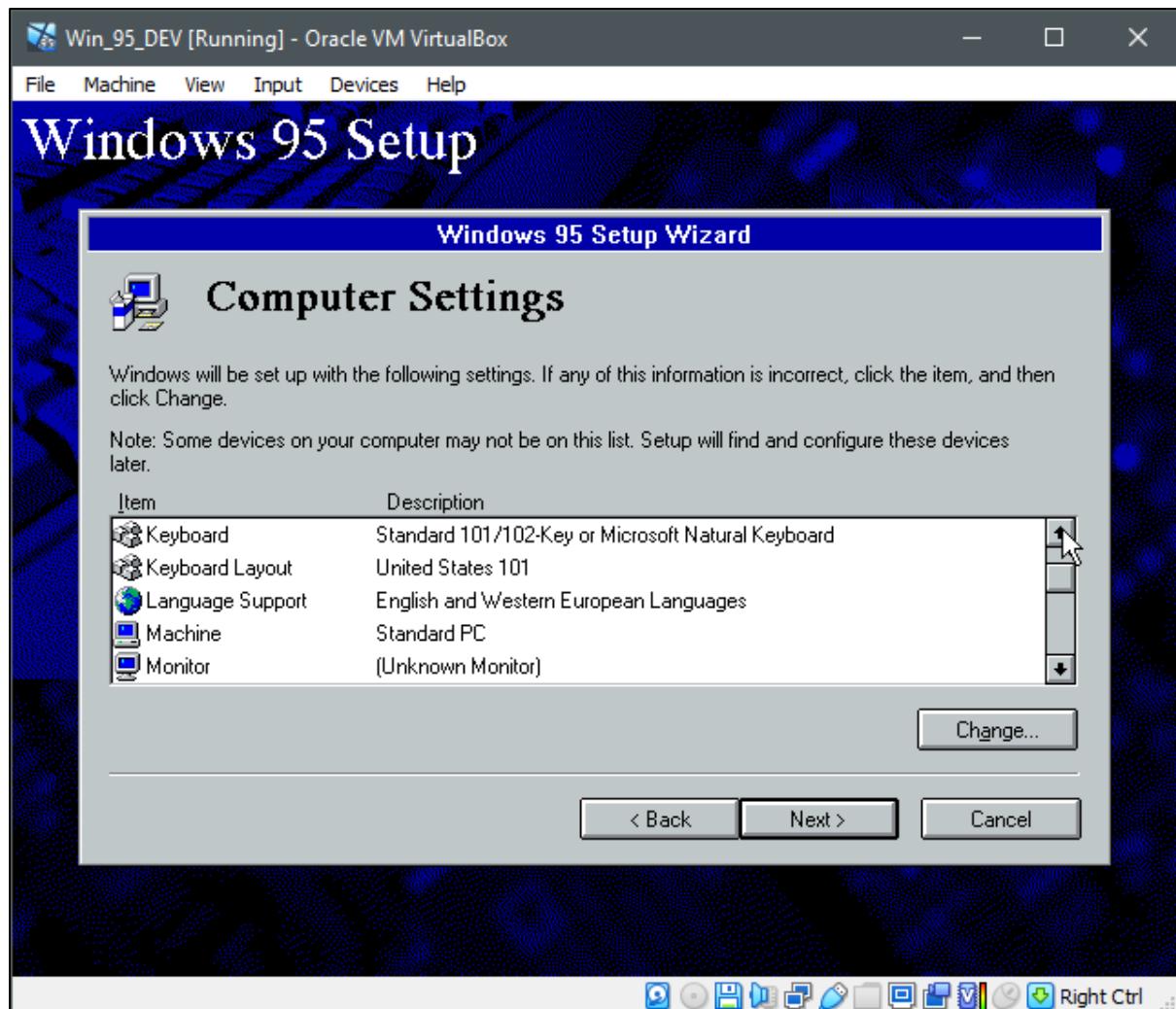


You will now be asked to check your network computer information. You can change the computer name and description if you wish to avoid conflicts.

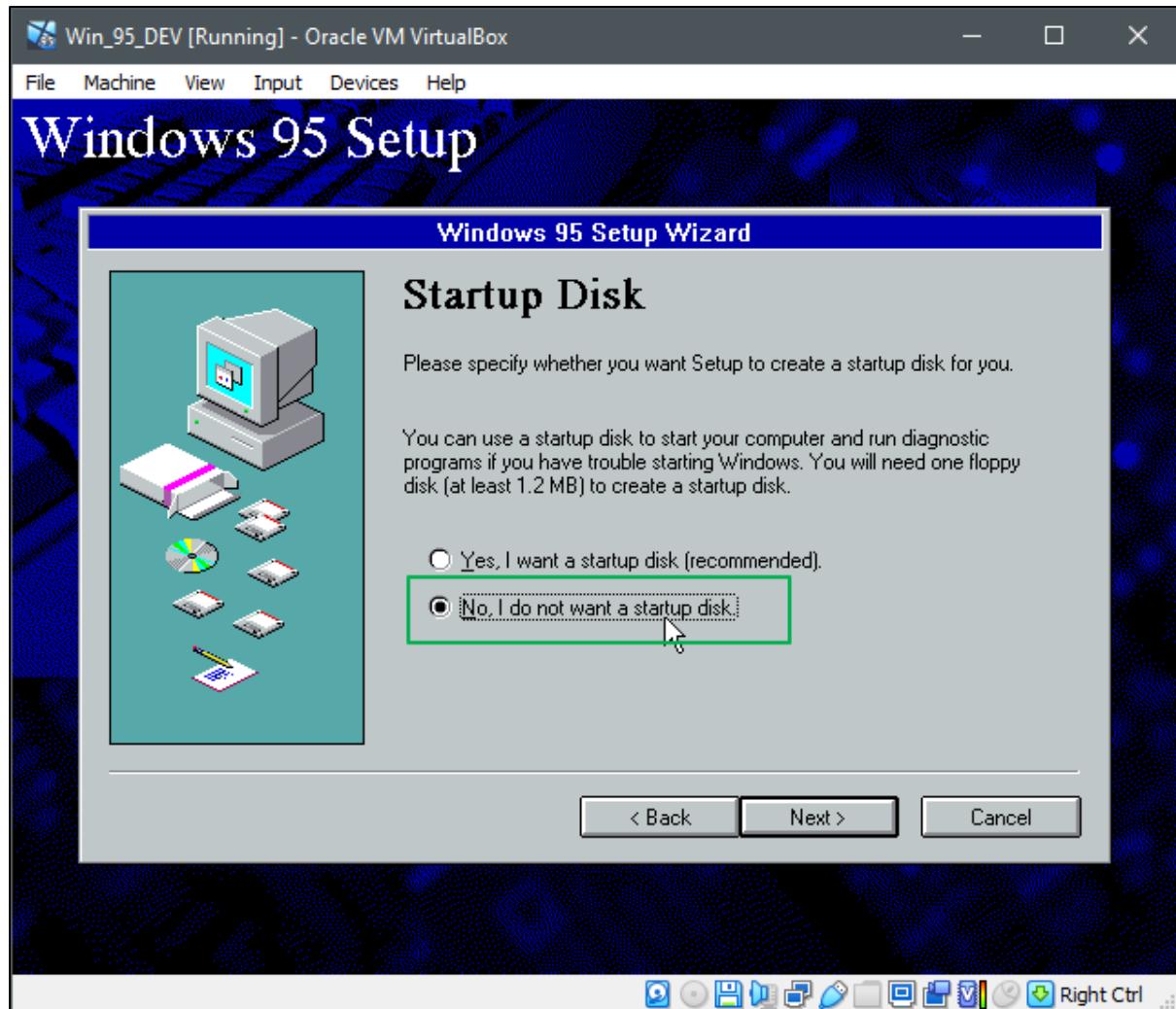




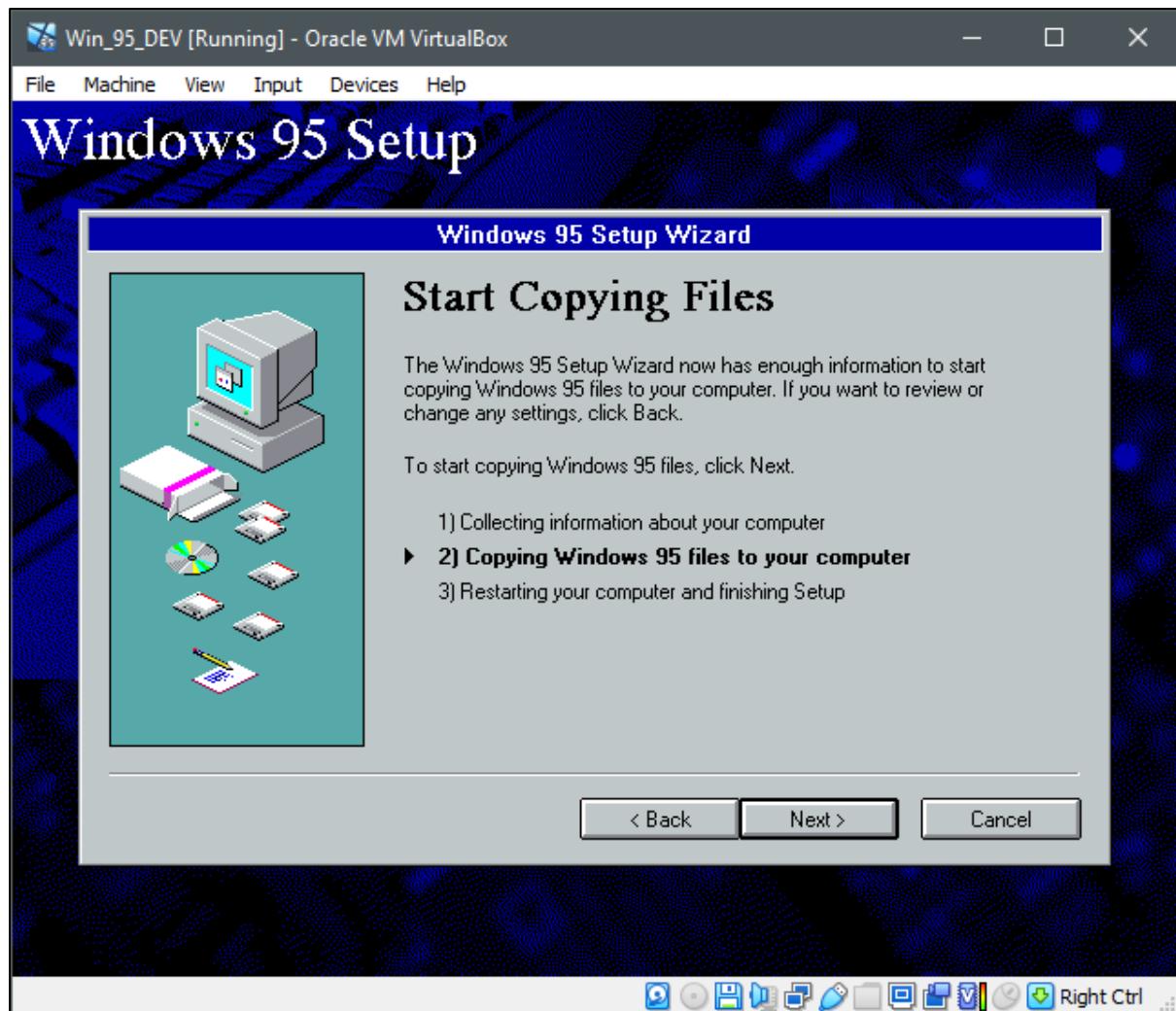
The following is the list of found setting. You can use the back button, or select the item to change any feature that you think is incorrect. Select next to continue.

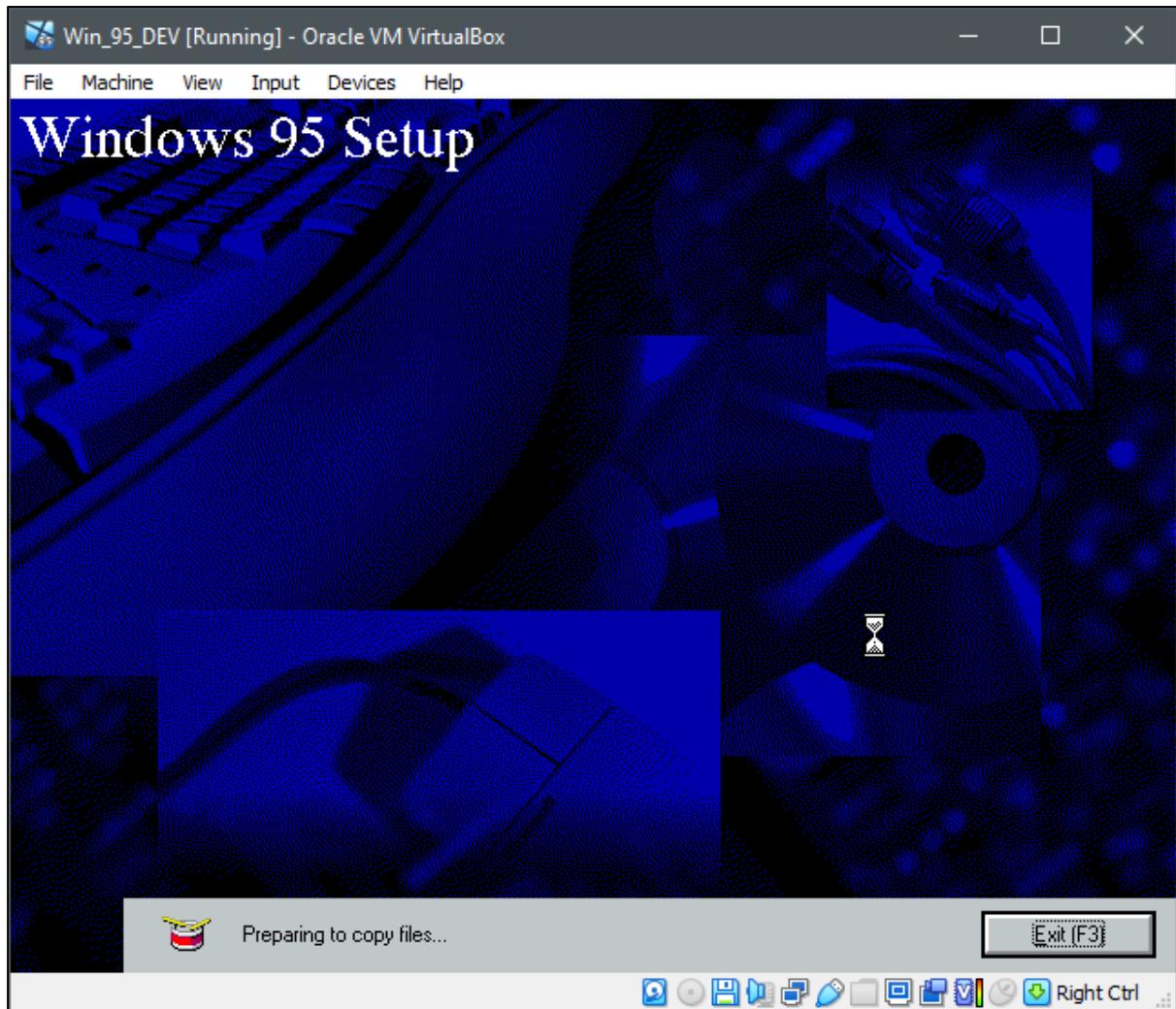


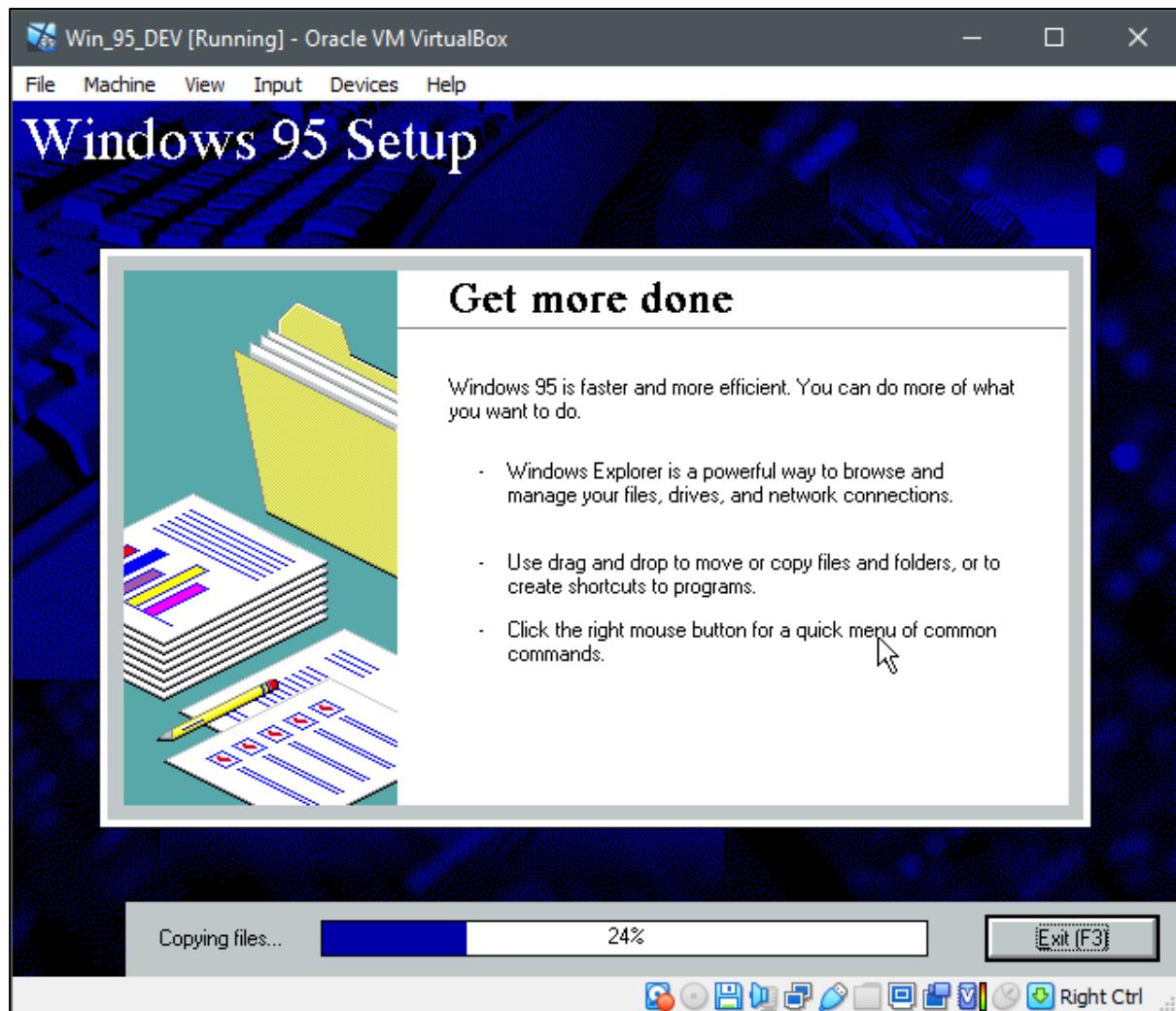
Select No, I do not want a startup disk. We already have the Windows 98 boot disk :)



We have completed setting up the custom components and are now ready to begin copying the files to disk. Select Next to continue. This stage will take a little while to complete.





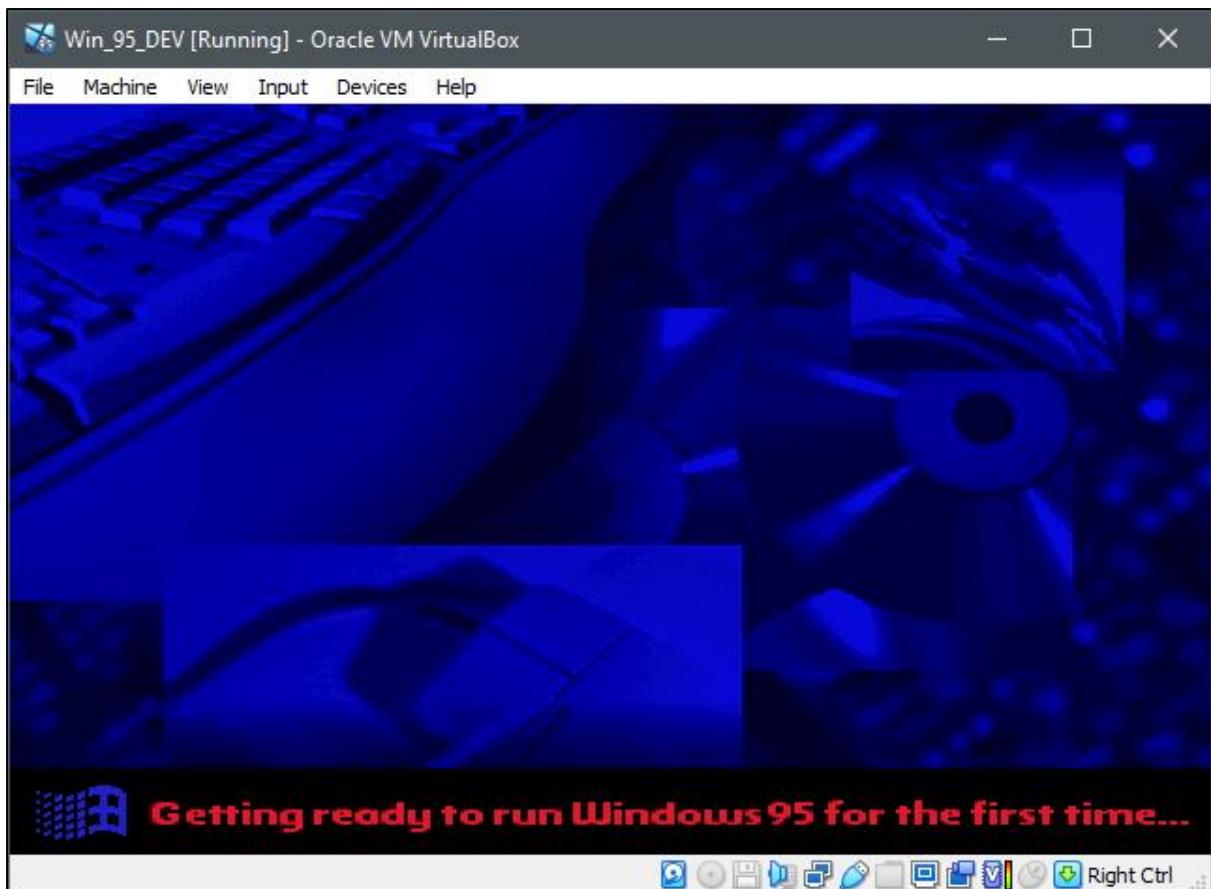


Next you will be asked to restart your computer. After the restart this will be the first run of Windows 95.

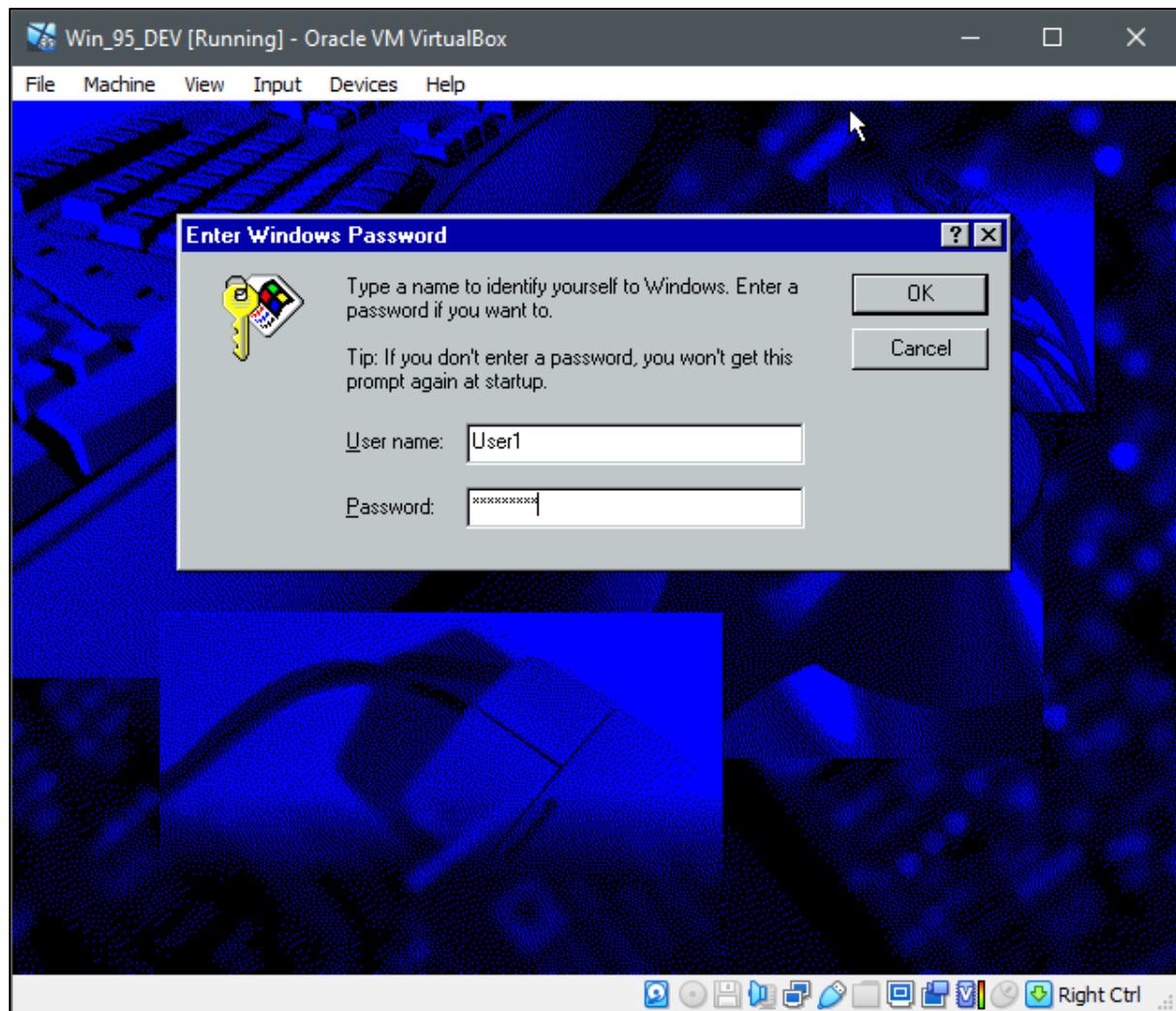
Remove the Floppy drive image from VirtualBox before the restart. If you don't it will boot from the floppy drive into DOS. If that happens remove the floppy drive and reboot the machine again.



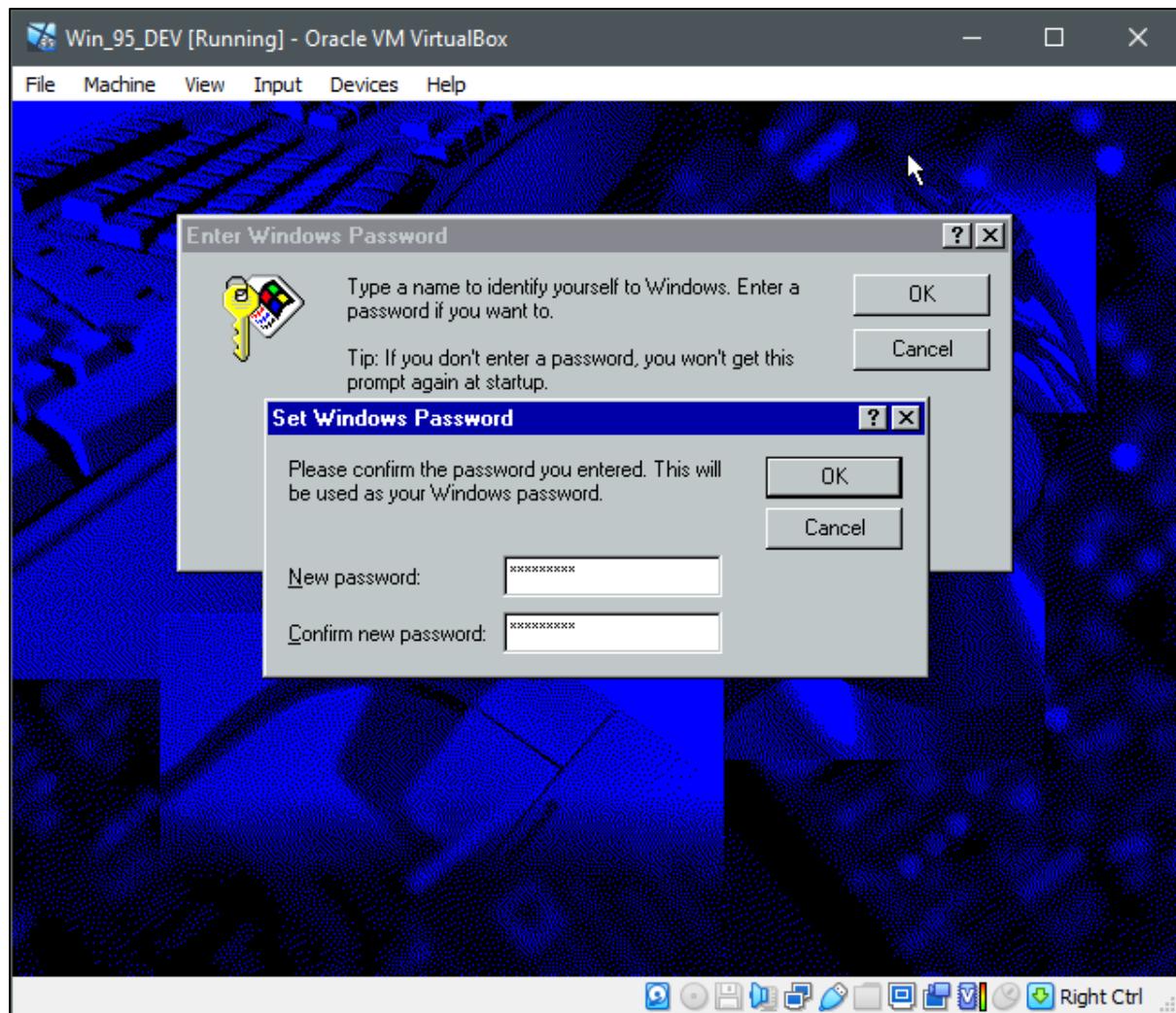
Windows 95 first start.



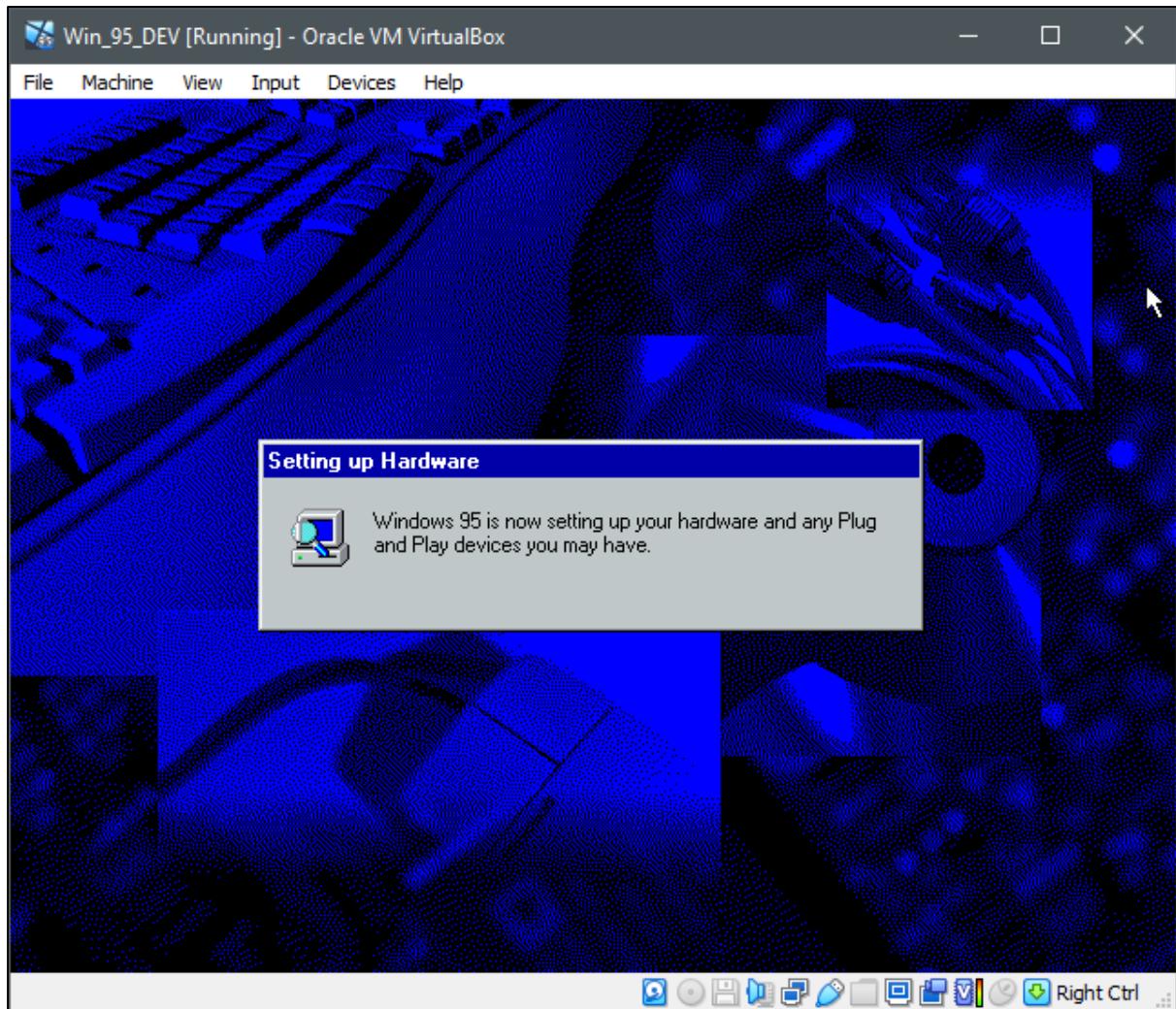
Enter the user name you first used during the setup and create a password.

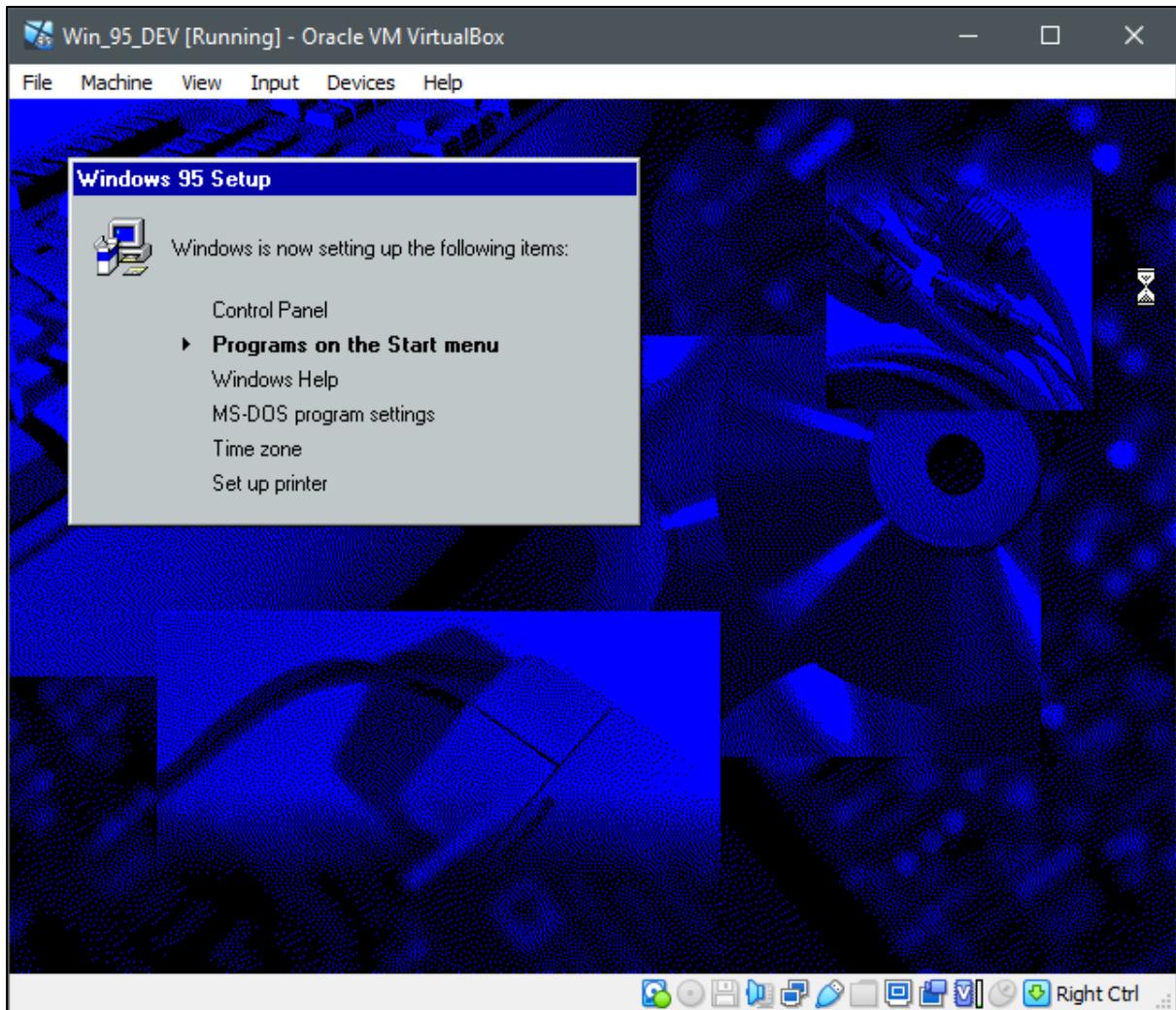


Next confirm your new password and click OK to continue.

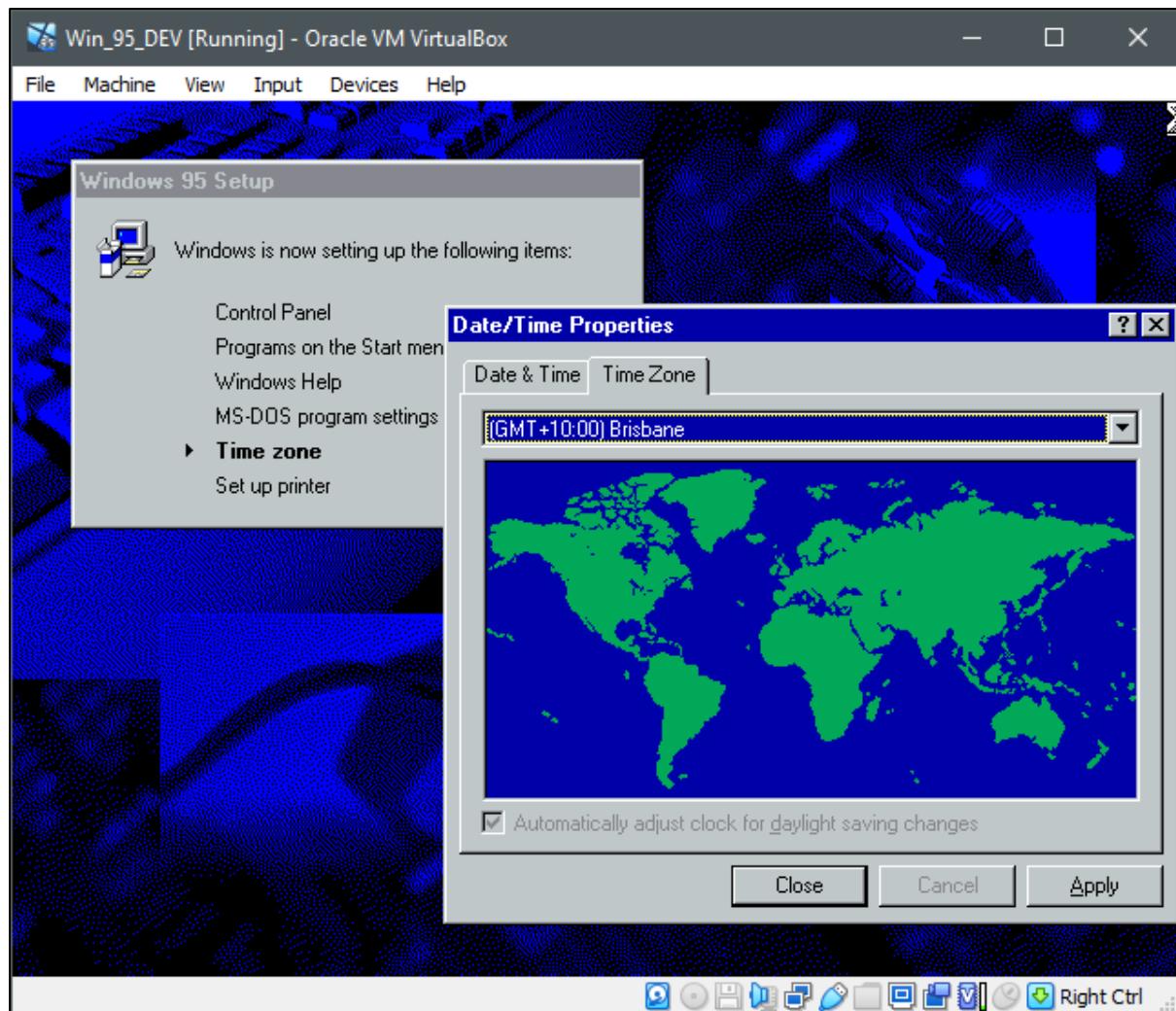


Windows will now continue with the final setup tasks.

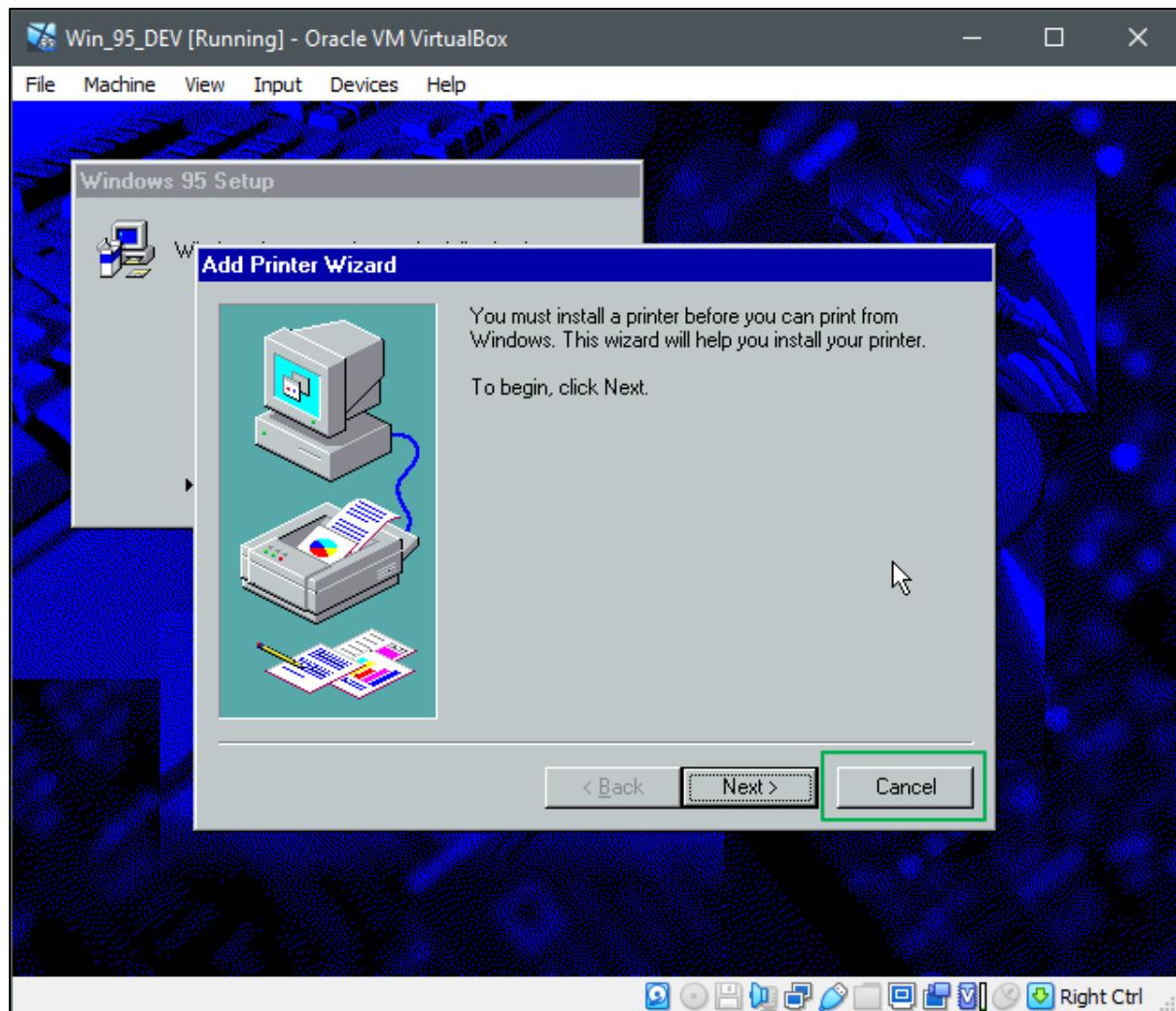




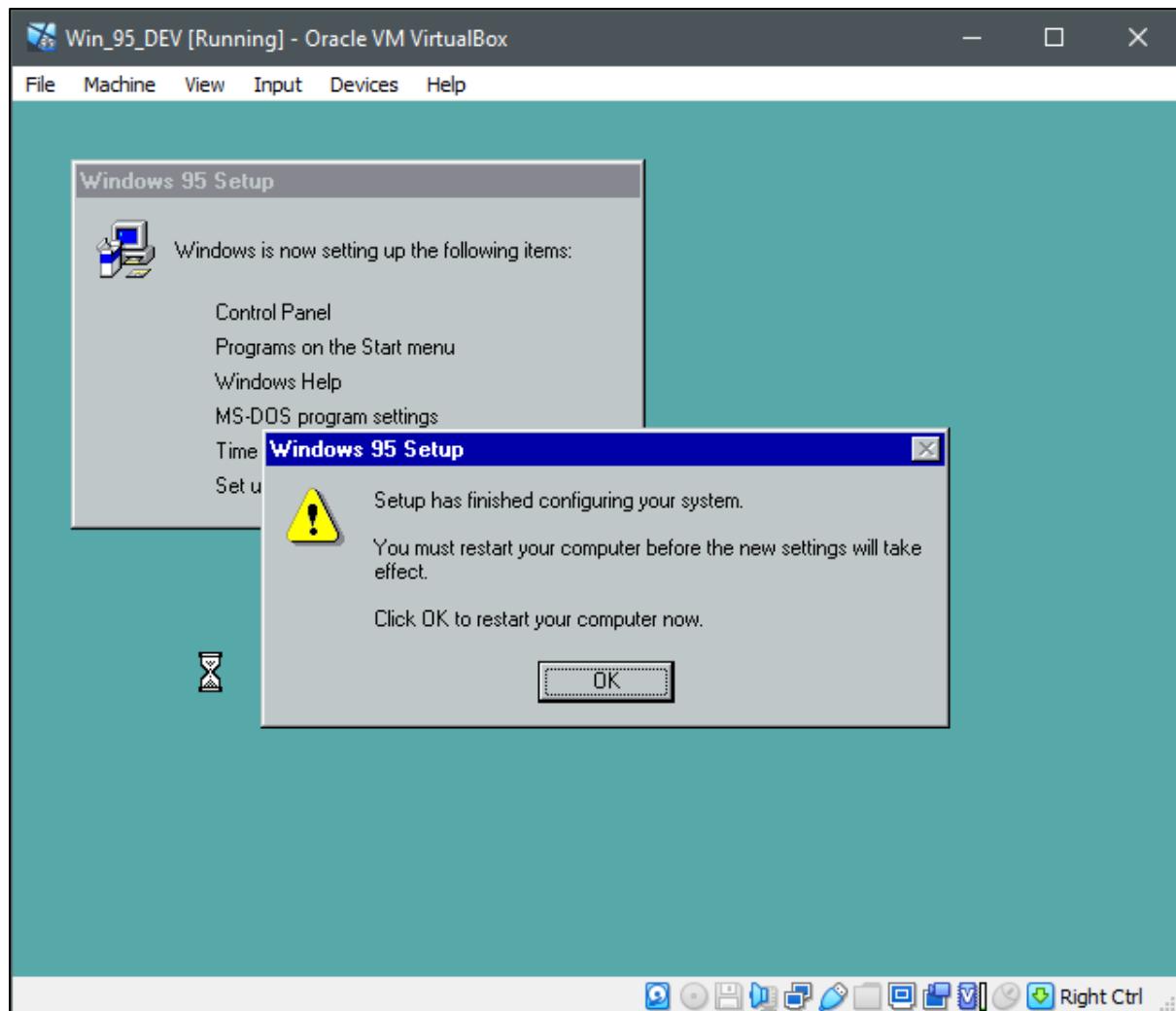
Select the correct time zone from the drop down menu, and then click on “Apply” followed by “OK”.



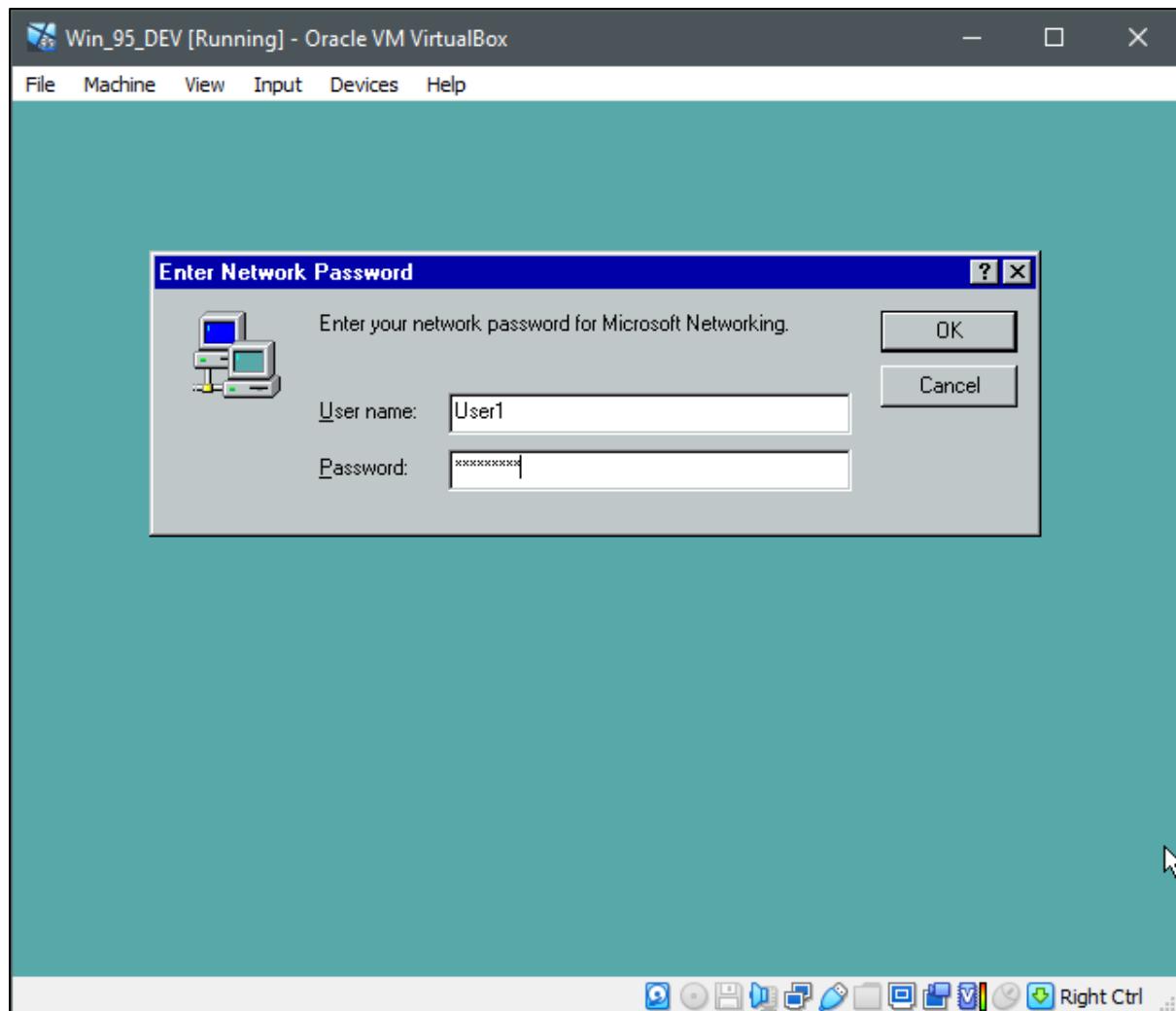
It is OK to select “Cancel” to skip setting up a printer.



You will now be asked for a final restart. Select OK to continue and restart the computer.

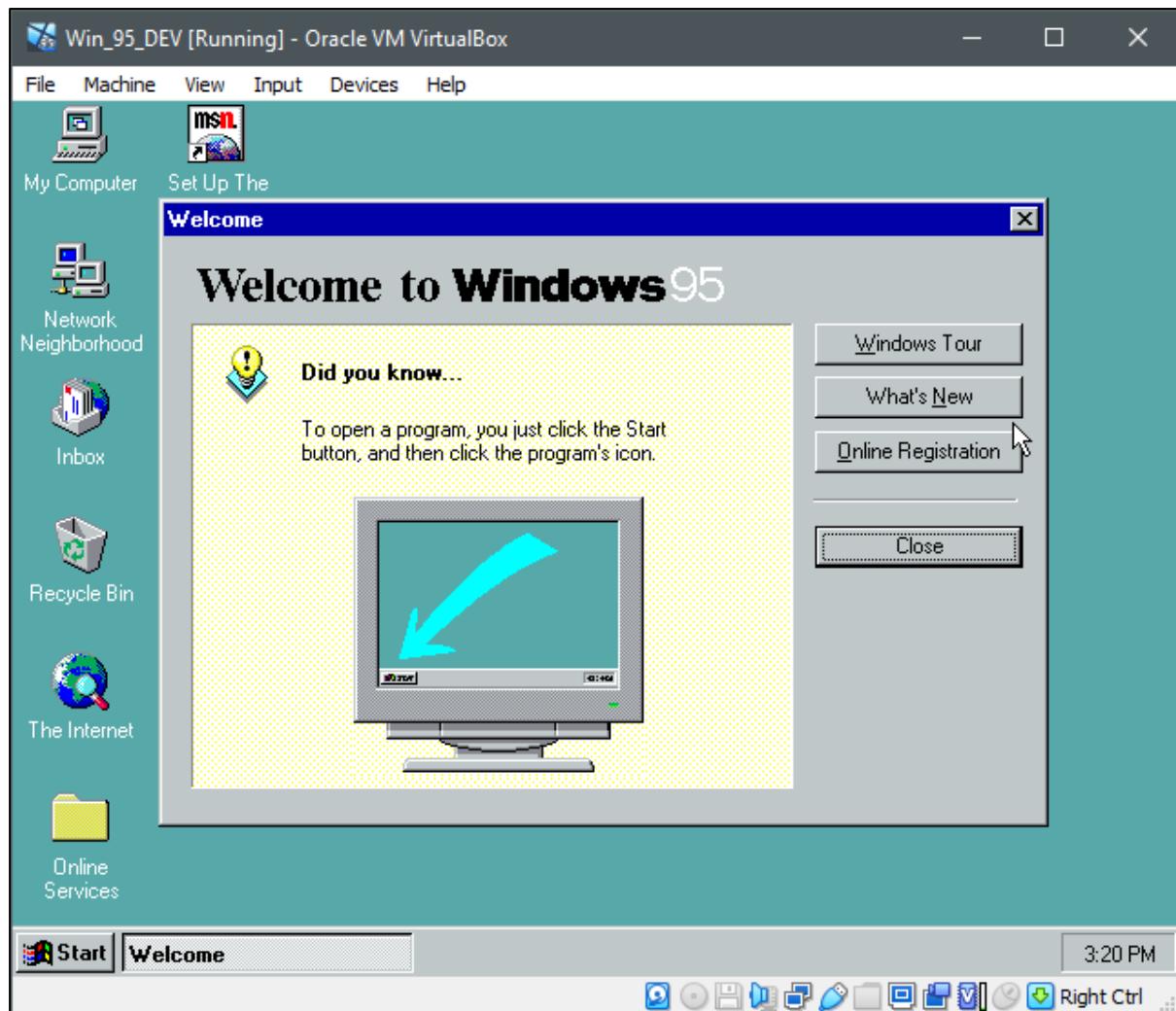


After the restart enter your User name and password to log in to Windows.



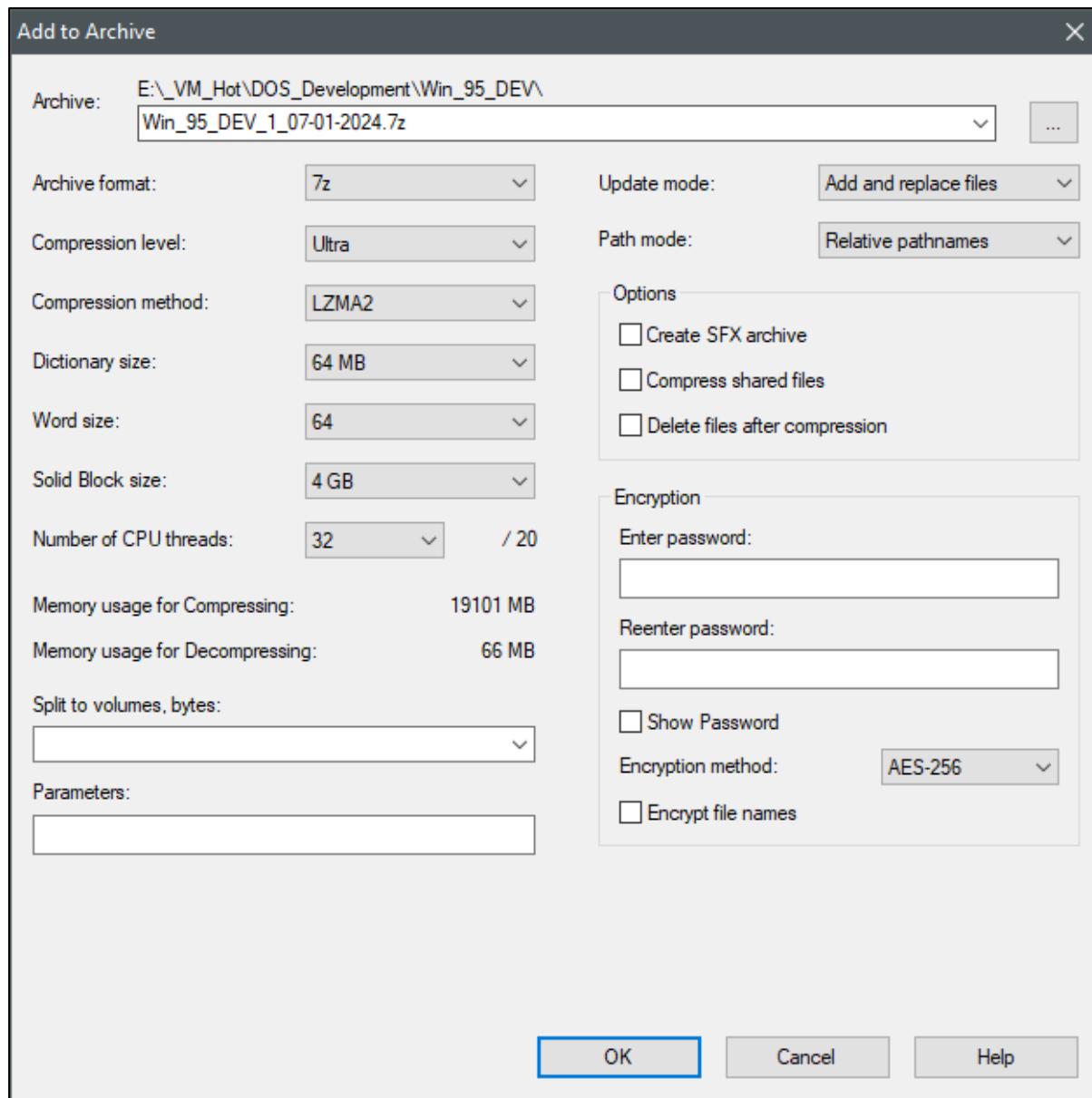
You can close the windows tour if you are already familiar with Windows 95.

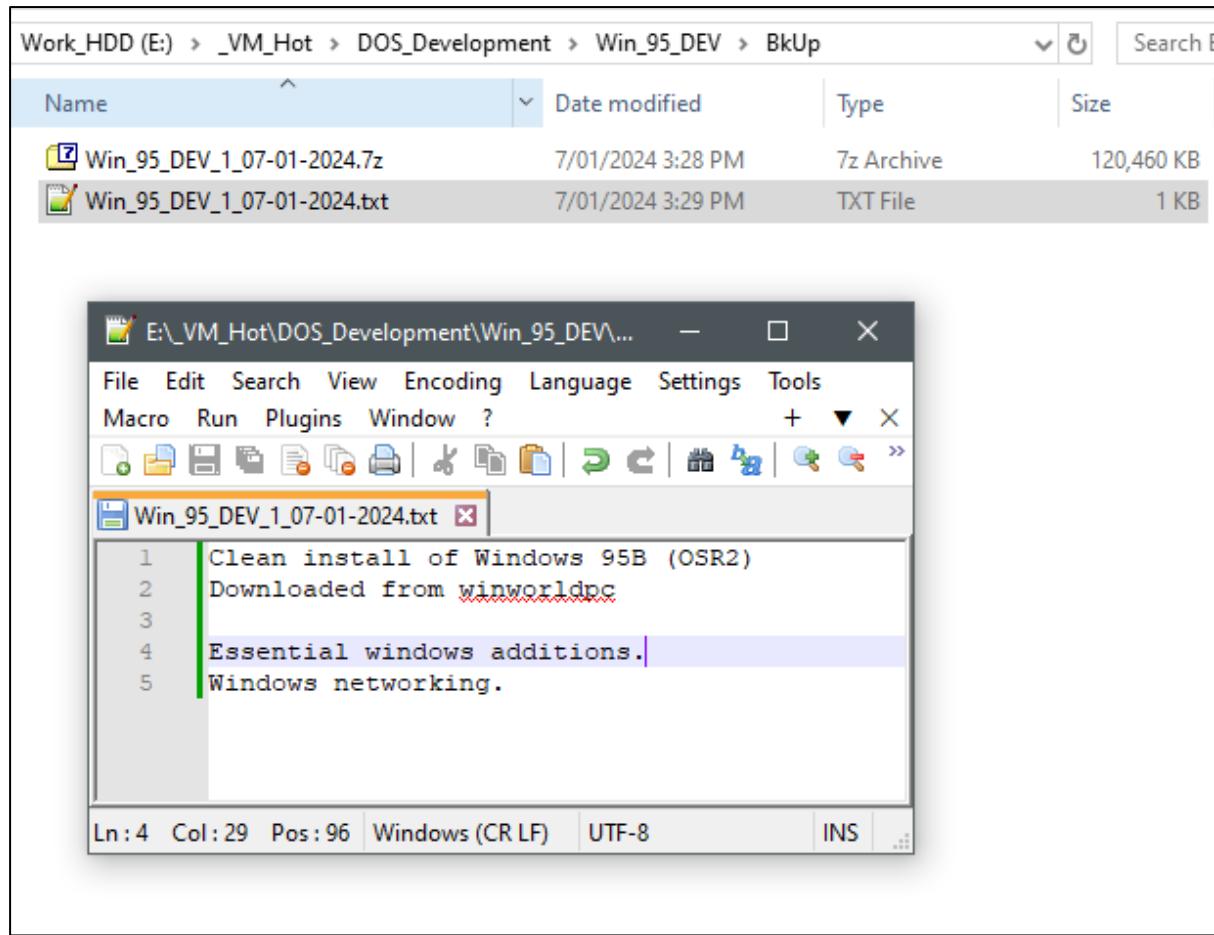
The next time you log in you may be presented with the windows tour again and you will have the opportunity to select "[/]do not show again".



At this point the Windows 95B install is complete.

I would recommend shutting down Windows 95 and creating an archive of the Primary VHD in your VirtualBox folder. You can use this as a clean restore point should you make mistakes setting up the next stages. I normally use 7-zip for this task and apply a time stamp to the archive name. Store the archive in a BkUp folder and keep a note of what was installed for later reference.





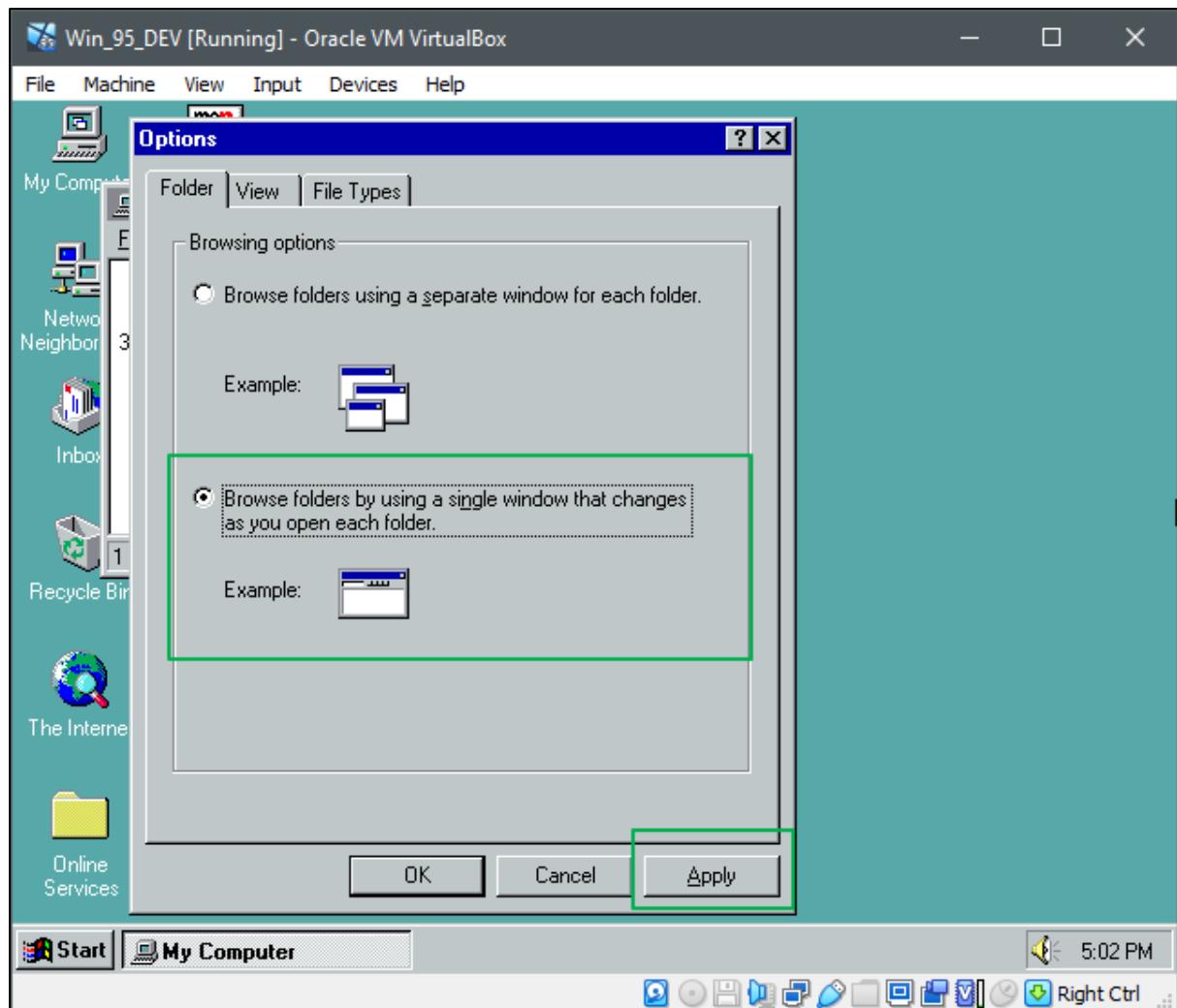
### Quick usability tweaks

#### Explorer

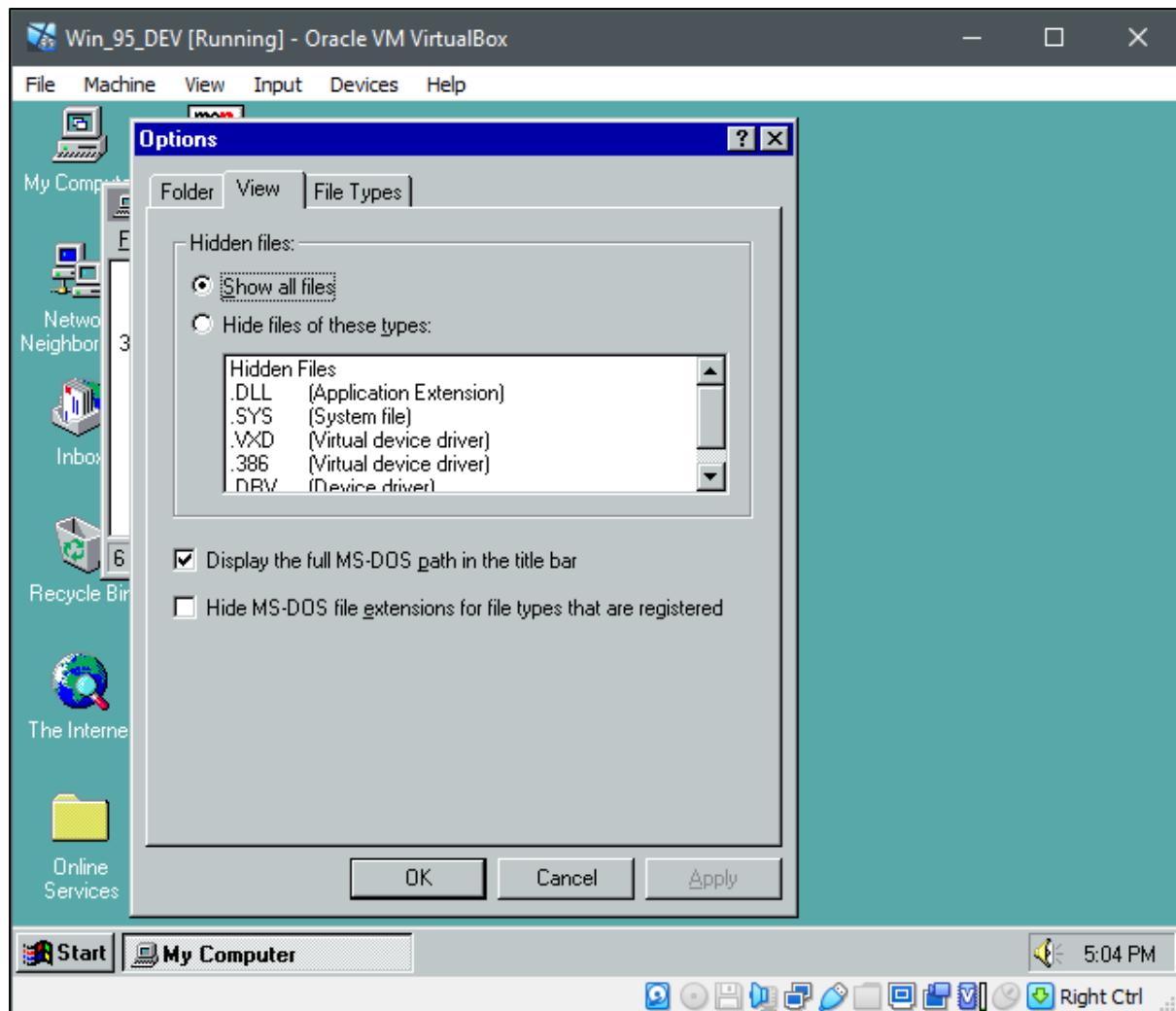
Open MyComputer in the file manager.

Under “View” in the menu bar select “Options”.

Change the folder setting to open in a single window.



Select the View tab and change the following 3 settings.



Remember to select **apply**, and then OK to close the settings.

From the explorer window select “View” from the menu bar and make sure “Tool Bar” is selected.

Windows 95 is known for not keeping the explorer settings. Set the above setting then close the explorer window and reopen. You may have to do this a number of times to get a view setting to stick in place.

### Fix Mouse – scroll

Windows 95 did not come with default drivers for a modern scroll wheel mouse so you will need to install a 3rd party driver to access the advanced mouse controls.

IBM Mouse Suite V 1.69

(ScrollPoint II mouse for Windows 95/98/NT 4.0 - QJ6Z34US - IBM IntelliStation Z Pro (Type 6866))

“qj6z34us.exe”

<https://www.ibm.com/support/pages/scrollpoint-ii-mouse-windows-9598nt-40-qj6z34us-ibm-intellistation-z-pro-type-6866>

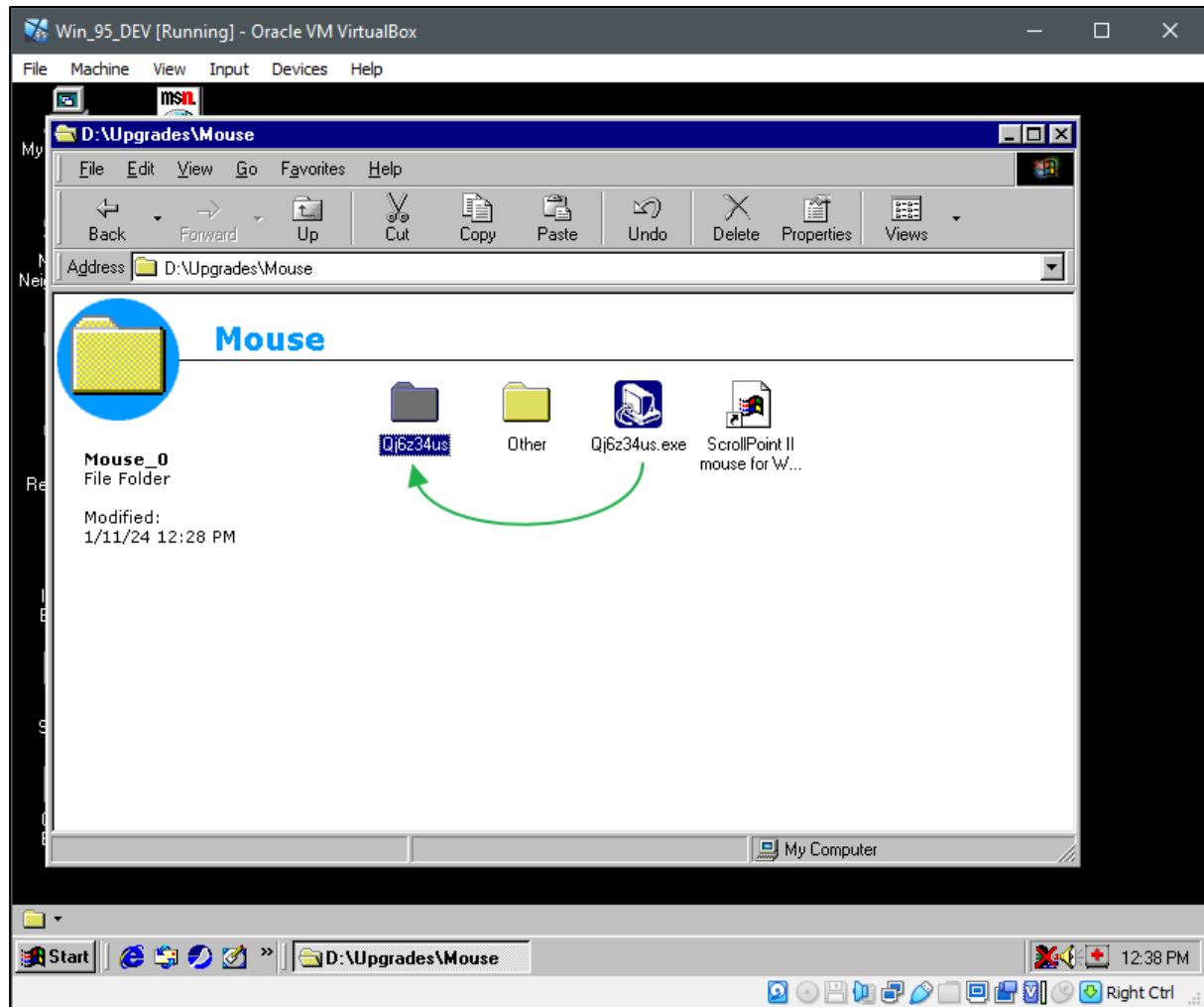
## A Beginners Guide To DOS Programming

There are other mouse drives and versions of this IBM driver that may also work, but I have found the qj6z34us.exe to work well.

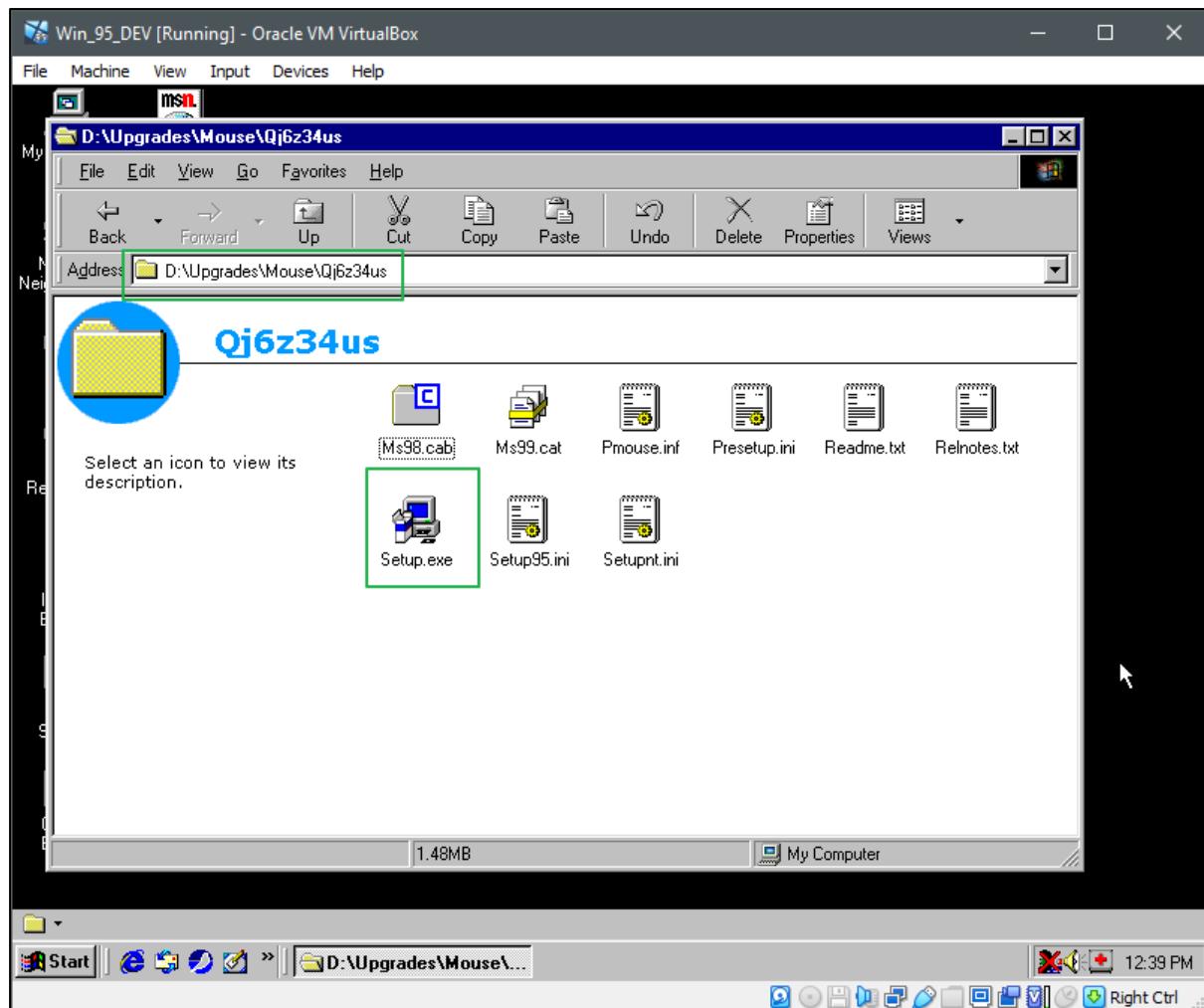
Unpack the qj6z34us.exe archive with 7-Zip to a directory \qj6z34us.

Transfer the \qj6z34us directory to your Windows 95 drive and run the installer.

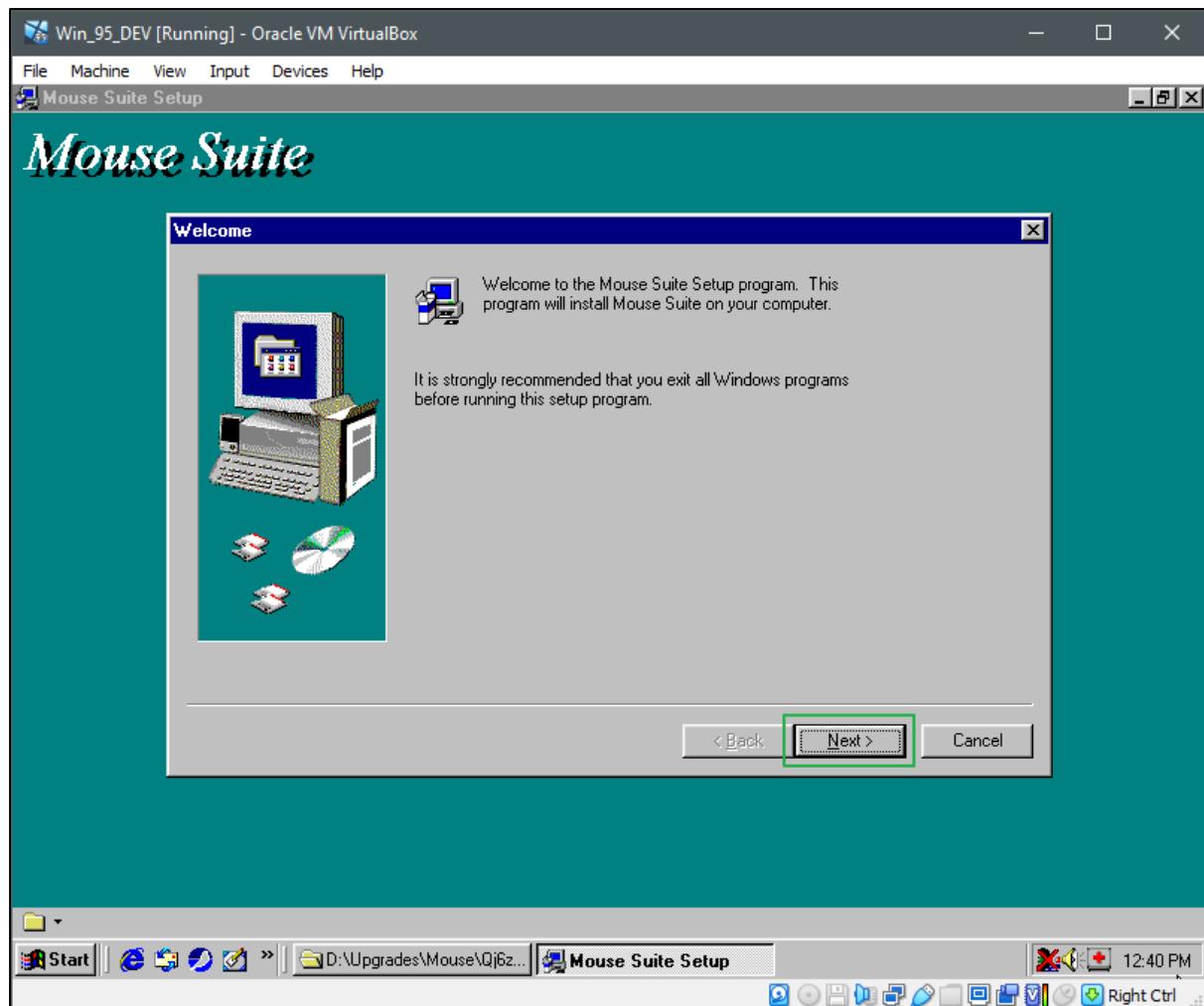
Note you can run the qj6z34us.exe in later upgraded windows. The only difference is the archive will unpack to the \qj6z34us directory on the C: drive before running the actual install setup.



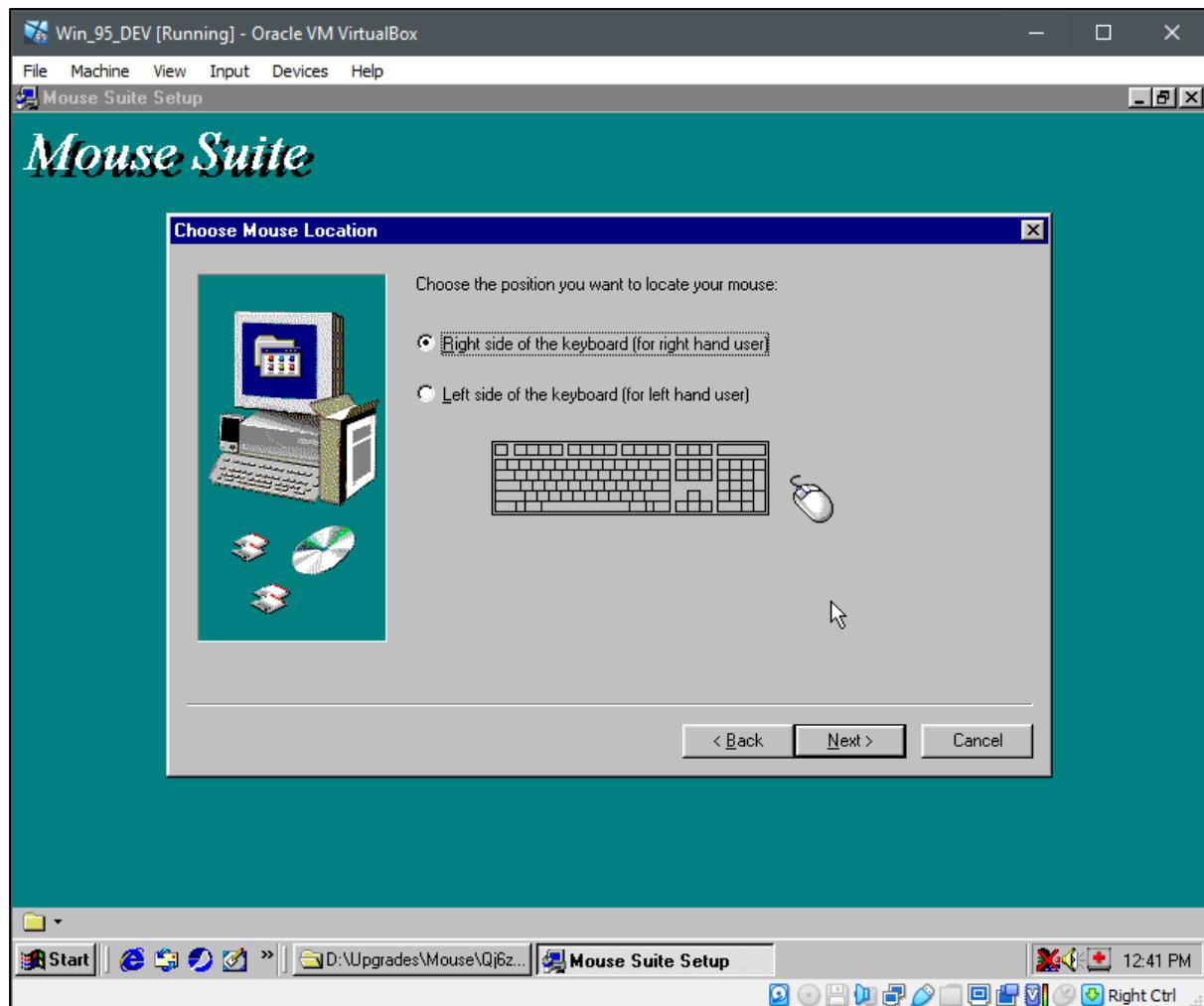
If you have any issue running the installer, Unpack the installer archive with 7-zip and run the setup.exe file.

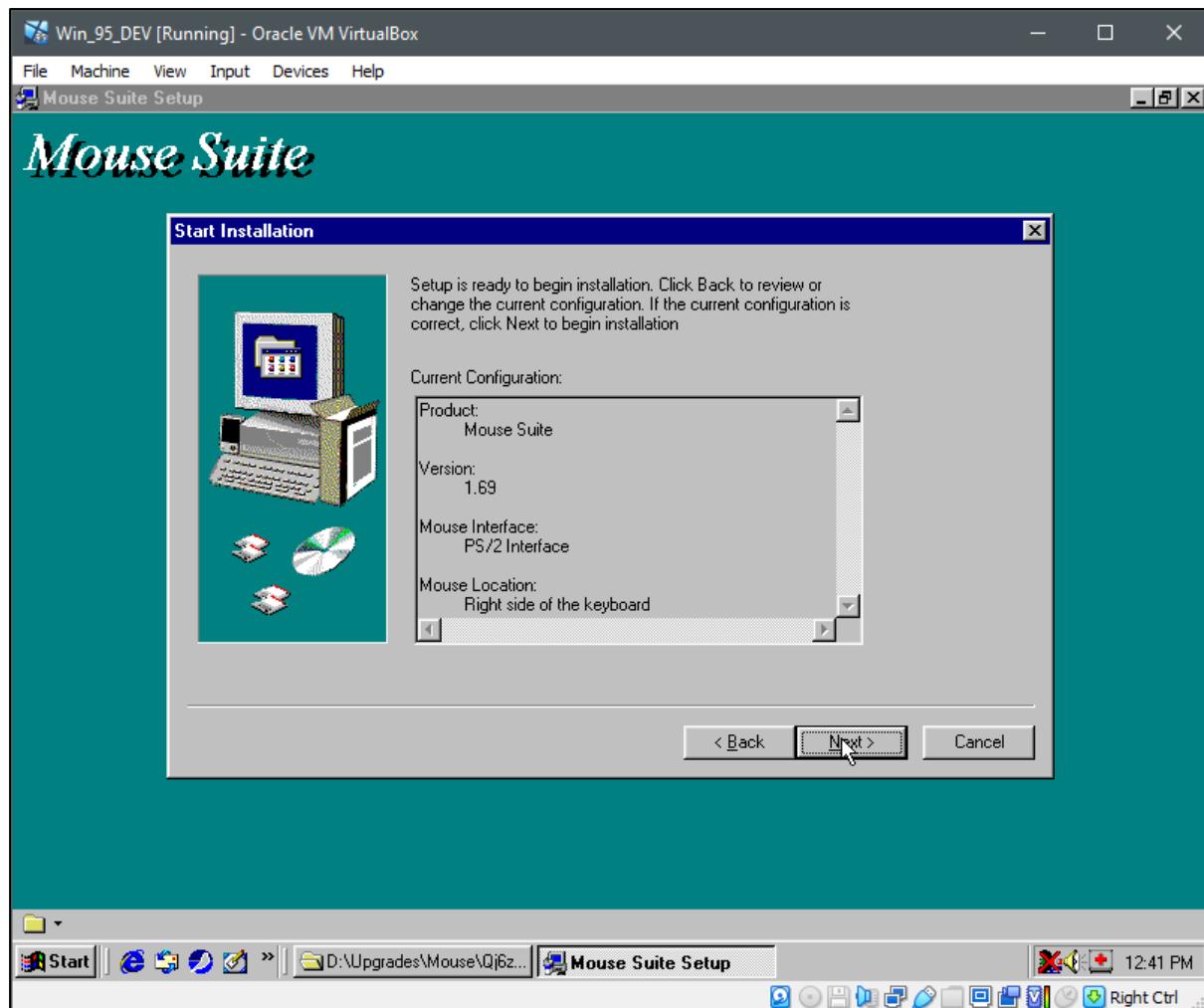


Follow the default install prompts.

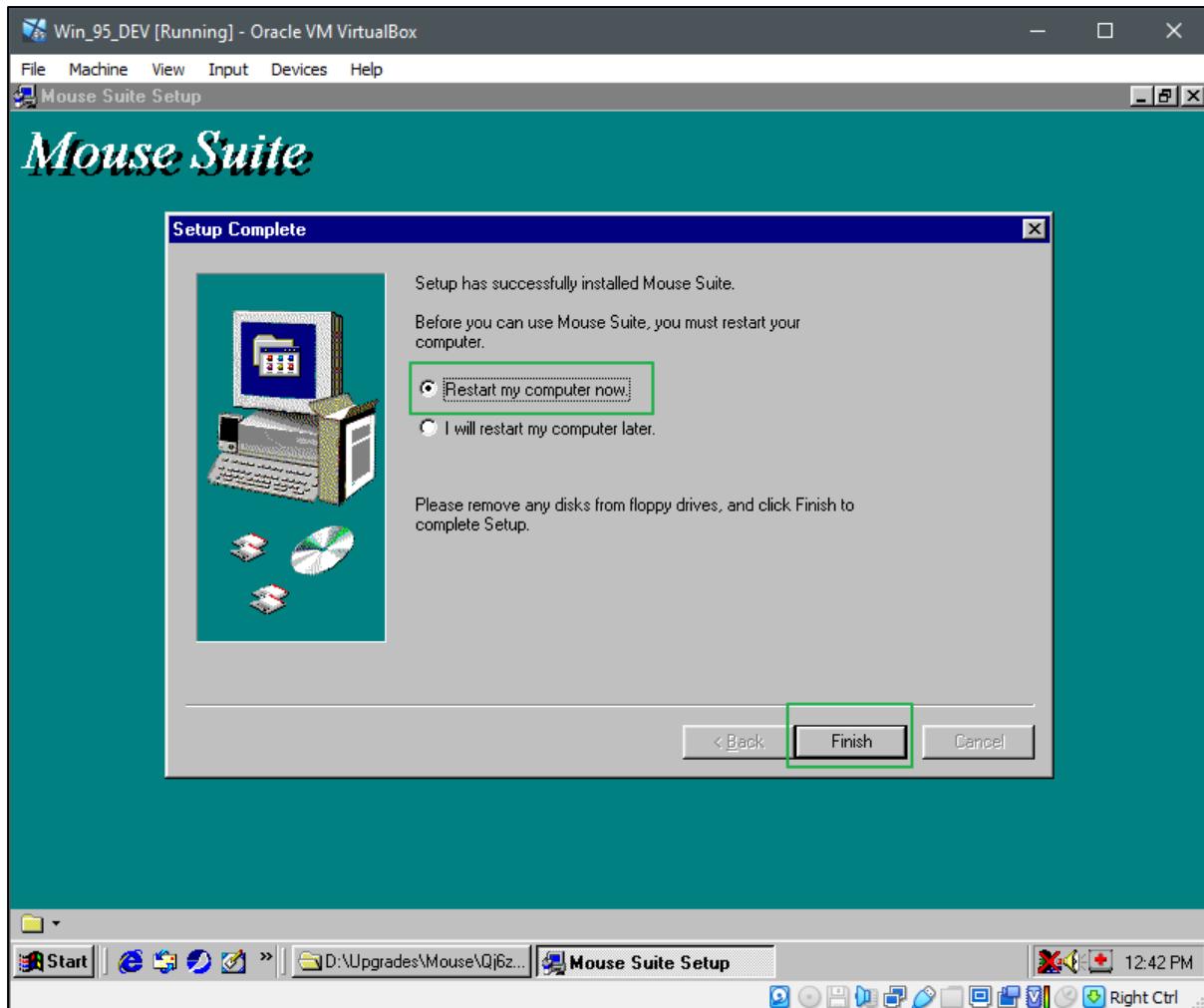


Select Right or Left handed.



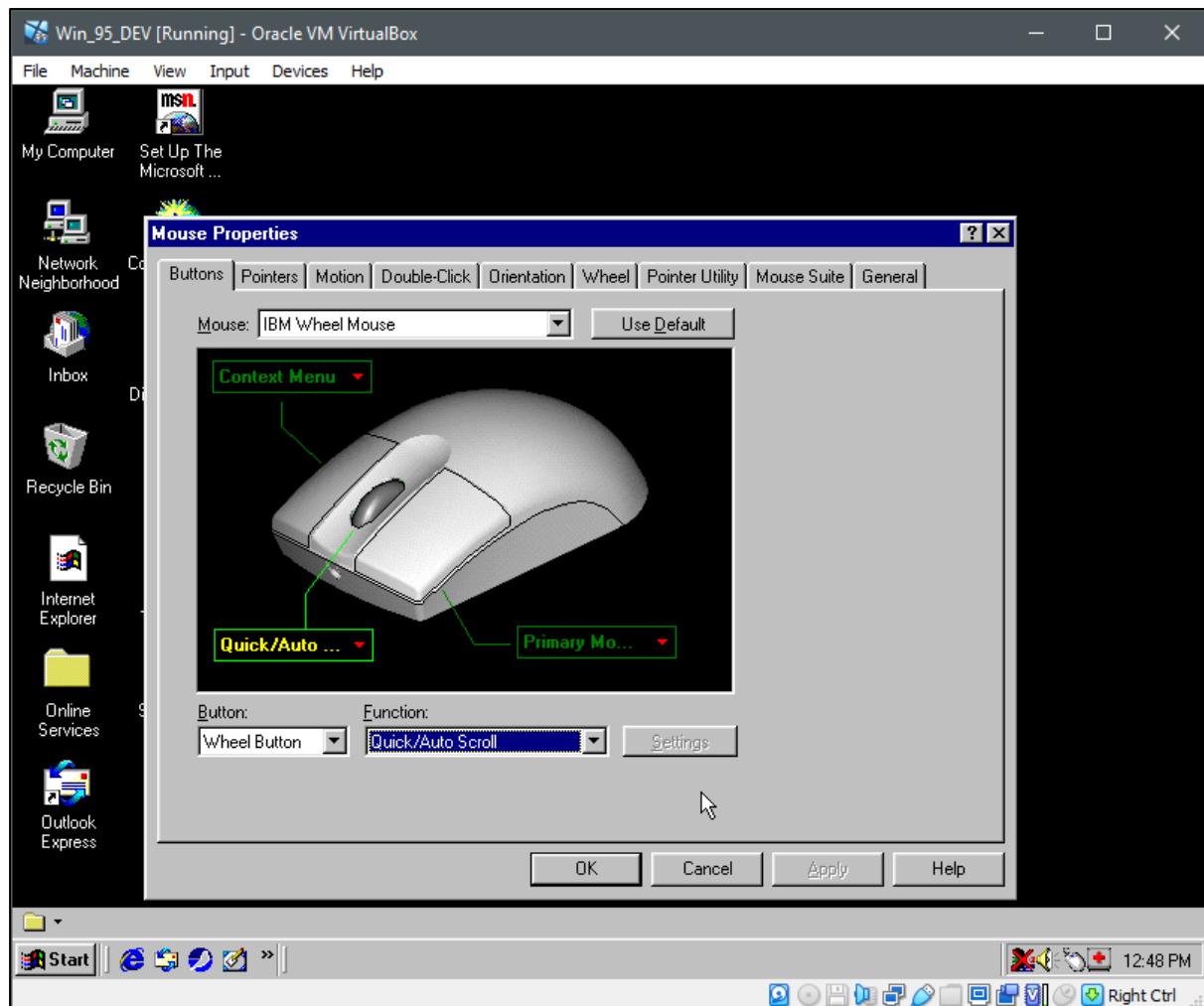


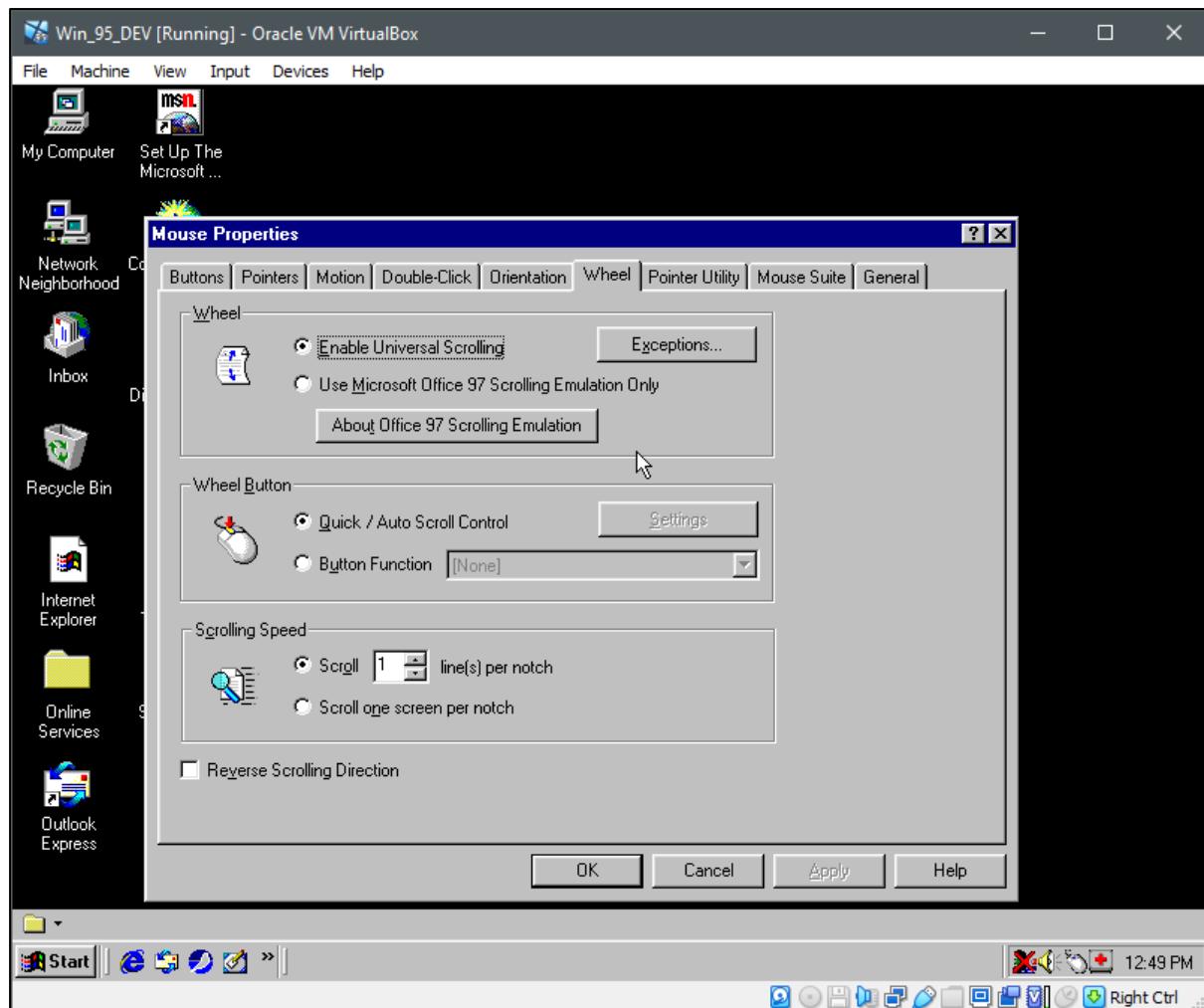
Restart the Windows 95 computer.



The new mouse drivers will be activated during the restart. You should now have a mouse icon in the system tray that will allow you do adjust the advanced mouse setting including the scroll wheel.







### Windows 95 Display drivers fix

Note: A version for DOS (FreeDOS) is also available in the package if you need to correct FreeDOS display issues in VirtualBox.

SciTech Display Doctor contains all of the required drivers to correct the known display problems when running Windows 9x on VirtualBox, because we can't install the VirtualBox additions (Containing the drivers).

The following guide assumes you are using the VBoxSVGA or VBoxVGA controller.

You can download this application from a number of sources. What you are looking for is the "scitech-display-doctor-7.iso" which contains the "scitech-display-doctor-win-7.0.0.340-beta.exe" Install application. Registration keys will be included in the archive.

An older version (6.53) is available at:

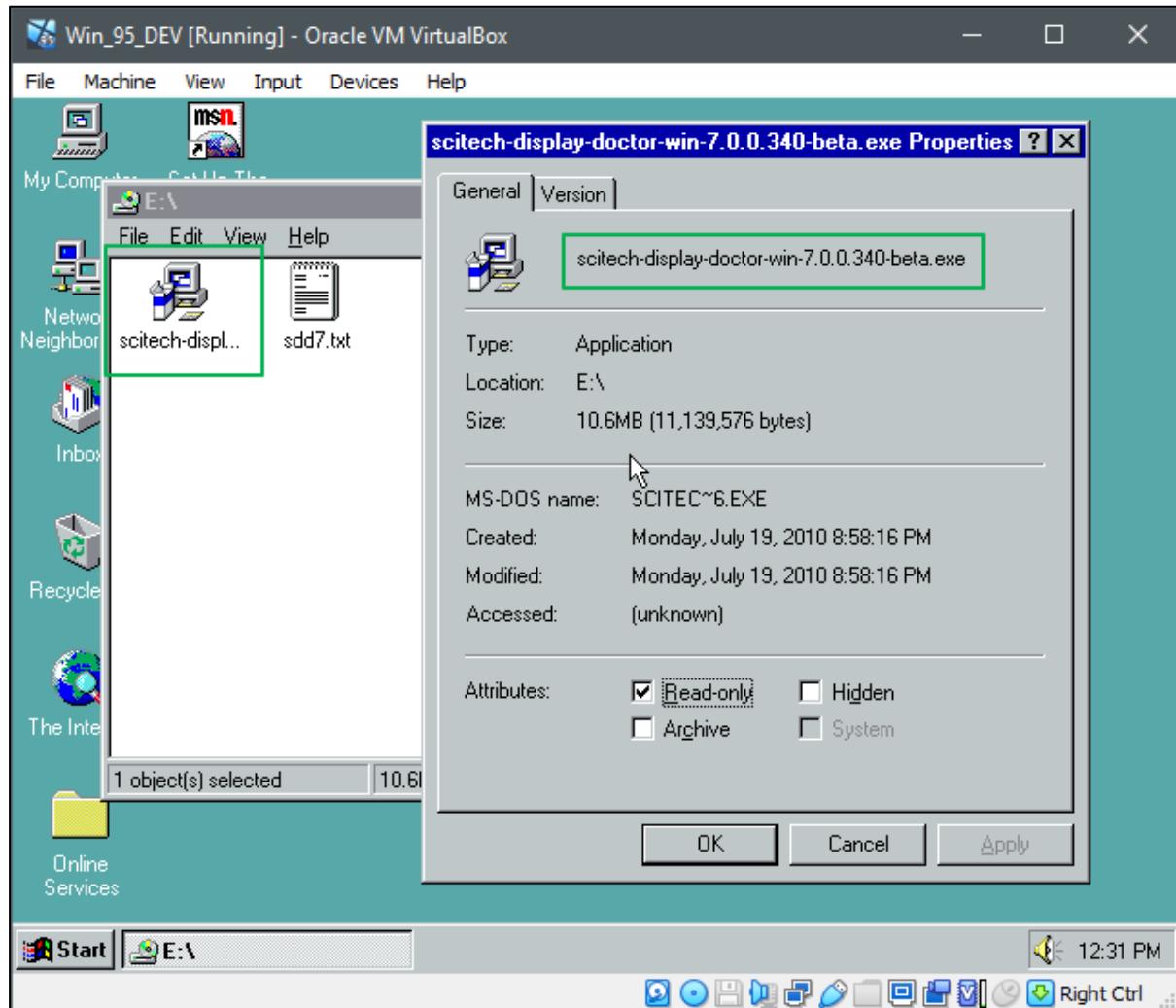
<https://www.mdgx.com/speed.htm>

Under "VESA BIOS Extensions (VBE)" File "sdd653-w.exe".

I will be using “scitech-display-doctor-win-7.0.0.340-beta.exe”

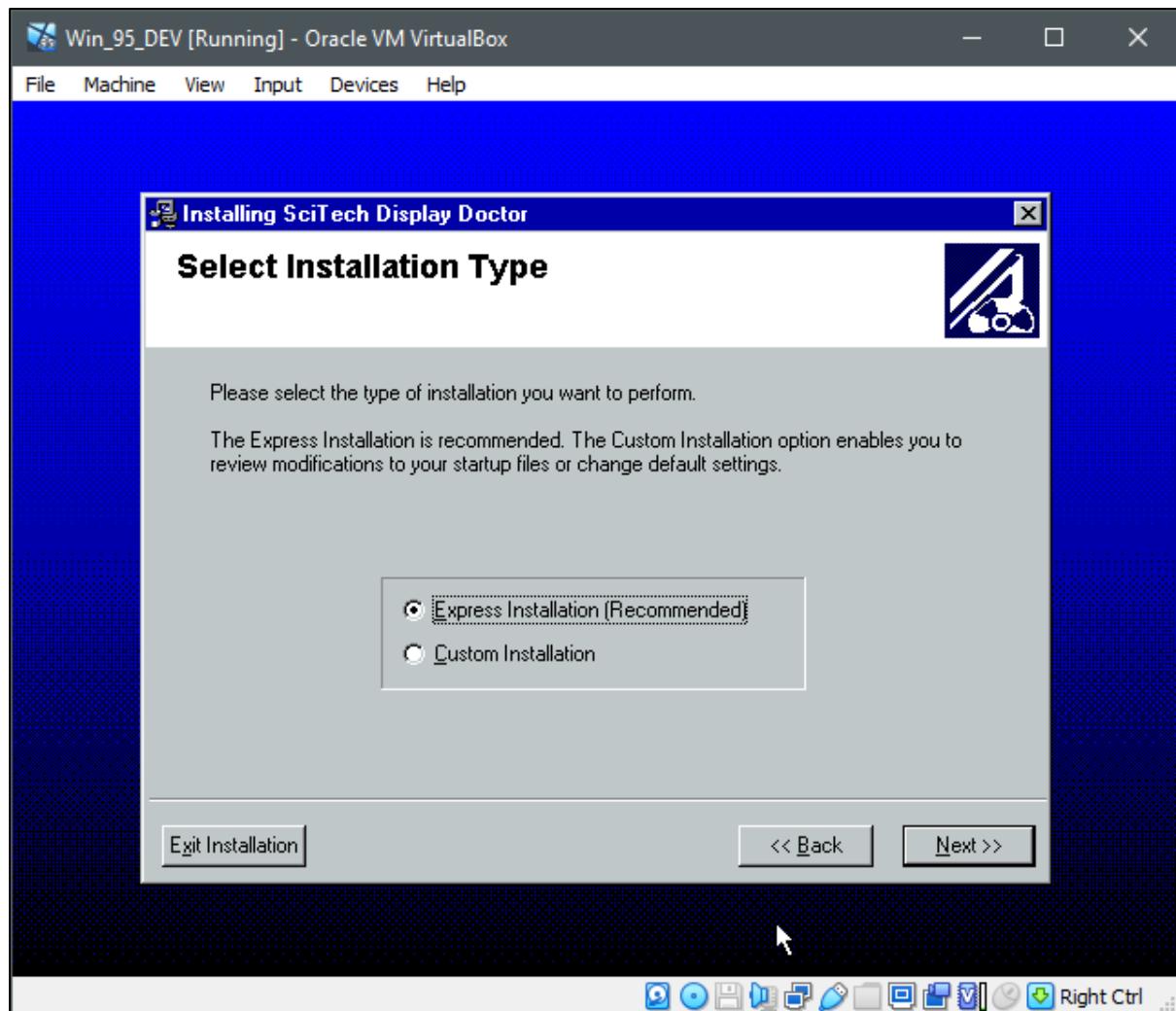
Start the windows 95 virtual machine and then mount the “scitech-display-doctor-7.iso” in the VirtualBox devices.

From Windows 95 navigate to the root of the CD drive and run the installer.

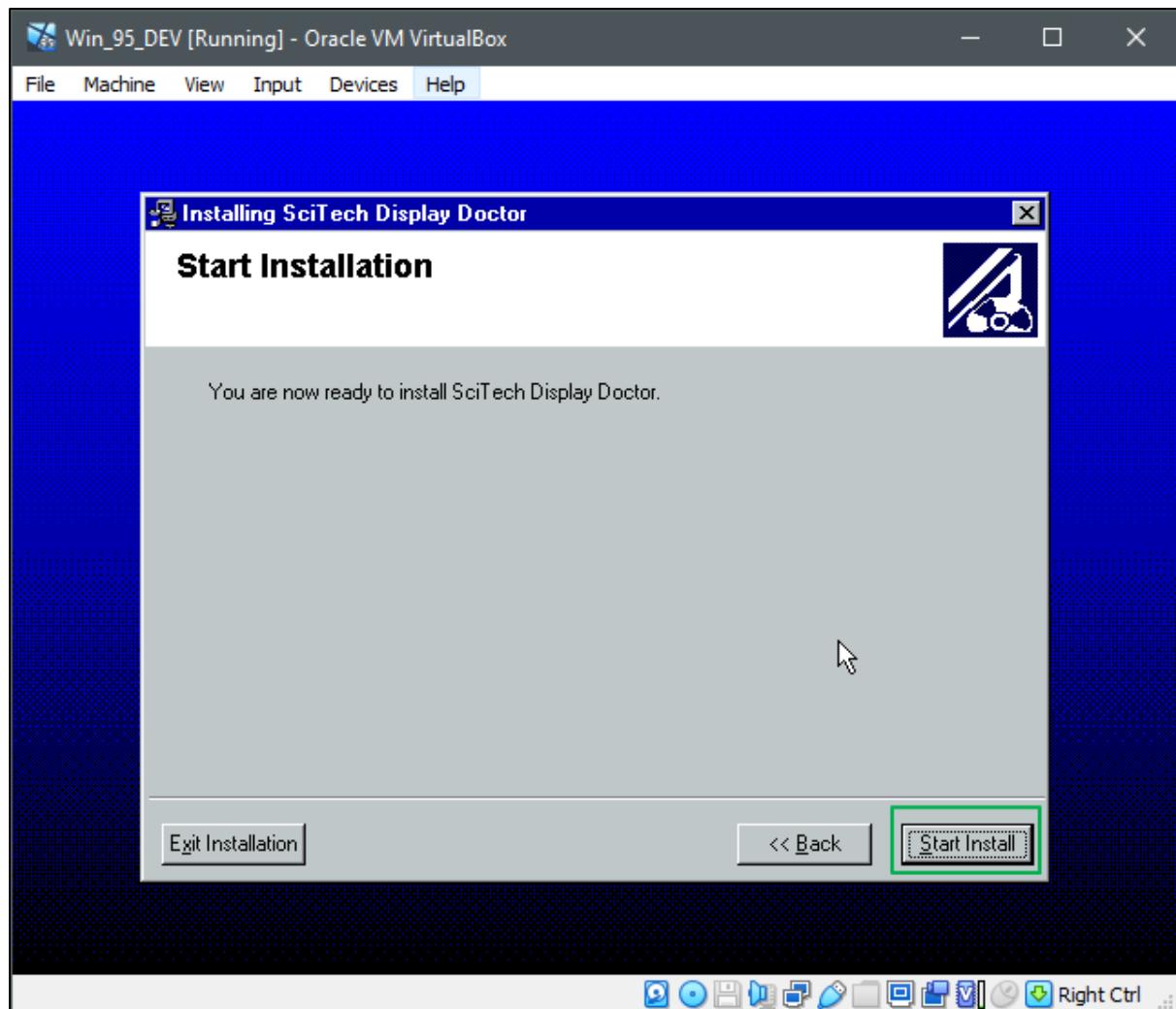


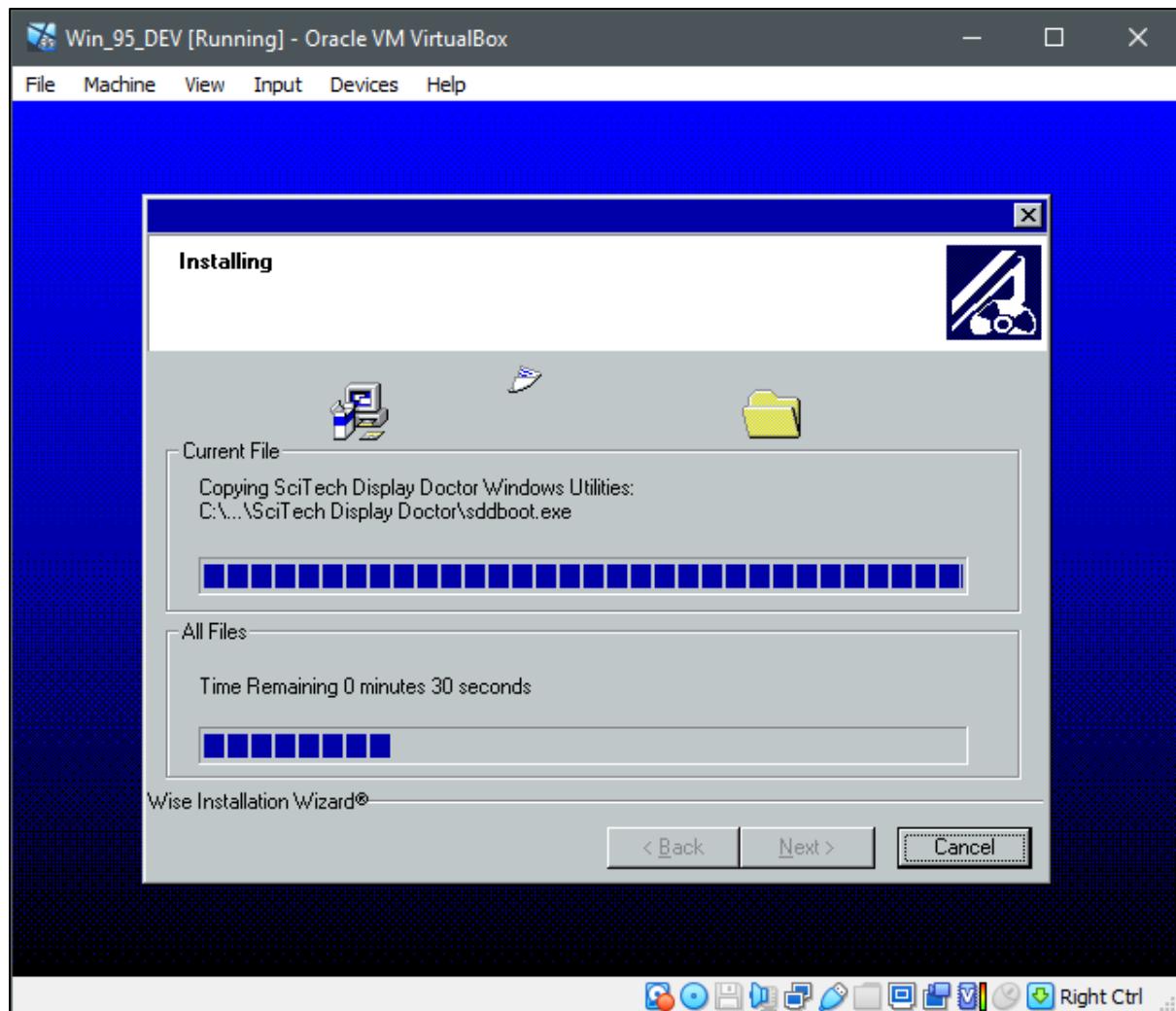
Be sure to have the serial key available.

Choose the Express Installation.

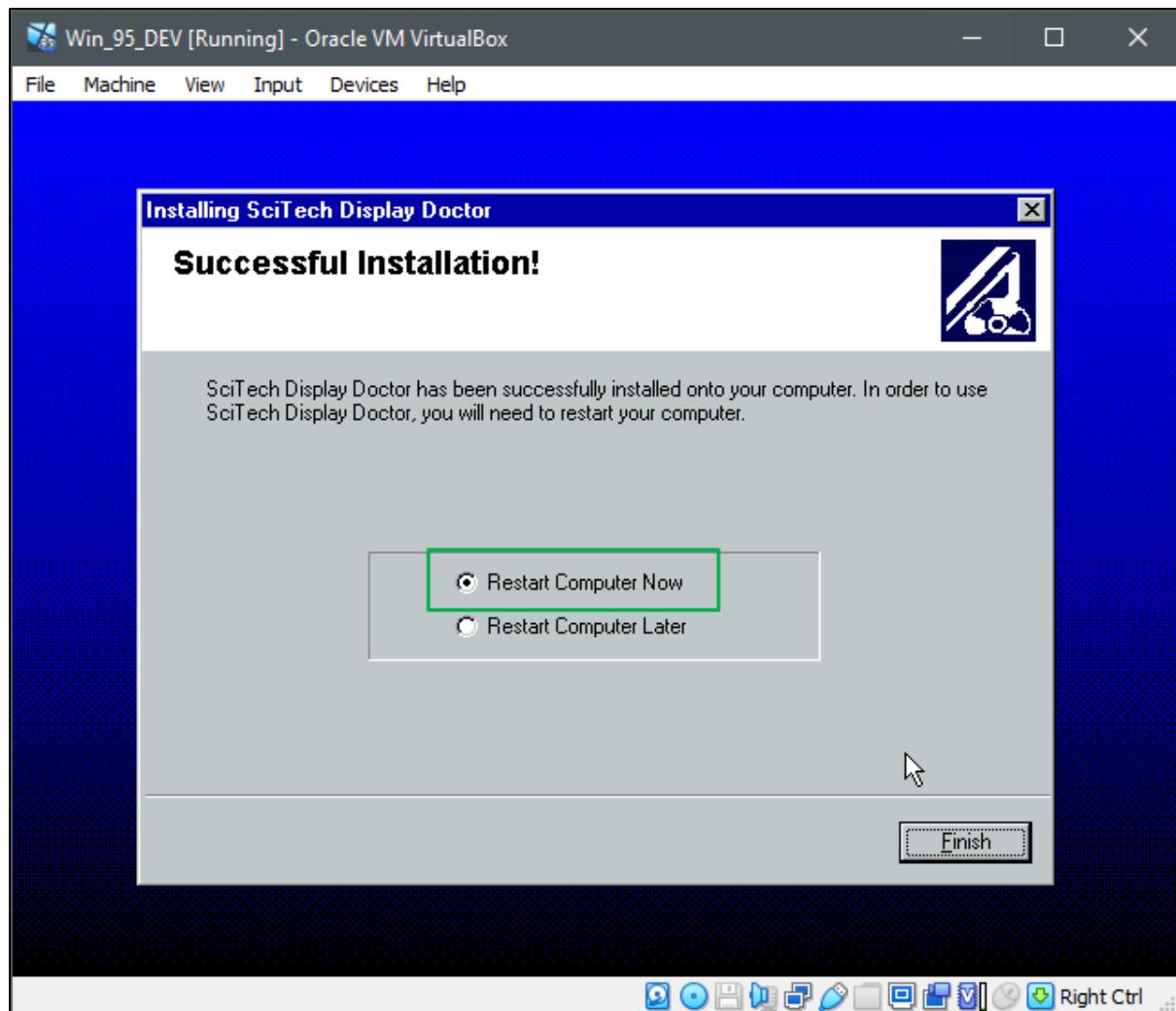


Start the Install.



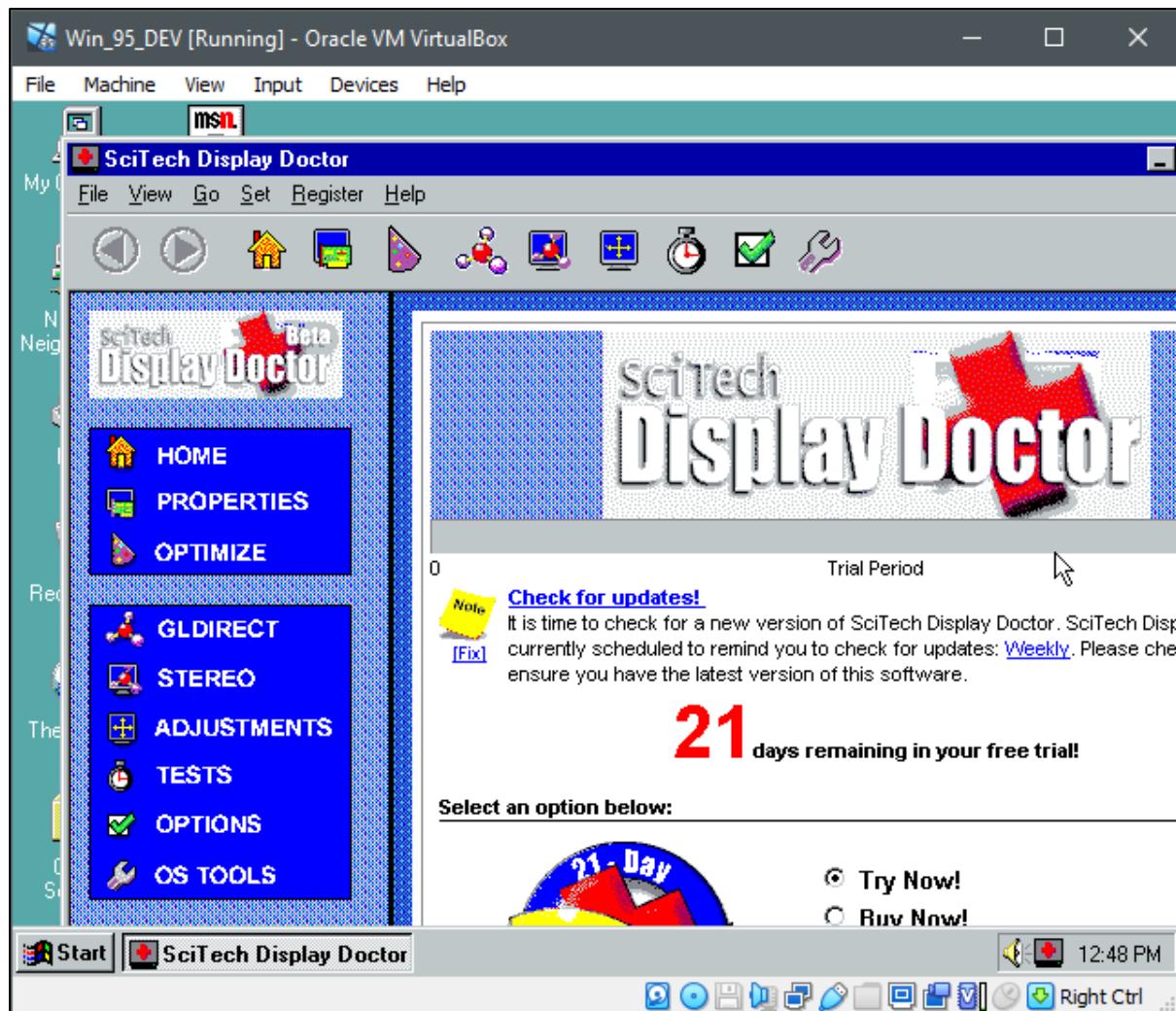


Restart Windows 95 when prompted.



You may receive an incorrect colour depth warning at start up, just click OK and ignore it.

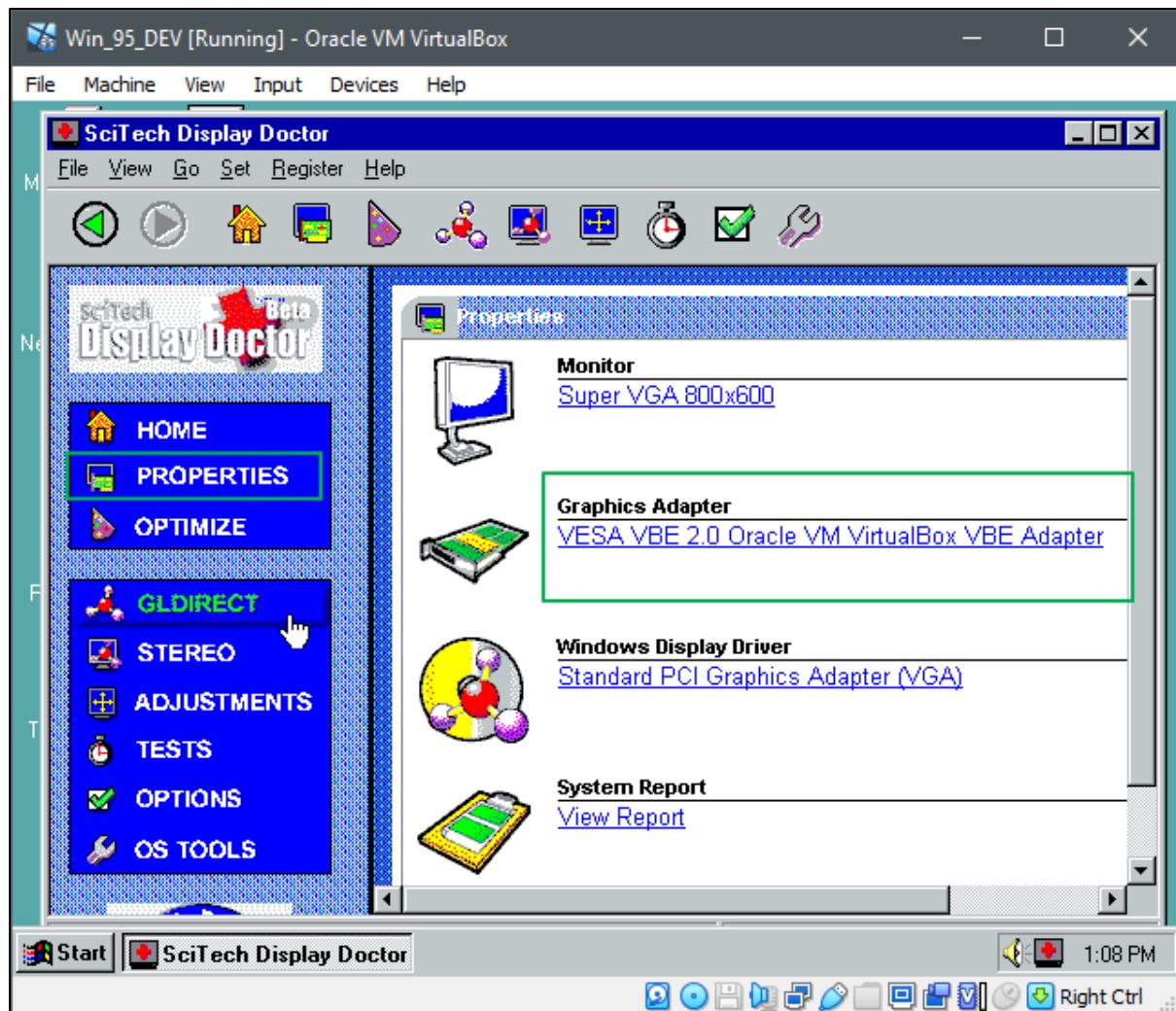
The SciTech Screen will appear a little bit large on first start, but the settings are still accessible.



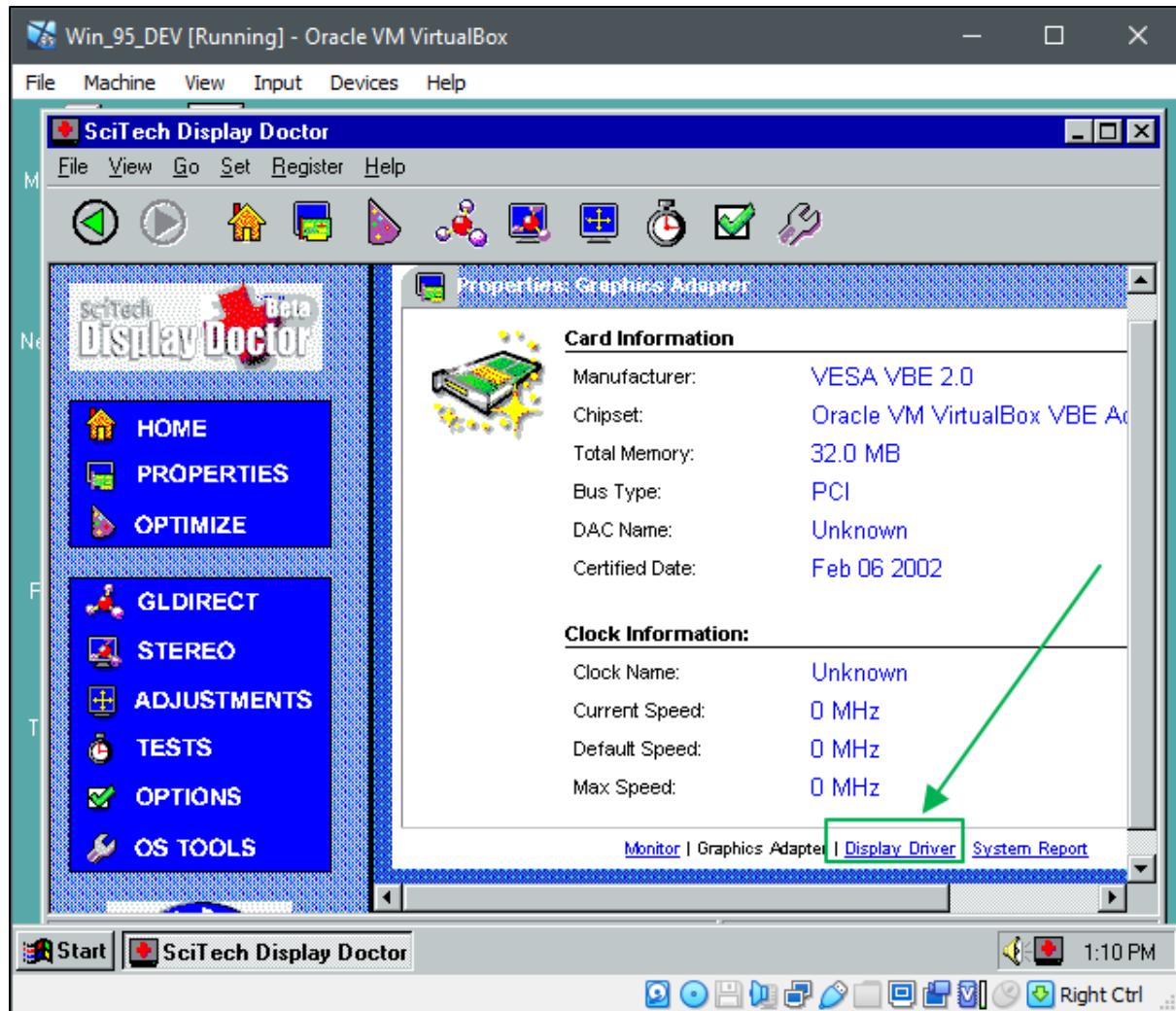
You can resize the window a little to access some of the options.

Go to “OPTIONS” in the left menu and then select “Properties” then click on “Graphics Adaptor” link.

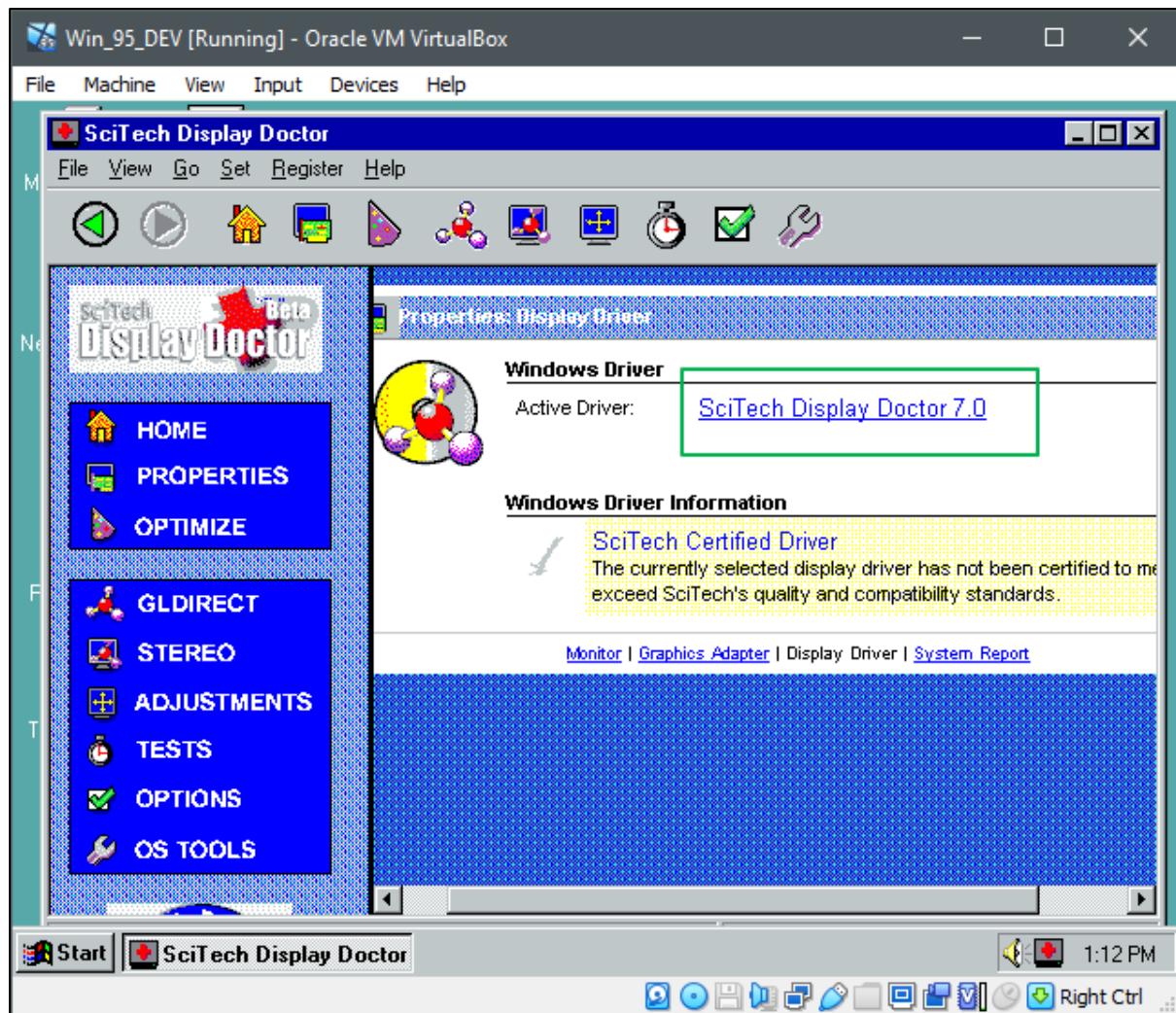
Your display may show no Graphics adaptor present.



At the lower section of the next screen select Display Driver.



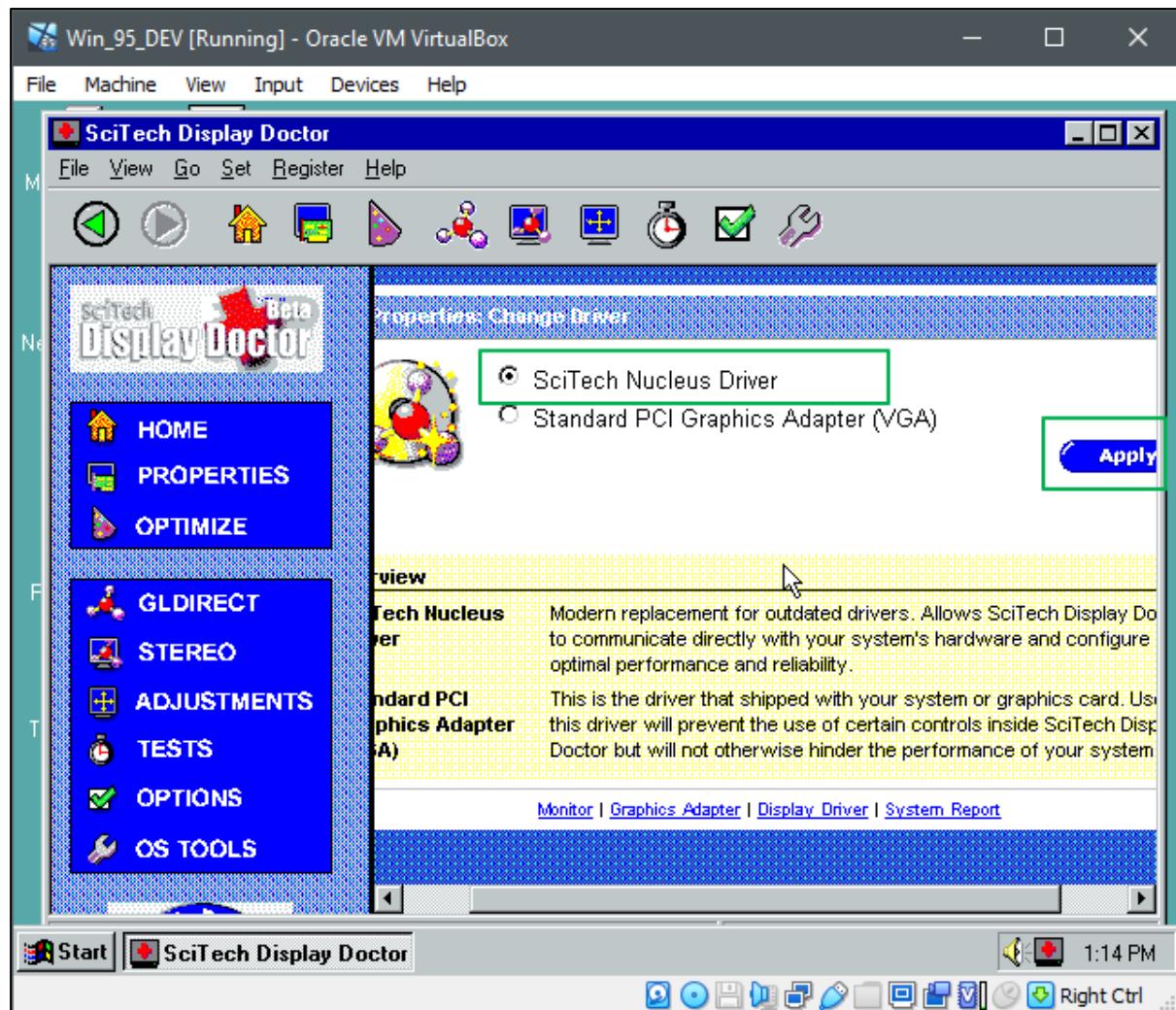
In the next screen will be a generic or unknown driver. Click on the text link.

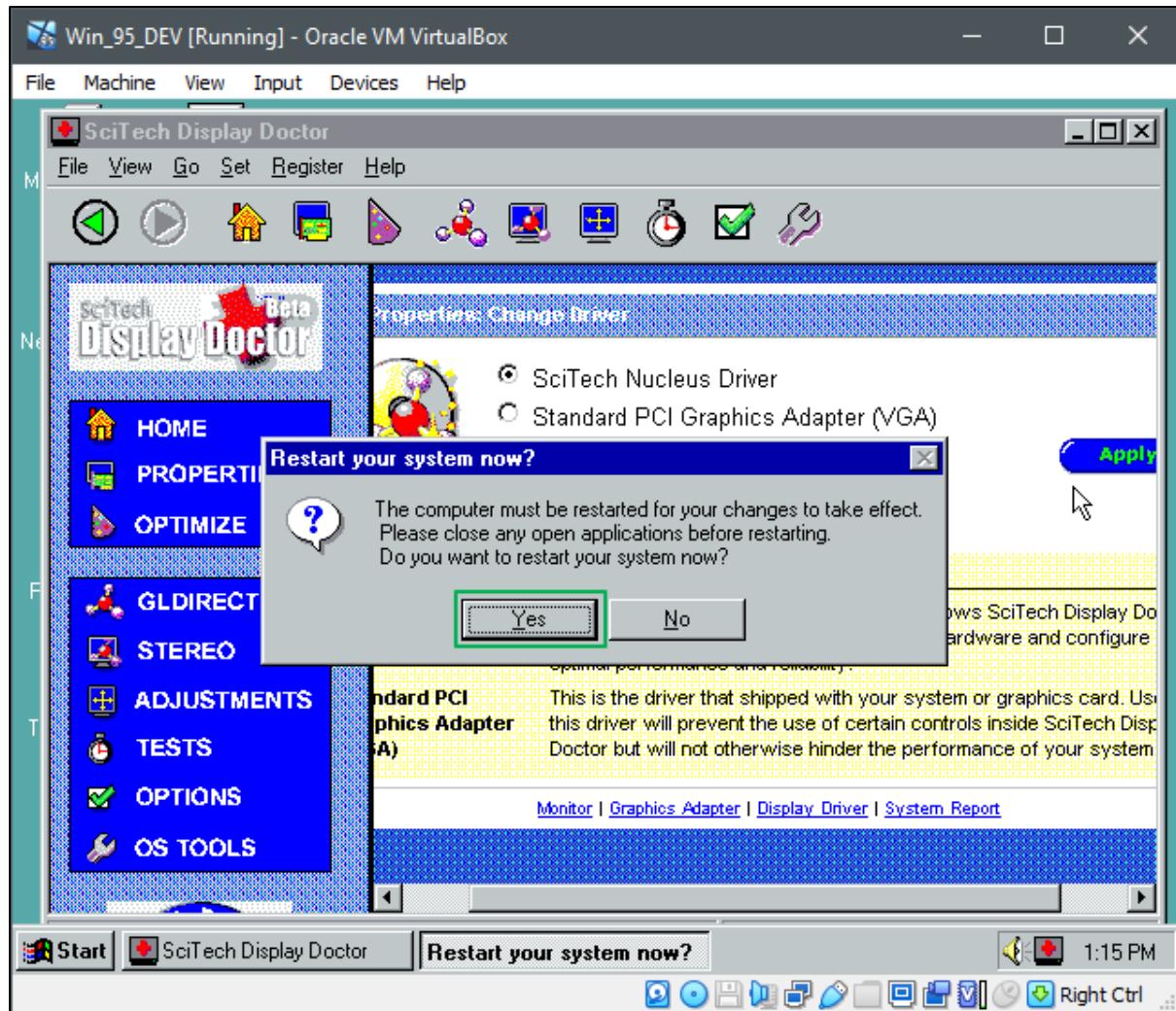


The next screen is the screen you will be looking for. It can also be found through the Options menus.

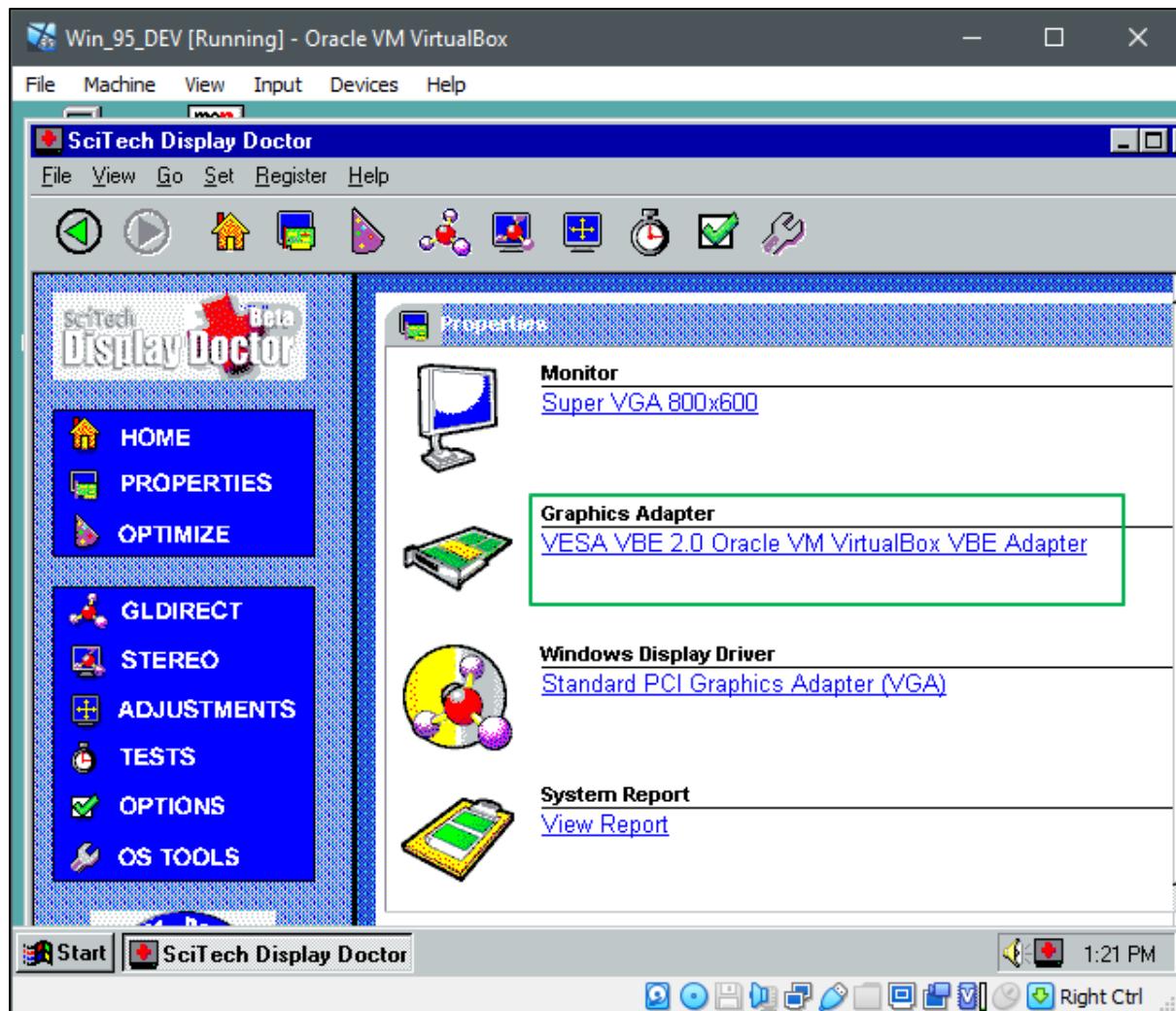
Change the setting to the SciTech Nucleus Driver. And then click apply.

You will be asked to restart your computer. Carry out the restart.



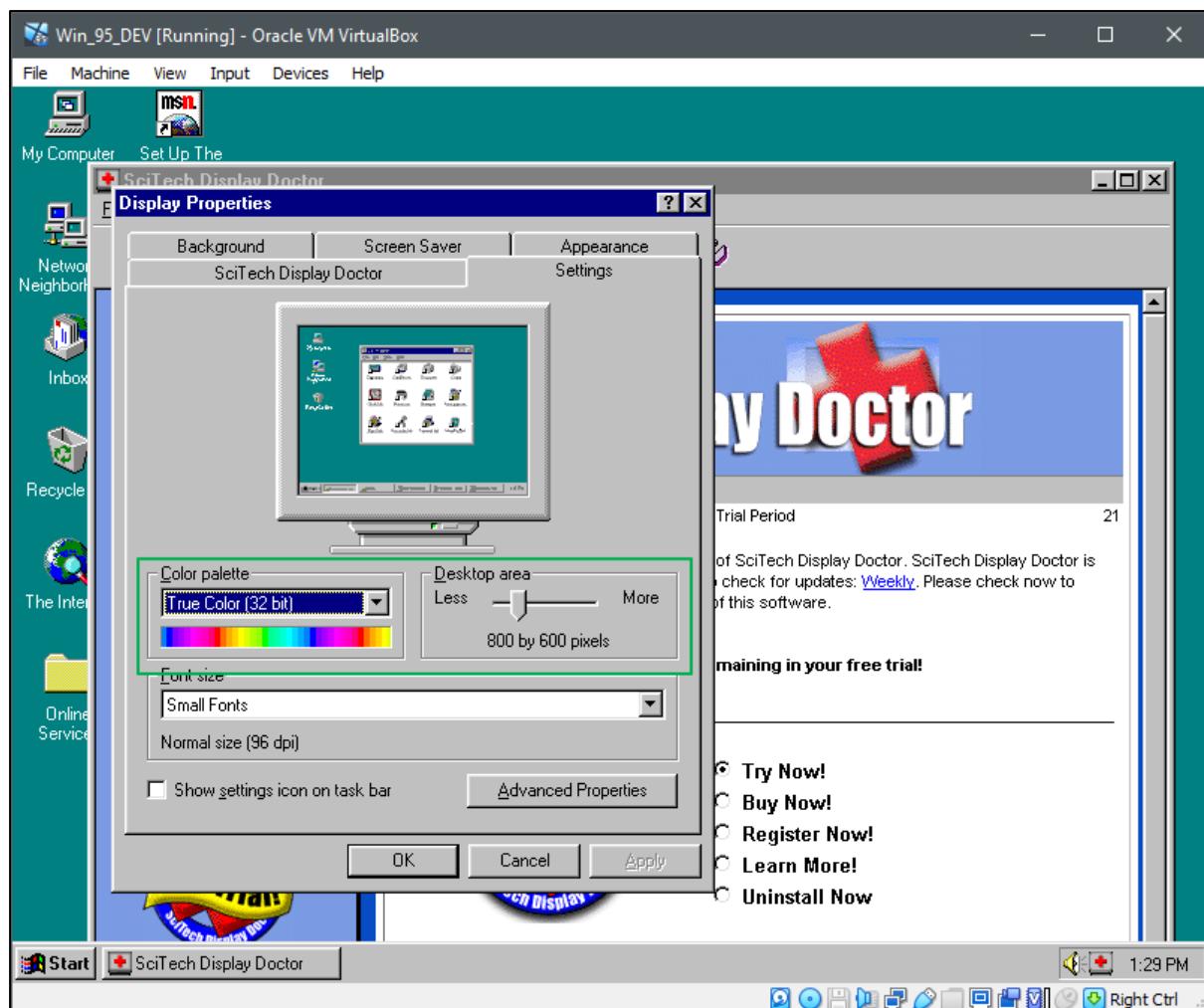


After the restart you will have the VESA VBE 2.0 Oracle VM VirtualBox VBE Adaptor available.



If you right click on the desktop and select properties you should now have more colours and display sizes available.

I have done this using the default VBoxVGA adaptor, but the above works well for the VBosSVGA adaptor as well.



SciTech Display Doctor is a very advanced display driver management system with many optional tweaks and features. The above should get you through the essential idea of selecting the best adaptor and driver to get a more viewable sized desktop. From here you can experiment and fine tune if you wish :)

Don't forget to register the application with the provided key.

The main window always opens when logging into Windows. This can be corrected with the following registry change.

### Removing Splash Screen:

To remove splash screen, go to the taskbar and click run

Type "regedit.exe"

Go to directory:[HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run]

"SystemTray"="SysTray.Exe"

Delete: "Check for SDD updates"="C:\Program Files\SciTech Display Doctor\sddboot.exe

### Windows 95 Upgrades

This section can be a little bit difficult as there is no single correct way to upgrade. Windows updates at the time had many patches fixes and update that can be difficult to establish in the update time line. Often Update/fixes can conflict with other fixes. Typically most windows updates were provides as part of the Internet Explorer updates, so IE is the standard update path. So the basic order will include upgrading Internet explorer to version 5.5 including Active desktop and the Windows Explorer file manager.

Many of the other updates, patches and fixes for windows 95B will only be required when absolutely needed by a particular hardware need or software application need.

Take some time to look at the many guides available on the internet before proceeding. I would recommend MDGX as the best source.

<https://www.mdgx.com/web.htm#OSR2>

<https://web.archive.org/web/20170828234629/https://www.mdgx.com/web.htm#OSR2>

The following patch does not work with the VirtualBox Guest OS, so I would caution against using it.

~~Download OSR2 SP1 1.05 Full [51.1 MB]. File name: OSR2SP1.EXE~~

Windows 95 can make use of USB but it can be difficult to get working. This is essentially a Windows 98 driver patched for Windows 95.

USBSTR95 [196 KB, free, English]. (Optional)

File name: USBSTR95.EXE

I have never encountered an issue with the CPU Speeds in VirtualBox but if you do have problems the following fix can help.

FIX95CPU [5.67 MB, free, English].

File Name: FIX95CPU.ZIP

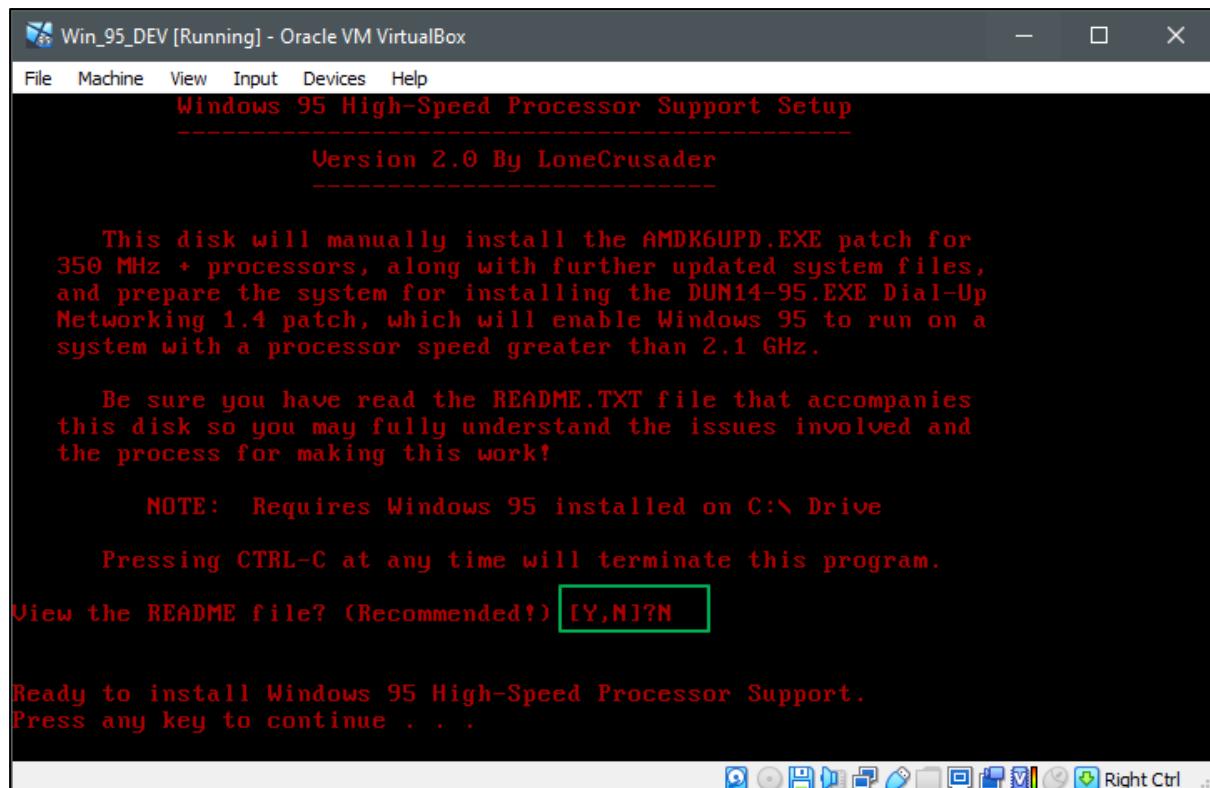
Be sure to unpack the “FIX95CPU.ZIP” archive. You will need the FIX95CPU.IMA

Copy the dun14-95.exe to your second storage drive in the VirtualBox VHD.

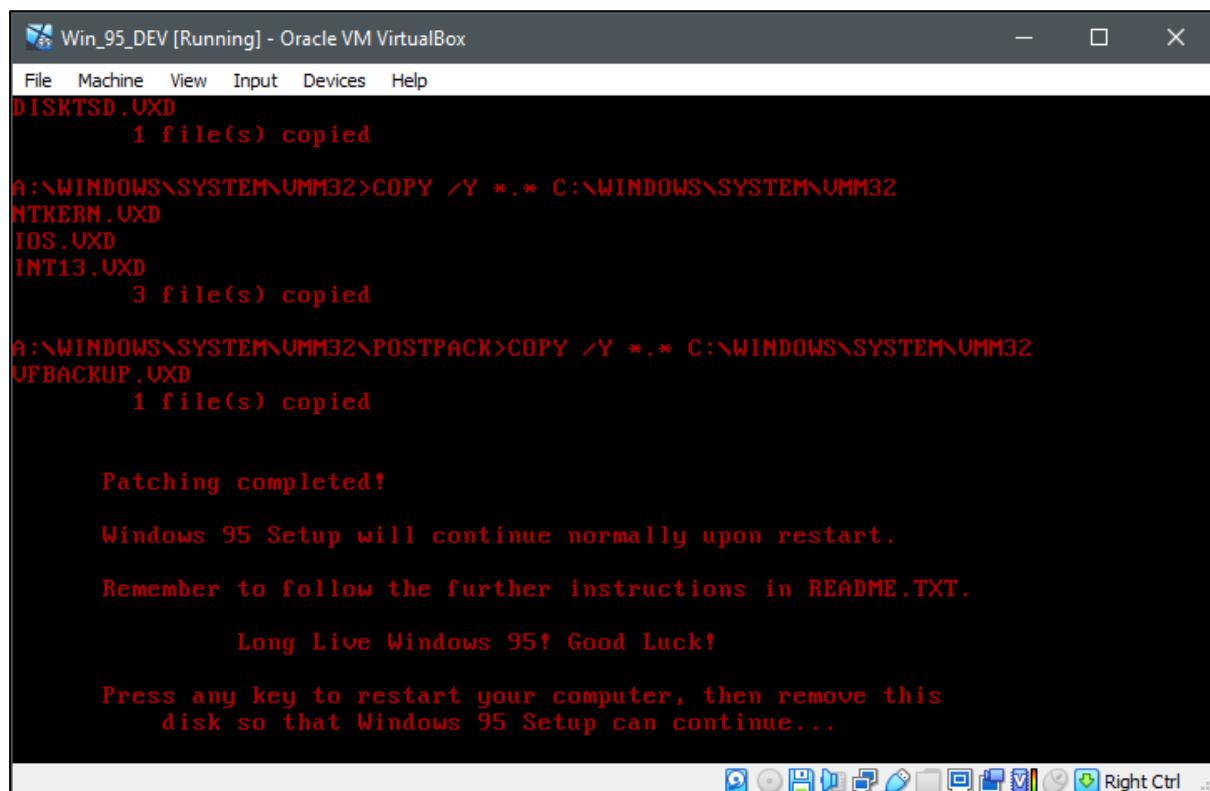
Insert/Mount the “FIX95CPU.IMA” from VirtualBox as a floppy drive.

Restart the Windows 95 virtual machine. The Fix95cpu patch will start automatically.

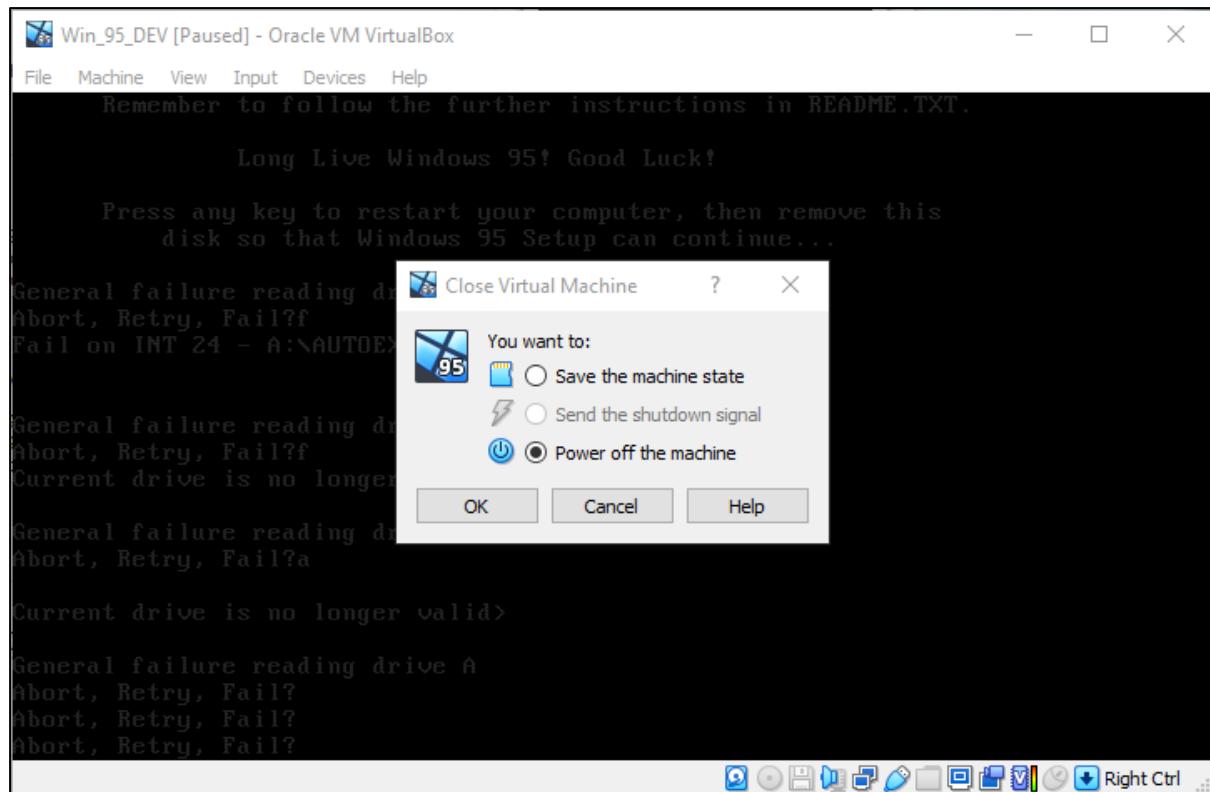
Select N (No) to continue with the patch install, and then press any key to continue.



After the patch install press any key to reboot Windows 95. It will attempt to reboot from the floppy disk. So unmount the FIX95CPU.IMA from VirtualBox and then power off the machine.

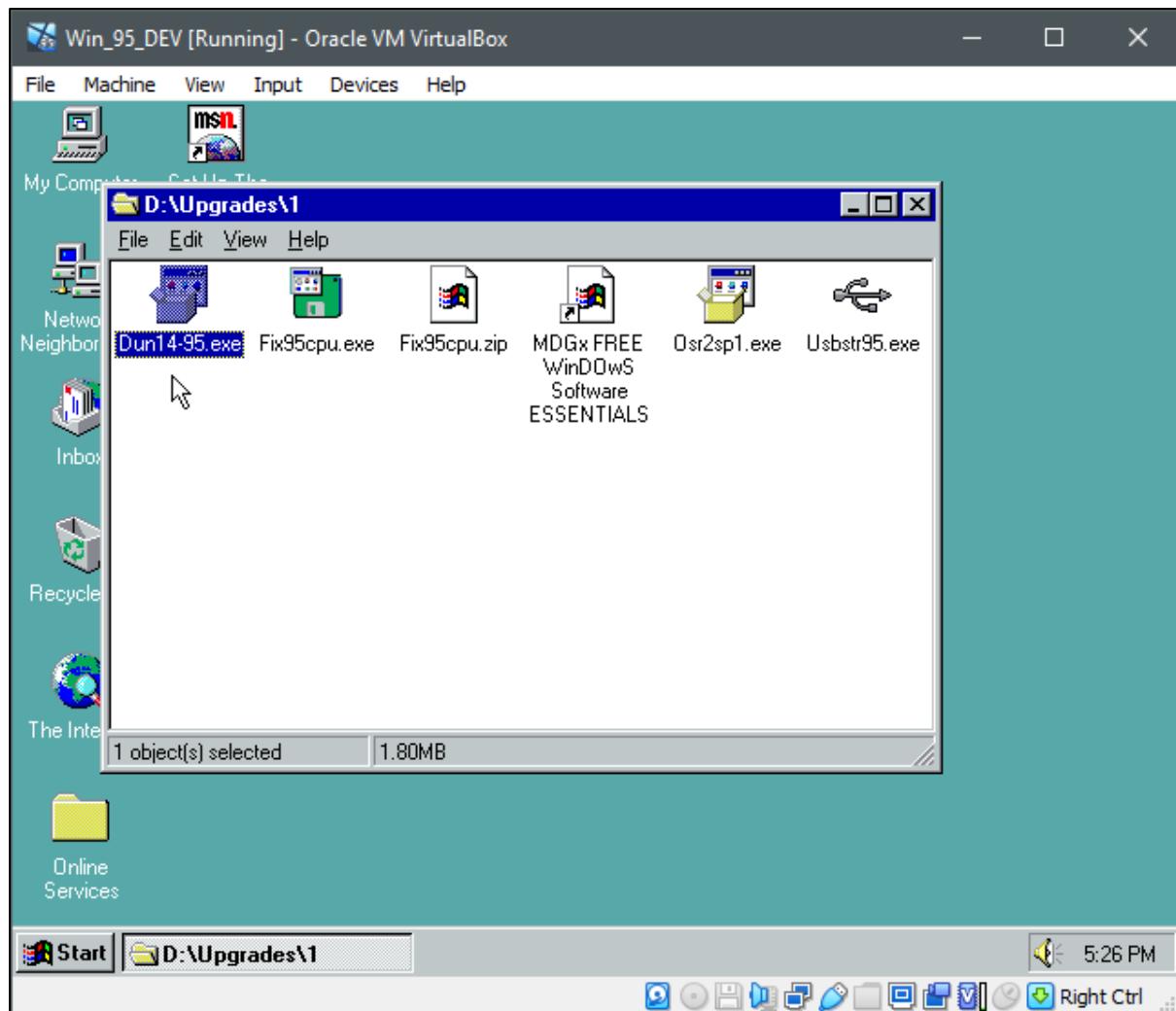


Unmount the FIX95CPU.IMA from VirtualBox and then power off the machine.

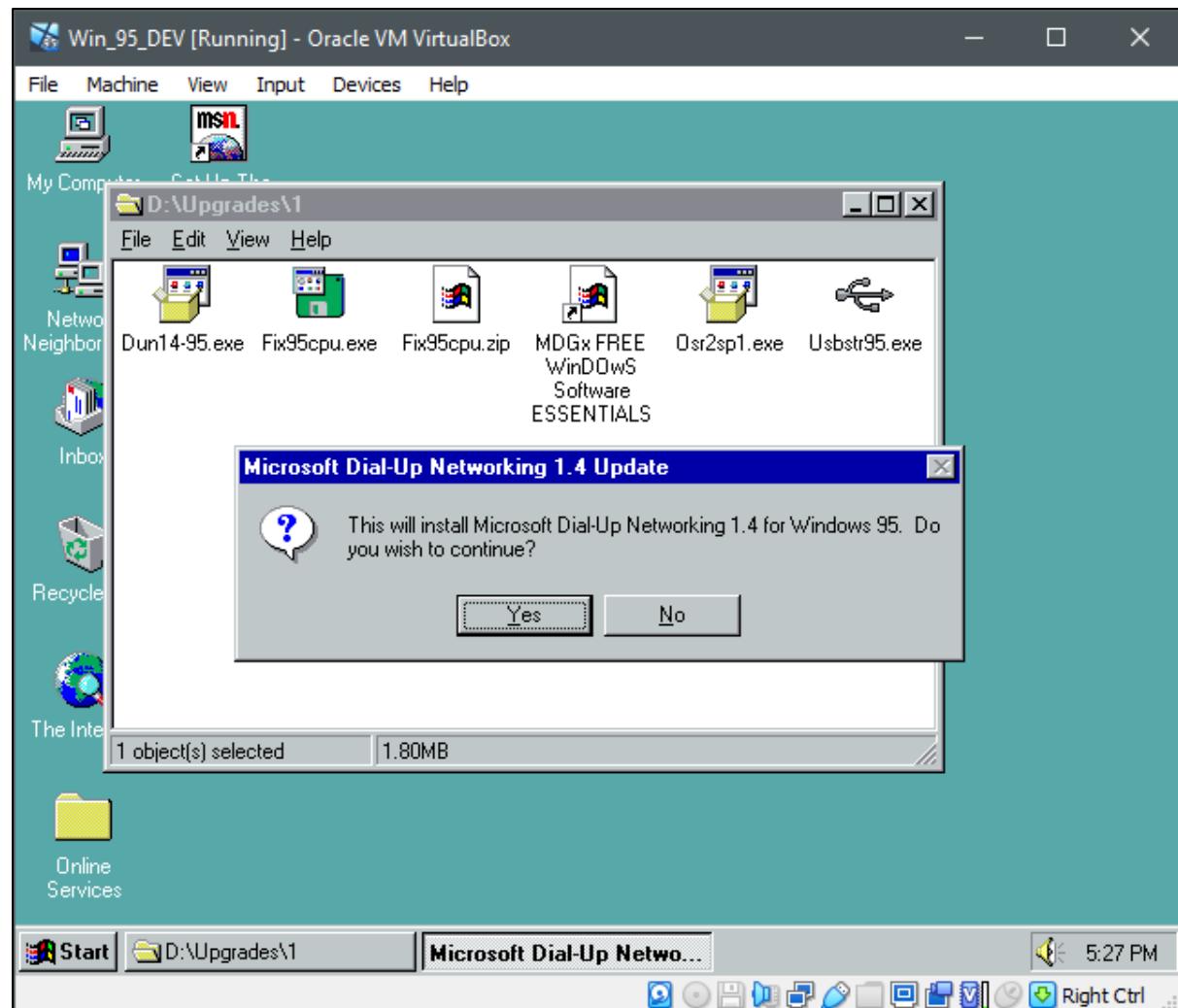


Restart the Windows 95 virtual machine.

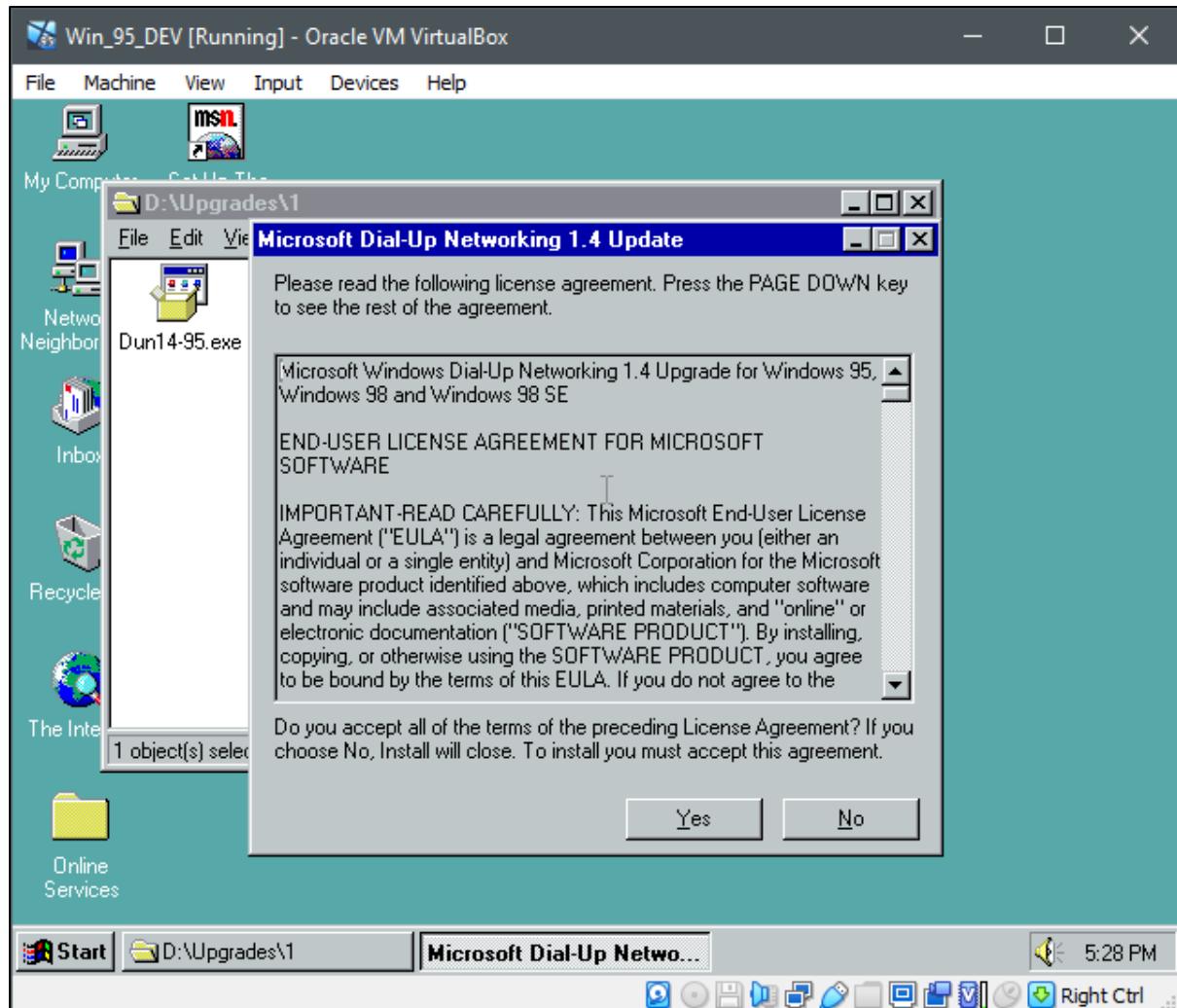
Navigate to the location where you placed the above files and install Dun14-95.exe.



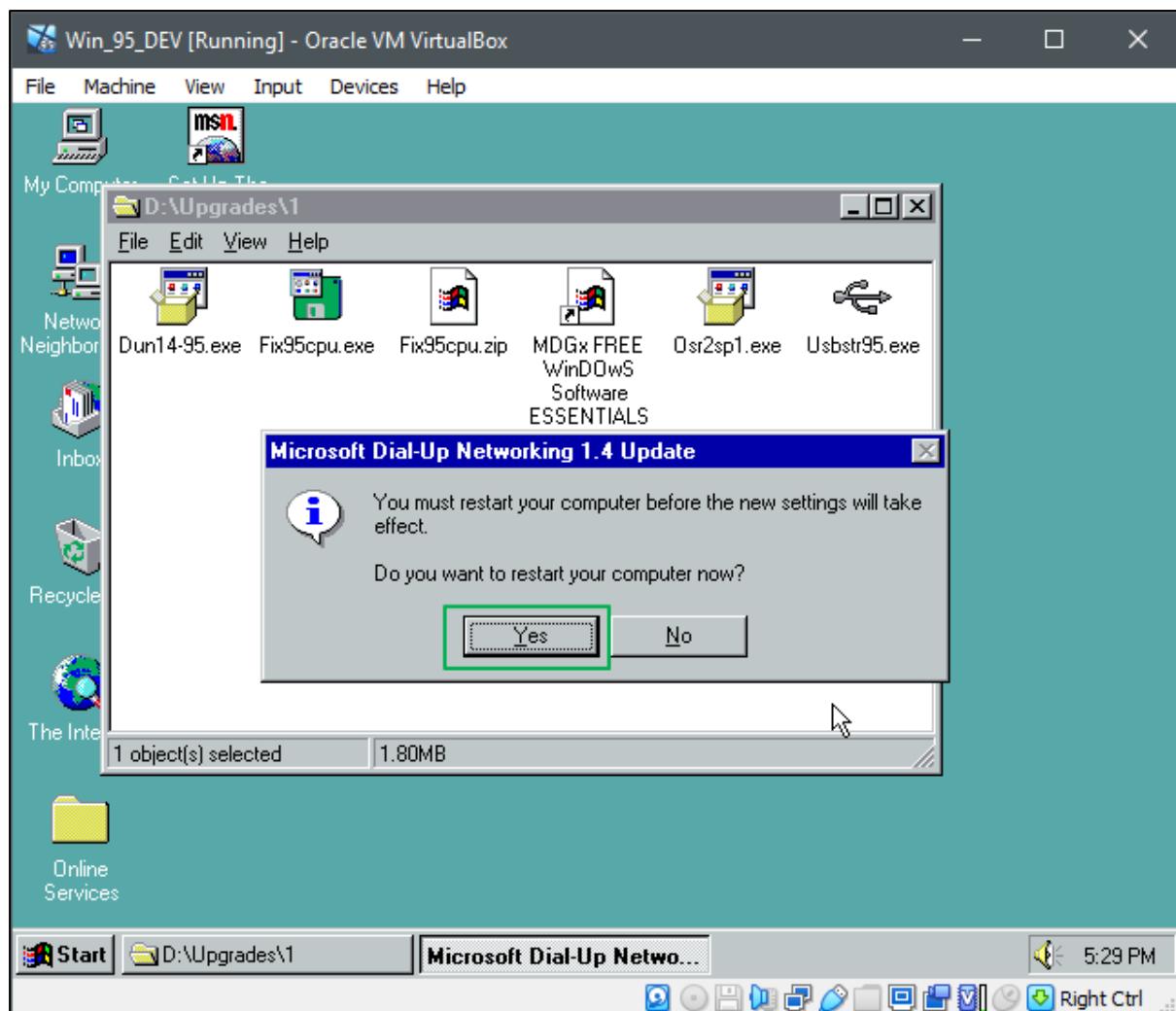
Select Yes to continue.



Accept the EULA and click Yes to continue.



Select Yes to restart the Windows 95 virtual machine.



After the reboot, additional files will be updated.

You will require another restart. Select Yes to restart the computer again. A further update will be made to the configuration files at start up.

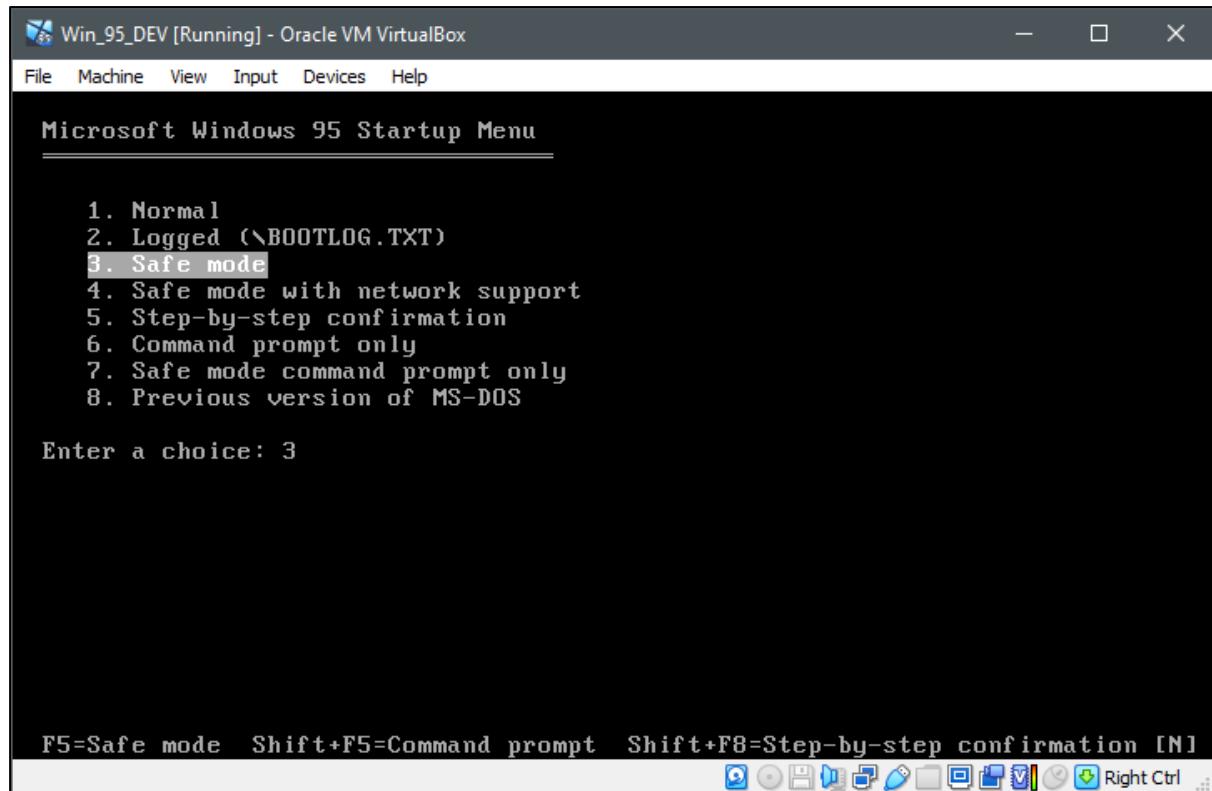
When finished this has corrected the CPU speed limitations as well as the CPU speed limitation in the Dial up networking system files. This update has completed.

You can now progressively increase the CPU speed execution cap in the VirtualBox control panel for the machine. I usually don't go above 80-90% as windows 95 doesn't have a CPU idle state and will run that VirtualBox real hardware CPU at 100%.

### Multiple Floppy drive issue

After the FIX95CPU you may notice multiple floppy drives showing in the system. This just requires a clean-up of the drive enumerators.

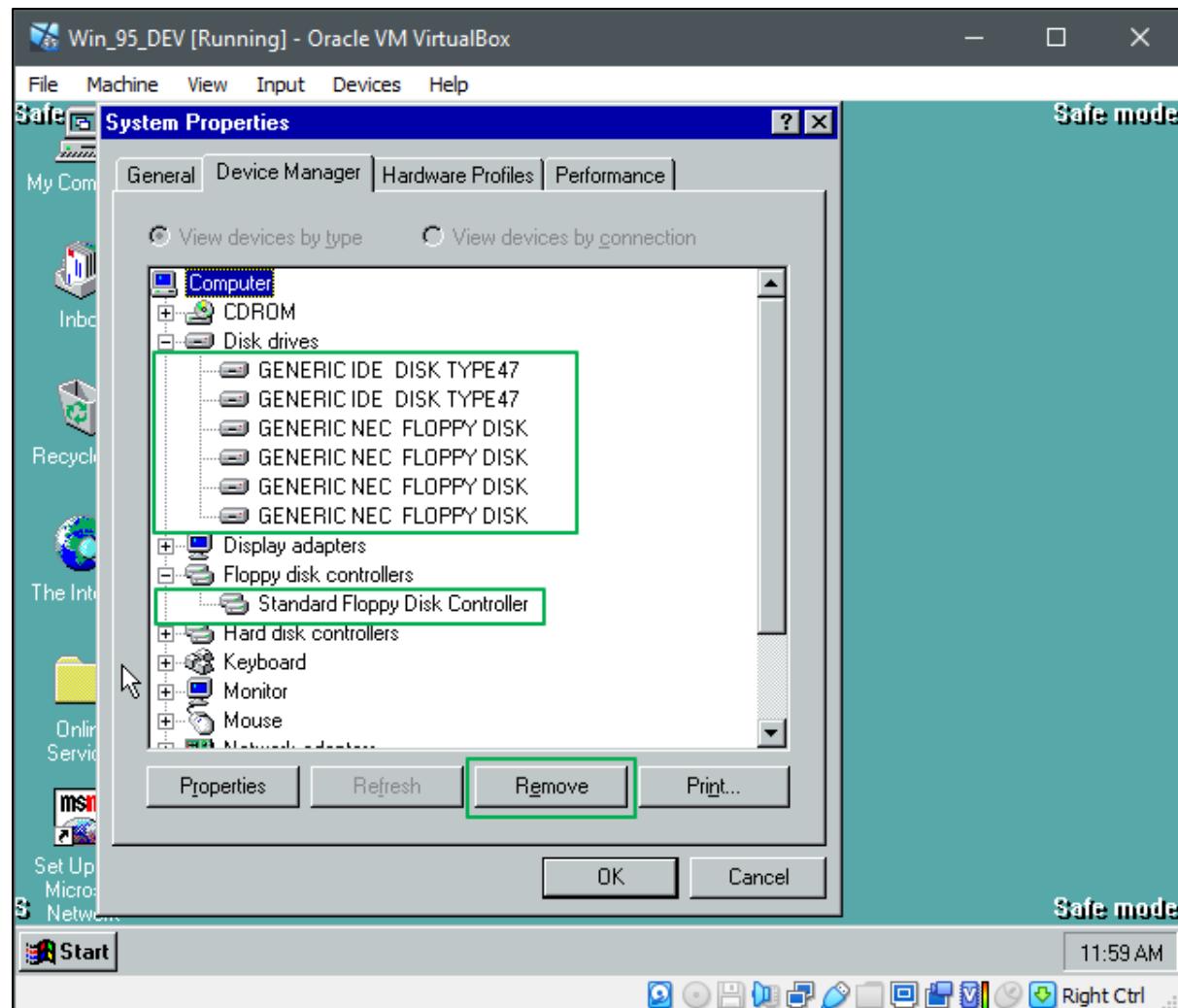
Reboot the computer in "Safe Mode" by pressing F8 at the BOIS screen and selecting "3. Safe Mode".



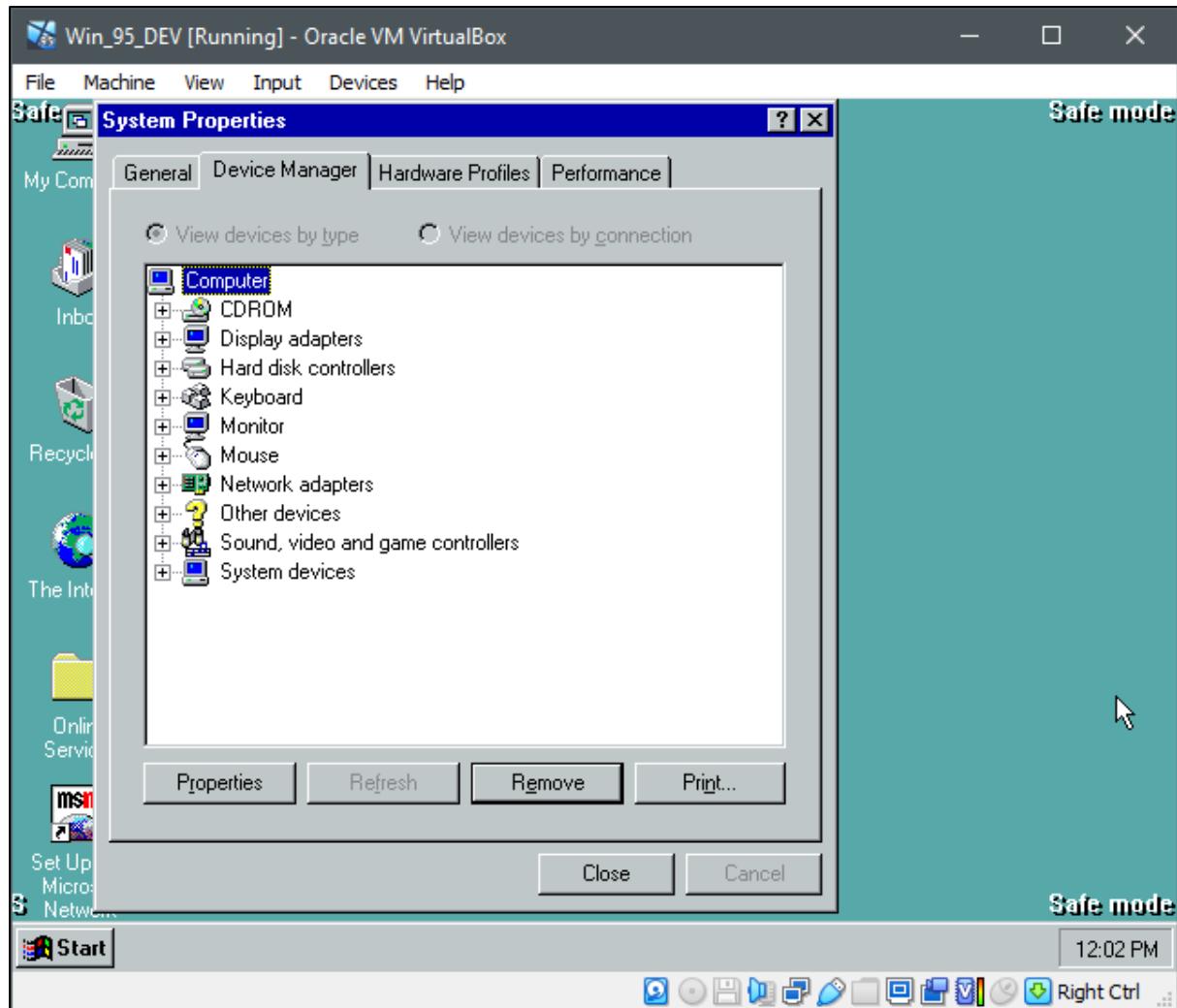
Right click on my computer and select Properties. Change to the Device manager tab.

Expand [+]“Disk drives” and [+]“Floppy disk controllers”.

Remove all of the disk drives and the controllers shown in green.

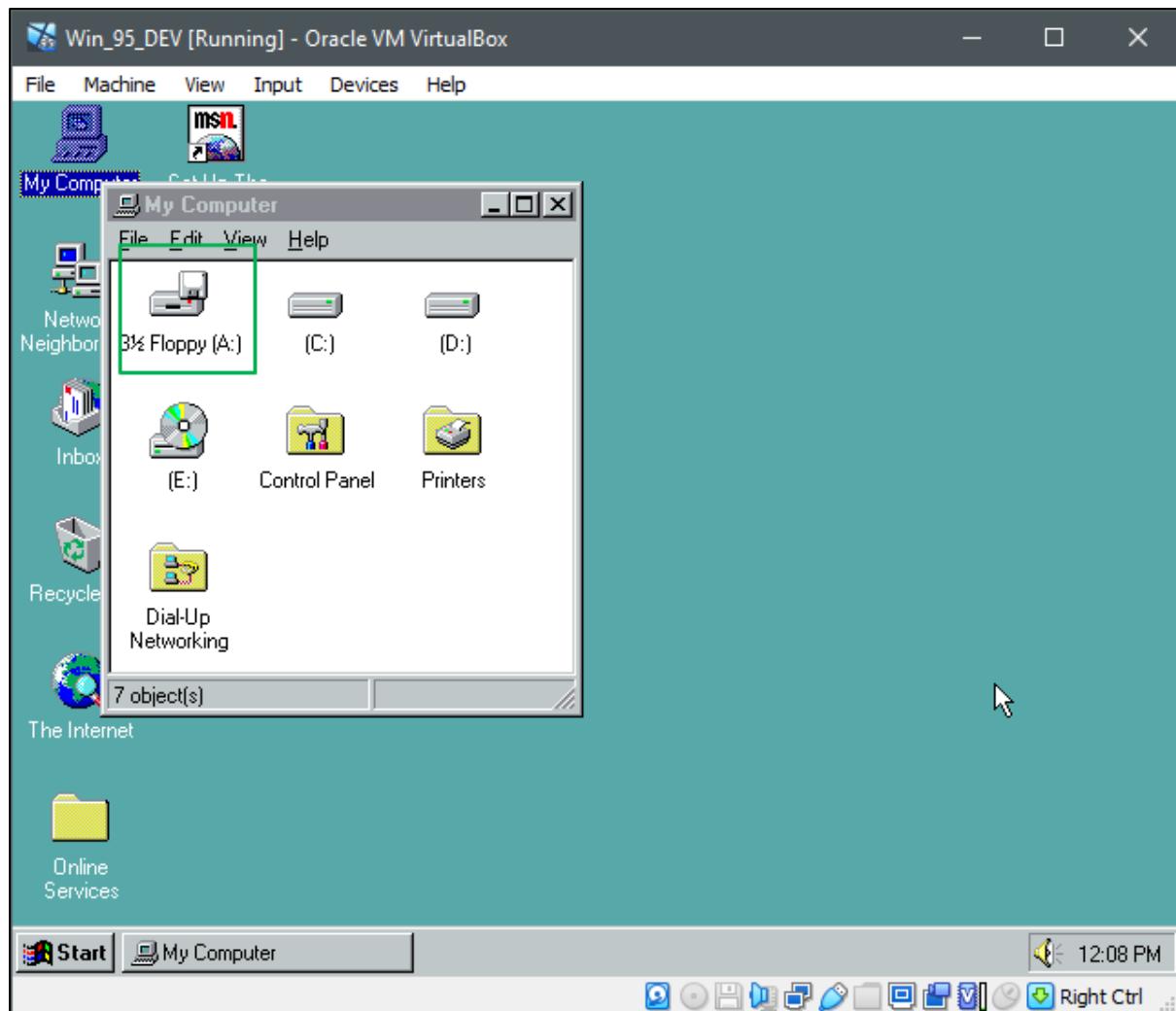


The Expand [+]“Disk drives” and [+]“Floppy disk controllers” will no longer be visible.



Restart the computer in normal mode.

You should now only see a single floppy drive.



Due to the way VirtualBox controls floppy drives, Windows 95 does not always recognise them correctly or may show multiple drive letters for the same drive. Remember VirtualBox does not officially support Windows 95. Removing the device driver allows the driver to be reinstalled as well as rebuilding the enumeration of current devices.

### Update Internet Explorer

Internet explorer from version 4.0 to 5.5 is an integrated component of the Windows 95 system. Updating Internet explorer is the primary Windows 95 desktop and shell update path and cannot be avoided. As such each version applies system updates that are required to install later IE version and it is necessary to install each in order. I have skipped a few releases in the following list that I feel can be skipped, but if you encounter problems collect all IE version from 4.0 to 5.5 and install each one in order. Be sure to find the version for Win32 (95) and don't use the Win-NT versions as they are different. Winworldpc, oldversion and some other sites will have the full IE series available for download.

<https://www.my-internet-explorer.com/ie5/>

<https://archive.org/details/internet-explorer-5-5.5>

In this process I am going to assume you are confident in installing Window 95 applications. I will only highlight steps that are of importance.

It is worthwhile making a temporary backup of the VirtualBox VHD file between each install for safety rollback recovery.

Unpack and copy all of the following updates to the W95 drive accept the ISOs. The ISOs will be mounted from VirtualBox as a CD-ROM.

IE4.0 (copy to HDD)

IE 4.01 SP1 1998 (ISO) Windows 95C (OSR2.5) IE updates begin here.

IE 4.01 SP2 (Copy to HDD)

IE 5.0 (EXE Web Installer) The English version is only a web installer, so don't use this.

"Internet Explorer 5-5.5.iso" from archive.org

IE 5 (Internet Explorer 5-5.5.iso)

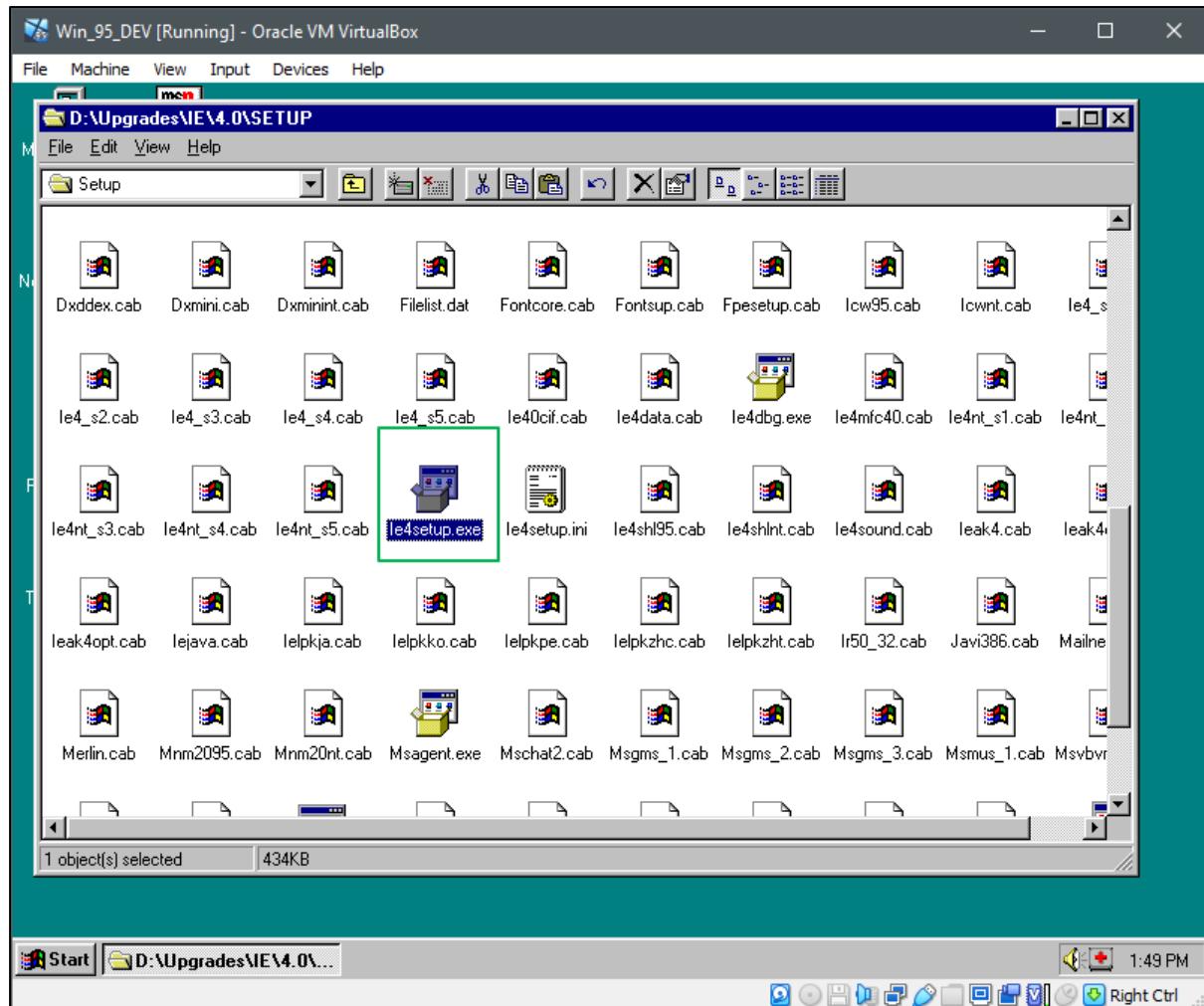
IE 5.01 SP2 (Internet Explorer 5-5.5.iso)

IE 5.5 (Internet Explorer 5-5.5.iso)

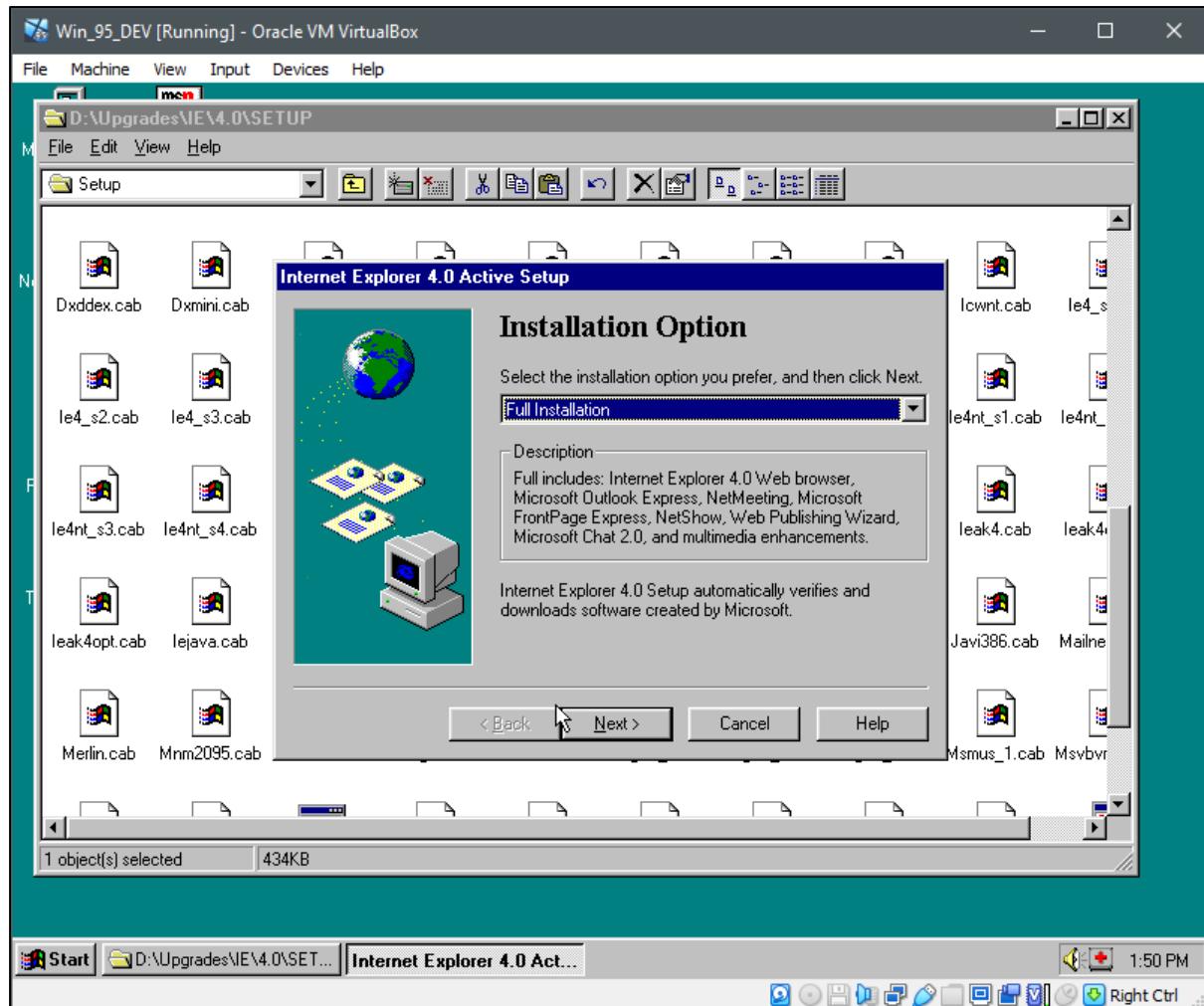
#### **IE 4.0**

Navigate to the unpacked IE directory and find the installer ie4setup.exe and run the installer.

**Warning! This install is the only opportunity to include the Active Desktop component!**

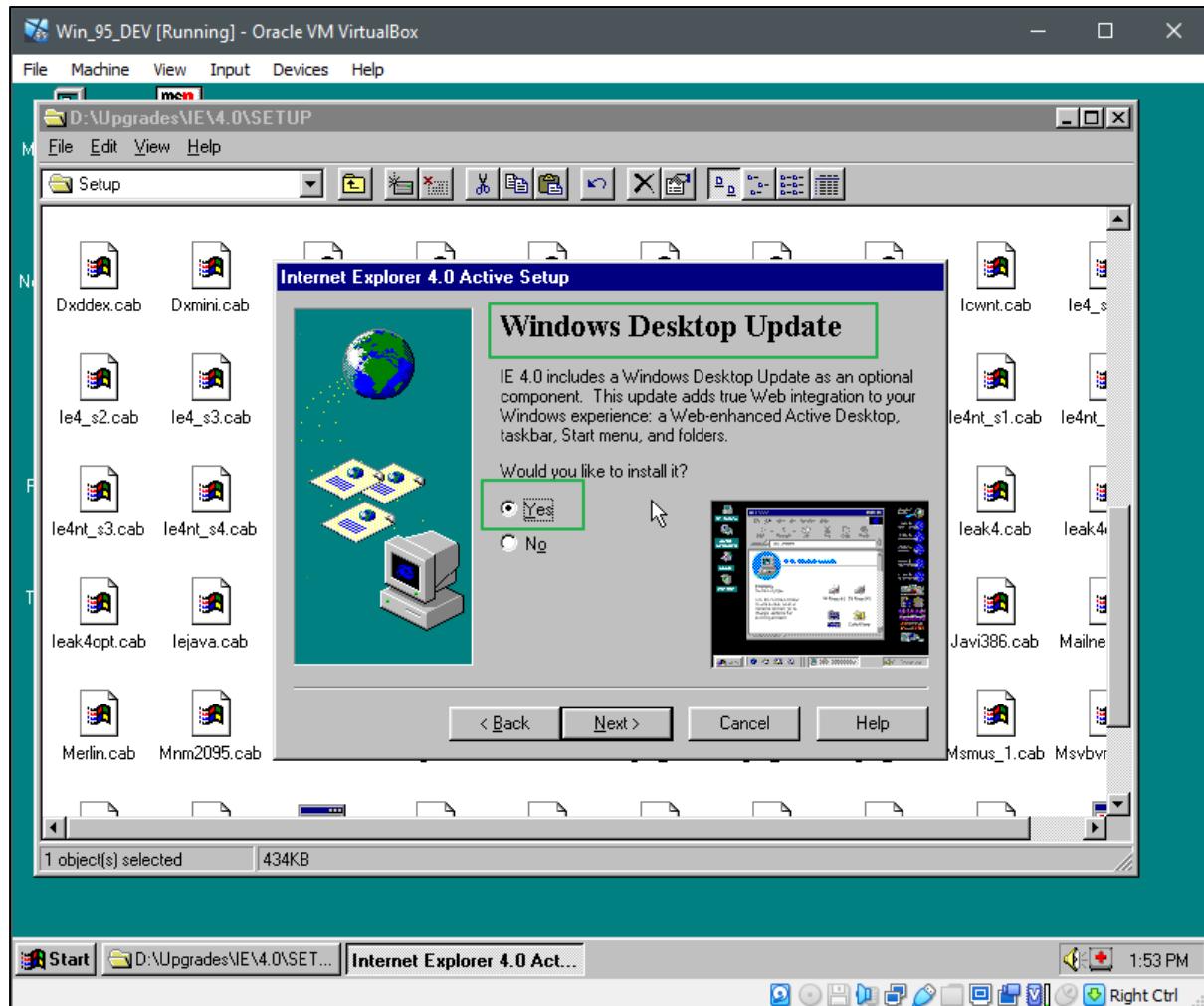


I would recommend the Full installation option as this will update a number of system files so you don't have to update them manually later.

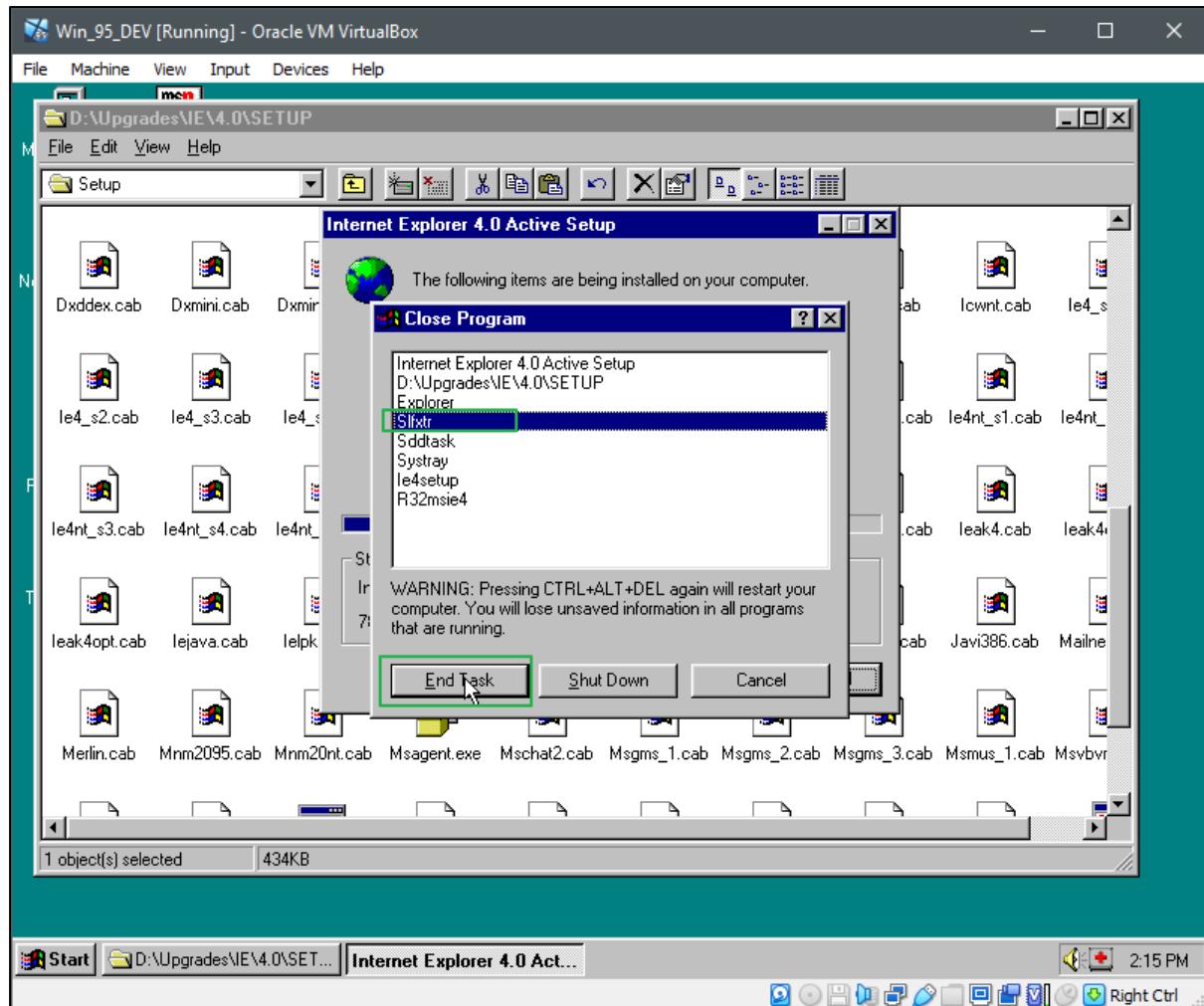


I would also recommend installing the active Desktop. This is the default for all Windows 98 through to Windows XP installs.

**You will not have a second opportunity to install this at any future point!**

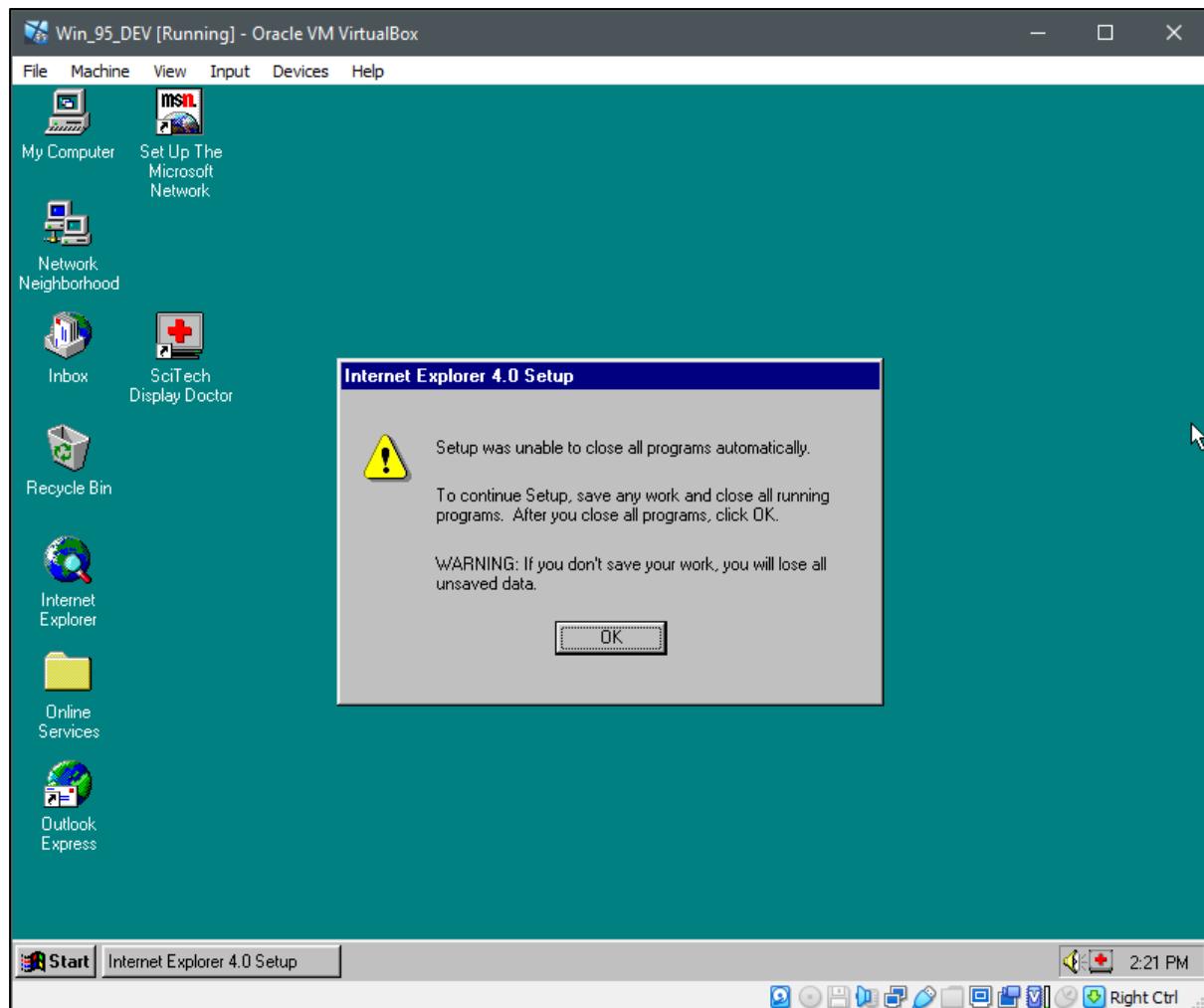


The multimedia components install may freeze at 78%. Open the Program manager by inserting “Ctrl + Alt + Del” from the VirtualBox manager. Kill the Slfxtr process. This is a compatibility issue in the underlying i386/i486 hardware virtualisation in VirtualBox (VirtualBox does not support Win9x OSs).



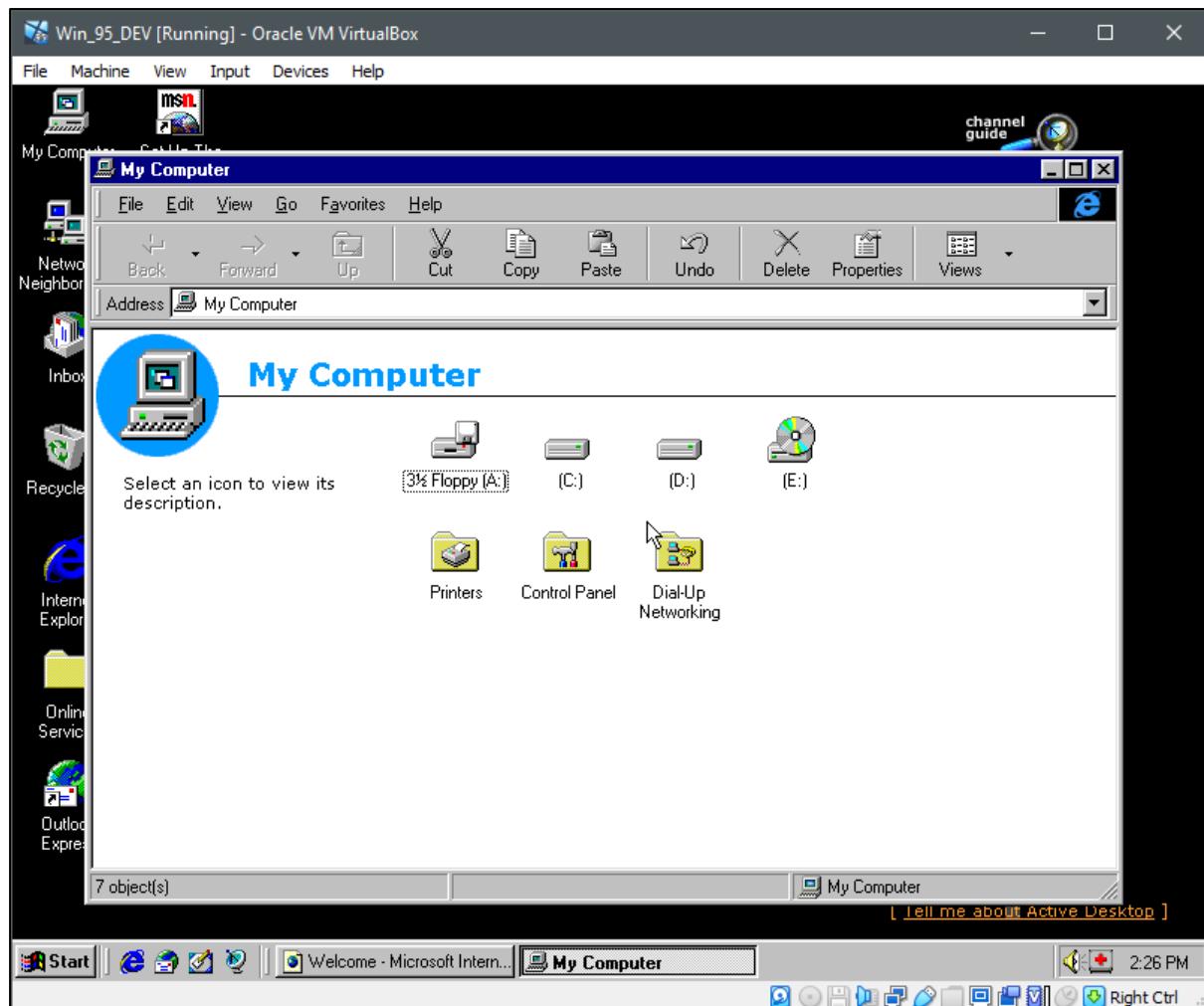
The install will complete after killing the above task. It just means that the 3<sup>rd</sup> party Real Player multimedia player doesn't get installed.

The IE setup will ask you to close all applications and restart the Desktop UI so that it can install the required system components.



The desktop UI has now been replaced with an IE driven explorer UI. The old file manager has now also been replaced with an “Explorer” file manager based on the IE engine and runtimes.

Windows desktop UI environment and Internet Explorer are now an integrated system component.



Shut down the computer and restart before continuing to the next IE version install. Note that Windows 95 regularly gets stuck at the please wait while your computer shuts down screen after this point. You will need to manually turn it off with the VirtualBox “File -> Close” option.

### IE 4.01 SP1 1998 (ISO)

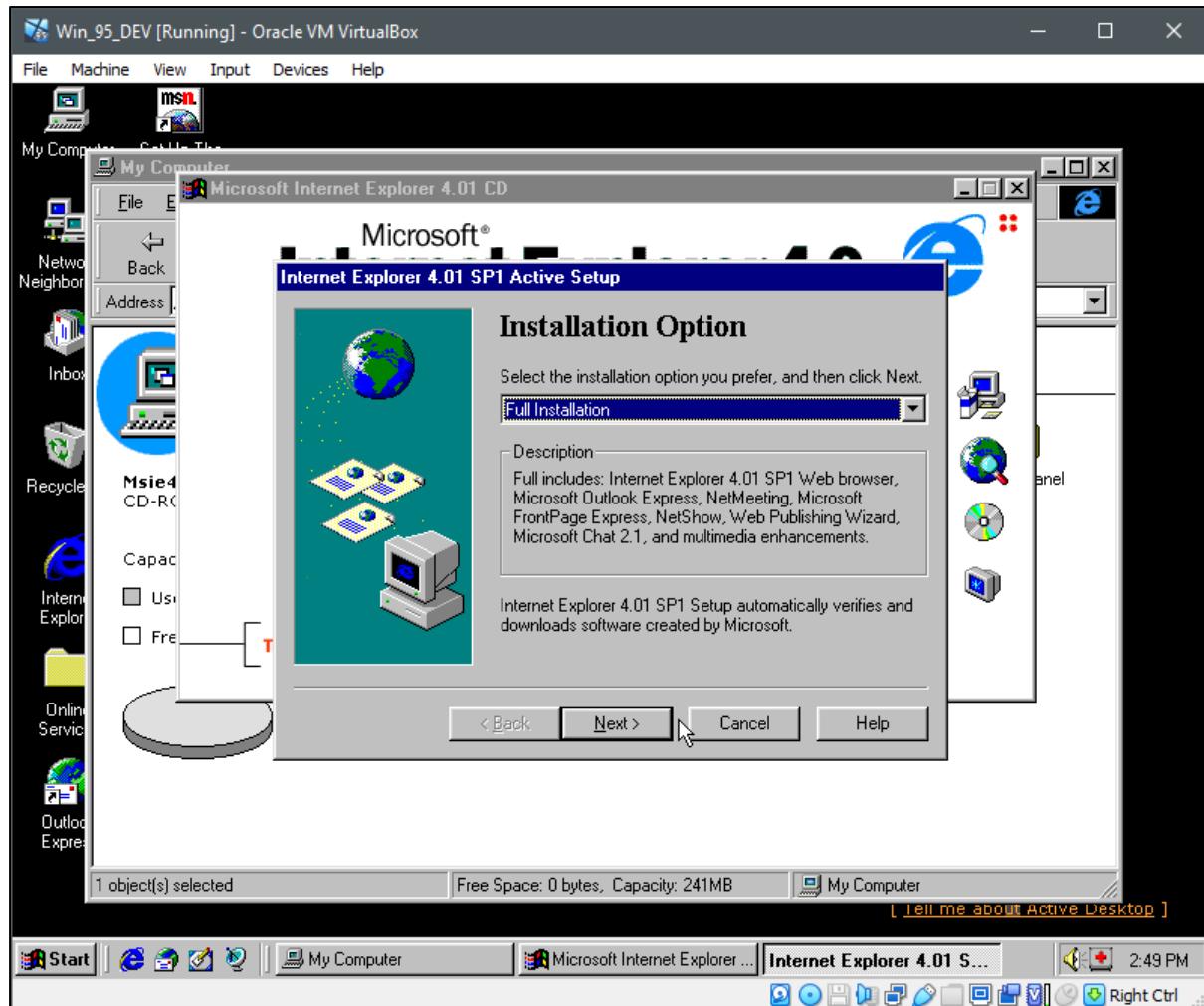
Mount the “MSIE401SP1A.ISO” from the VirtualBox menu, and navigate to the CD drive.

If you double click on the drive **Autoplay** is the default option and the installer application will begin. If the setup does not begin it can be found at i386\setup.exe

Select the “Install Internet Explorer 4.01 option”.



As with the IE 4.0 install select Full Installation so we can include additional system updates.



With this update all of the components will be installed correctly including the Multimedia section.

You will need to perform a restart so close any application and Explorer windows before continuing.



After the restart Open the CD-ROM IE installer again and select the Add-On Components.

It is really up to you if you think that you need any of the extra components in the list but I would install the "Additional Explorer Enhancements".

Macromedia Shockwave Flash may be useful only if you intend on working with old flash games or other Flash products.

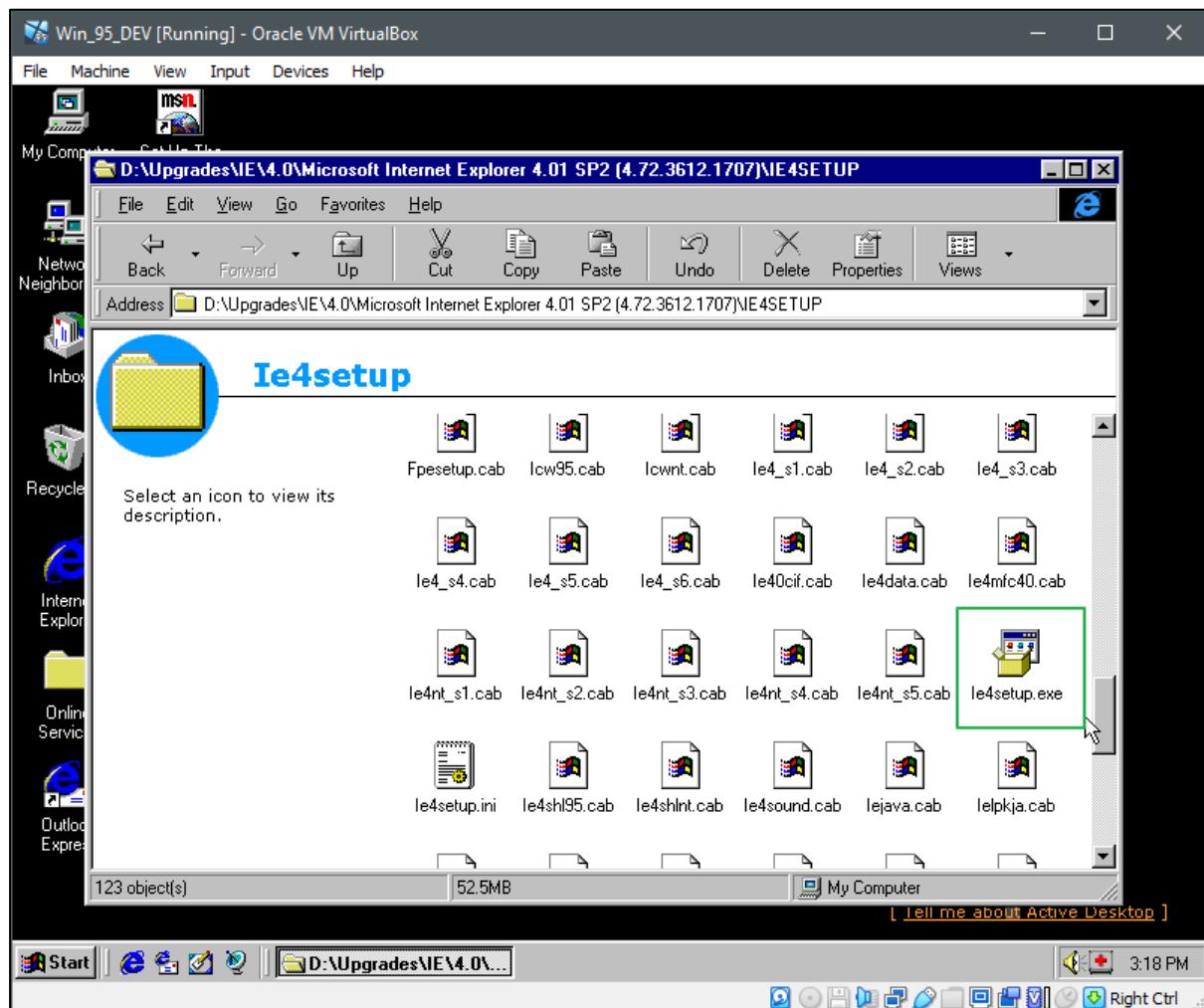
You will need to click the "Install Now" button to install the selected items from the CD.

After the install is complete you will need to restart the computer.

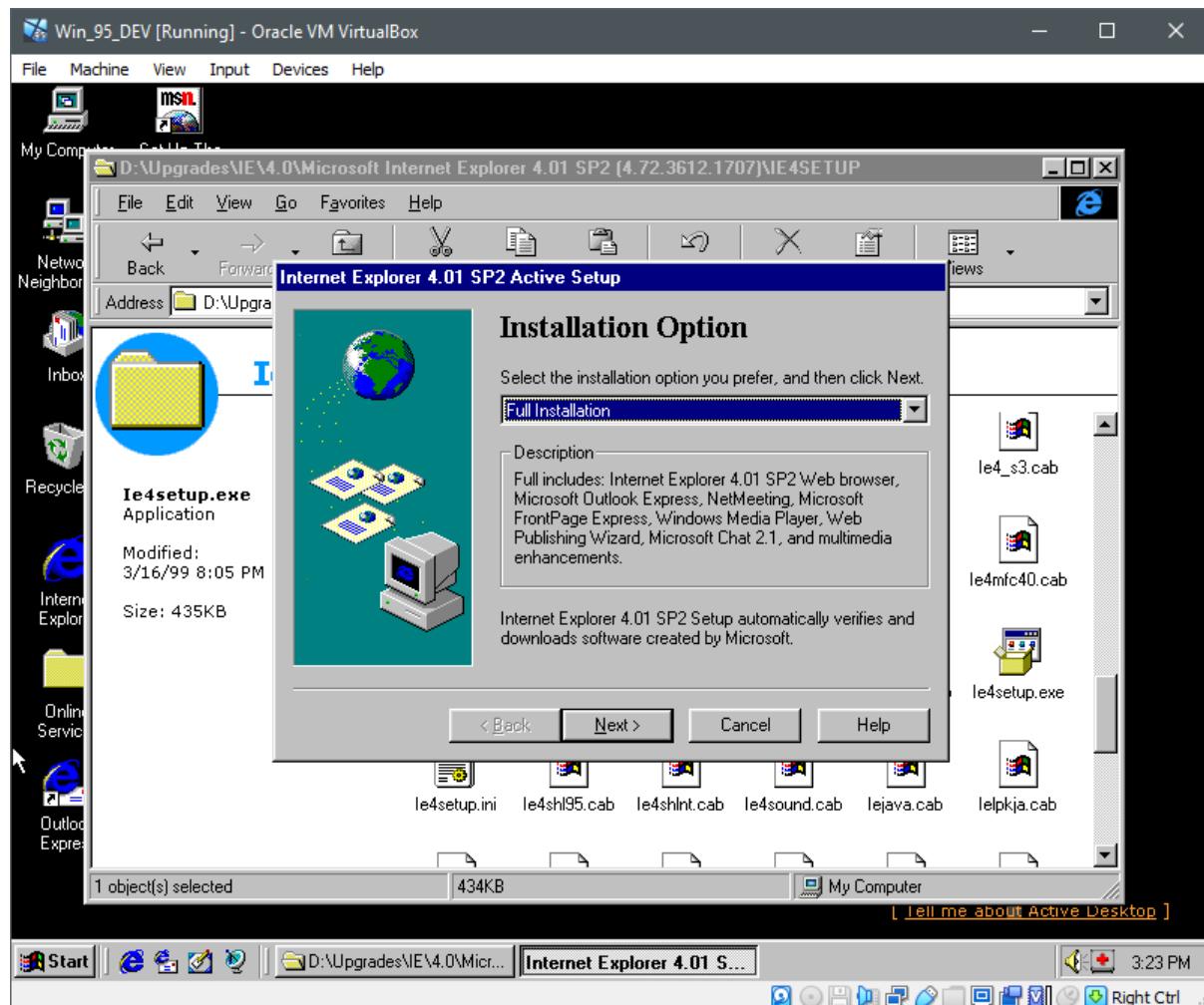
After the restart close out any open Internet explorer windows and unmount the CD/ISO from VirtualBox.

### IE 4.01 SP2

Navigate to the location where you placed the unpacked “Microsoft Internet Explorer 4.01 SP2 (4.72.3612.1707)” directory and locate IE4SETUP.EXE executable and run the installer.



Follow the install prompts as for the previous versions. Remember to again select “Full Installation”.



After the installation has completed you will be asked to restart the computer to finish the install of some components.

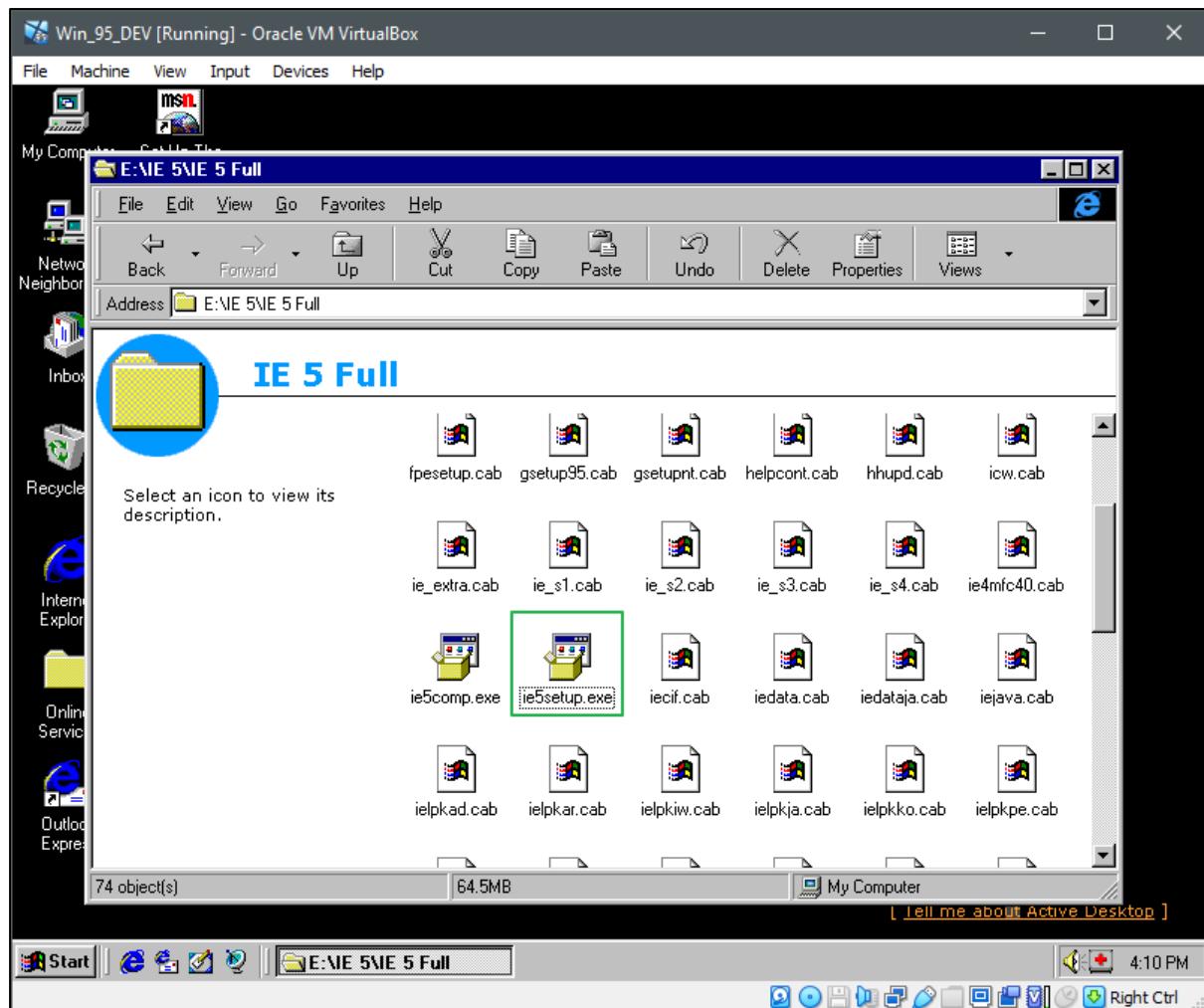
### IE 5.0 (Internet Explorer 5-5.5.iso – full Offline)

NOTE: The common download is a web installer which attempts to download the non-existent files from Microsoft. The full IE 5.0 offline install can be difficult to find.

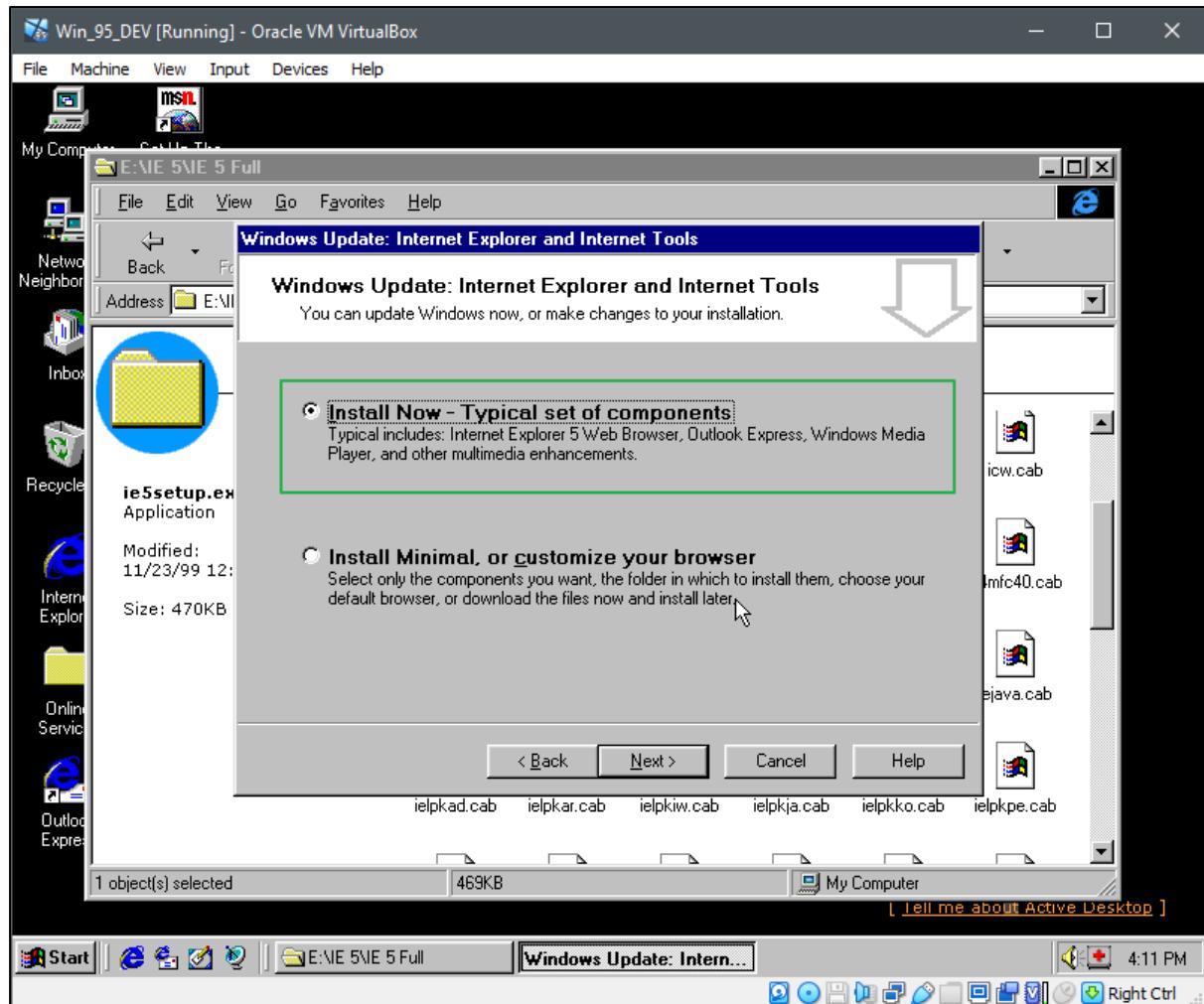
Mount the “Internet Explorer 5-5.5.iso” from VirtualBox.

Navigate to the CD directory in Windows 95 and find Drive:\IE-5\IE 5 Full\ie5setup.exe

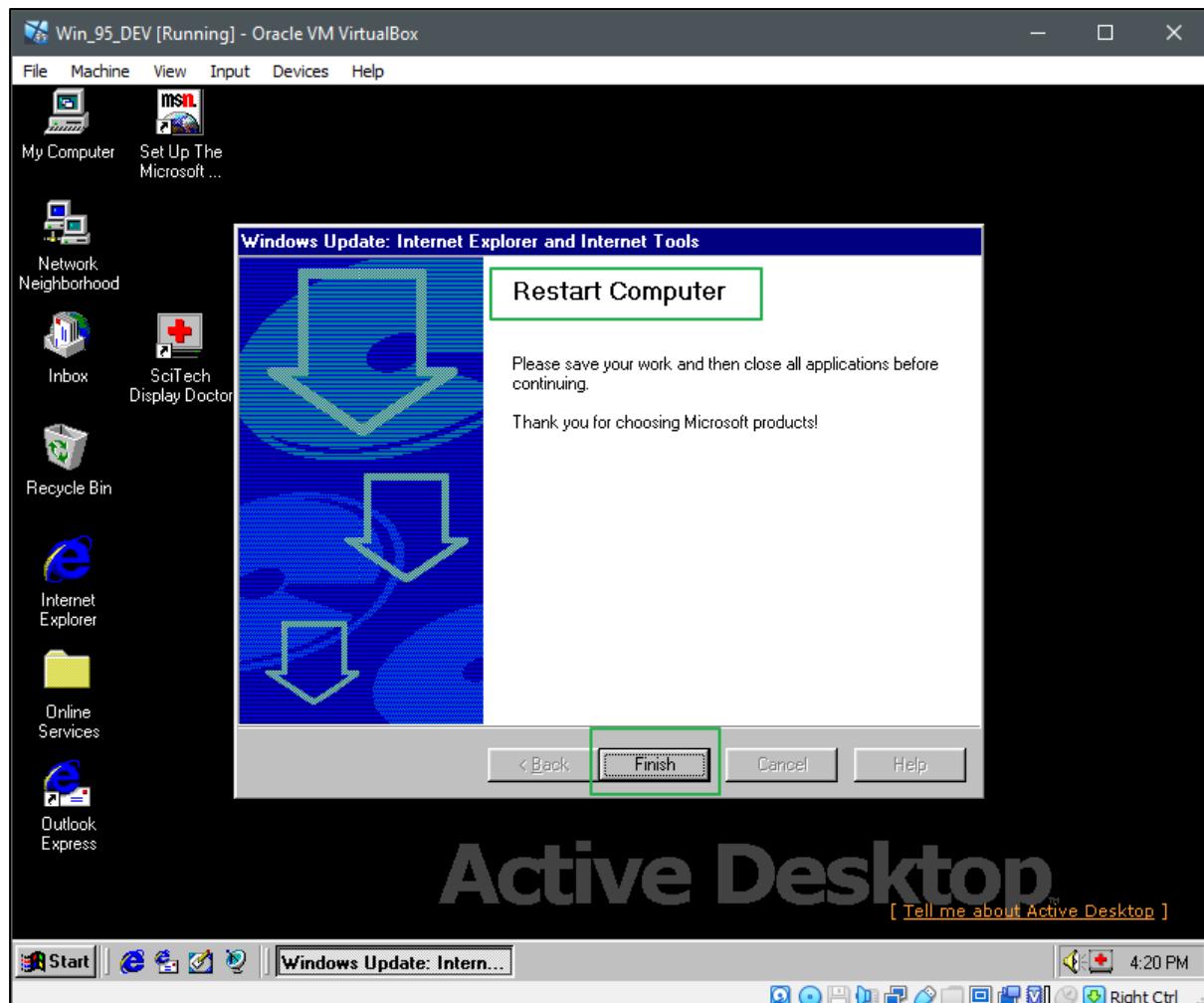
Run the installer.



Install the Typical set of components. This is essentially a full install.



After the install has completed you will be asked to restart the computer.



After the restart the system will update some of the configurations. After this the IE 5.0 Install is complete.

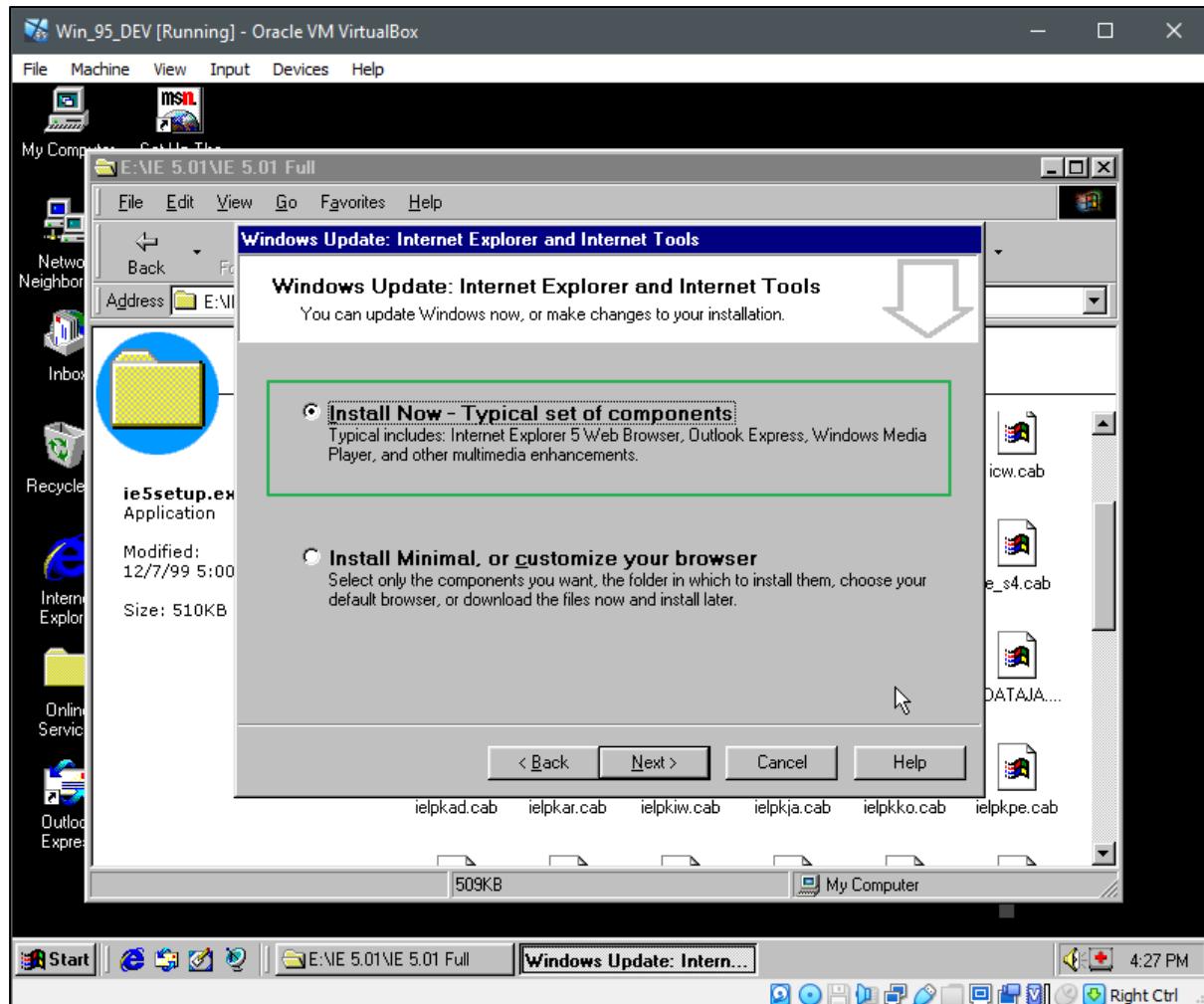
### IE 5.01

“Internet Explorer 5-5.5.iso”

Mount the “Internet Explorer 5-5.5.iso” from VirtualBox.

Navigate to the CD directory in Windows 95 and find Drive:\IE 5.01\IE 5.01 Full\ie5setup.exe

Run the installer.



Restart the computer when instructed. After the final configurations are finished proceed to the 5.01 SP2 install.

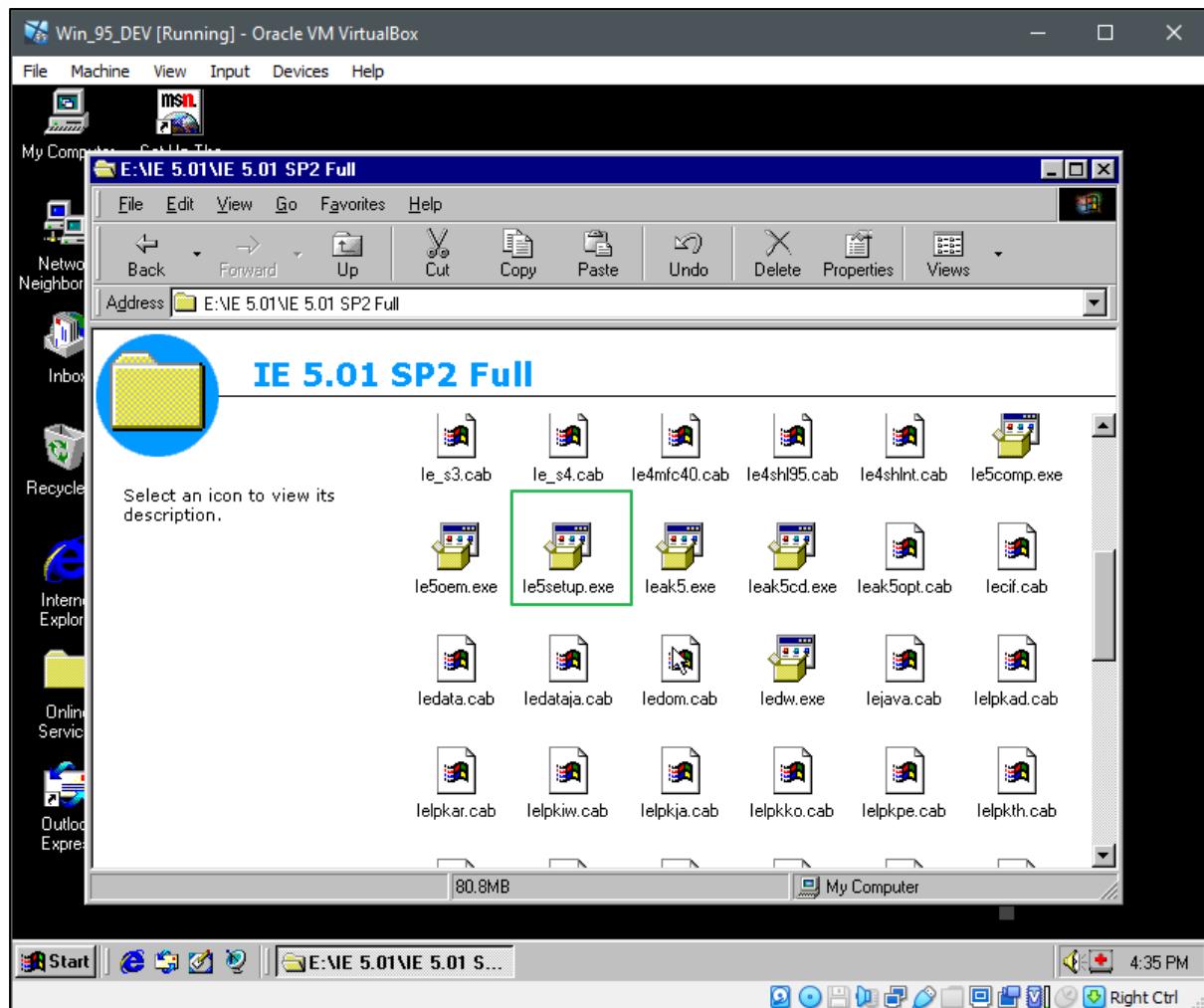
### IE 5.01 SP2

"Internet Explorer 5-5.5.iso"

Mount the "Internet Explorer 5-5.5.iso" from VirtualBox.

Navigate to the CD directory in Windows 95 and find Drive:\IE-5\IE 5.01 SP2 Full\ie5setup.exe

Run the installer.



The following install processes are the same as the previous IE 5.x installs. When the restart has completed move on to the next IE 5.5 version.

### IE 5.5

"Internet Explorer 5-5.5.iso"

Mount the "Internet Explorer 5-5.5.iso" from VirtualBox.

Navigate to the CD directory in Windows 95 and find Drive:\IE 5.5\IE 5.5 Full\ie5setup.exe

Run the installer.

Done...

### IE 5.5 SP1

"Internet Explorer 5-5.5.iso"

Mount the "Internet Explorer 5-5.5.iso" from VirtualBox.

Navigate to the CD directory in Windows 95 and find Drive:\IE 5.5\IE 5.5 SP1 Full\ie5setup.exe

Run the installer.

Done...

### IE 5.5 SP2

“Internet Explorer 5-5.5.iso”

Mount the “Internet Explorer 5-5.5.iso” from VirtualBox.

Navigate to the CD directory in Windows 95 and find Drive:\IE 5.5\IE 5.5 SP2 Full\ie5setup.exe

Run the installer.

Done...

Unmount the “Internet Explorer 5-5.5.iso” image from VirtualBox.

At this point most of the windows components have been update to the latest versions before the Windows 95 End of life. Most functionality is somewhat close to early editions of Windows 98.

There will still be some runtimes required to be updated such as Visual Basic, C and C++ runtimes as well as DirectX. On occasion you may run into an issue where an individual component may need to be upgraded for an application as well as some patches for hardware specific issues. Otherwise the Windows 95 Install is near as complete as it can get except for the DirectX install.

See winworldpc and other sites if a specific patch or update is required.

<https://winworldpc.com/product/windows-95/patches>

It is worthwhile doing another safety back up of the Virtual machine VHD at this point. After all of the main components are installed we can do a clean-up of the left over install components before creating a final clean install image of the updated Windows 95 for future recovery or to clone for additional virtual machine clients. Note at this point my Windows 95 install is taking up about 300MB of space on the C: drive.

## Runtime Libraries

### DirectX

DirectX V8.0a is the last version that can run on Windows 95. v8.0a is newer than v8.0.

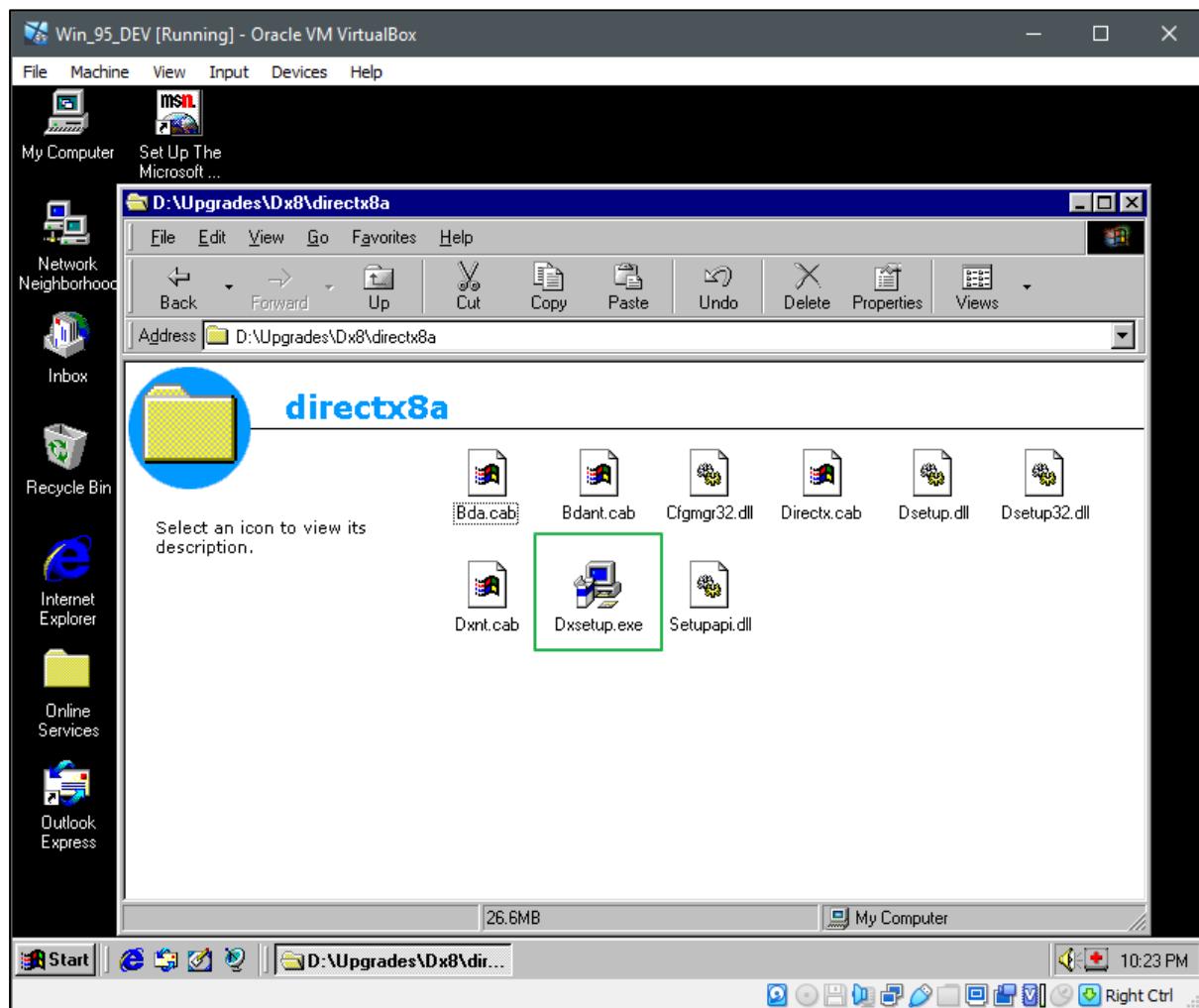
### DX8.0a

<http://www.oldversion.com/windows/directx-8-0a>

“directx80a.exe”

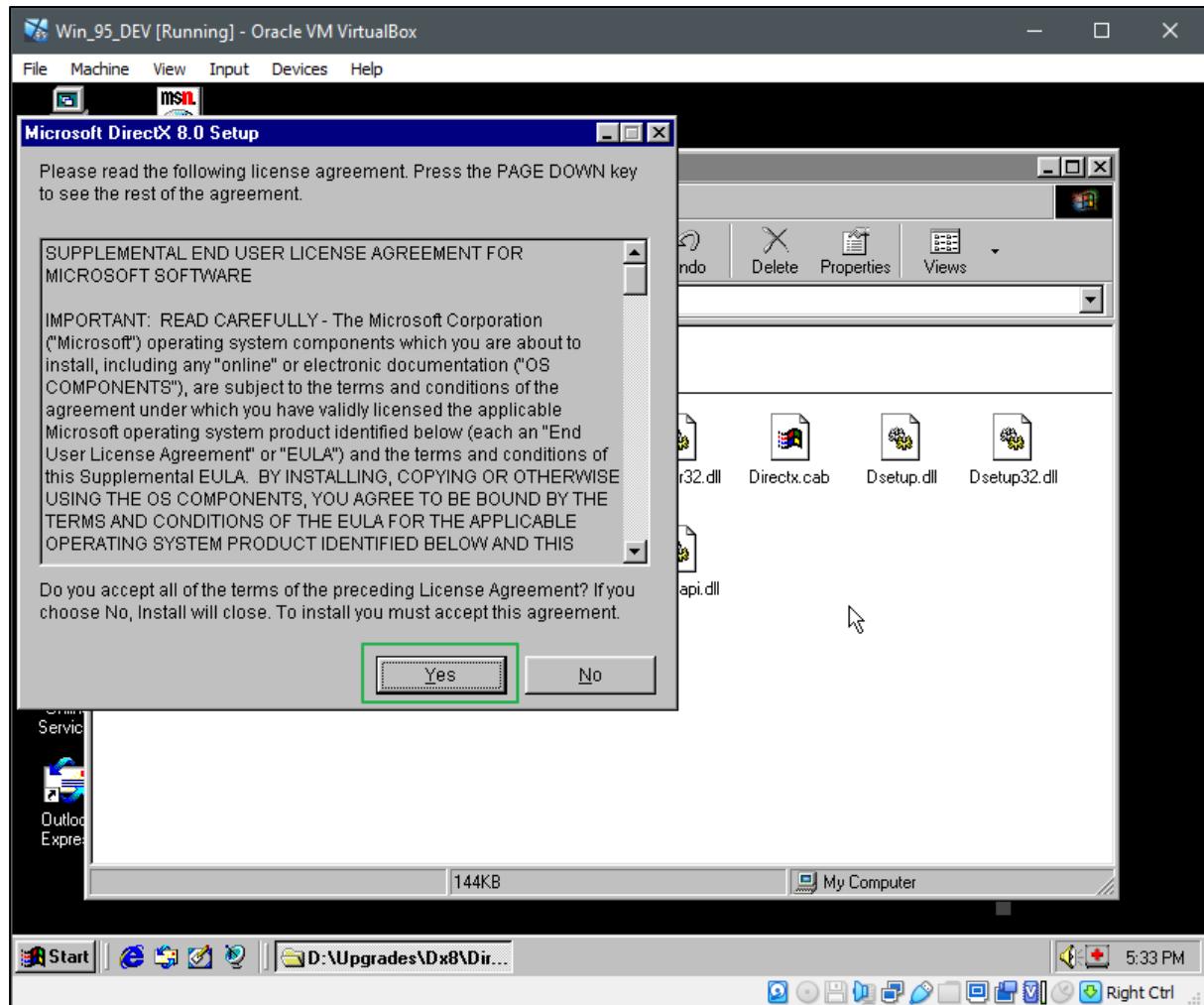
Unpack the \redist directory from the archive “directx80a.exe” and transfer the \direcxt8a directory it to your Windows 95 VHD.

Navigate to the location of the \direcxt8a installer directory in Windows 95 and run the “dxsetup.exe” file.



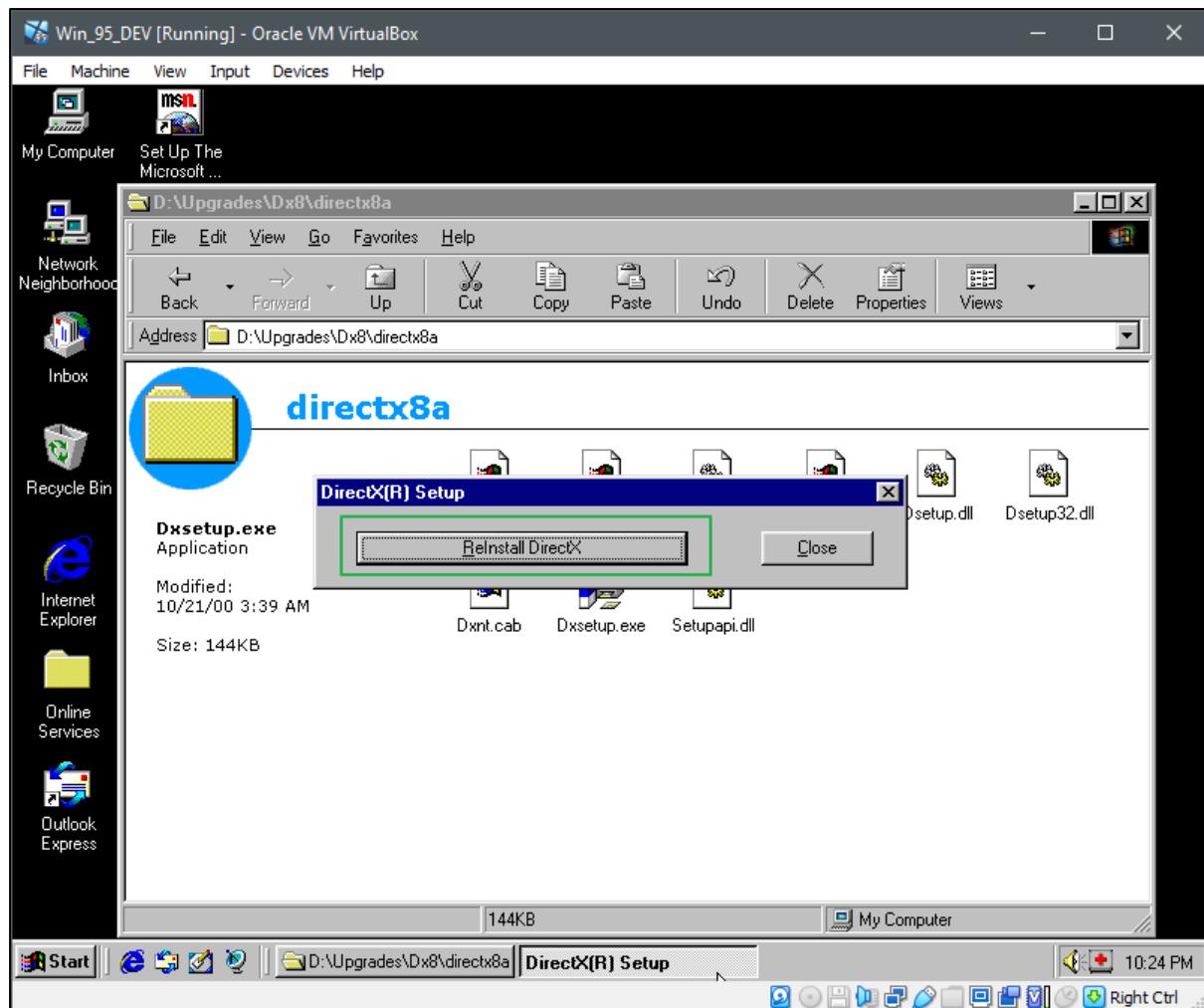
Accept the EULA.

## A Beginners Guide To DOS Programming

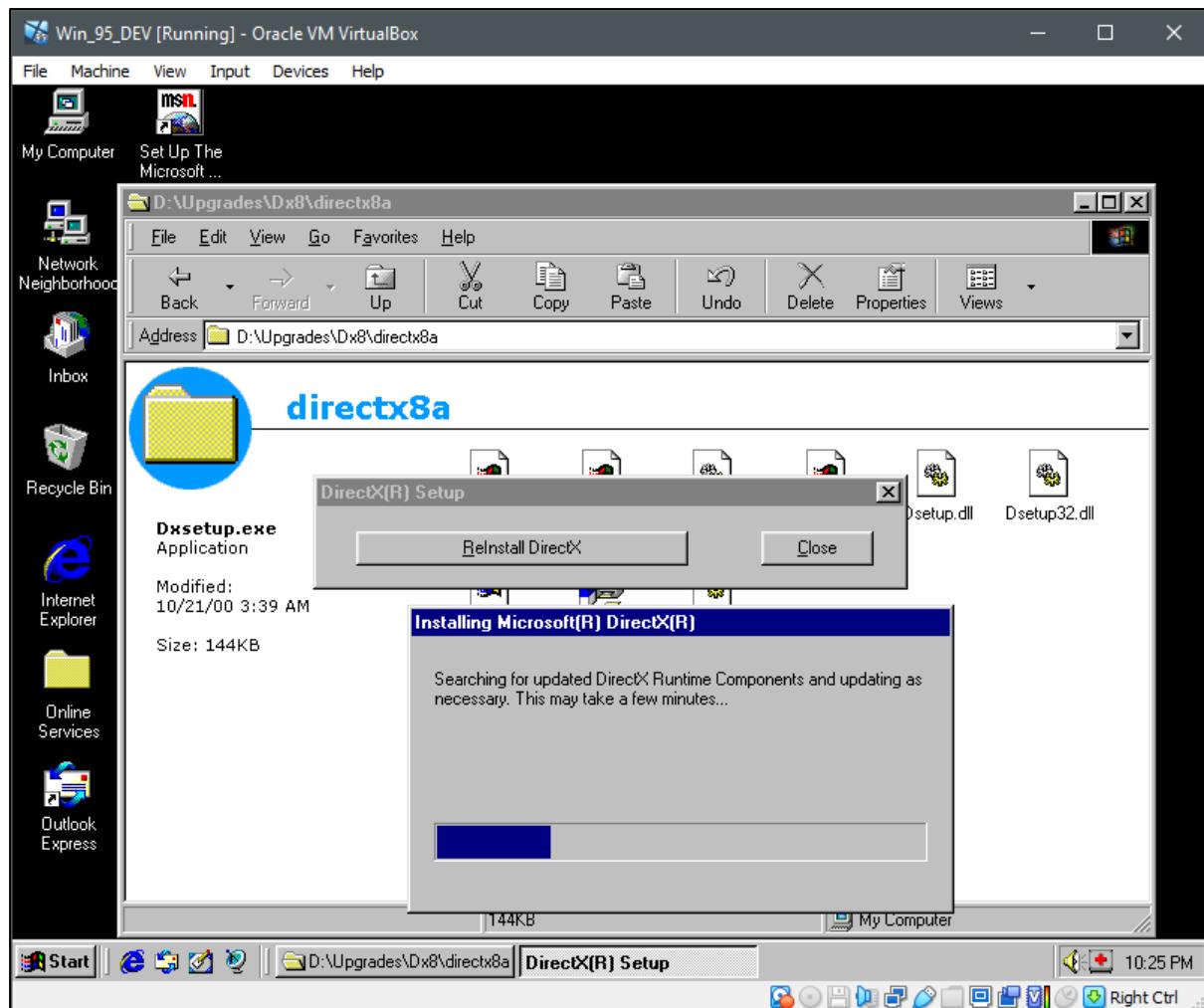


Install Directx 8.0a.

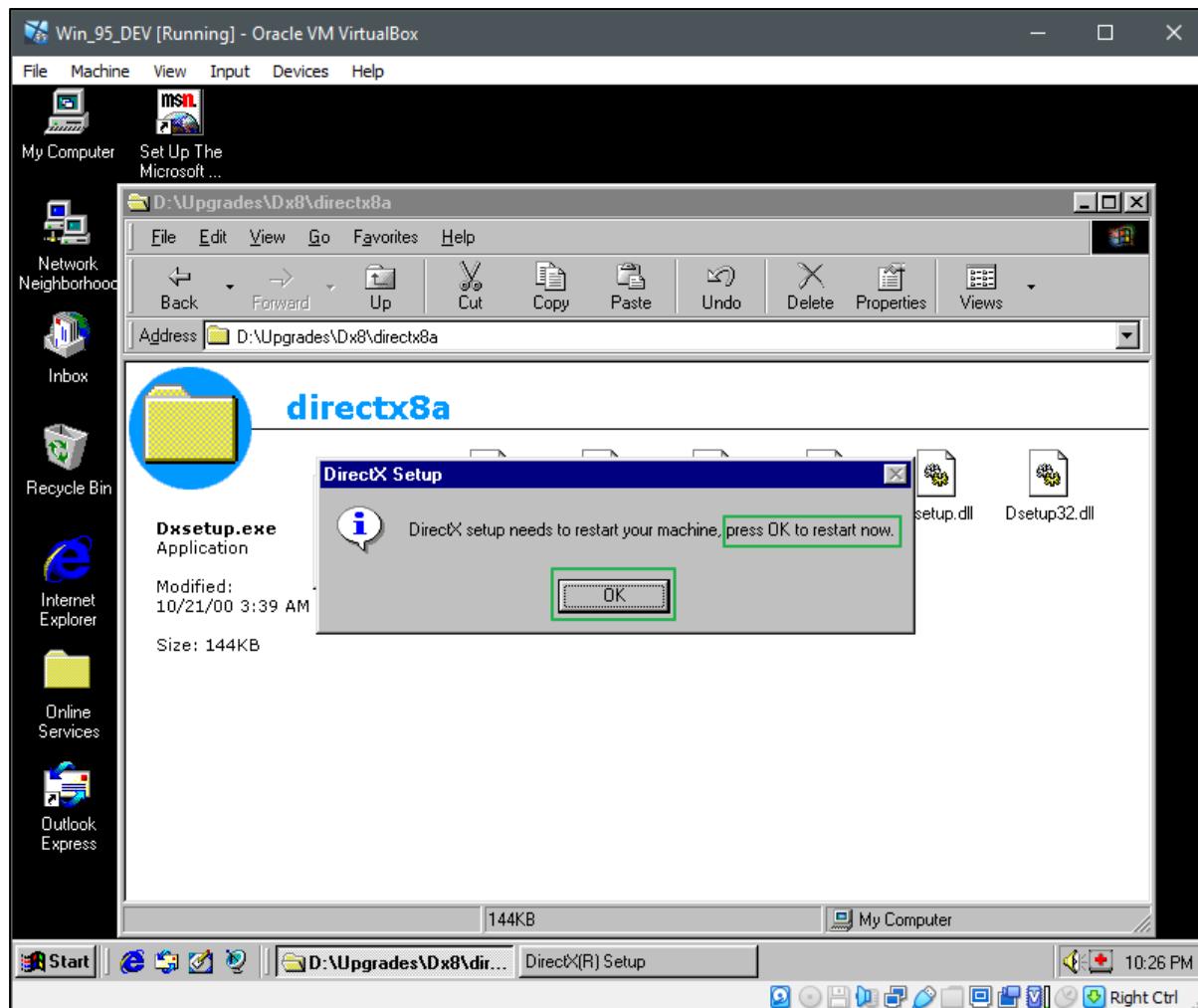
## A Beginners Guide To DOS Programming



## A Beginners Guide To DOS Programming



Restart Windows 95 when asked.



After the restart the DirectX 8.0a install has completed.

### Microsoft Visual C(++) 6 Runtime Files

"vc6redistsetup\_enu.exe"

Only install if required.

[https://web.archive.org/web/20070226050026/https://download.microsoft.com/download/vc60pro/update/1/w9xnt4/en-us/vc6redistsetup\\_enu.exe](https://web.archive.org/web/20070226050026/https://download.microsoft.com/download/vc60pro/update/1/w9xnt4/en-us/vc6redistsetup_enu.exe)

You will need to unpack the downloaded archive to access the installer.

**NOTE:** As you progressively work with Windows 9x you will encounter version issues with the many DLL libraries. This is unavoidable and what was referred to at the time as "DLL Hell". There were so many alternative versions of the DLLs including updates and fixes that it can be difficult to find a match for the application that is failing.

In some instances you will find patches and updates that you can apply to the system, but this can

often leave you with another application that requires a different DLL version. What can you do? If the application you are attempting to run requires specific runtime DLL files and the application exist in its own directory away from the system path you can place the DLLs next to the application executable. Windows 9x will look in the application folder for DLLs before checking the system path if not found with the application. Use tools such as Depends V22 to help find what library dependencies are needed. I have provided more information on using Dependency Walker in the Development environments section.

### VisualBasic Runtime libraries

Only install if required.

vb3run.exe

vb4run.exe

Msvbvm50.exe

VBRUN60.exe

VB6.0-KB290887-X86.exe

You can download a combined installer pack here. If it is not available then you may need to source the individual files.

<https://sourceforge.net/projects/vb6extendedruntime/>

<http://ftpmirror.your.org/pub/misc/ftp.microsoft.com/Softlib/MSLFILES/>

I would recommend making a clean Windows 95 full upgrade backup of the VirtualBox primary VHD drive containing Windows 95 at this stage. Don't forget to keep a text file with a description of what components were installed for future reference.

---

## Essential Applications

Before going too far into the large number of utilities and applications available for Windows 95 I have to cover some of the essential convenience applications required to perform some common tasks. Windows 95 comes quite "Bare Bones" out of the box, so I will cover some of the basics required to make using Windows 95 a little easier.

### Archives

It is difficult to avoid working with archive files. Most downloaded files will be in a zip or rar archive and Windows 95 does not contain any native tools for this.

### 7-Zip 9.20

7-Zip 9.20 is the last version that will work on Windows 95 and is capable of working with the majority of common archive types.

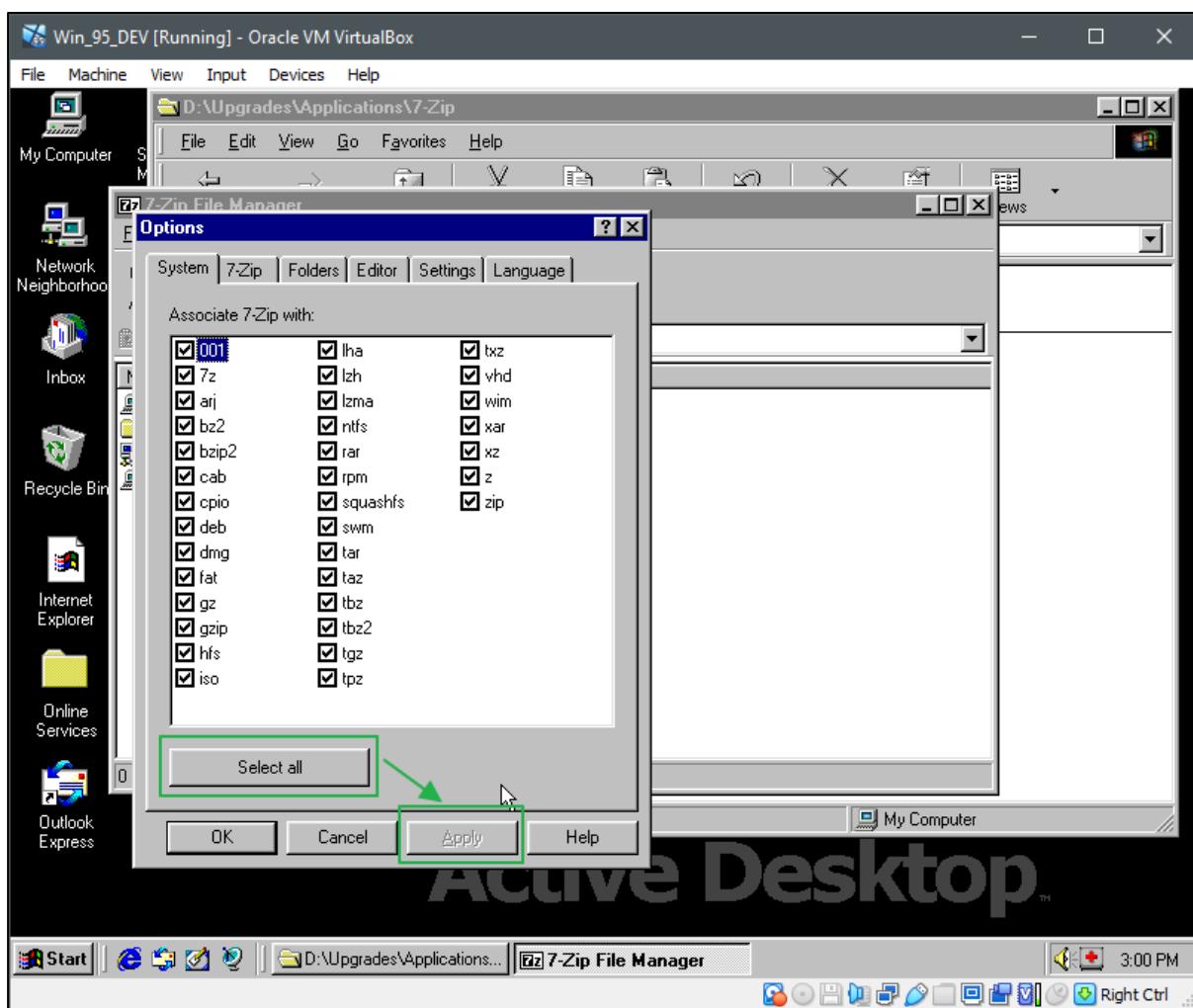
<https://sourceforge.net/projects/sevenzip/files/7-Zip/9.20/>

“7z920.exe”

Transfer the “7z920.exe” application installer to your Windows 95 drive, and run the installer.

After the install has completed Open the 7-Zip file manager from the start menu and then select “Tools -> Options”.

Under the tab “System” Select all to associate archive files with z-Zip. This will allow you to manage most archive files directly in the directory in which it is located.



Note: After installing Notepad++ you can also select NP++ as the default text editor for 7-Zip under the Editor tab in Options.

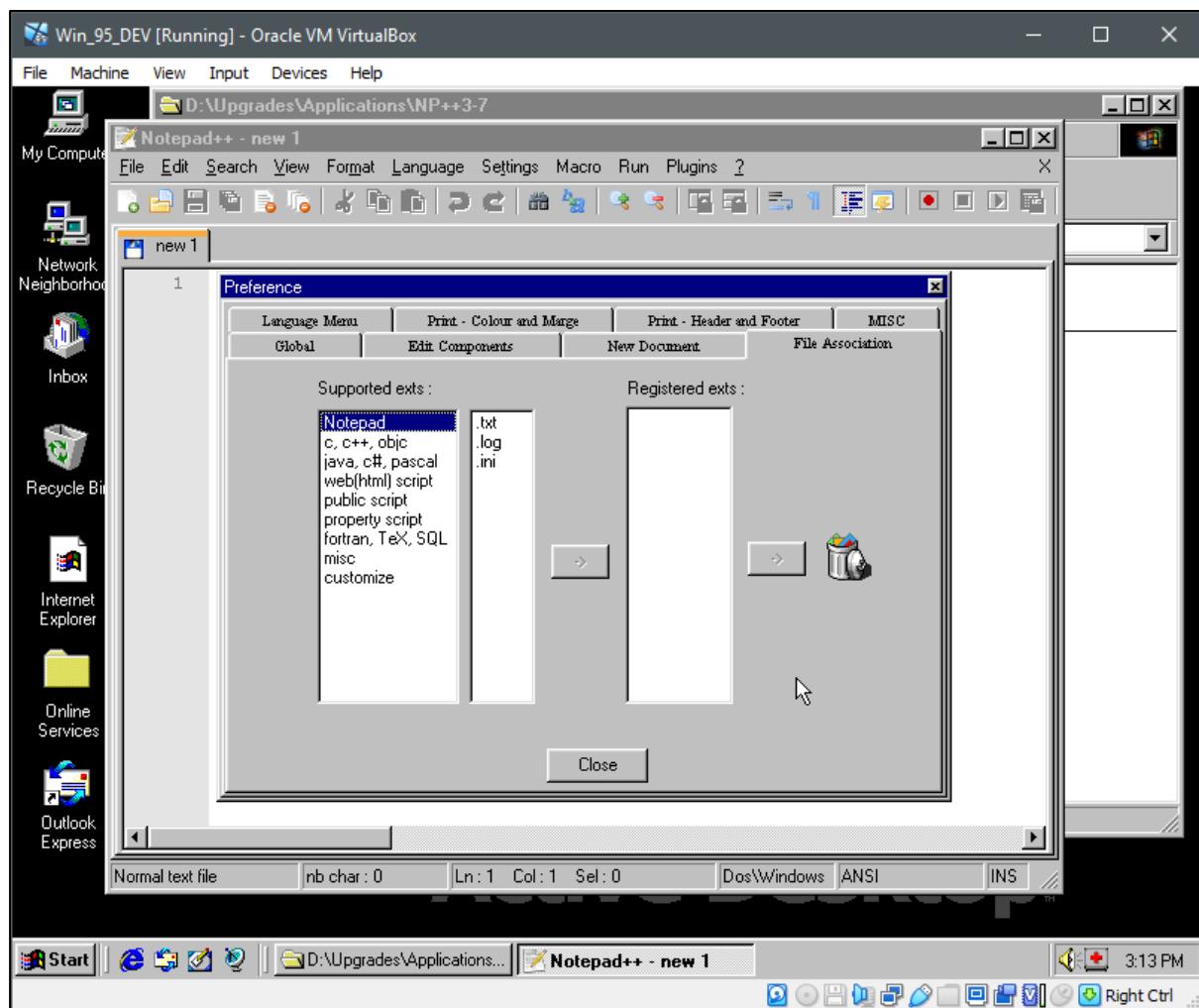
### Notepad++ v3.1 safe (3.7 max)

Some changes were made between 3.1 and 3.7. I have found V 3.7 to work well after the IE upgrades.

<http://www.oldversion.com/windows/notepad-3-7>

Transfer the “3.7\_npp.3.7.Installer.exe” application installer to your Windows 95 drive, and run the installer and use the defaults.

After the install open the “Settings -> Preferences” menu and navigate to the File Association tab. You can select and move wanted file extensions to the right hand section to automatically open the file type on double click. V3.7 does not have the “Open with Notepad++” context menu entry although it can be hacked in through the registry to set it as an option. I typically open Notepad++ and drag the file on to the editor.



### MS Word Viewer

Note that MS WordPad is installed by default when first installing Windows 95 and can be found under the Accessories section in the start menu.

If you need to view old RTF or DOC files a MS Word Viewer application installer can be found on the Windows 95 Install CD under Drive:\other\wordview\setup.exe.

You can install it directly from the CD-ROM.

Note that you have to click on the large icon to begin the installation.



If you really need some form of Word Processor, Abiword 2.9.4 is the last version that will work on Windows 95. You can find a copy at OldVersion

<http://www.oldversion.com/windows/abiword-2-9-4>

If you need to read Acrobat PDF files you can find old versions at winworldpc.

<https://winworldpc.com/product/adobe-acrobat/4x>

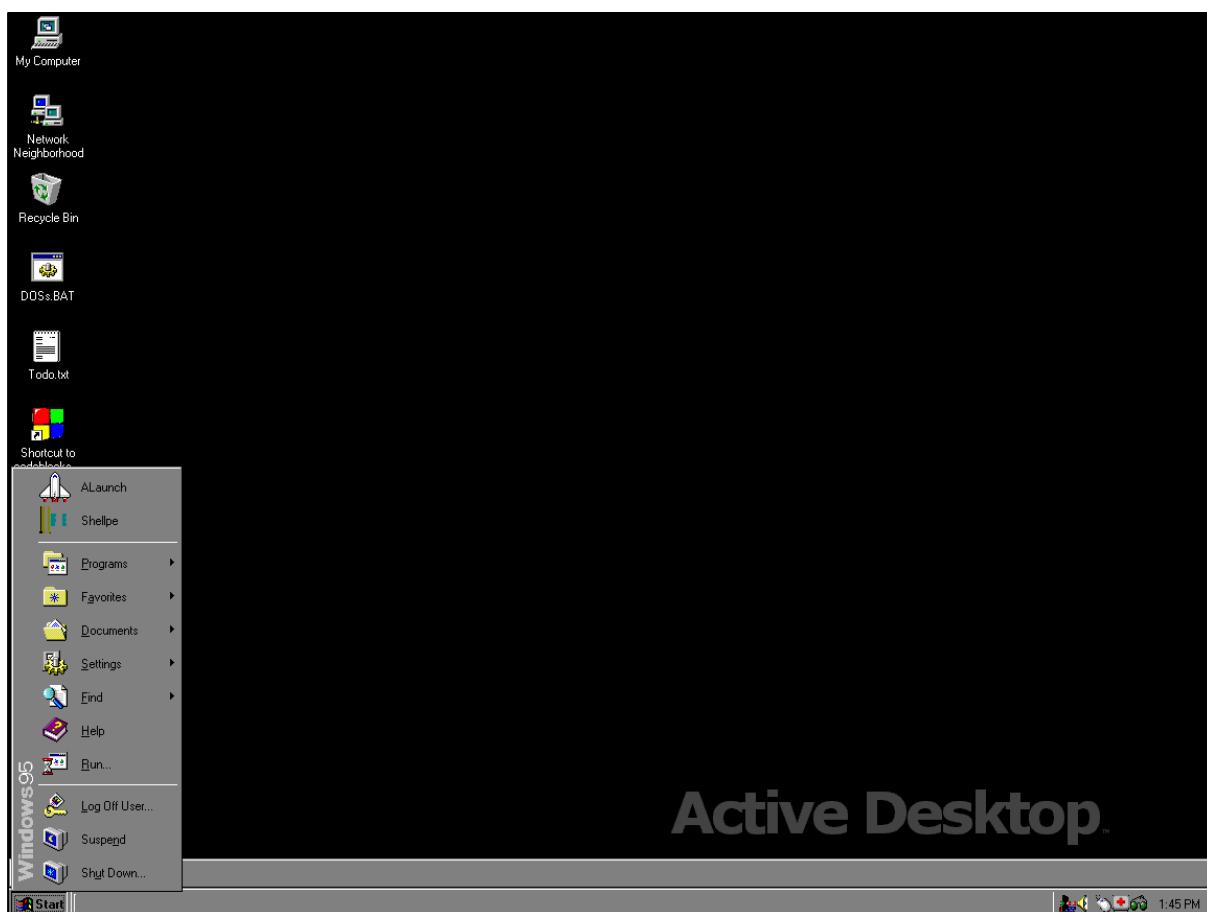
Other old versions of PDF readers such as Sumatra PDF 0.8.1 will also work. I would not install a PDF reader unless it is absolutely necessary.

### Advanced Launcher v1.4

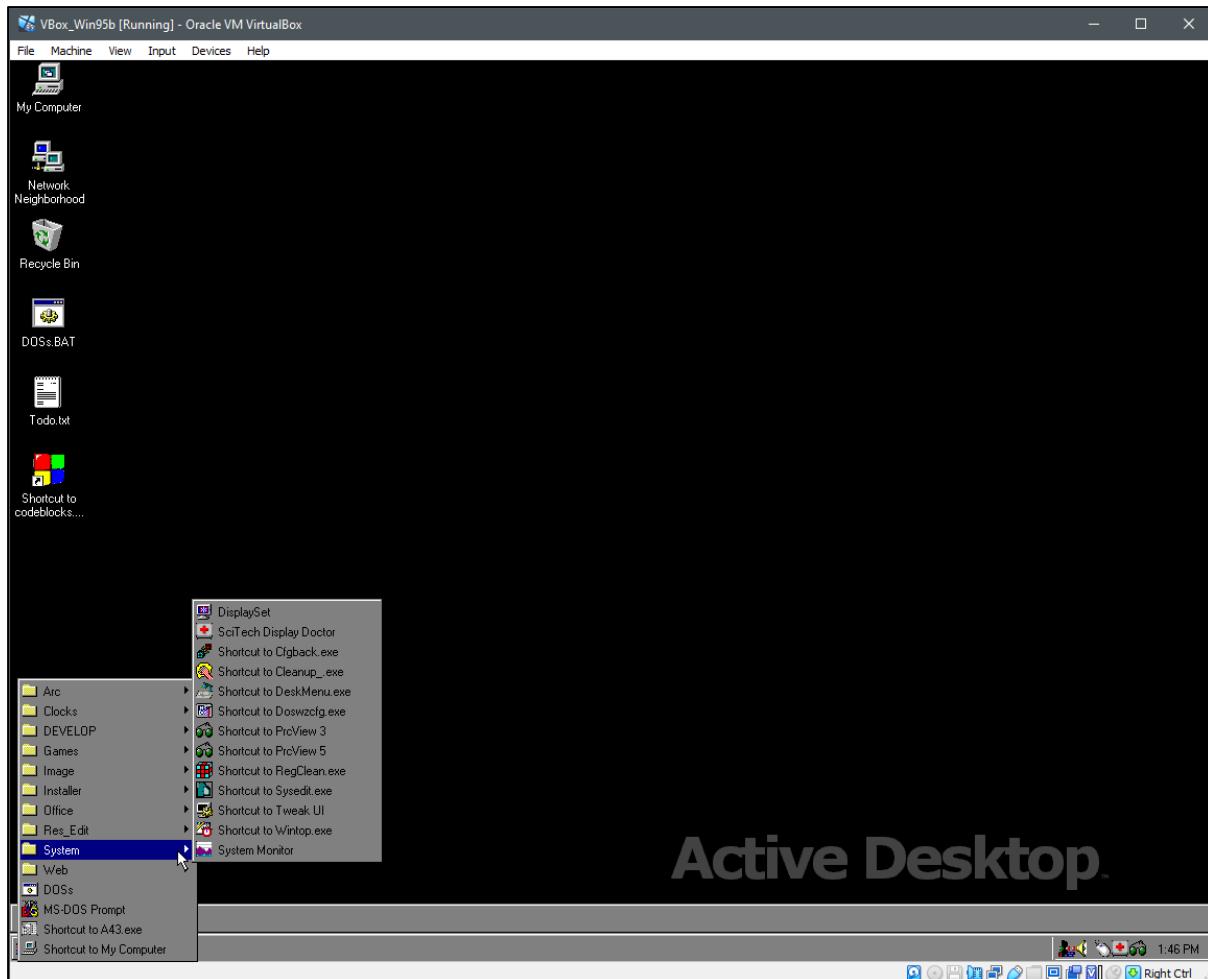
Advanced Launcher is an optional convenience tool. By this point you will likely have a mix of desktop icons scattered about and a number of start menu application entries. Later versions of Windows have a built in “Quick Launch” Toolbar to help organise your list of common application icons, but this does not yet exist in Windows 95. Advanced Launcher adds an additional tray above the Windows application tray which then allows you to create groups of quick launch menus. I find it convenient to keep the clutter off the Windows desktop and have my common applications organised. You can leave it turned off or start the menu when you need to.

<https://www.alentum.com/alaunch/>

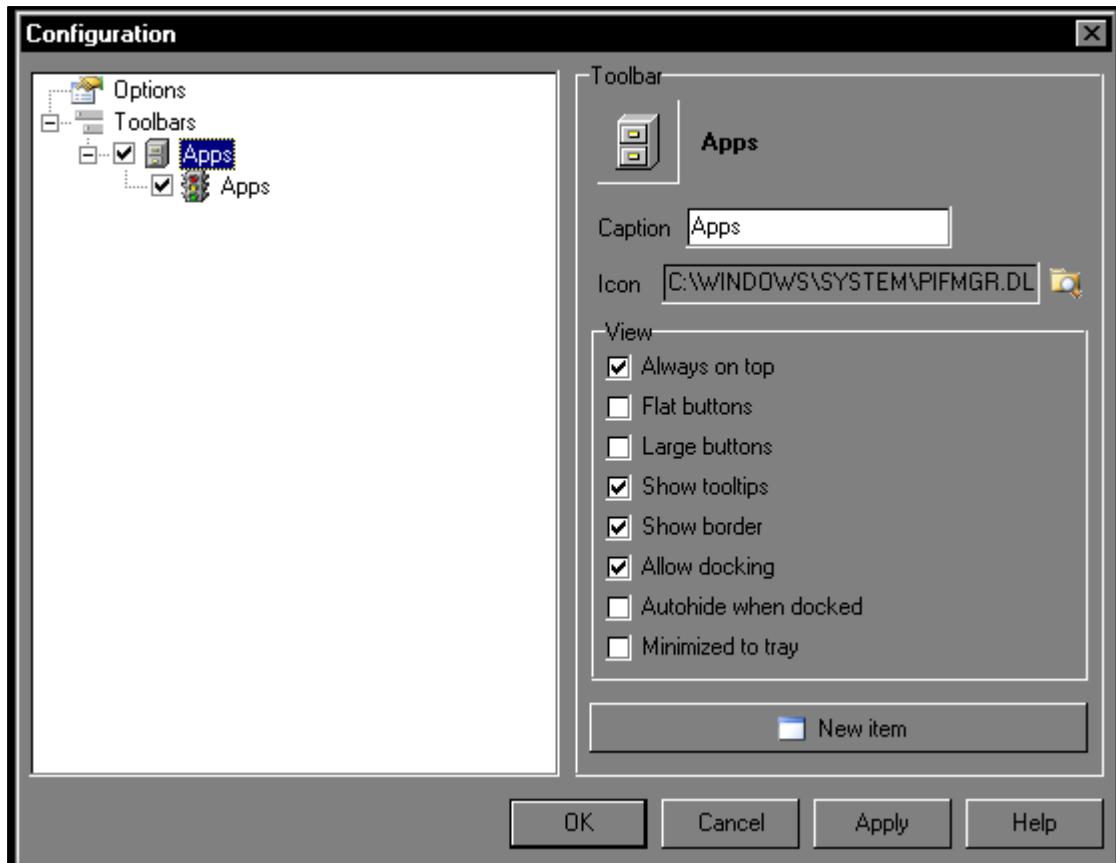
“ALaunchSetup.exe”



## A Beginners Guide To DOS Programming

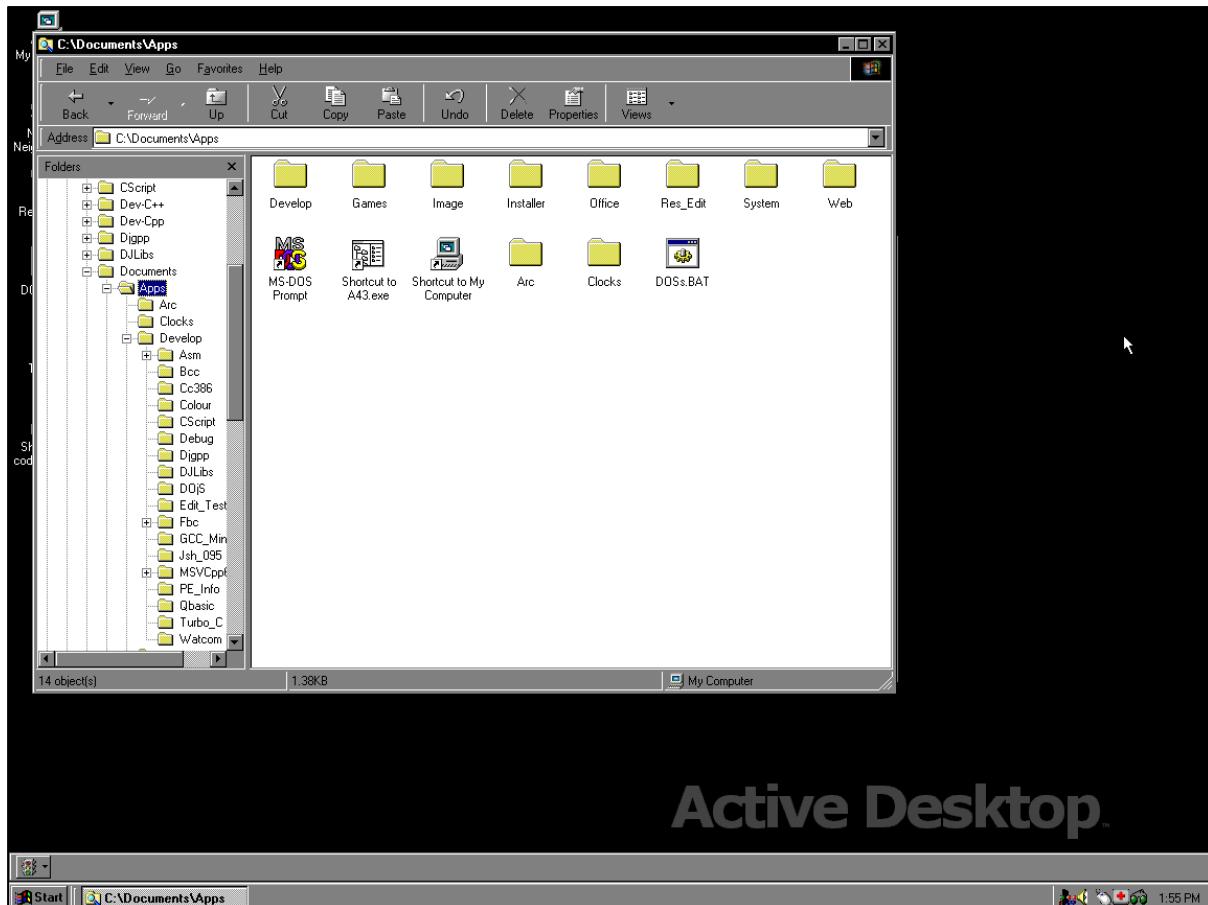


After Installing Advanced Launch you will need to become familiar with the Configuration window. You can move the Launch bar to any place on the desktop.

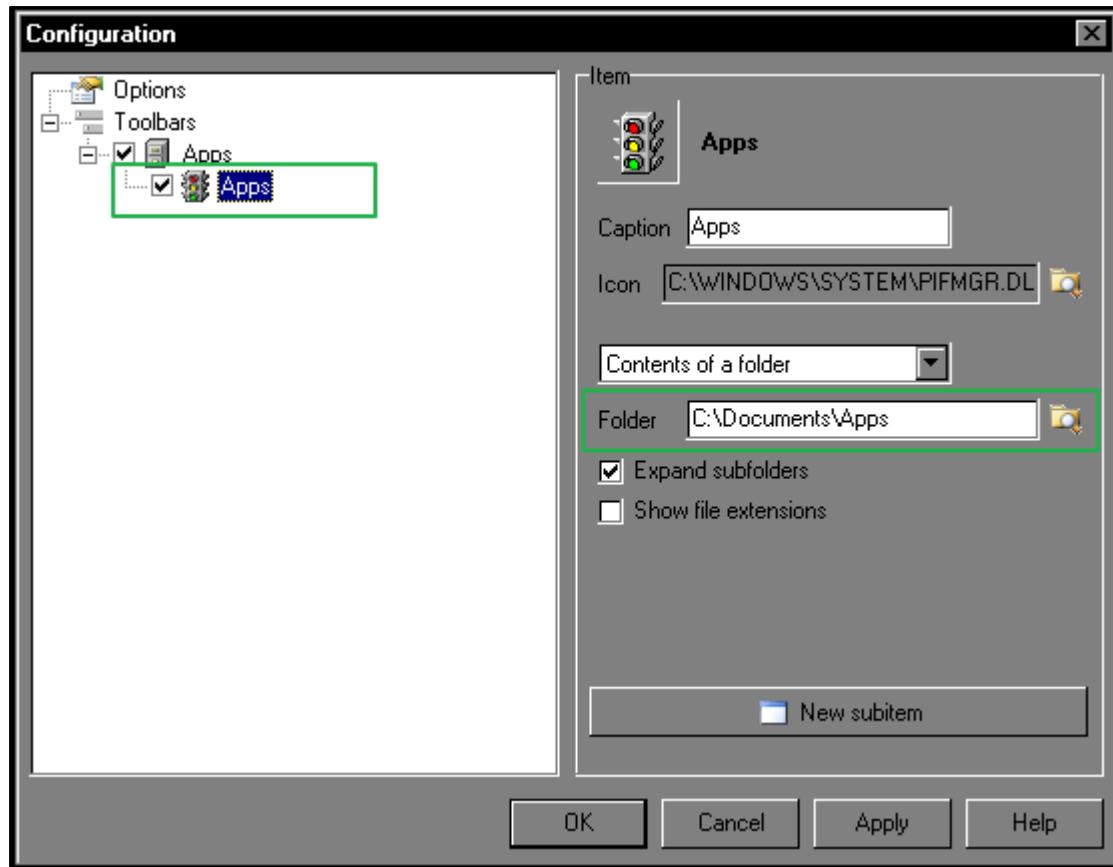


To create an organised App menu with groups Create a root folder in your hard drive and child folders named in groups. Place all of your application shortcuts into the folder where you want it to display on the menu. This is the same as for Later Windows Quick launch toolbars.

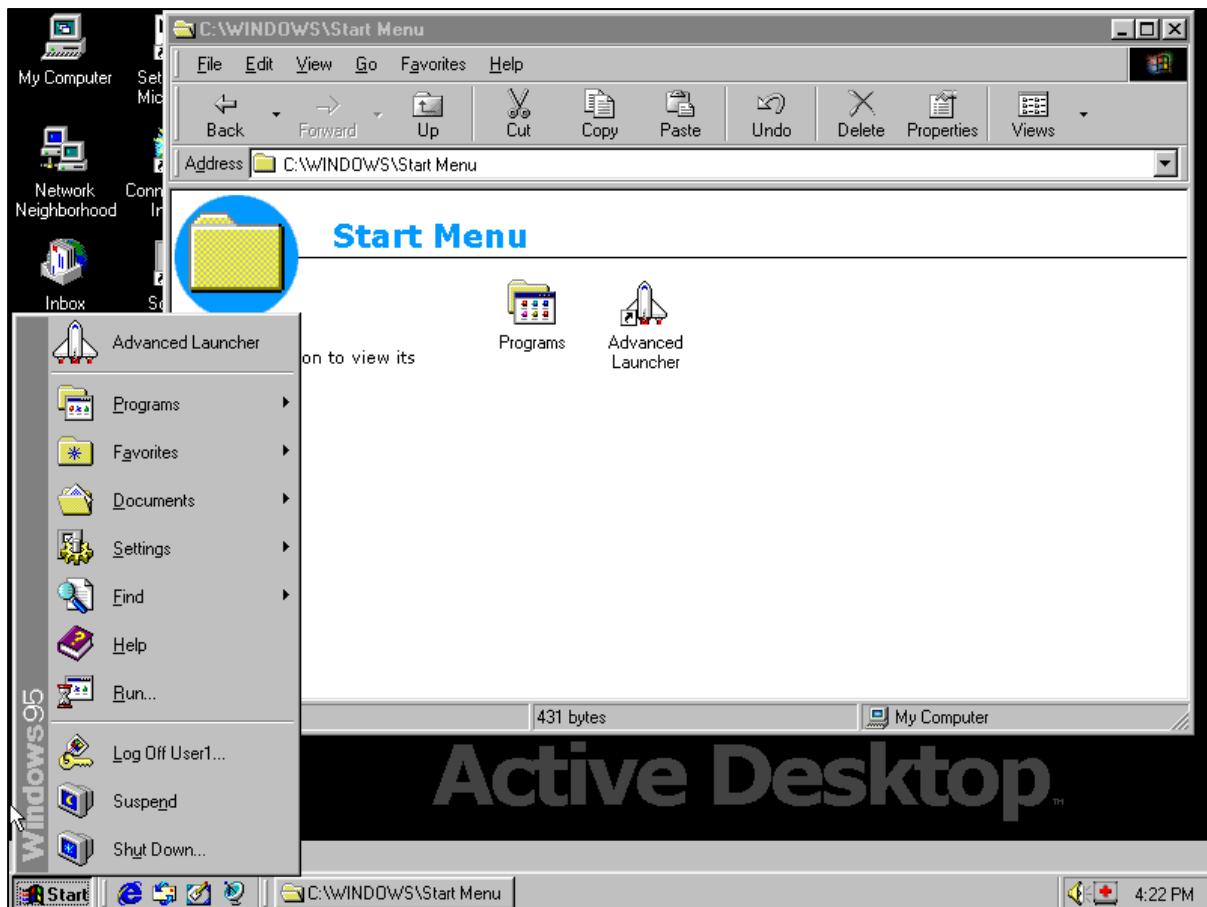
I currently have mine in C:\[My ]Documents\Apps\\*.\*



Finally point you menu entry to the root folder with all of your shortcuts. Whenever you create a new shortcut, place it in the appropriate windows folder above.



If you want to show the ALaunch icon in the first start menu section like I have, place the ALaunch icon/shortcut in C:\WINDOWS\Start Menu

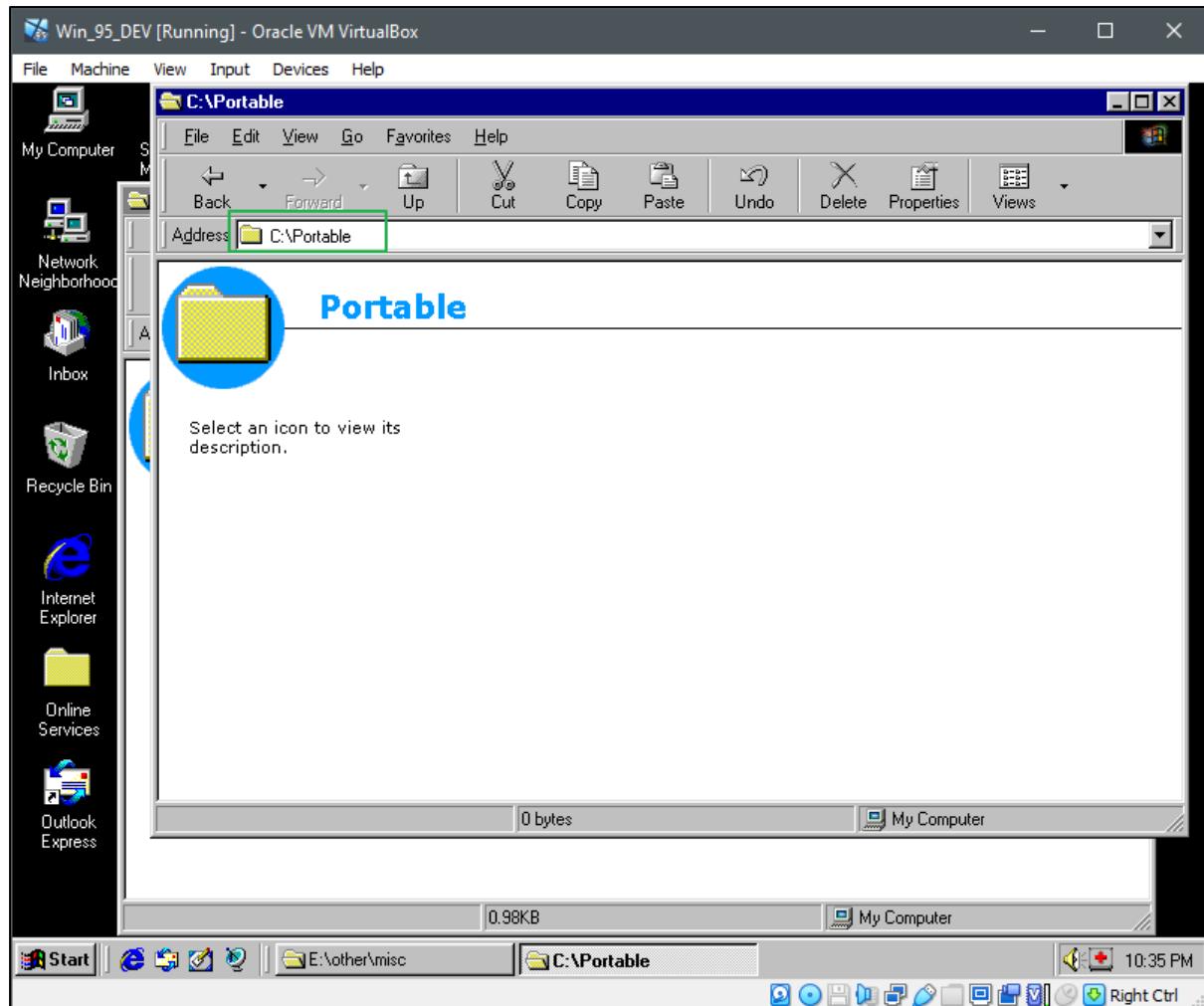


### Restore tools

Insert the original Windows 95 install CD ROM (ISO) from the virtualBox control panel. Navigate to the directory CD-Drive:\other\misc\

Open a second instance of the file manager (Explorer) and navigate to the root directory of the C: drive.

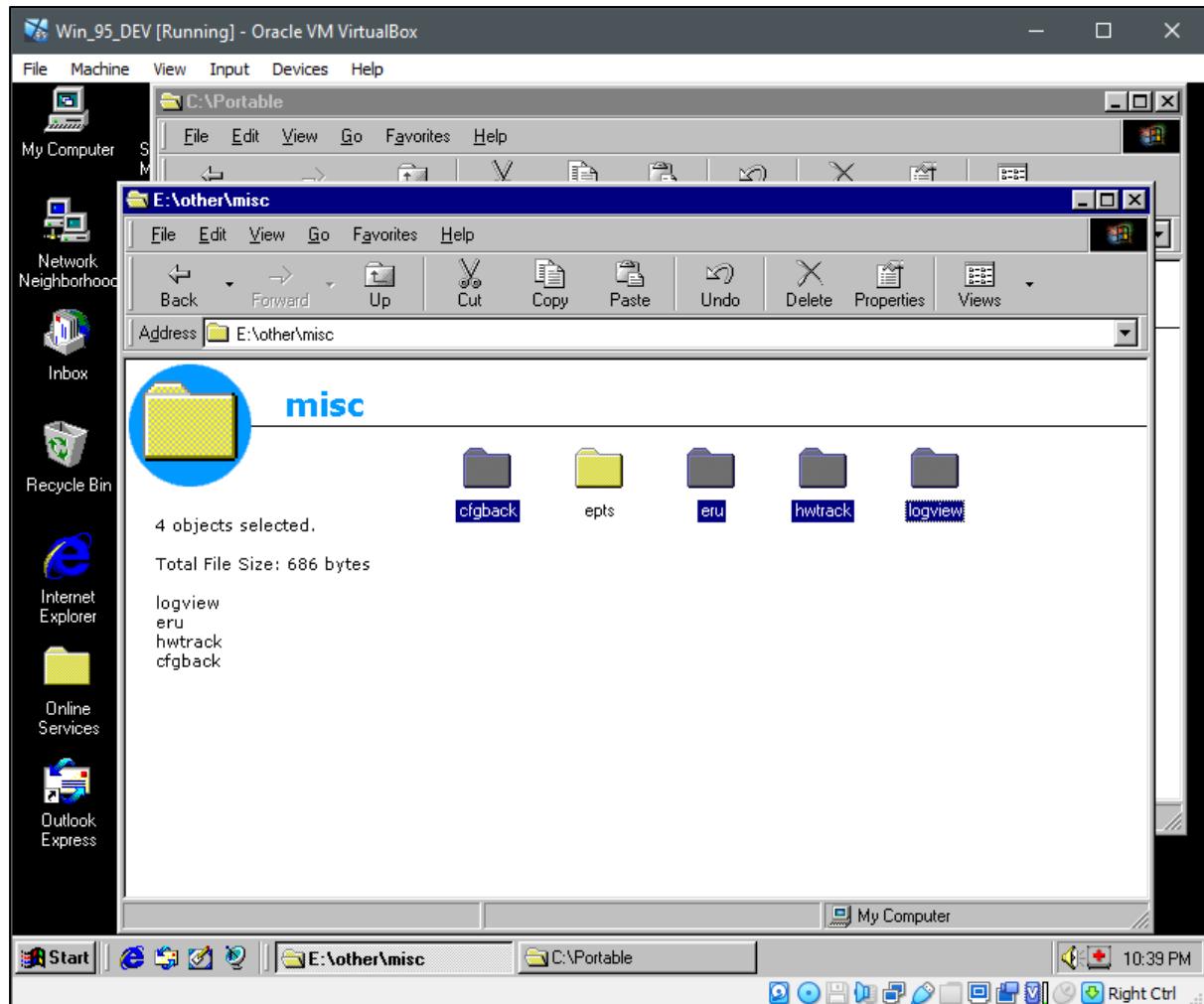
Create a new directory C:\Portable

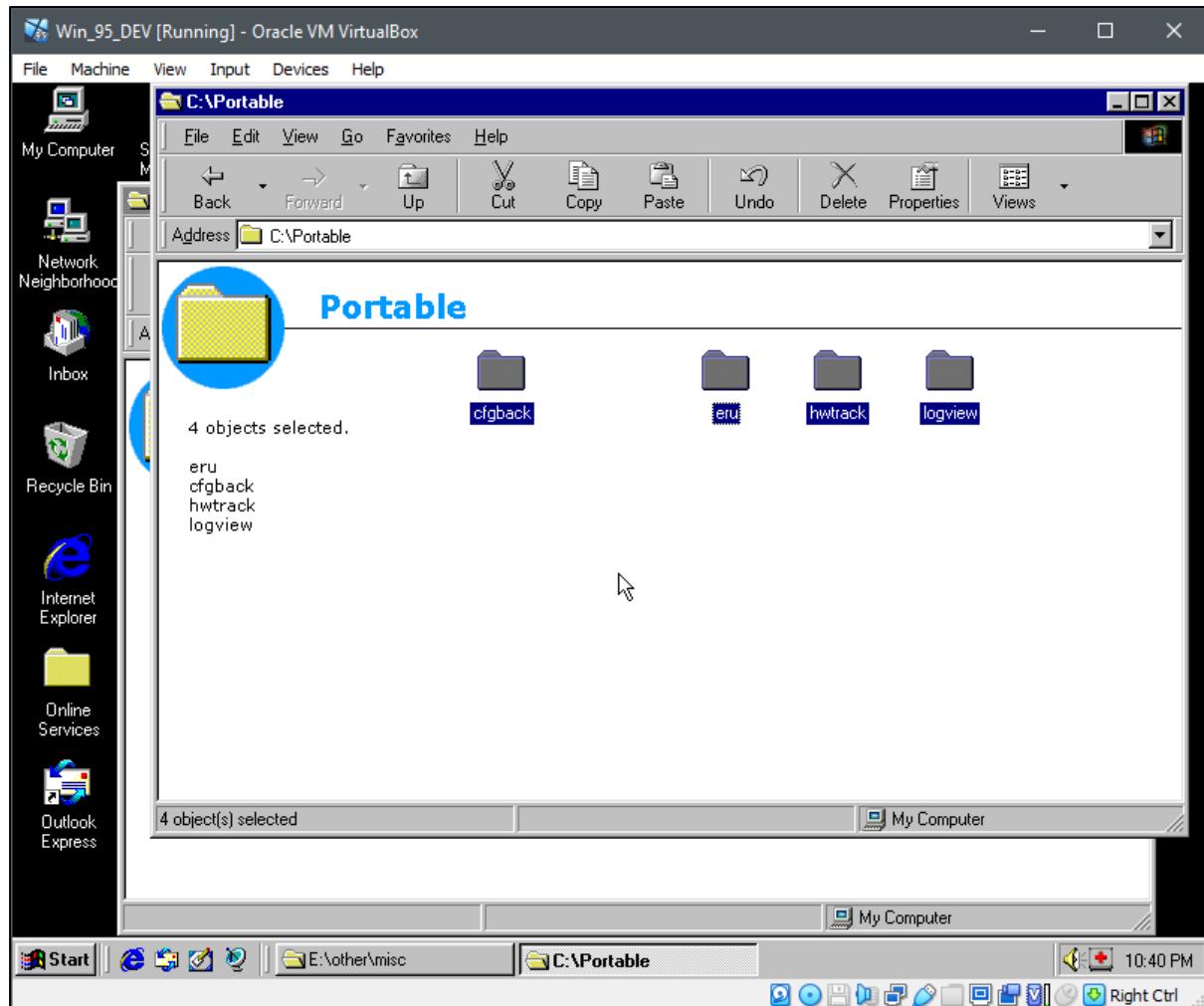


You can choose another name if you like. I use this directory for Non-installed or “Portable Applications”. This will often comprise of simple system utilities.

From the CD \misc directory select all directories except \epts and copy them the C:\Portable

## A Beginners Guide To DOS Programming



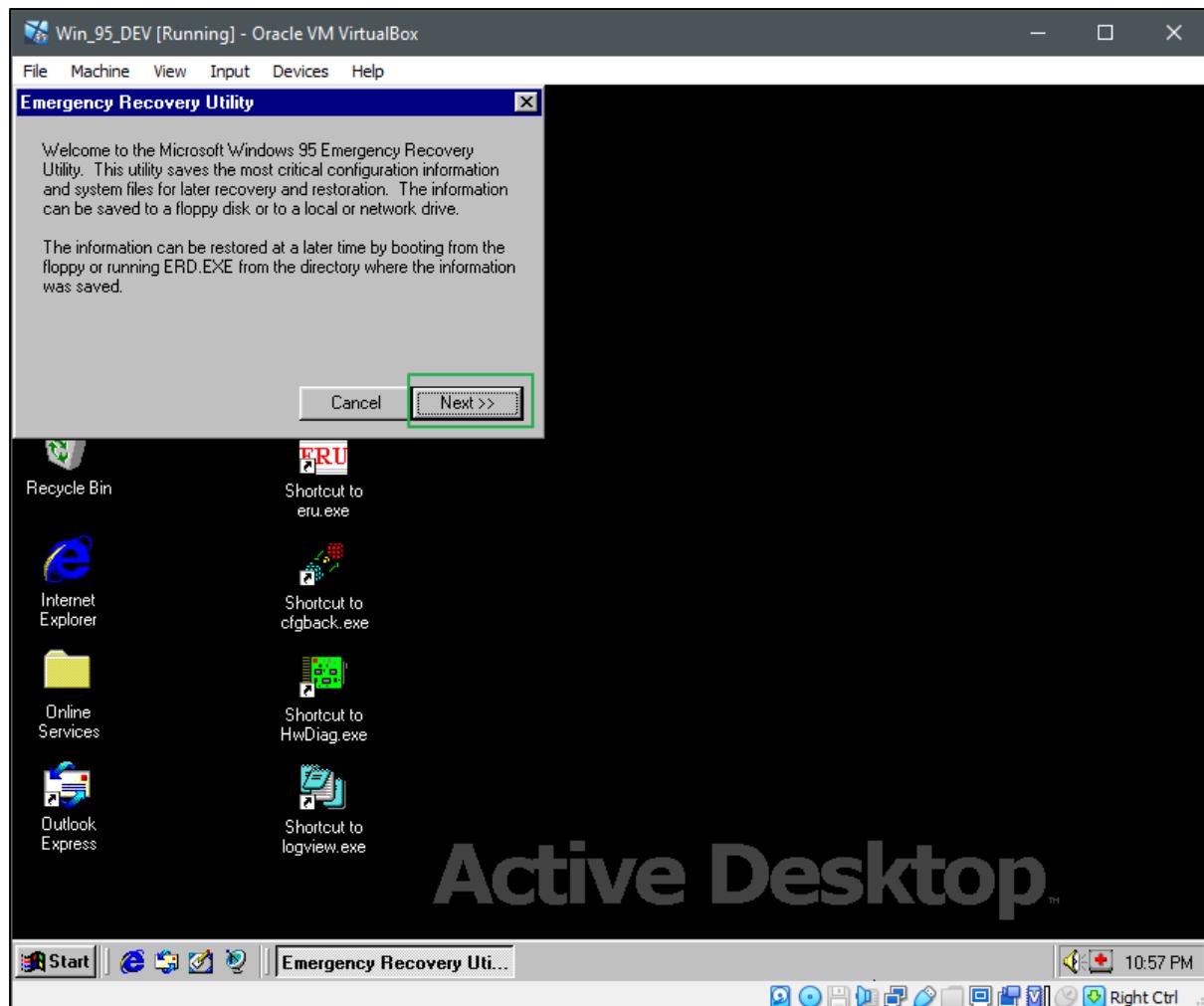


Read the notes and help files on each application. logview allows you to easily view the Windows logs in a single interface, hwtrack allows you to see the current hardware and system details.

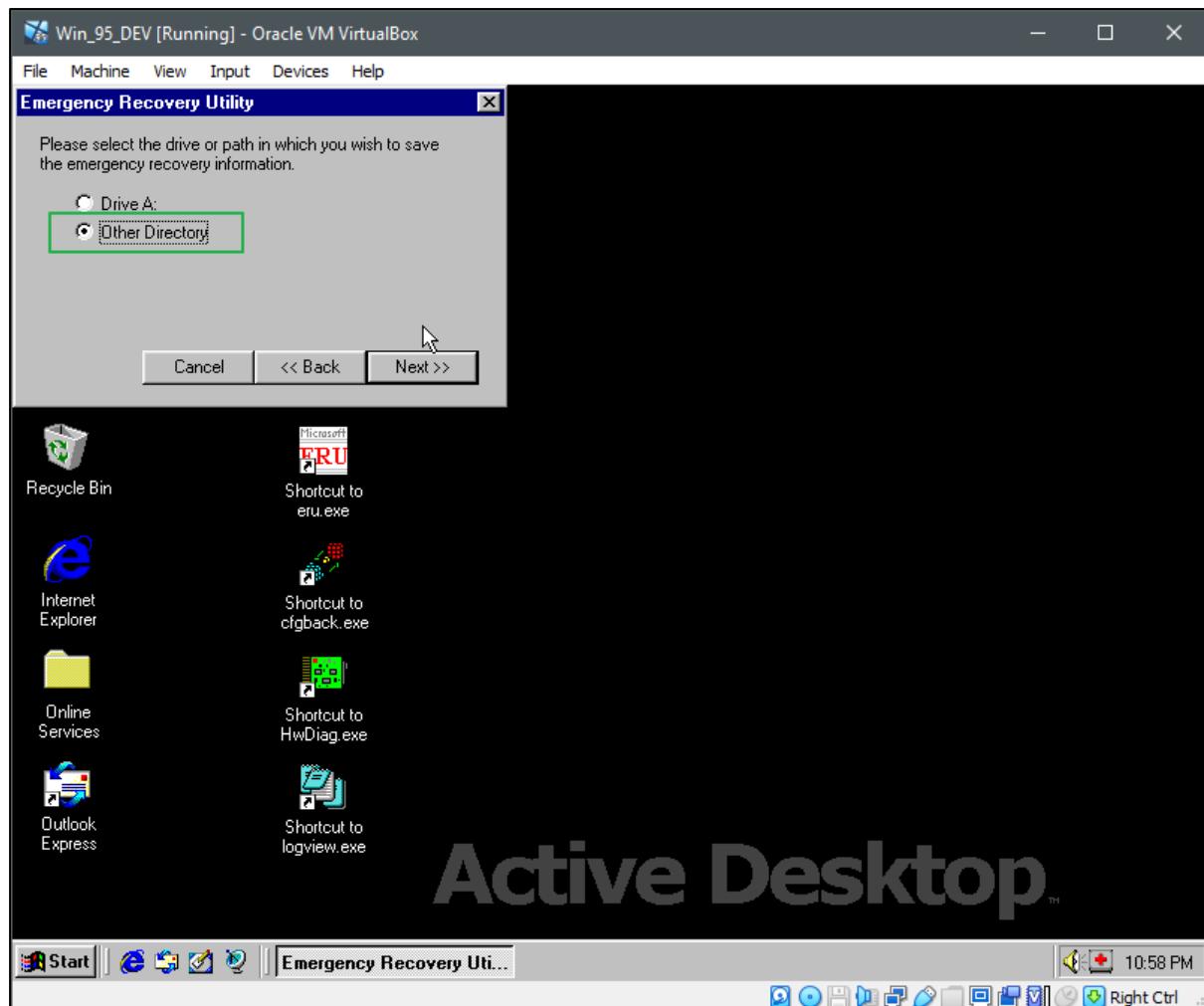
ERU and CFGBack are specialised recovery utilities. In case of a fatal system crash you can use these 2 tools from the command line via a DOS boot disk to recover your system. Note you will need to make your own shortcuts for portable applications. It is OK to place them on the desktop (Send to Desktop) for the moment, or place them in your ALaunch Quick launch folders.

### ERU

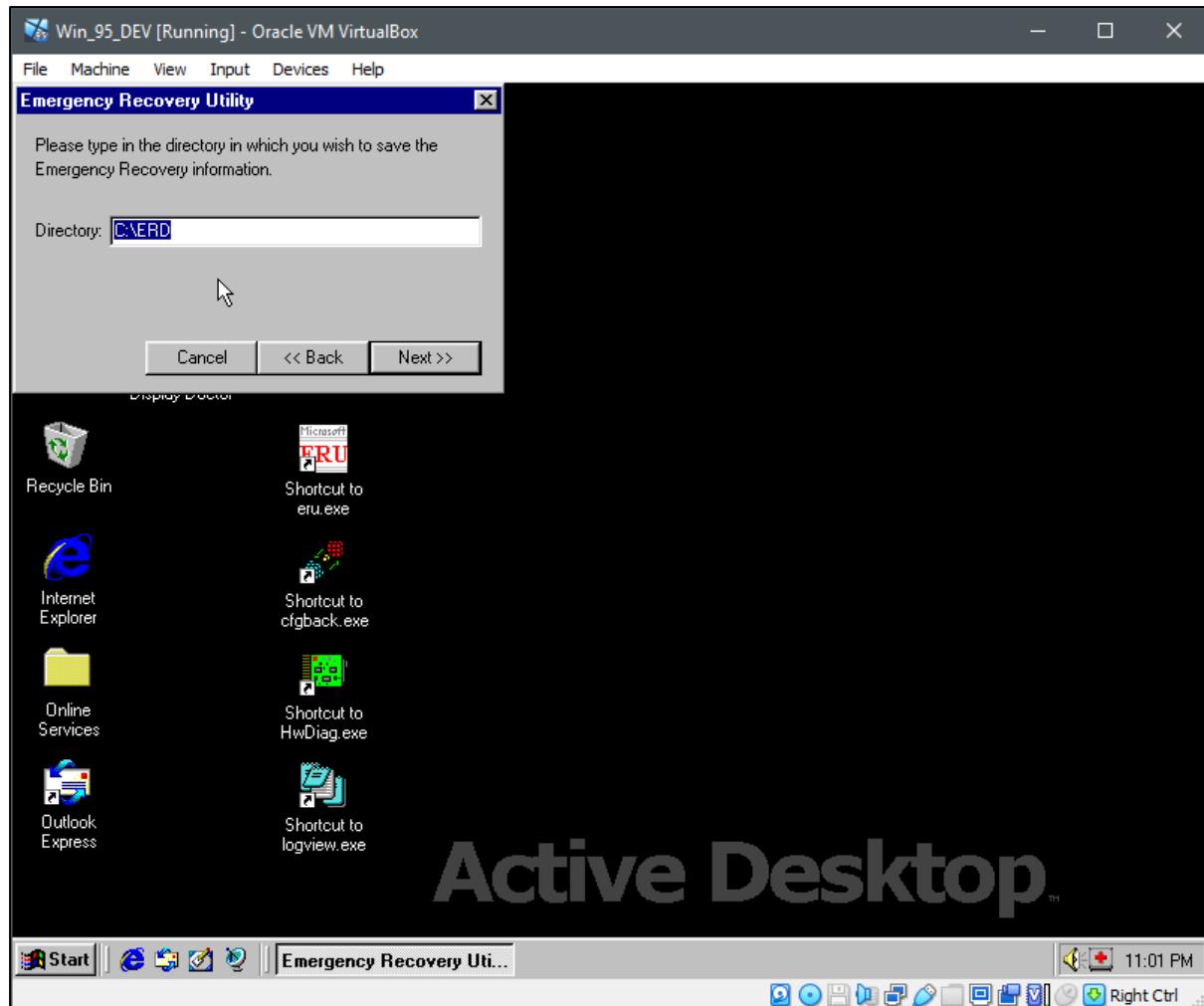
Run the ERU.exe and follow the prompts to create boot recovery set.



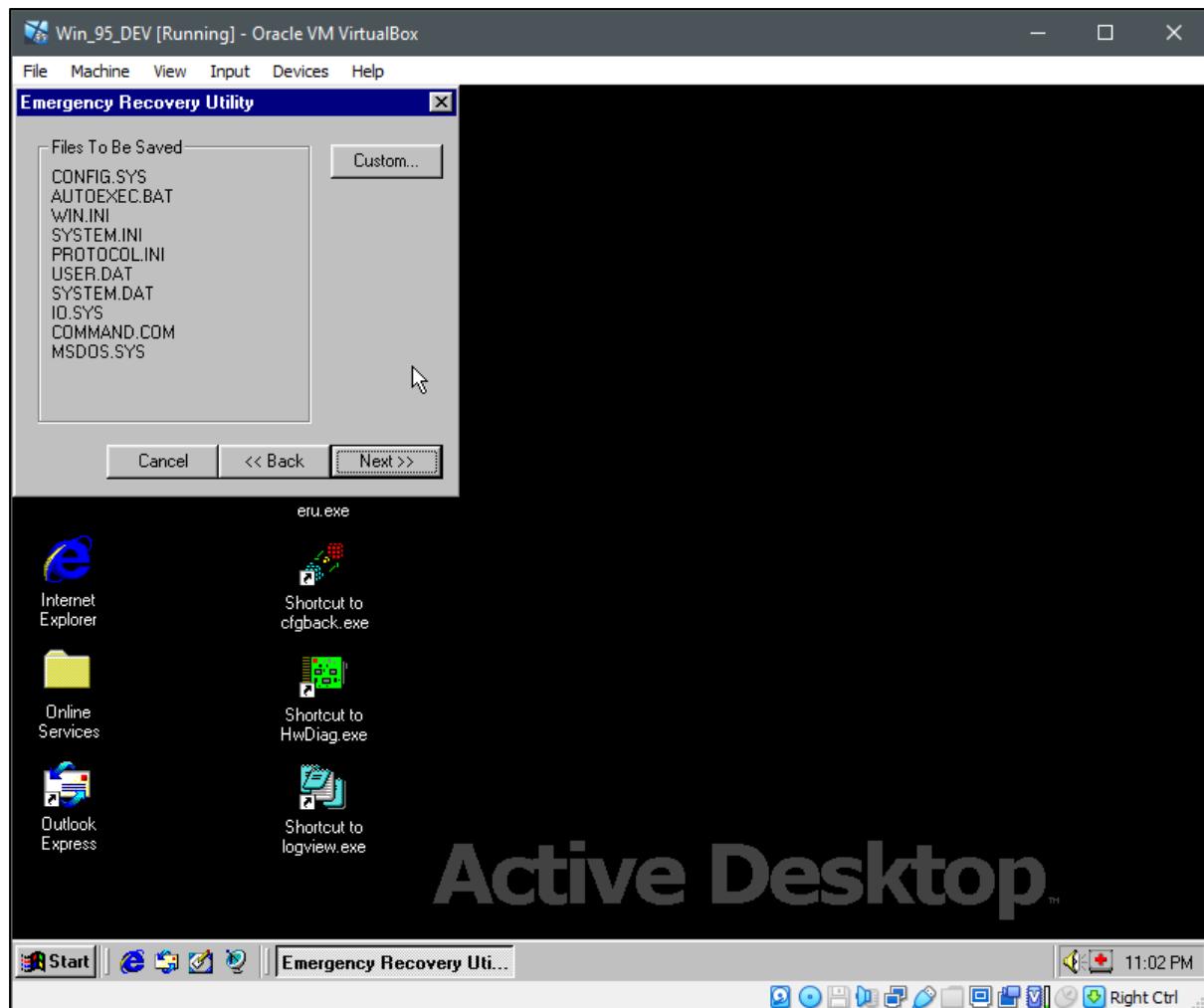
Choose Other Directory.



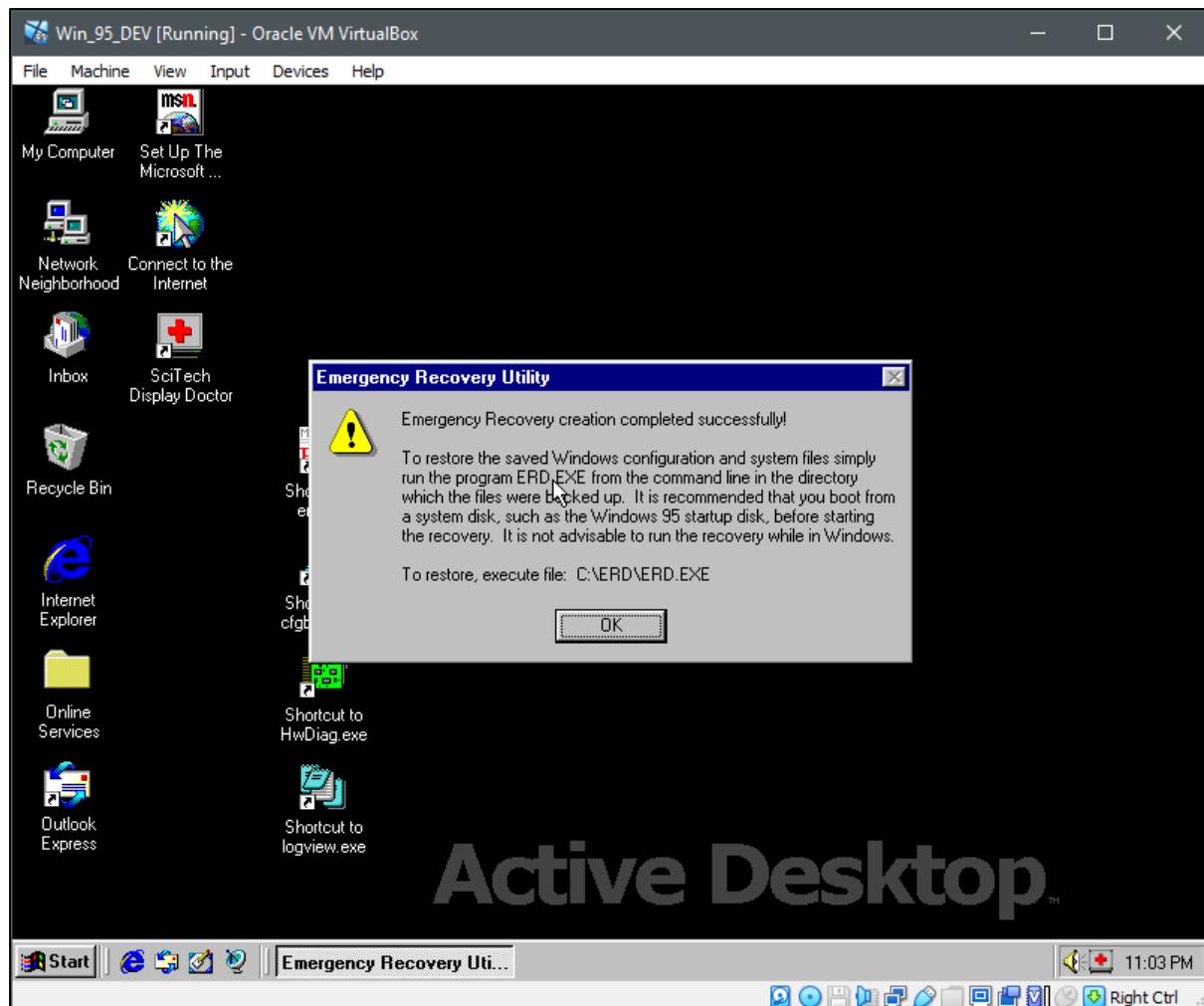
You can choose another location to store the recovery backup if you want. I find the ERD directory easy to find. It is also possible to make dated recovery backups by creating dated directories inside of ERD.



You will next be presented with the list of system and configuration files that will be saved. Click next to continue.

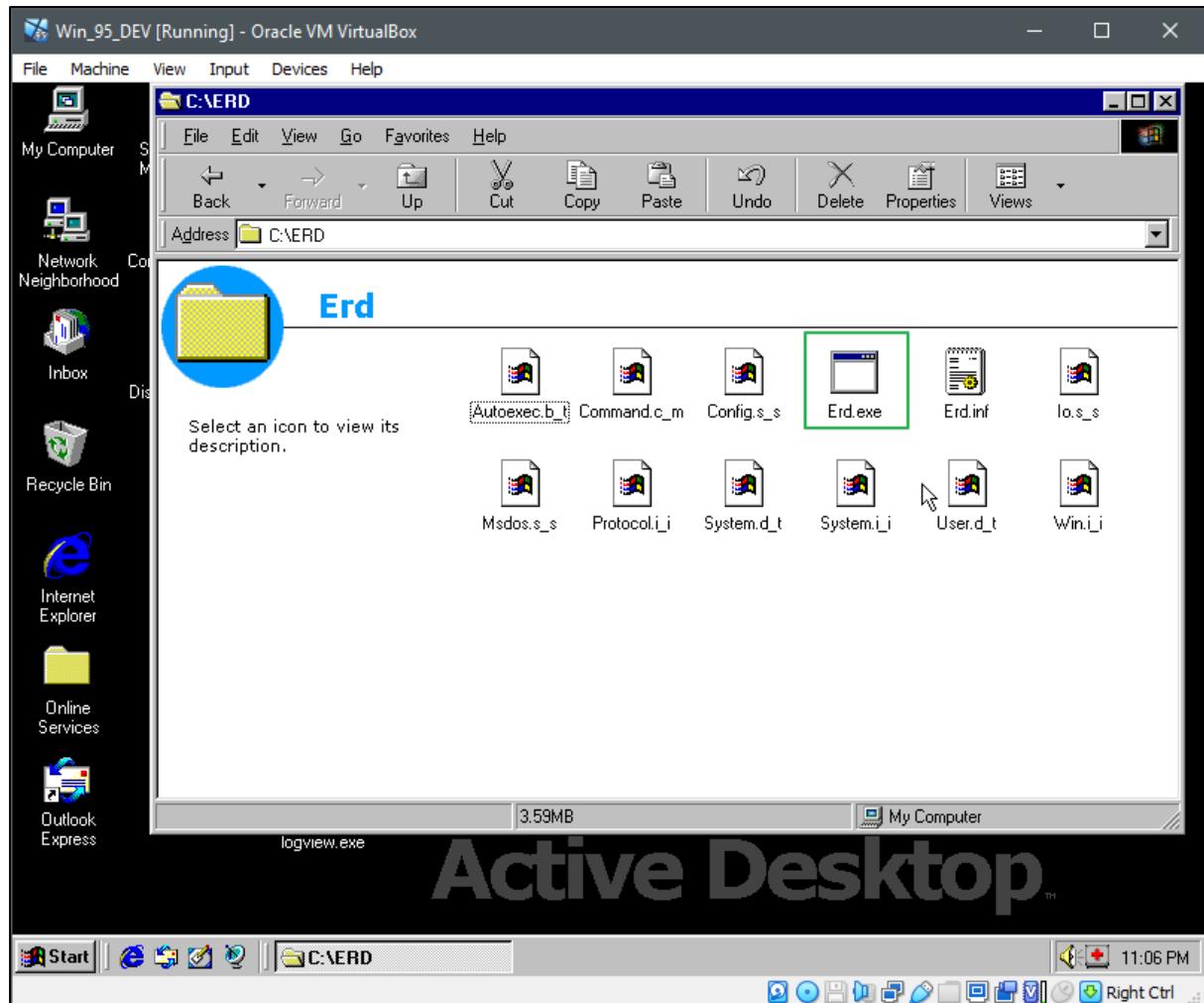


The recovery backup has been completed.



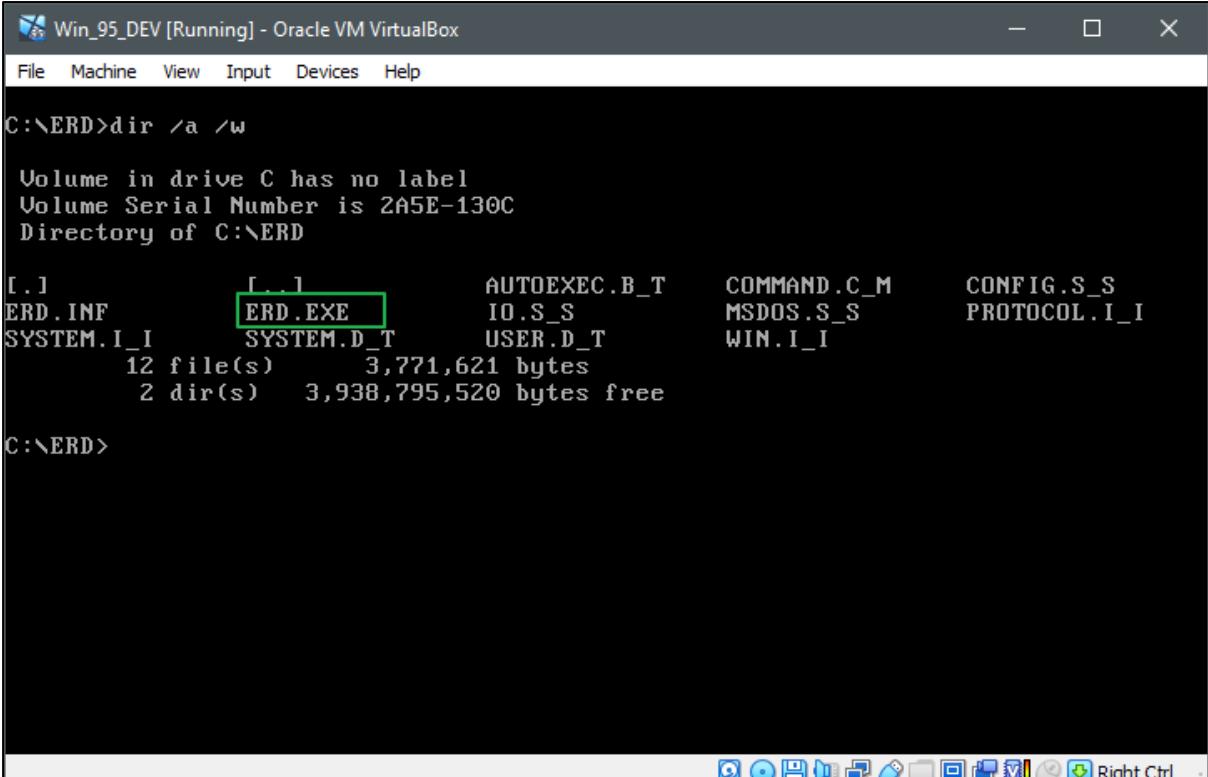
To recover an unbootable system, boot the system using a DOS start up disk and navigate to the ERD.EXE to restore the saved version of the backup files. Do not attempt to recover the files while running windows.

## A Beginners Guide To DOS Programming



Boot to DOS.

## A Beginners Guide To DOS Programming



```
Win_95_DEV [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

C:\ERD>dir /a /w

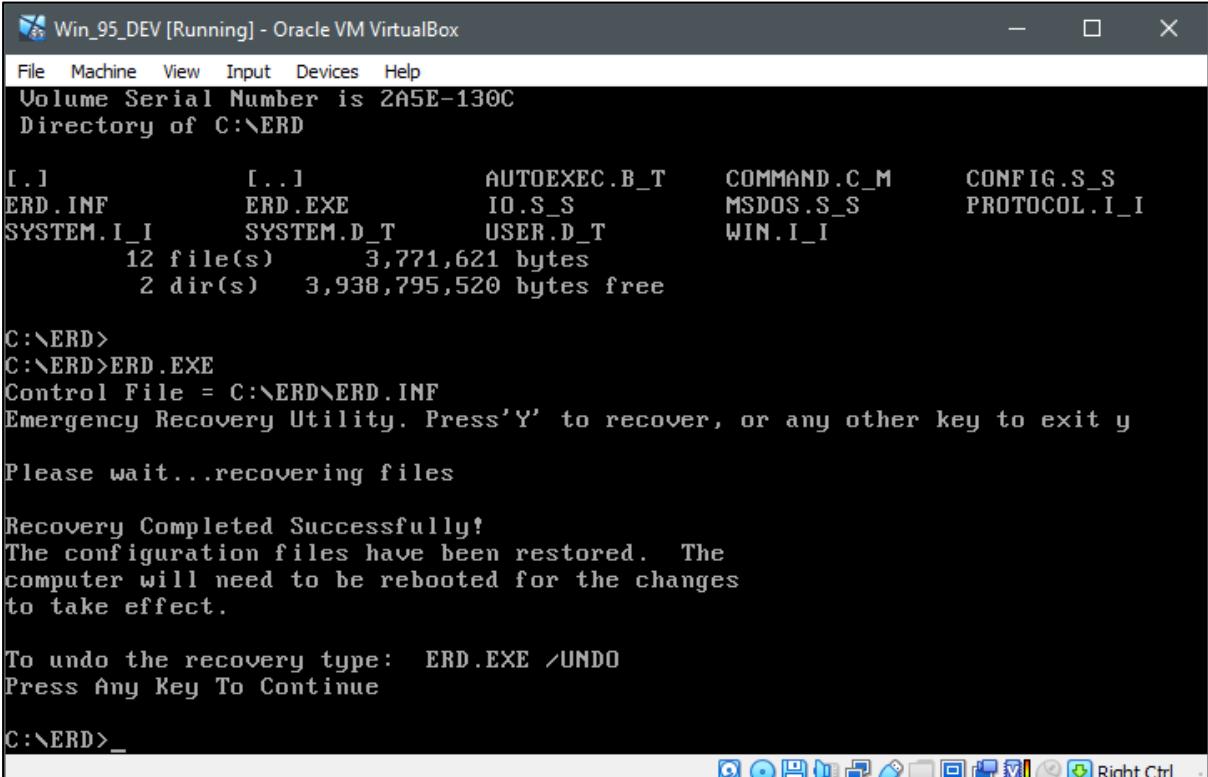
Volume in drive C has no label
Volume Serial Number is 2A5E-130C
Directory of C:\ERD

[.]      [...]      AUTOEXEC.B_T      COMMAND.C_M      CONFIG.S_S
ERD.INF    ERD.EXE      IO.S_S      MSDOS.S_S      PROTOCOL.I_I
SYSTEM.I_I   SYSTEM.D_T      USER.D_T      WIN.I_I

12 file(s)   3,771,621 bytes
2 dir(s)   3,938,795,520 bytes free

C:\ERD>
```

Run ERD.EXE



```
Win_95_DEV [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Volume Serial Number is 2A5E-130C
Directory of C:\ERD

[.]      [...]      AUTOEXEC.B_T      COMMAND.C_M      CONFIG.S_S
ERD.INF    ERD.EXE      IO.S_S      MSDOS.S_S      PROTOCOL.I_I
SYSTEM.I_I   SYSTEM.D_T      USER.D_T      WIN.I_I

12 file(s)   3,771,621 bytes
2 dir(s)   3,938,795,520 bytes free

C:\ERD>
C:\ERD>ERD.EXE
Control File = C:\ERD\ERD.INF
Emergency Recovery Utility. Press 'Y' to recover, or any other key to exit y
Please wait...recovering files

Recovery Completed Successfully!
The configuration files have been restored. The
computer will need to be rebooted for the changes
to take effect.

To undo the recovery type: ERD.EXE /UNDO
Press Any Key To Continue

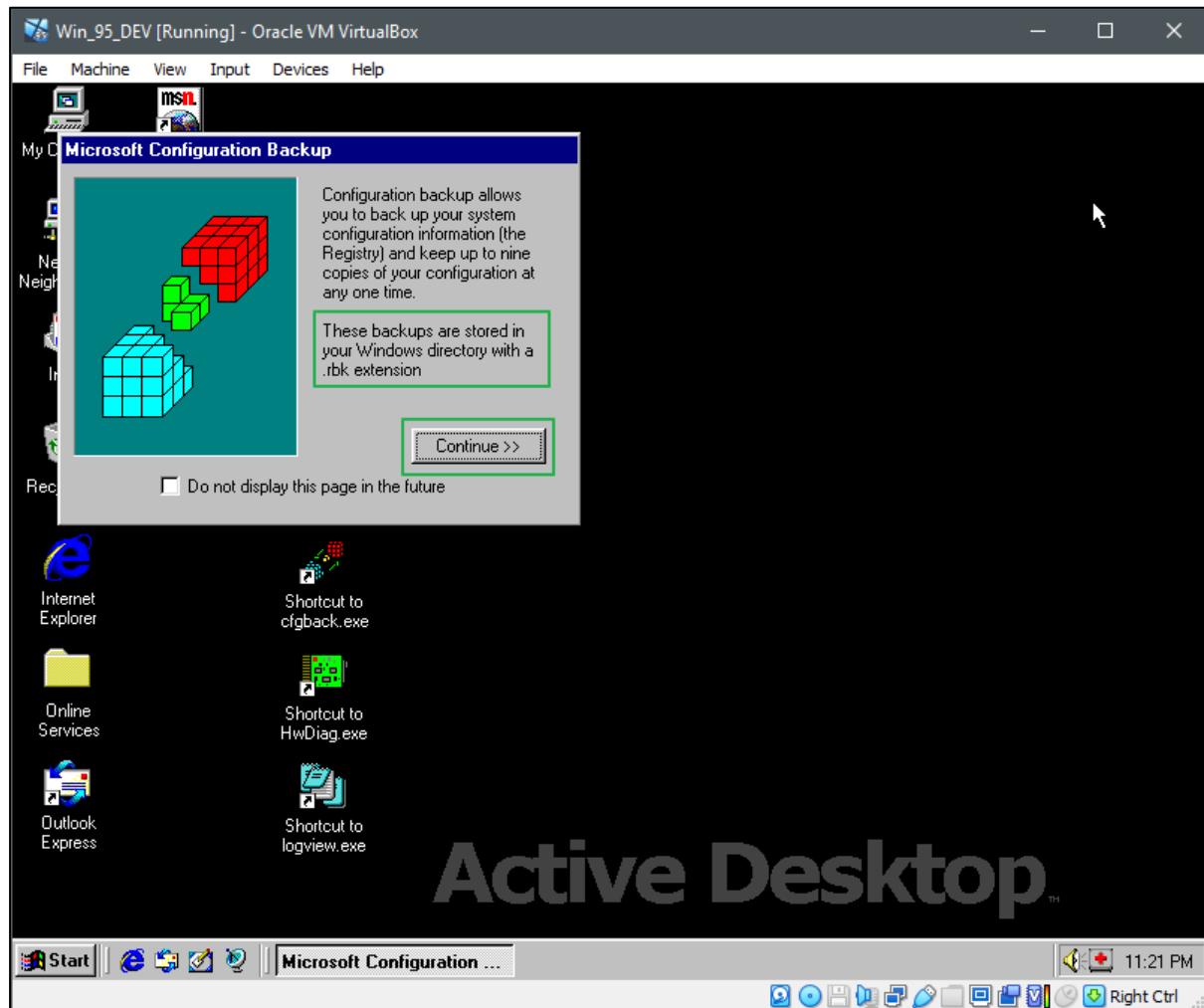
C:\ERD>_
```

The emergency recovery has completed. You can now try and reboot Windows 95.

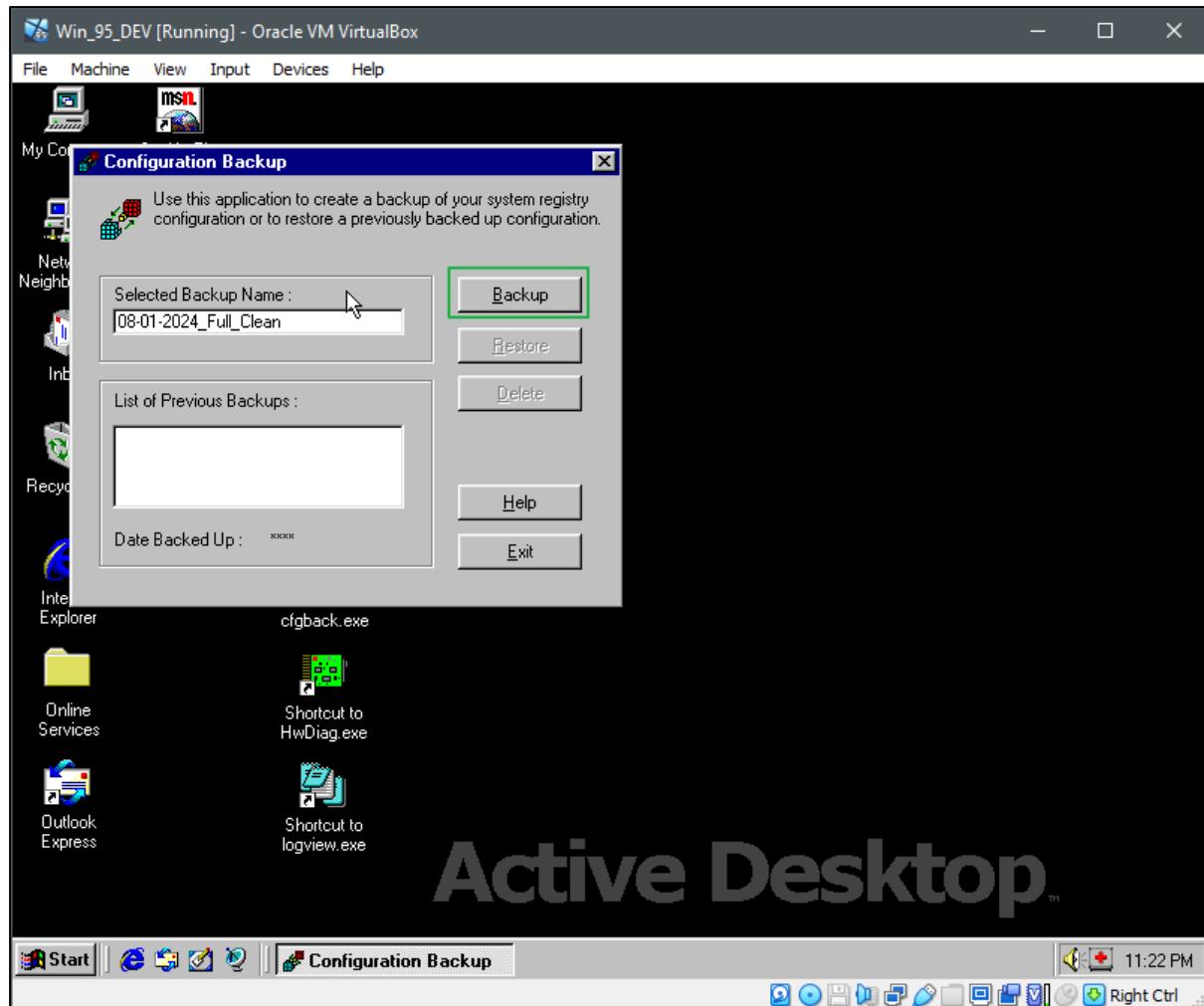
### CFGBack

Configuration Backup is a Windows utility used to back up and restore the system registry. This is similar to the ERU/ERD application but focuses entirely on the system registry.

Run the cfgback.exe

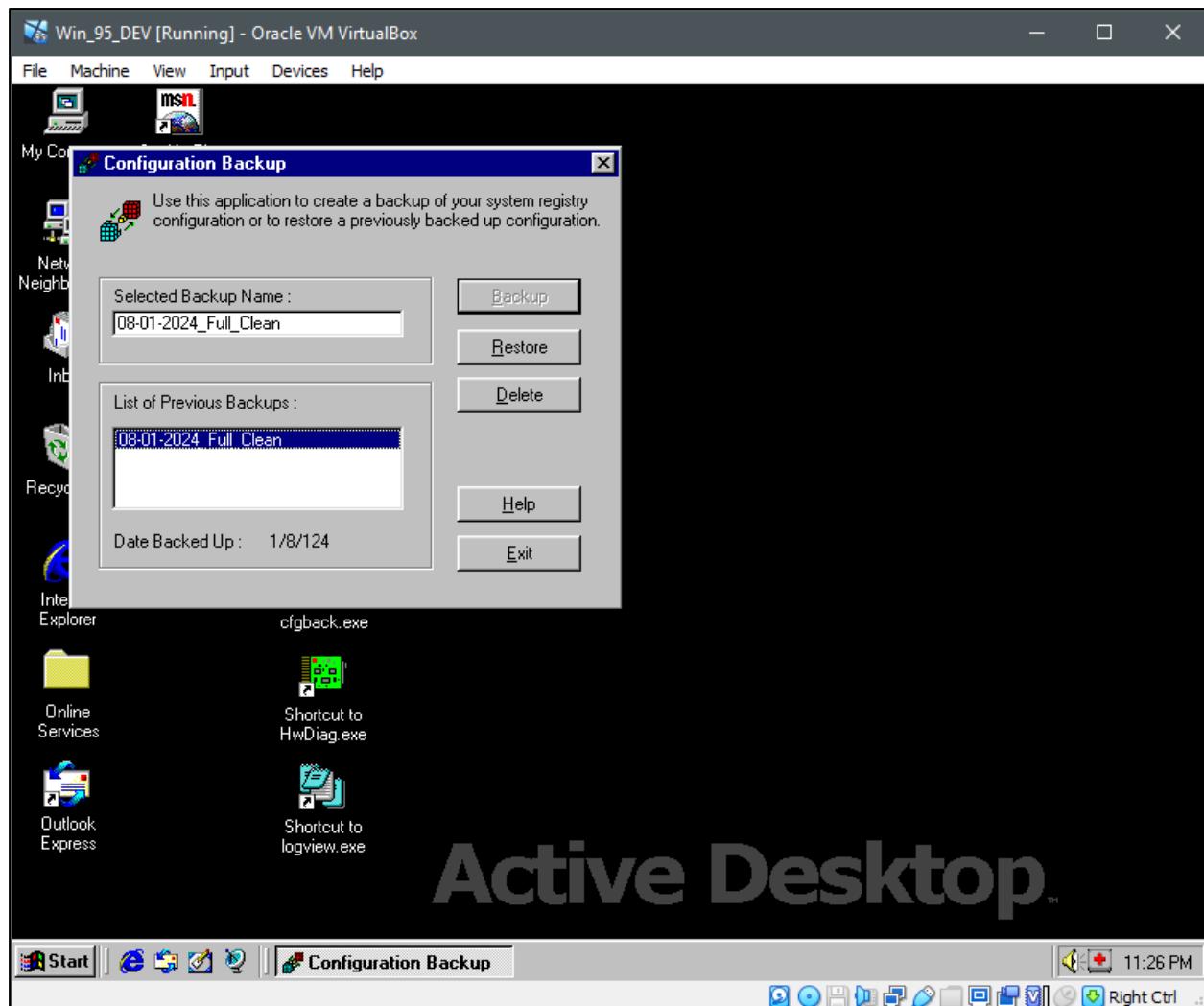


Select a backup name and then click Backup.



The backup is now created. To restore a backup select from the available backups and click restore.

Note that I would recommend limiting this to the most recent backup and use only for disaster recovery.



### System Clean-up

Windows 95 does not have any built in drive clean-up utilities like those that are found in later versions of Windows, so in many ways you are on your own with system clean-up and maintenance. Some people say that you can move the windows 98 Cleanmgr.exe application to Windows 95 but I have not tested this.

Some early version of other clean up utilities such as CCleaner are said to also work. I have found the following 2 tools do a reasonable job. Please take care using the RegClean utility as it can sometimes remove required registry entries, although this is rare.

Always make a backup of the current system before doing any registry or file clean-up tasks.

### Windows CleanUp! 4.5.2

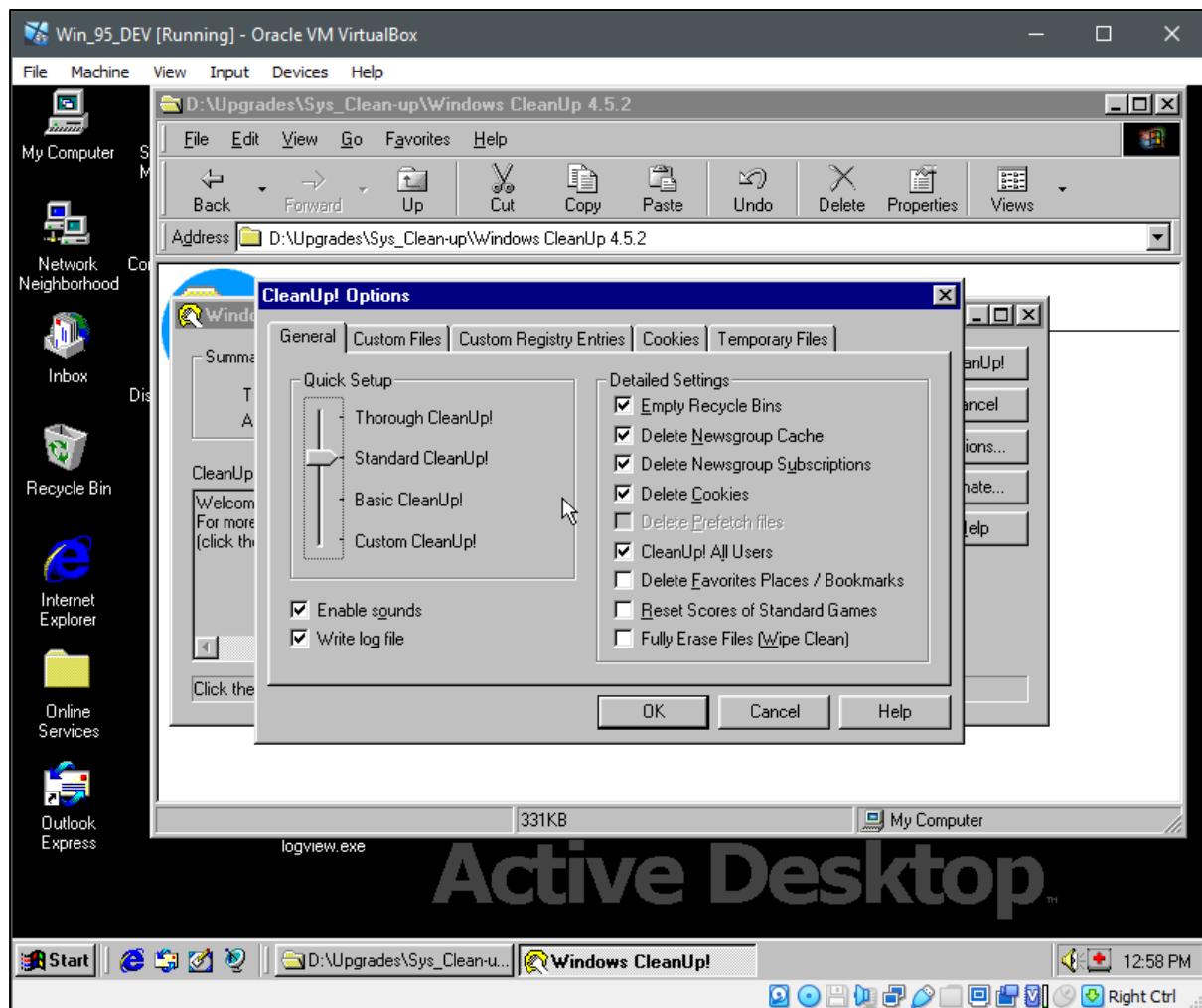
<http://cleanup.stevengould.org/>

“CleanUp452.exe”

Transfer the “CleanUp452.exe” to your Windows 95 system and install the application.

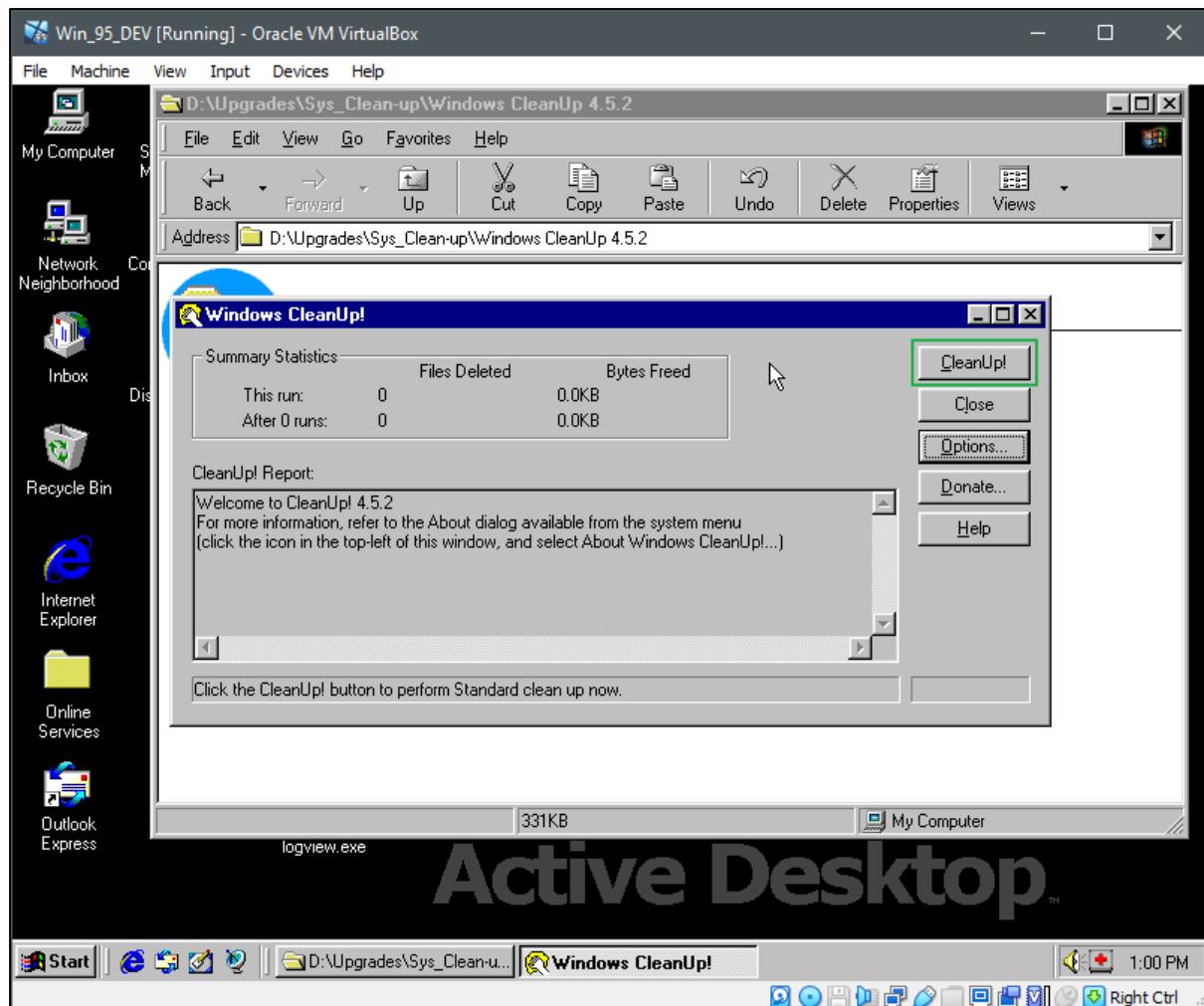
After the install run the application. CleanUp has many options that you can configure if needed. The default “Standard CleanUp!” will perform some basic cleaning except for some of the Windows Temp file locations. You may need to include these locations manually. When the application is first run it will do a demonstration, but not remove any files.

Open the “Options...” section and quickly become familiar with the available clean up options.

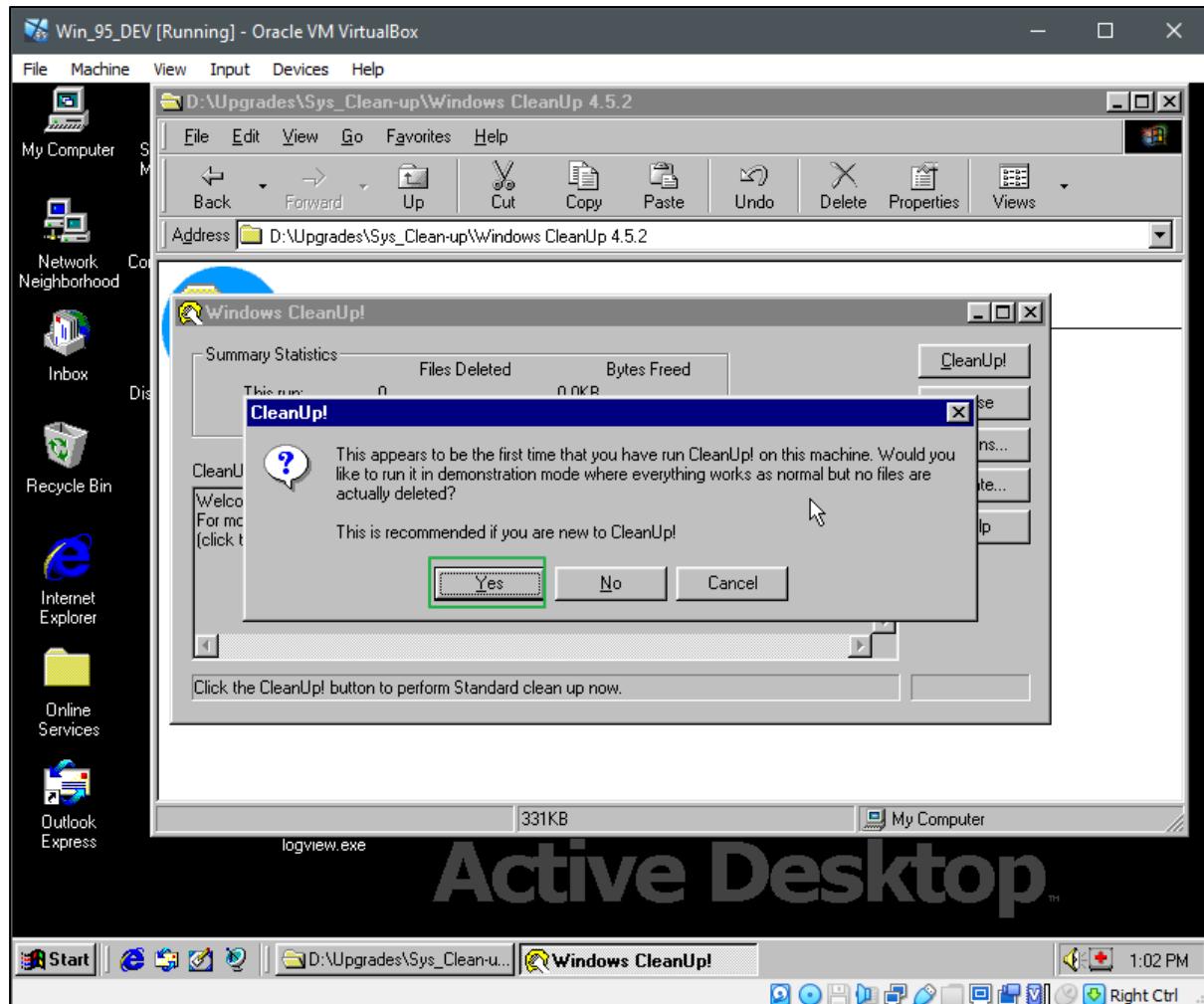


Next select “CleanUp!”

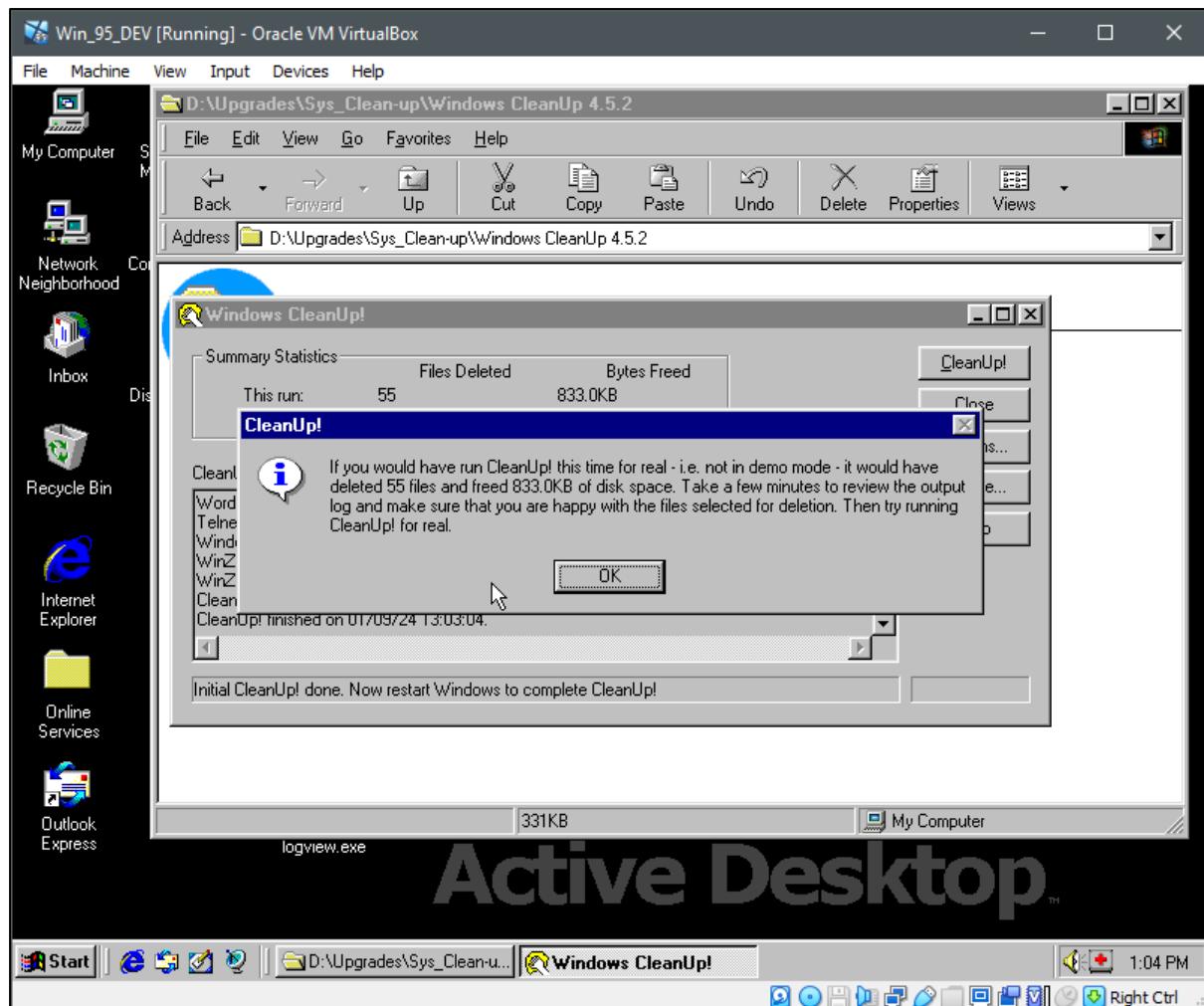
## A Beginners Guide To DOS Programming



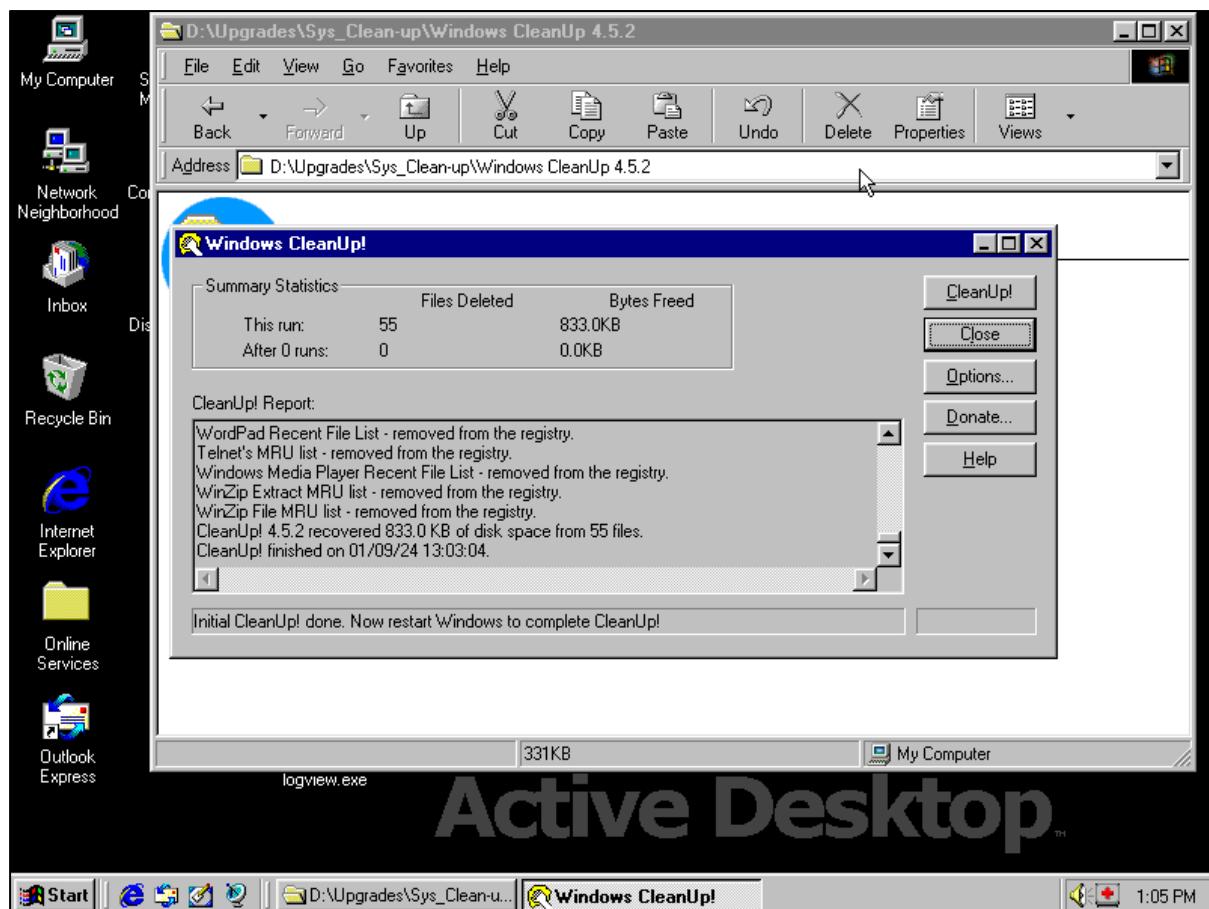
The choice for demonstration mode will show. Select yes to run in demonstration mode.



The demo mode gives you brief outline of what files would have been removed. You can use this to check what files are found for removal before deleting them.



The demo CleanUp report.



Close the application.

When you go to the start menu you will have the option to start the application in Demo mode or Real mode. The only difference is the demo mode starts the application with the Cleanup.exe /demo switch.

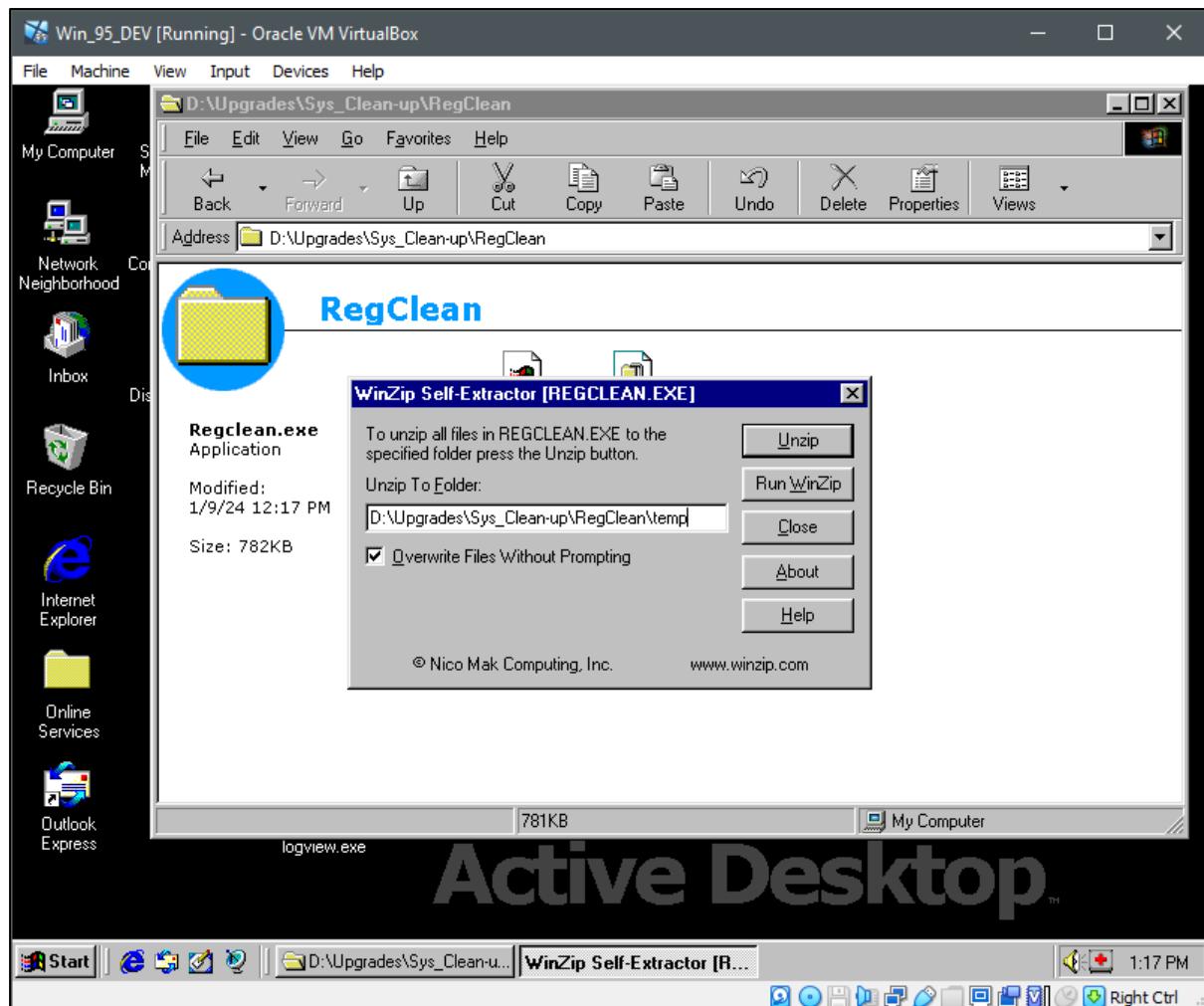
If you are happy with the file delete list in demo mode, then start in real mode and delete the files etc.

### Microsoft RegClean 4.1a Build 7364.1

[https://www.majorgeeks.com/files/details/microsoft\\_regclean.html](https://www.majorgeeks.com/files/details/microsoft_regclean.html)

Transfer the "Regclean.exe" archive to your Windows 95 drive.

Run the Regclean.exe and change the output to a temp directory in the current folder.



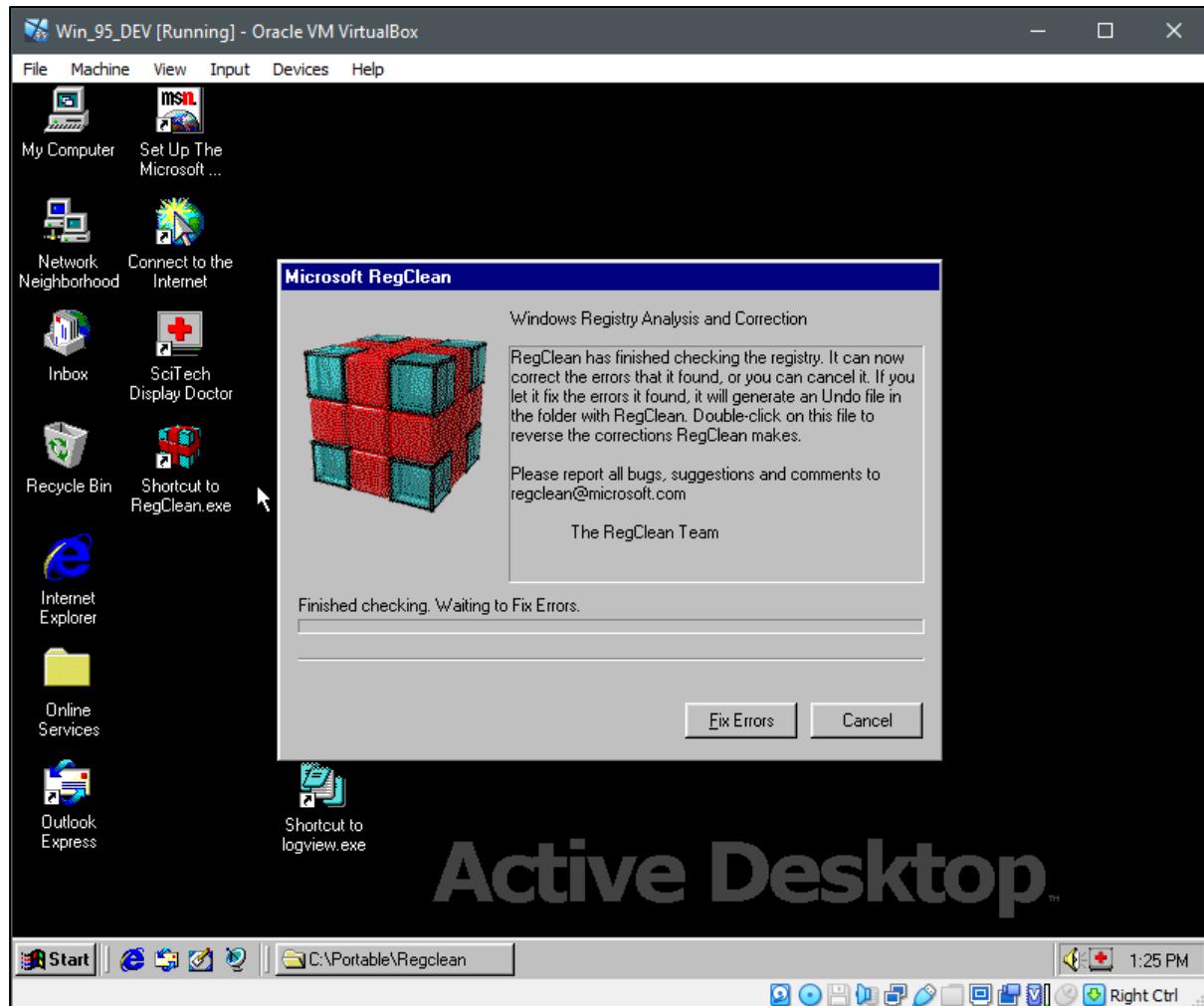
You will have a temp directory containing 3 files; The regclean.exe application, a help file and a system update if needed. Rename the temp file to Regclean and then copy the directory and files to the portable application folder in the C: drive.

C:\Portable\Regclean\regclean.exe

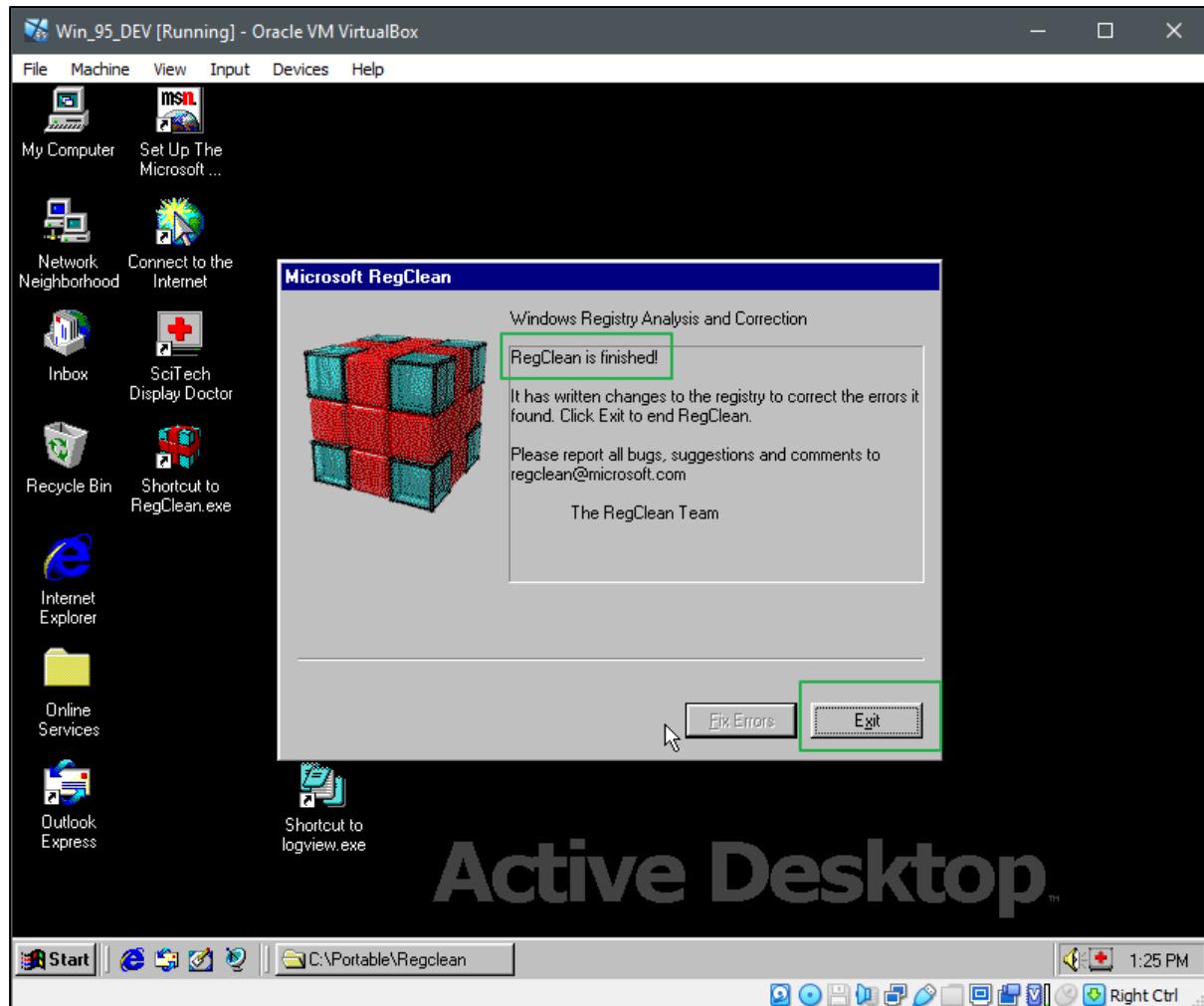
Create a shortcut to the Windows desktop with "Send To".

Run the RegClean application.

It will scan for errors but does not give a report of what entries will be deleted. Don't worry, the application will create a backup of the registry entries in the application folder. Select Fix Errors to continue.

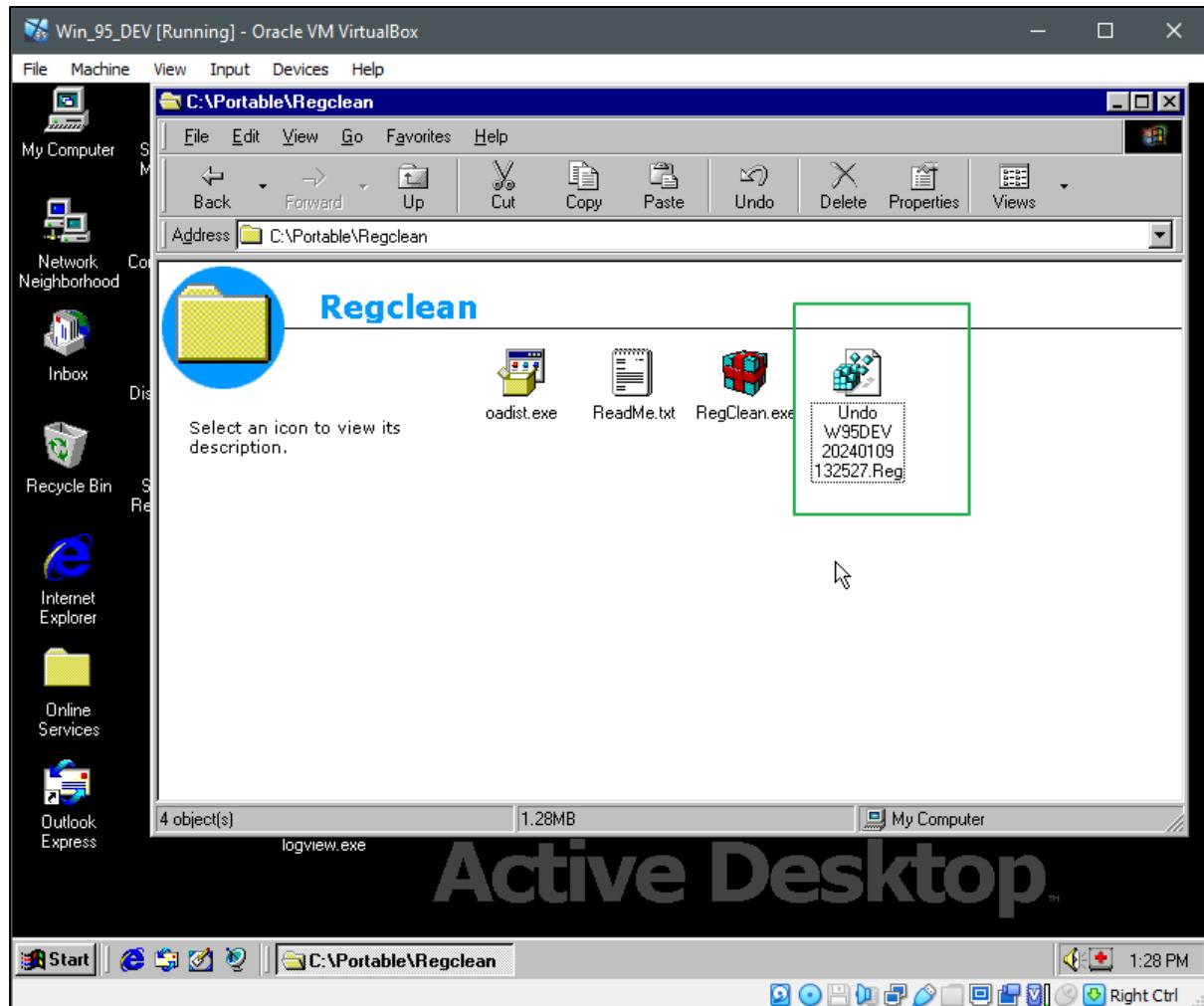


When finished you can Exit the application.

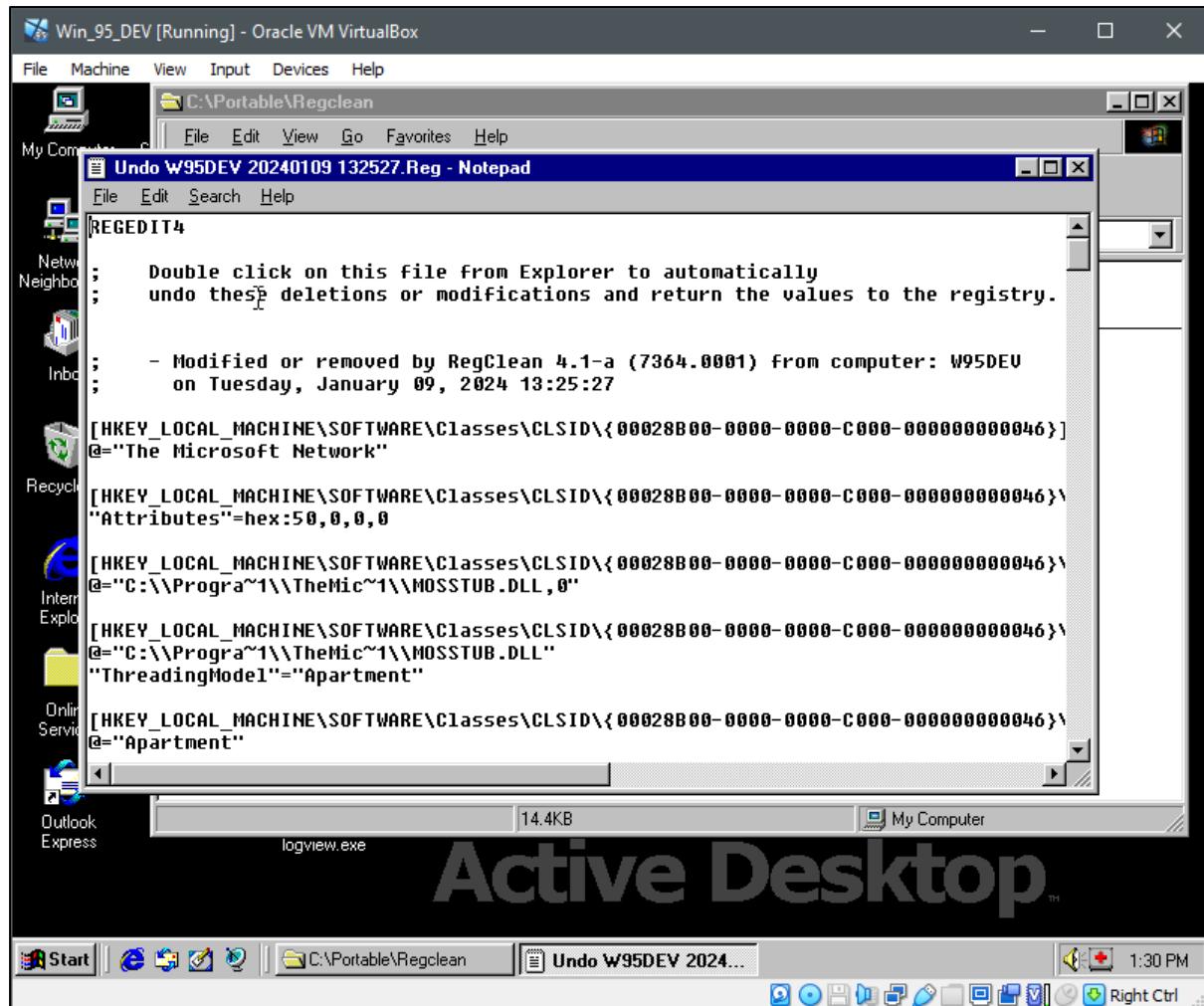


Navigate to the \Regclean application folder. You will find the registry undo/backup entry with a timestamp.

## A Beginners Guide To DOS Programming

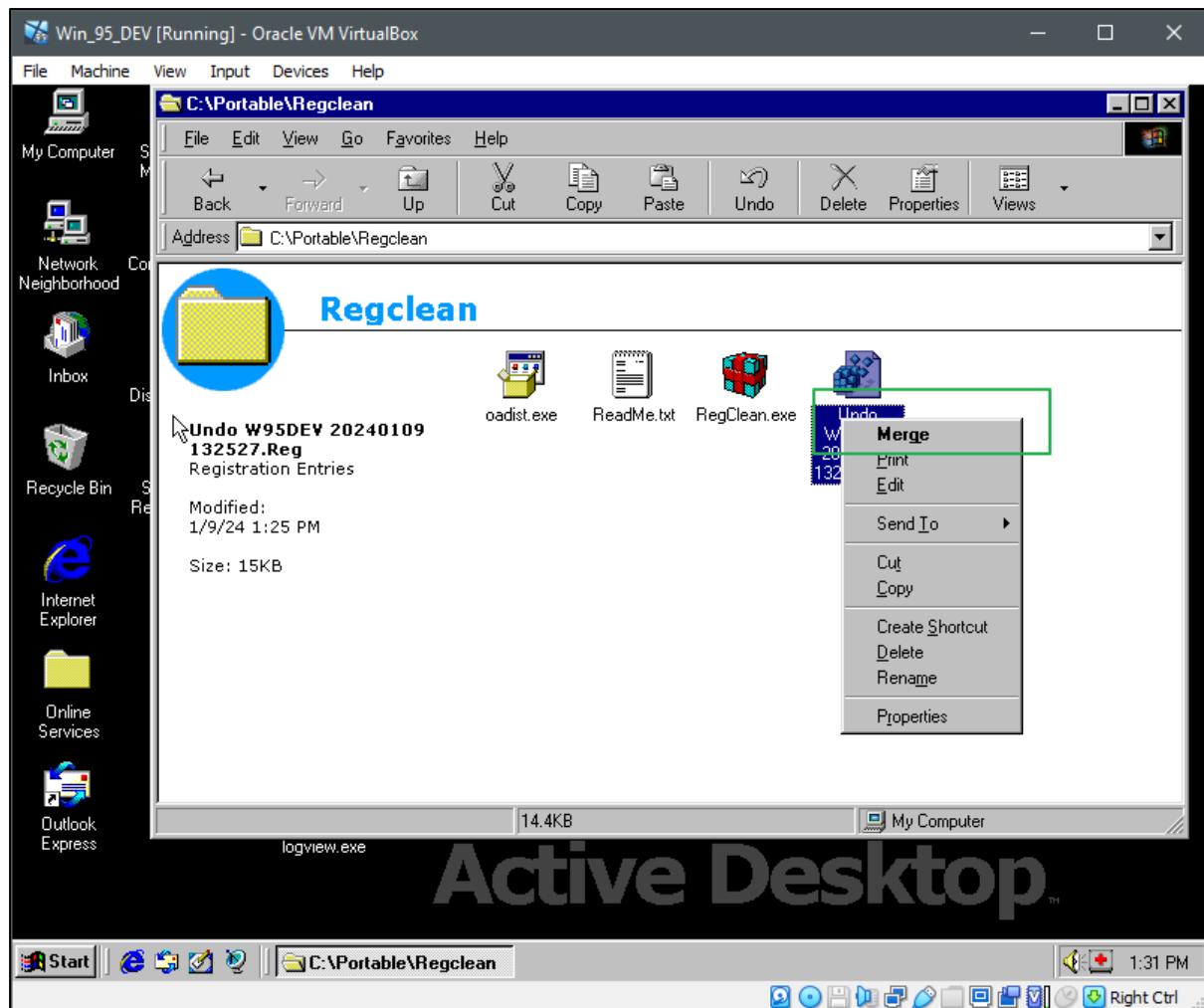


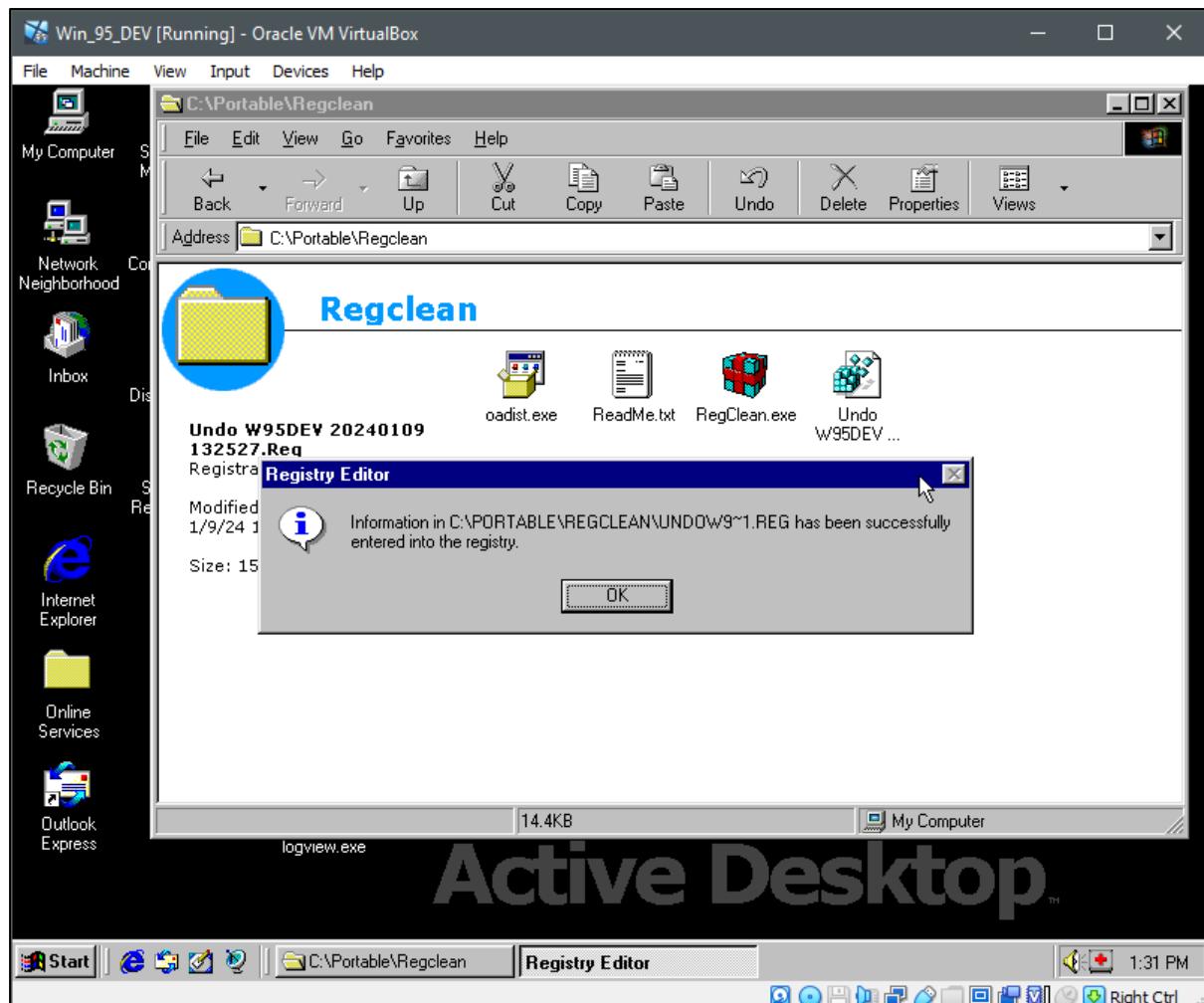
You can open the backup in a text editor to assess what entries were removed.



If you think that there was an issue you can restore the registry entries using the **Merge** option.

## A Beginners Guide To DOS Programming





Once you know the basic header format for a .reg text file it is quite easy to create a separate .reg file and copy over only the individual registry entries that you want to restore. There are a number of good “Text colouring/highlighting” text editors such as Notepad++ that can be useful for this.

### Manual cleaning.

Windows 95 keeps its primary temporary files in C:\WINDOWS\TEMP\\*.\*

There are other temporary directories such as msdownload.tmp and some application installers can use directories such as C:\TEMP

You can add some of these paths to the CleanUp application above if that is convenient, but please be very careful as you may delete important files or IE temporary system files.

Always double check what files are in the directory before deleting them.

### System tools

Some system tools are useful in helping to manage our Windows 95 Environment. There are many utilities and tools listed in the Windows 95 software archives that you can try but I find the following small list the most useful.

HwDiag.exe (Already copied from the Windows 95 install CD)

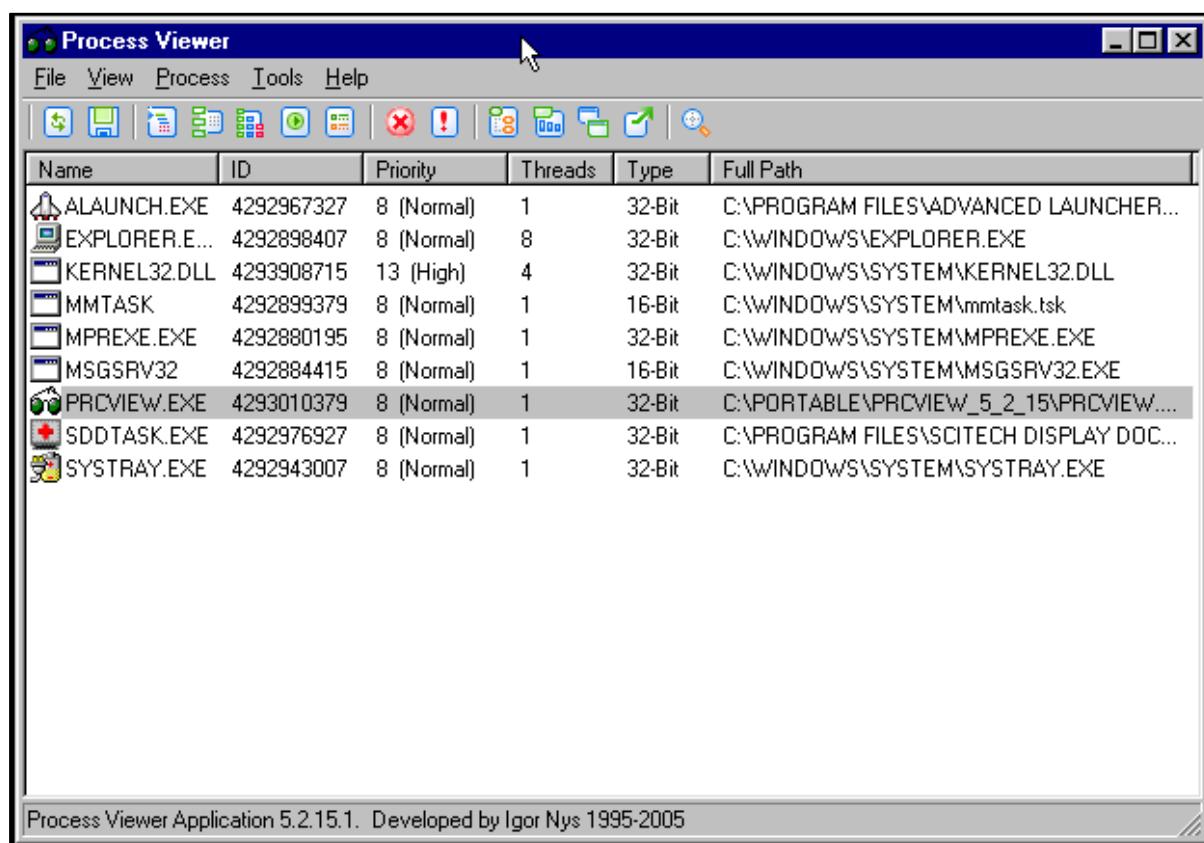
logview.exe (Already copied from the Windows 95 install CD)

### PrcView v5.2.15.1 (see also V 3.x.x)

PrcView allows you to view and manage processes running on your computer. It offers a lot of advanced capability not found in the Windows Task Manager.

<https://www.portablefreeware.com/index.php?id=406>

Unpack the PrcView\_5\_2\_15.zip archive and copy the unpacked directory to your C:\Portable directory. Create a shortcut on the desktop to PrcView.exe or place it on you ALaunch directory.



### Windows 95 Power Toys

A collection of useful helper application. You can install all of the applications from the installer, or unpack the installer archive and install individual components using the correct Name.inf file. An .inf

file is just a special Windows install script. I would recommend using the newer TweakUI v1.33 over the version that exists in the PowerToys archive.

“W95PowerToy.exe”

“tweakui\_v133.exe”

### **Windows 95 Kernel Toys**

A collection of useful helper applications. Upack the self extracting archive using 7-zip. Install individual components using the correct “Name.inf” file. An .inf file is just a special Windows install script.

“W95KrnIToys.exe”

You may have to hunt around on the internet to find the above 3 installer files as they are no longer stored on the Microsoft public servers.

[https://math.vanderbilt.edu/schectex/wincd\\_files/local\\_archive/powertoys/](https://math.vanderbilt.edu/schectex/wincd_files/local_archive/powertoys/)

These are a collection of useful tools for Windows 95 management tasks. TweakUI V1.33 is an updated version of the application found in the PowerToys installer.

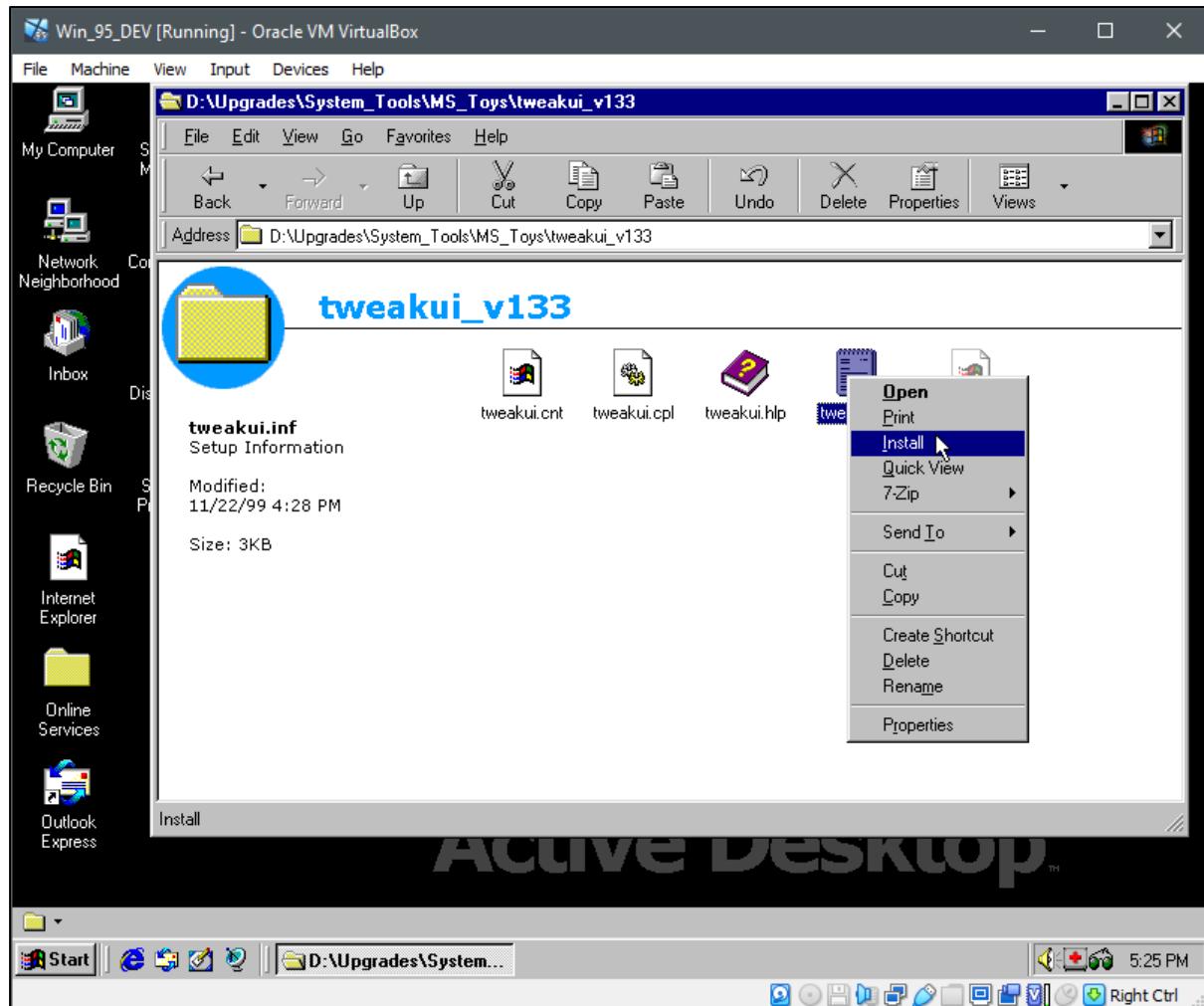
### **Installing the power toys**

I am only going to make use of (Install) a small number of the available power toys. You can read up on the documentation and install other components if you wish. You can also come back and install other components at a later time.

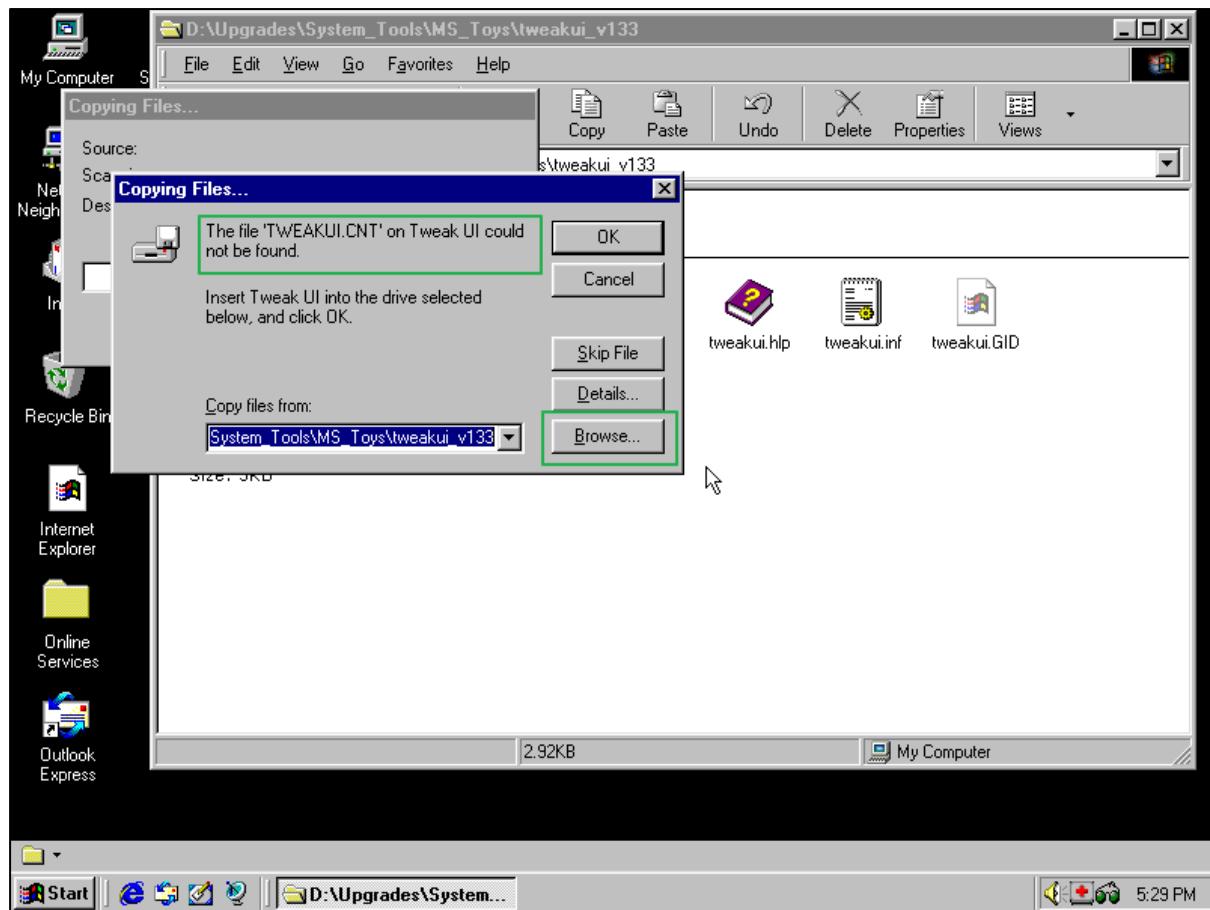
#### TweakUI V.133

Unpack the “tweakui\_v133.exe” archive, and locate the tweakui.inf file.

Right click on the INF file and select Install. This process will be the same for all other power toys.

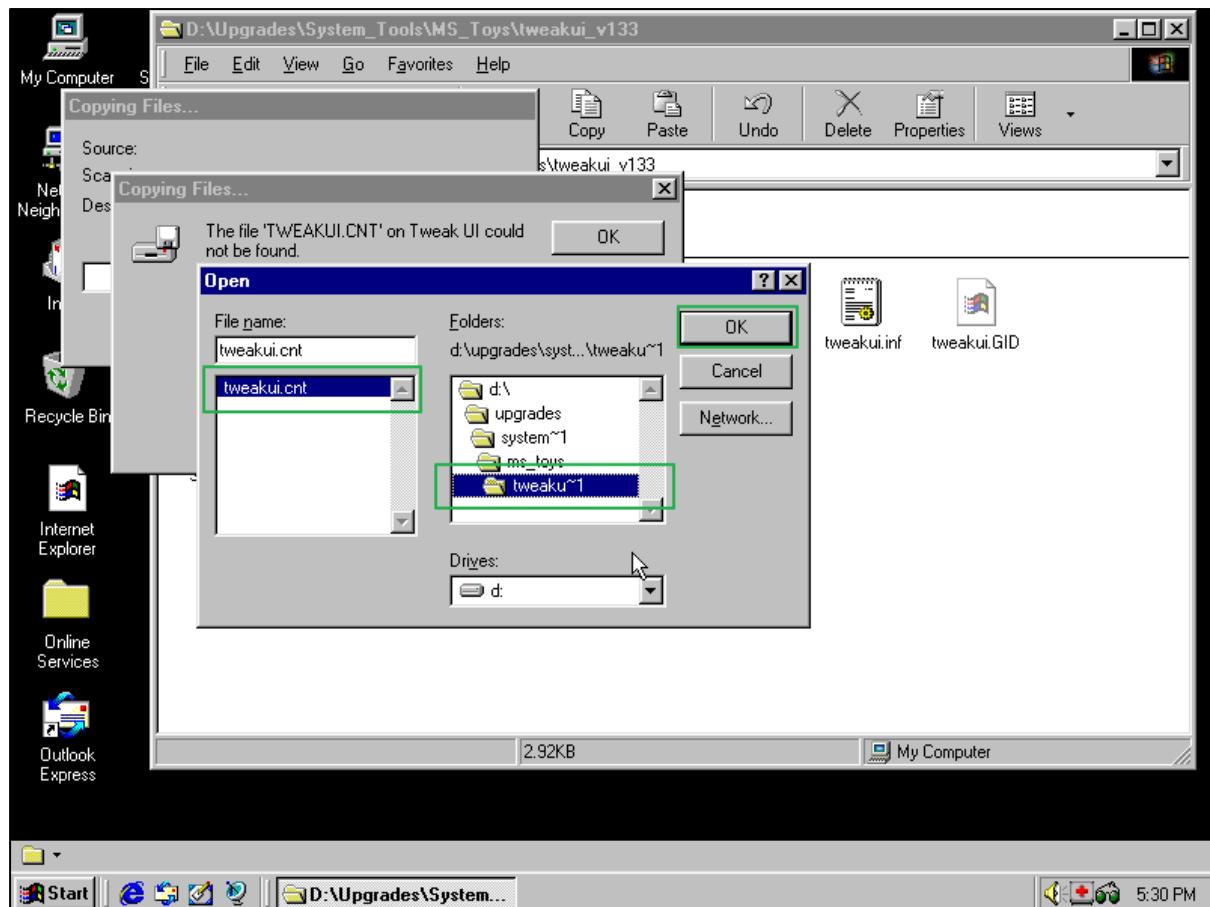


If you are met with a can't find file etc. Use the brows button to navigate to where the file is. The path shown will be correct but this is a common Windows 95 problem.

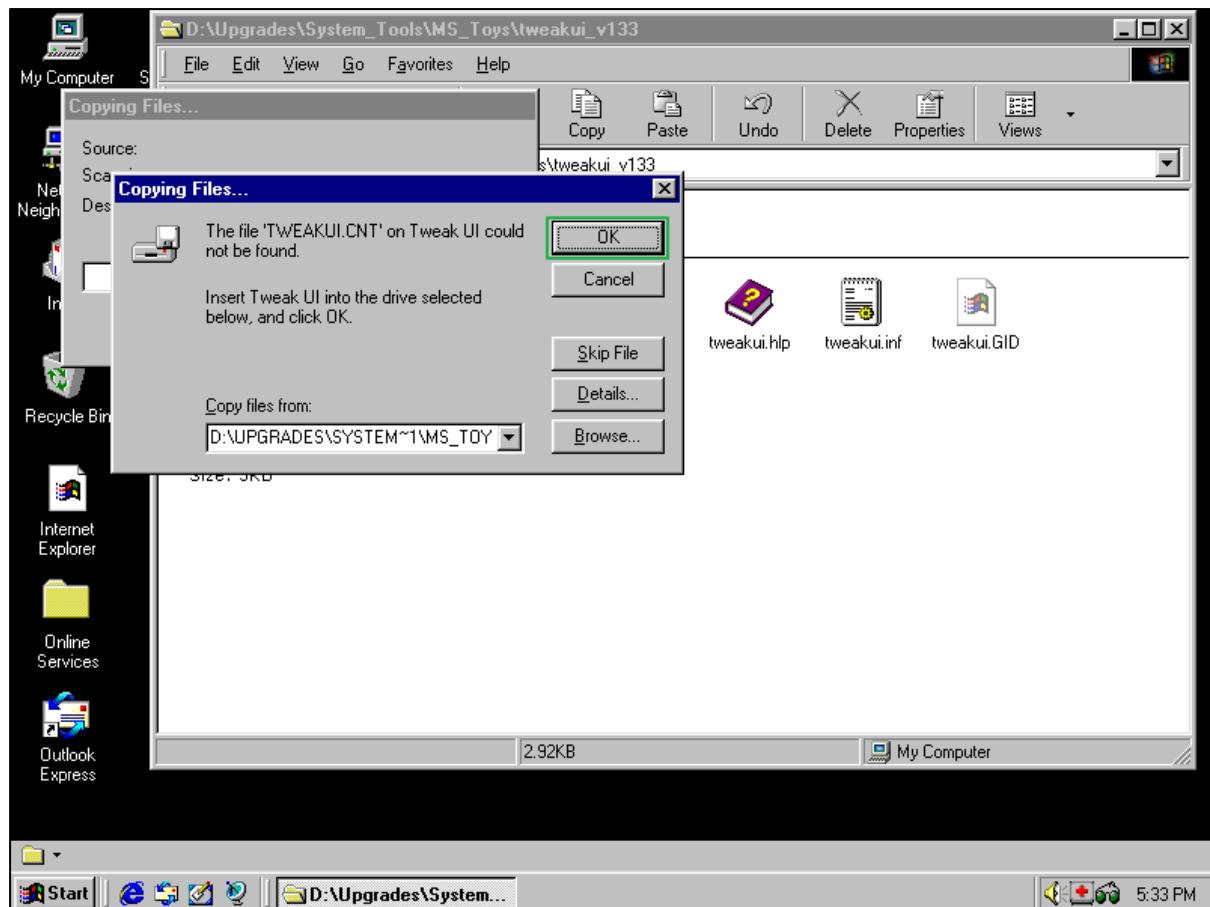


After you click Browse it will likely show the correct directory and the missing file. Just click OK if it is correct.

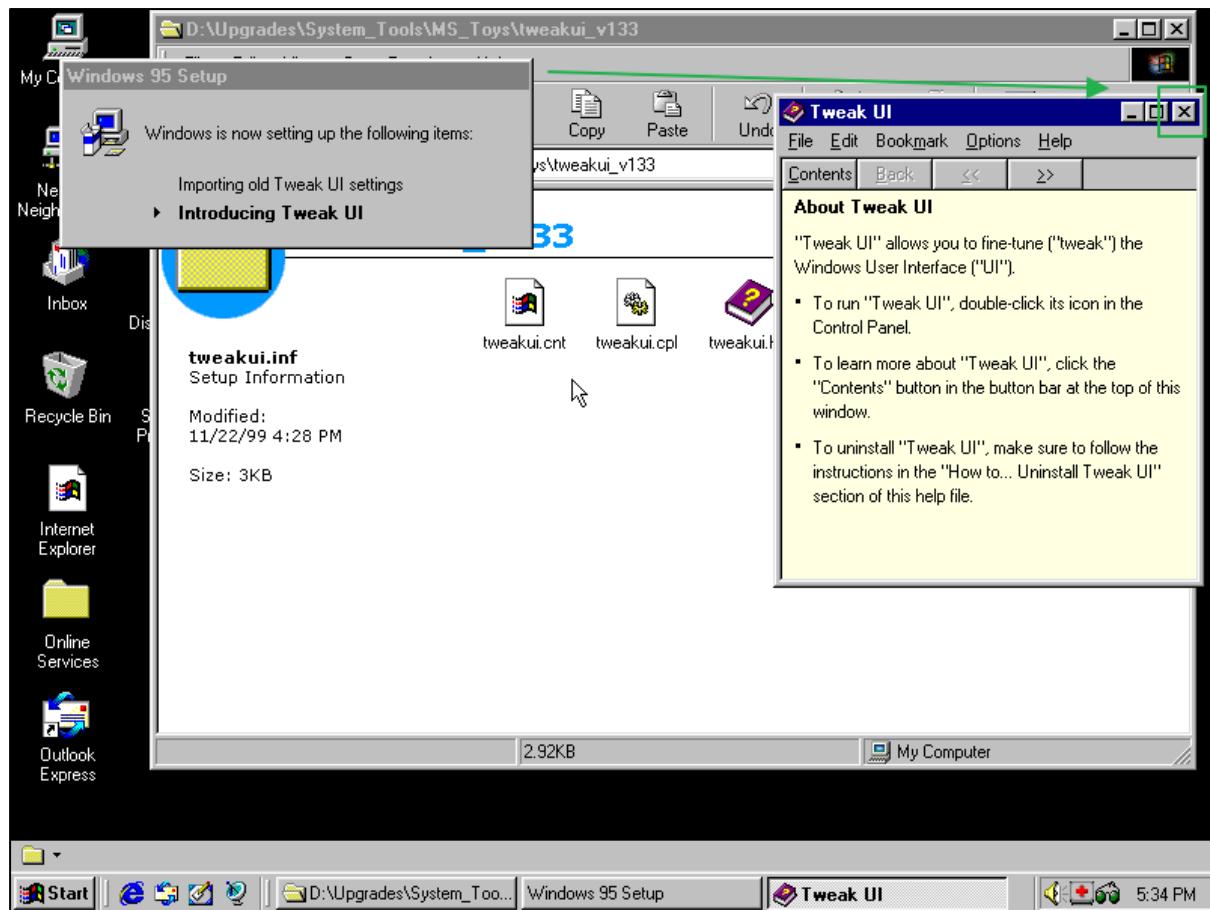
## A Beginners Guide To DOS Programming



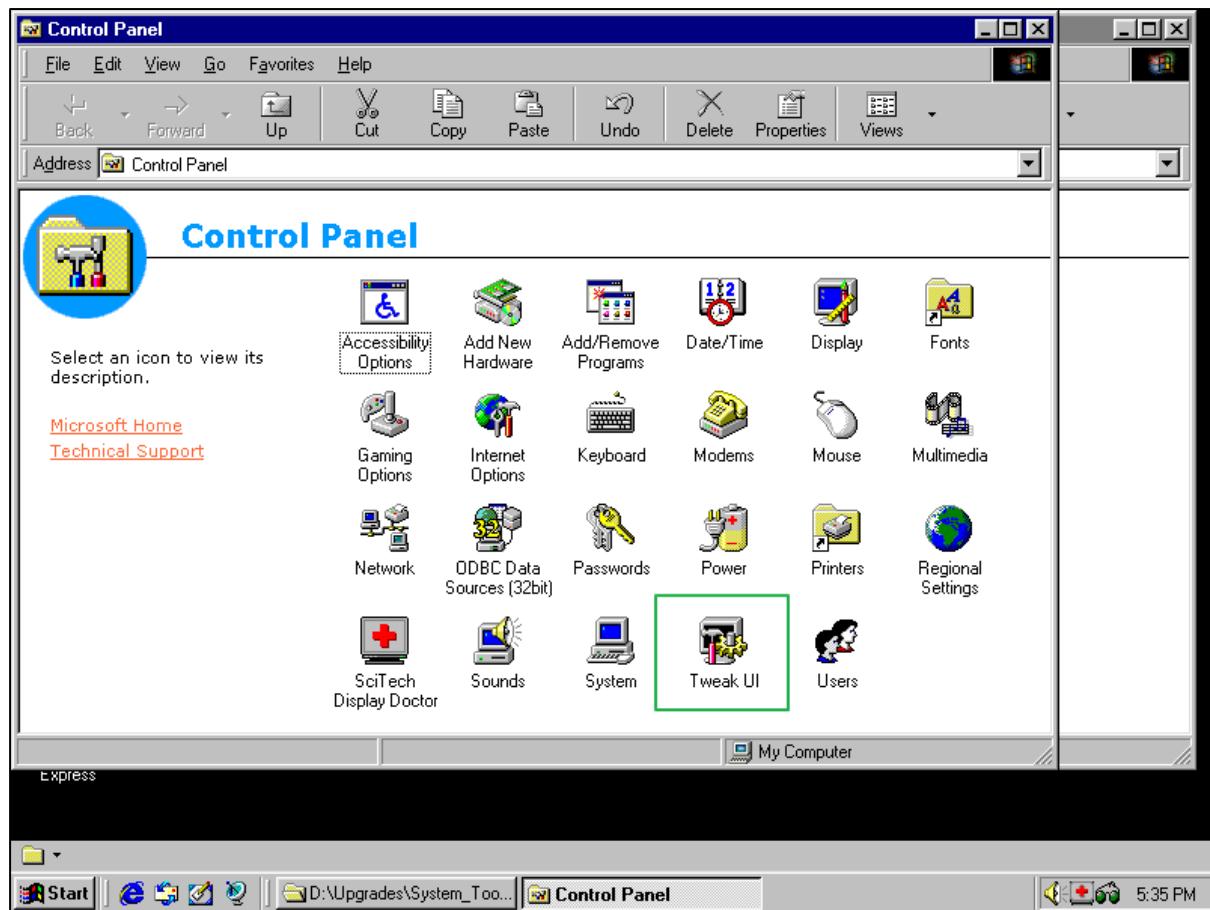
Then OK to continue installing the missing file.



Now the install completes. Close the help manual to complete the setup task.



Navigate to the Windows 95 Control panel and the TweakUI icon will be there.



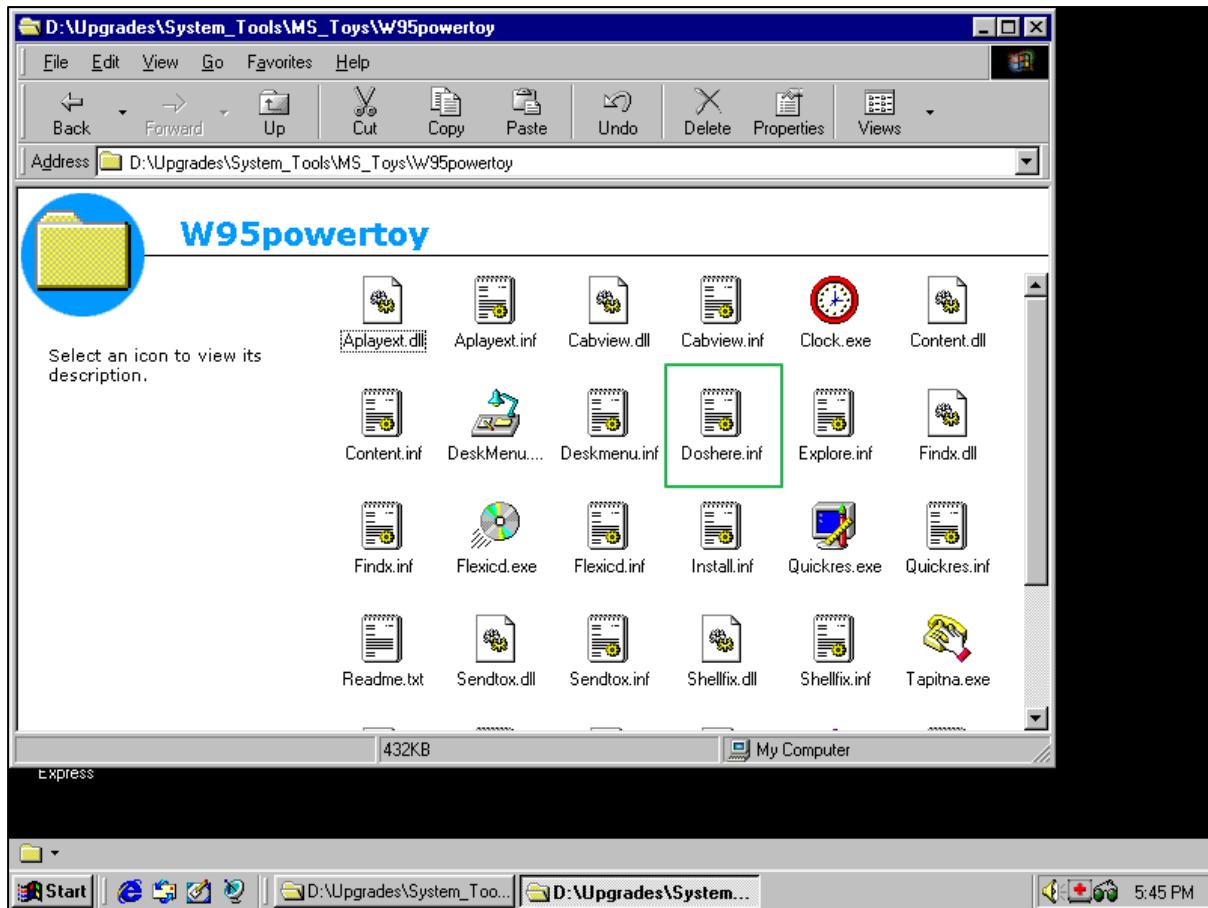
TweakUI gives access to many of the systems hidden settings without going into the registry.



Note if you right click on the TweakUI icon in the control panel you can send a shortcut to the desktop or use it in ALaunch.

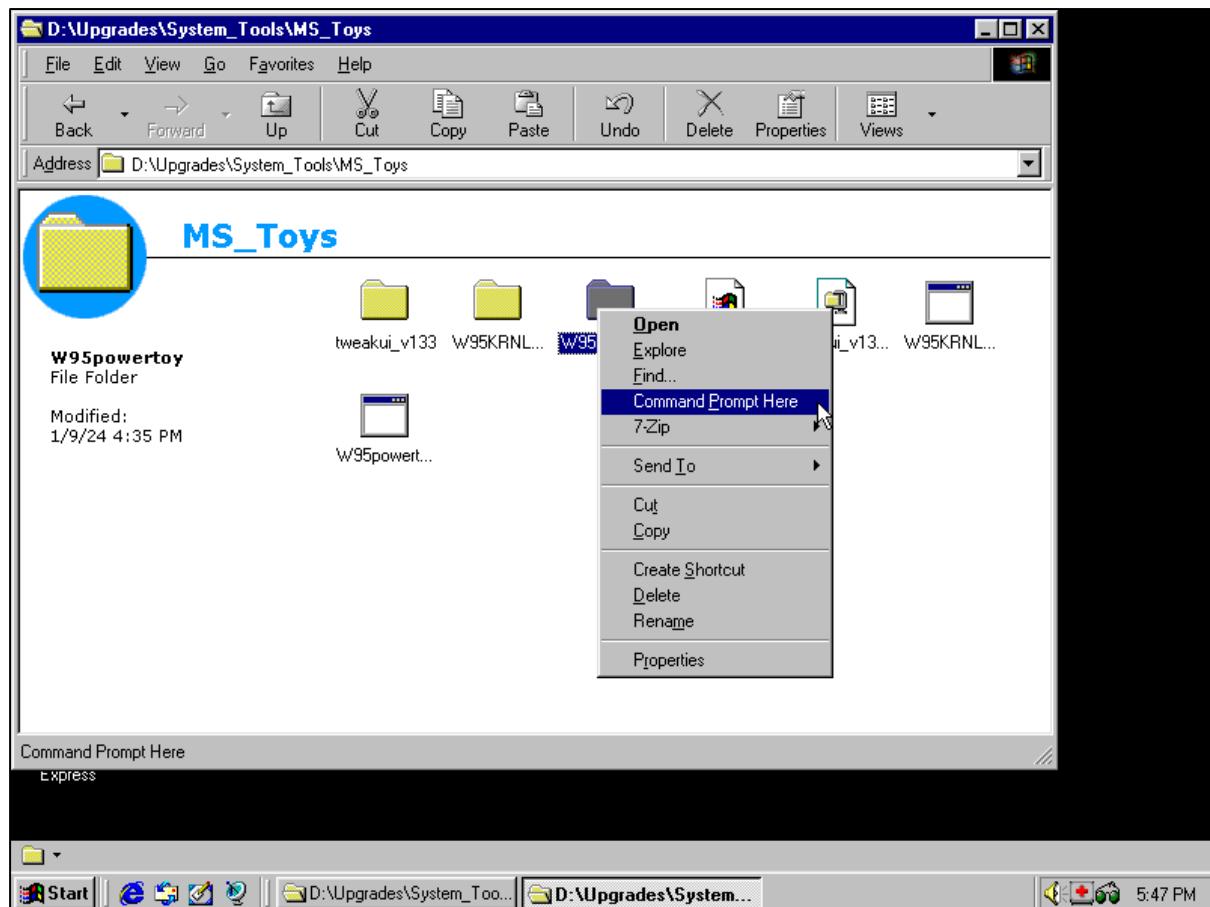
The next toy that I recommend is “Command Prompt Here”. This will allow you to open the windows command console from and folder location in Explorer.

Navigate to the W95powertoy directory and Install the “Doshere.inf”.



Navigate to the missing file as shown in the previous install for TweakUI.

Now when you right click on a folder you will have the option to open the command interpreter in that directory.



## Applications

### IrfanView 3.55 (4.44)

Universal image viewing application.

Requires "vc6redistsetup\_enu.exe" Visual C(++) runtime files.

"iview435\_setup.exe"

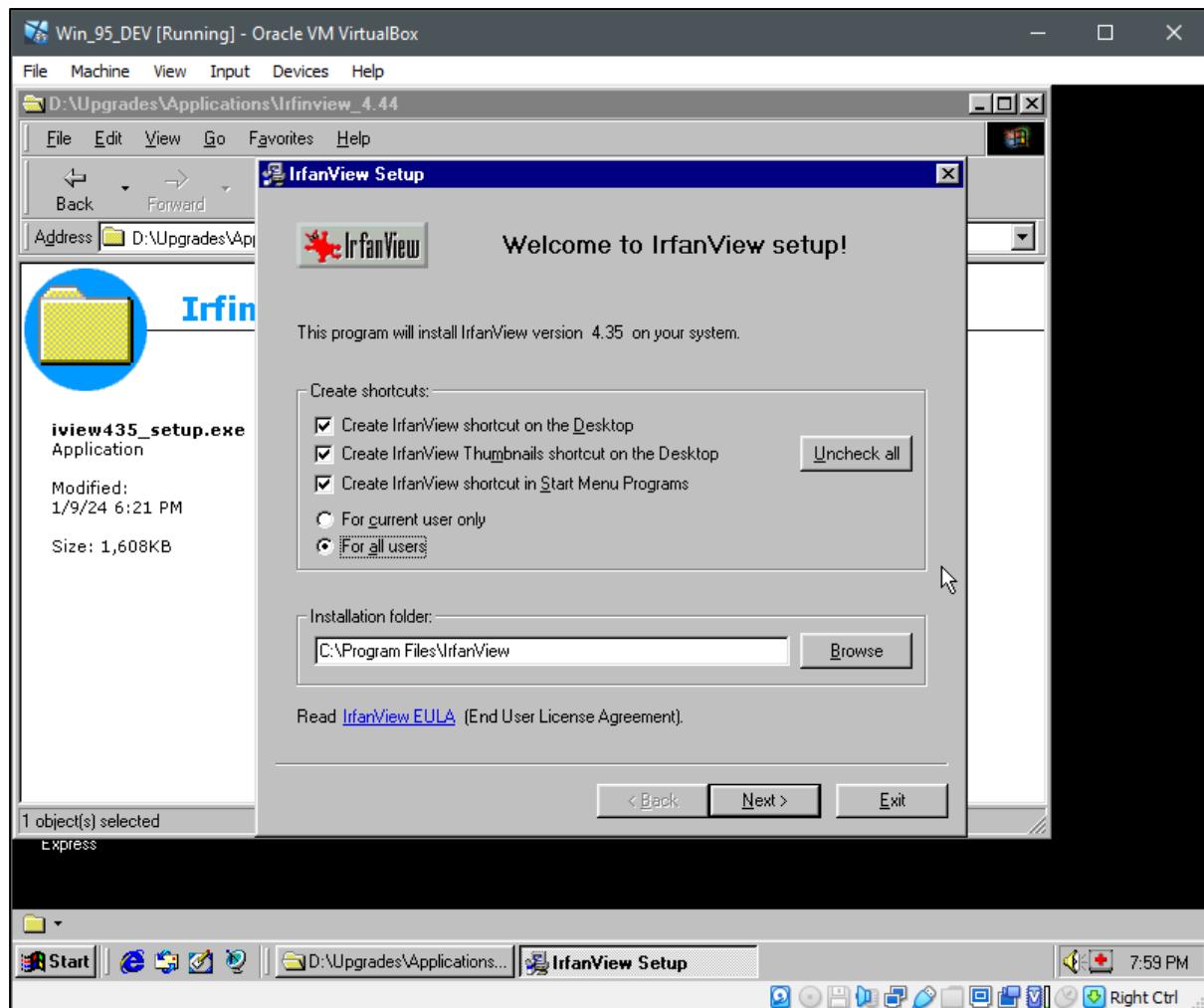
"iview444\_setup.exe"

IrfanView 4.44 will work on Windows 95 but does have some issues with dialogs such as the About window, but otherwise works OK as a picture viewer. If you have issues drop back to the V4.35 or earlier.

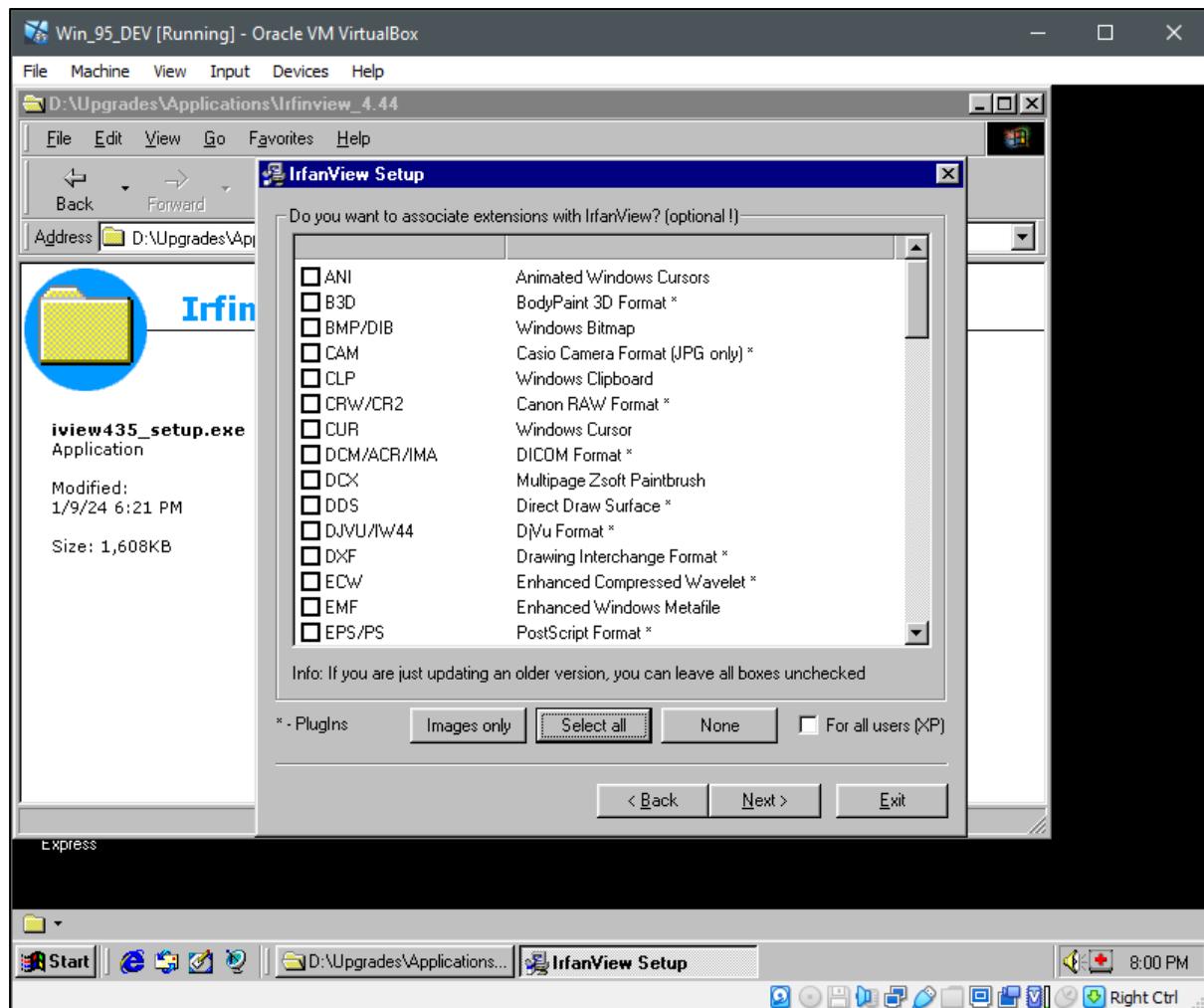
<https://www.philscomputerlab.com/vga-test-patterns.html>

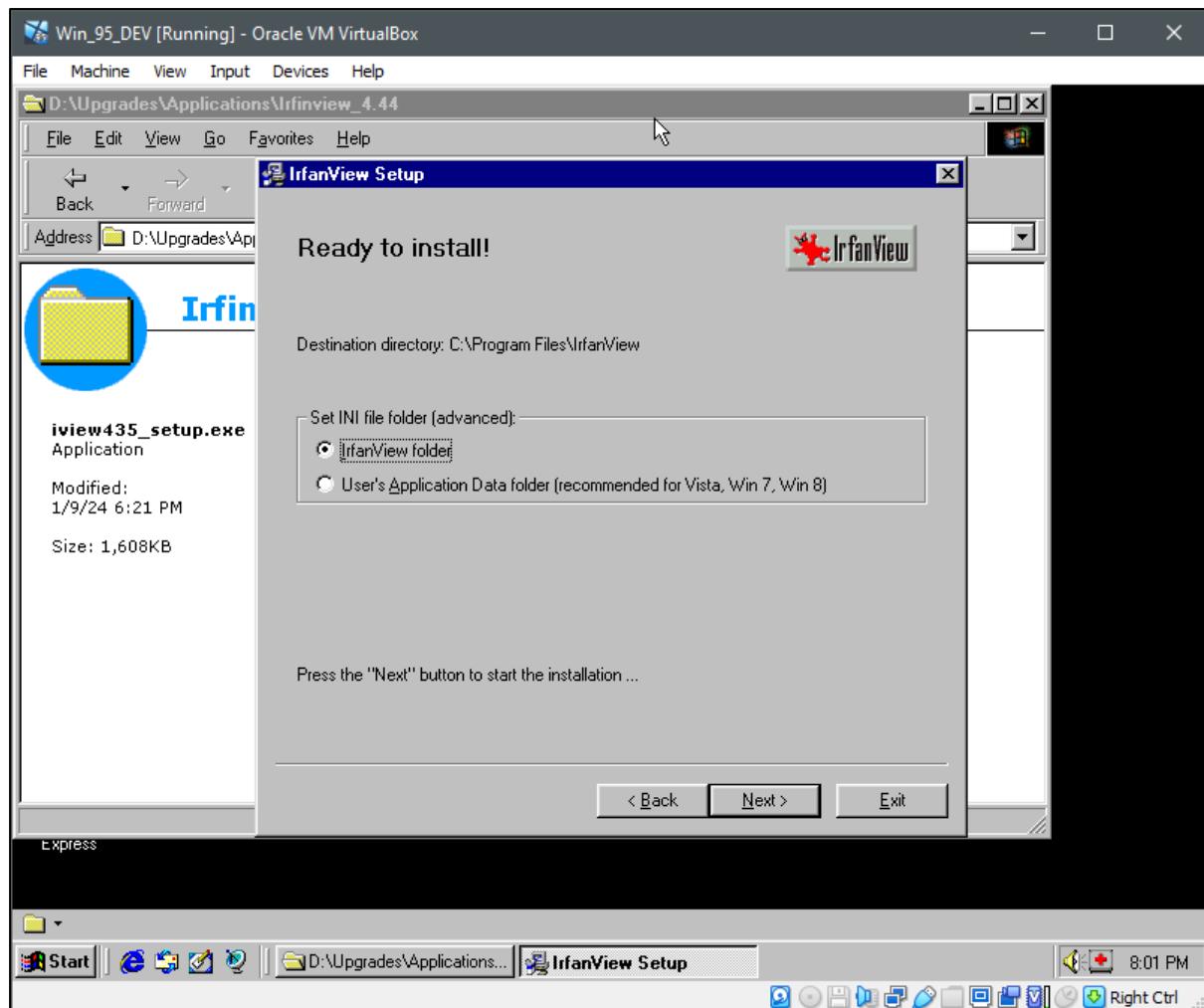
<http://www.oldversion.com/windows/irfanview/>

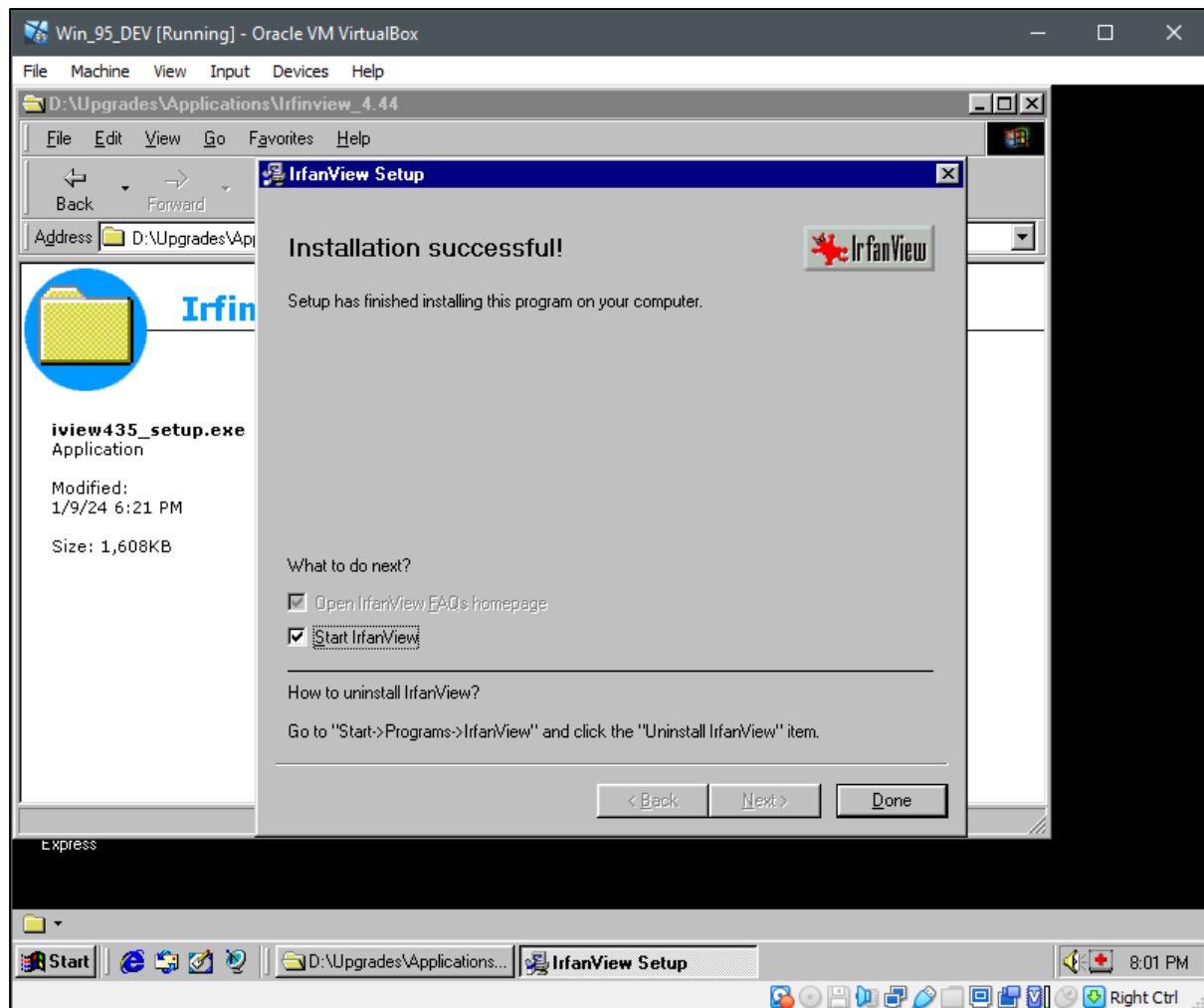
Run the Installer and follow the setup options.



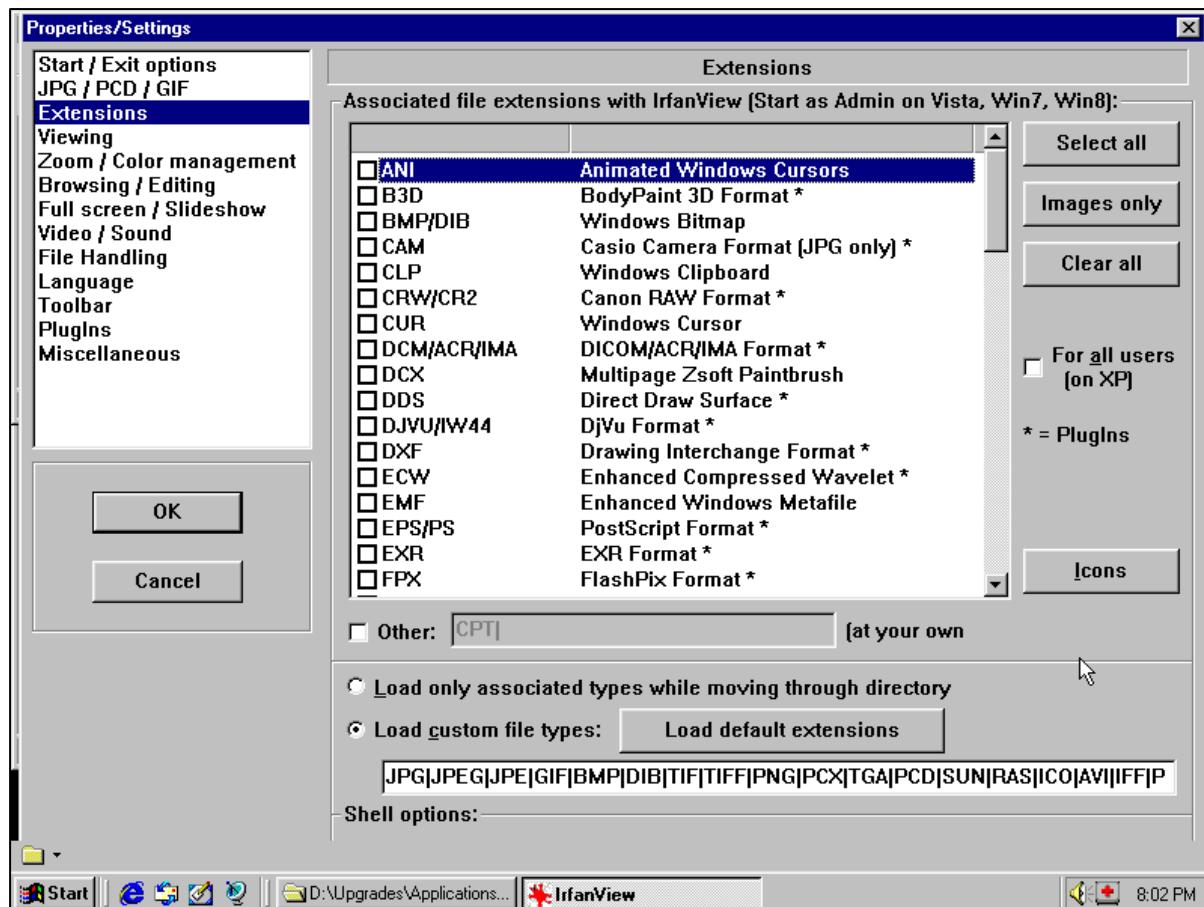
You can select file associations now, or later from the options menu.







The options menu after running Irfinview.



### Other Image viewers and tools

Gimp V1.2.0 (Very hard to find the compiled binaries)

MS Paint (Installed)

Many other Image viewers and editors including some from DOS.

### Browsers

Windows comes with IE as the default web browser. You can also use an old version of SeaMonkey that will run on Windows 95. There are other browsers such as OffByOne that will also work. Remember that browsers of this age are typically not secure, so use them with care and sparingly.

Tiny Personal Firewall Engine V 2.0.15 A (221001) will work and provide some level of protection on Windows 95.

### SeaMonkey V 1.09

SeaMonkey is a very lightweight web browser based on Mozilla.

<https://www.seamonkey-project.org/releases/seamonkey1.0.9/>

<https://www.seamonkey-project.org/releases/1.0.9>

Select the Windows (32bit) – Full Installer.

“seamonkey-1.0.9.en-US.win32.installer.exe”

The install is fairly straight forward so you can just follow the default prompts.

Complete will install email and other browser add-ons. I typically choose “Complete”.

### **Tiny Personal Firewall 2.0.15A**

Optional.

This is a free edition. It is a somewhat hands on firewall, meaning that you will have to manually allow and disallow certain connections and settings.

<http://www.oldversion.com/windows/tiny-personal-firewall-2-0-15a>

“tpfirewall2015a.exe”

The firewall application download does not include the user guide. You will need to search the webs for a copy of it.

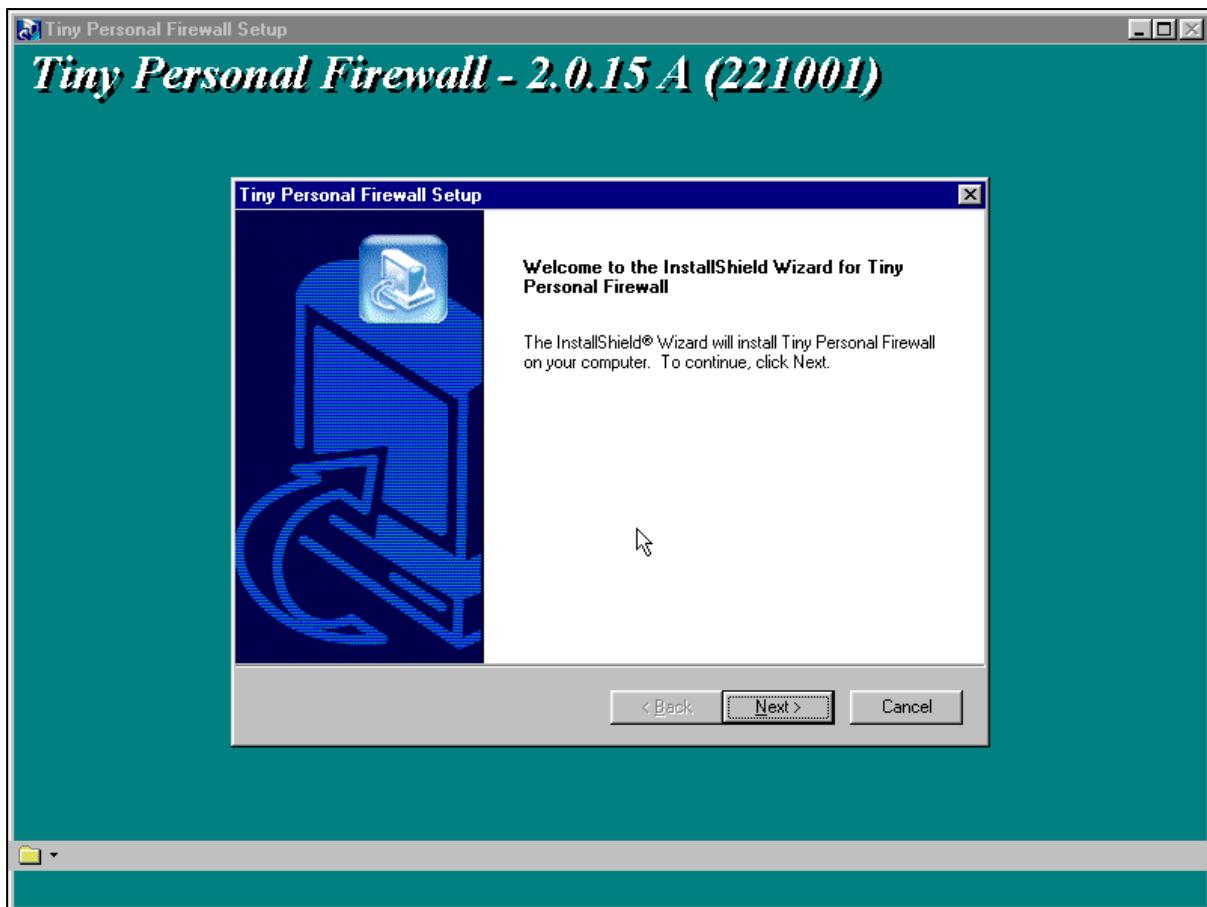
“User's Guide. Tiny Personal Firewall V.2”

“Tiny Personal Firewall 2 User's Guide.zip”

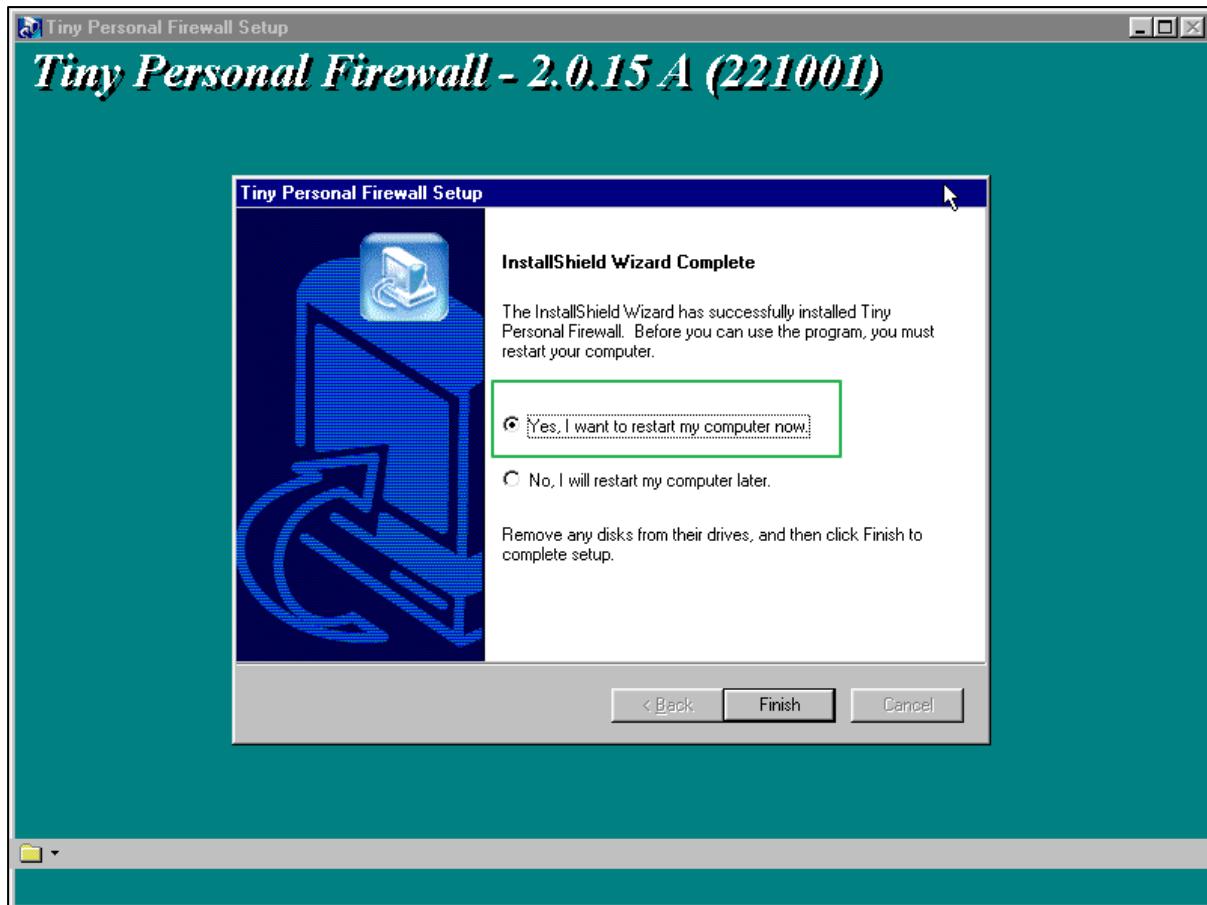
<https://vetusware.com/download/Tiny%20Personal%20Firewall%202.0.9/?id=5209>

<https://docplayer.net/21485376-User-s-guide-tiny-personal-firewall-v-2.html>

Run the downloaded file to install the firewall.



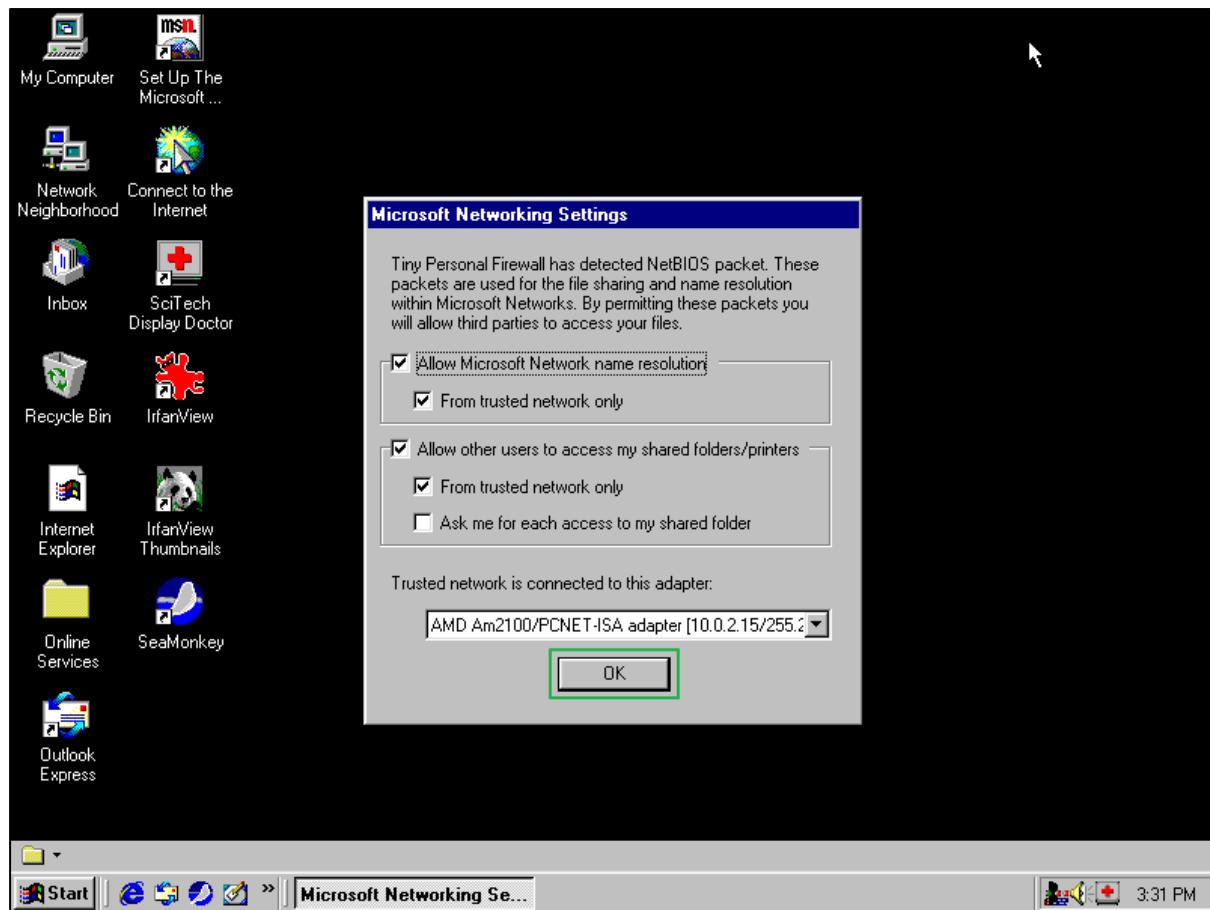
Follow the install with the default settings and then restart the computer.



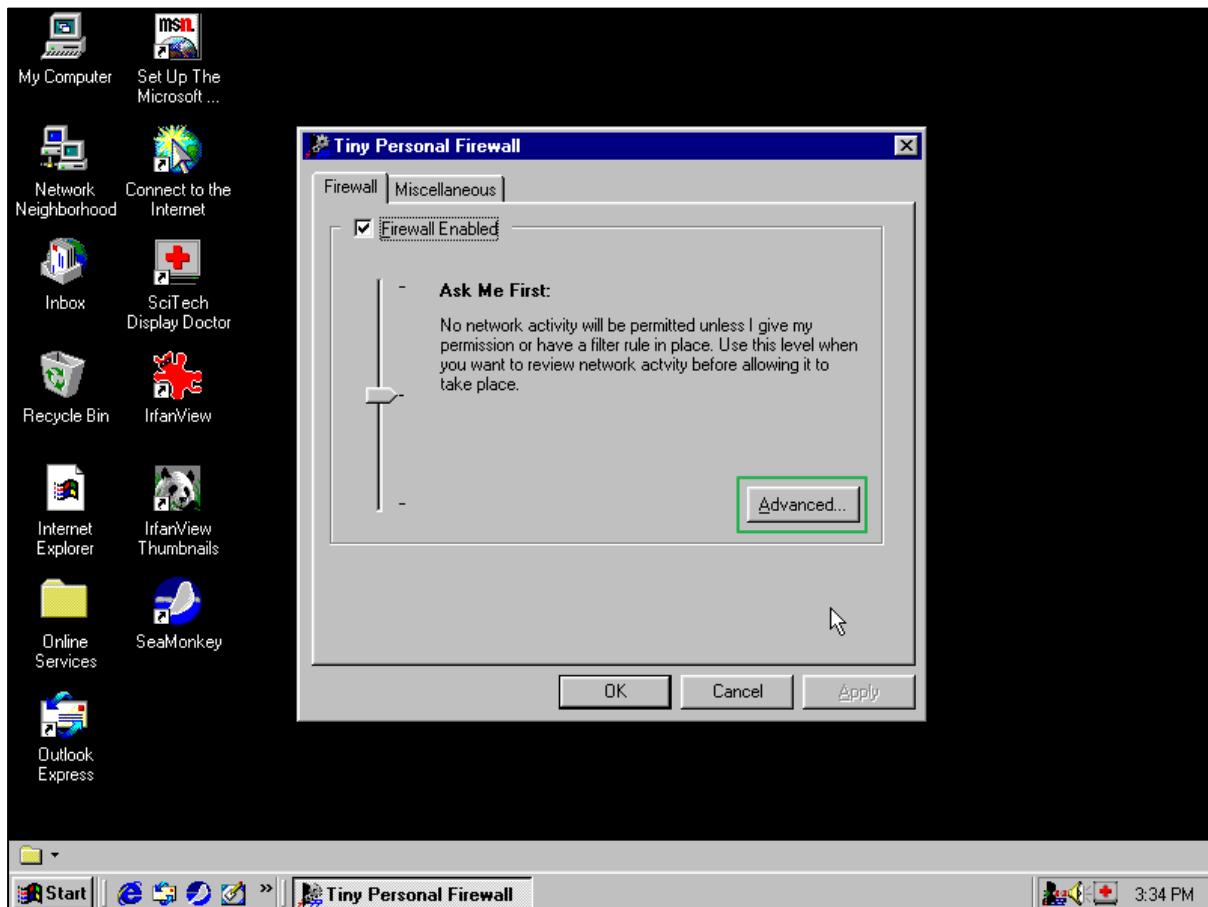
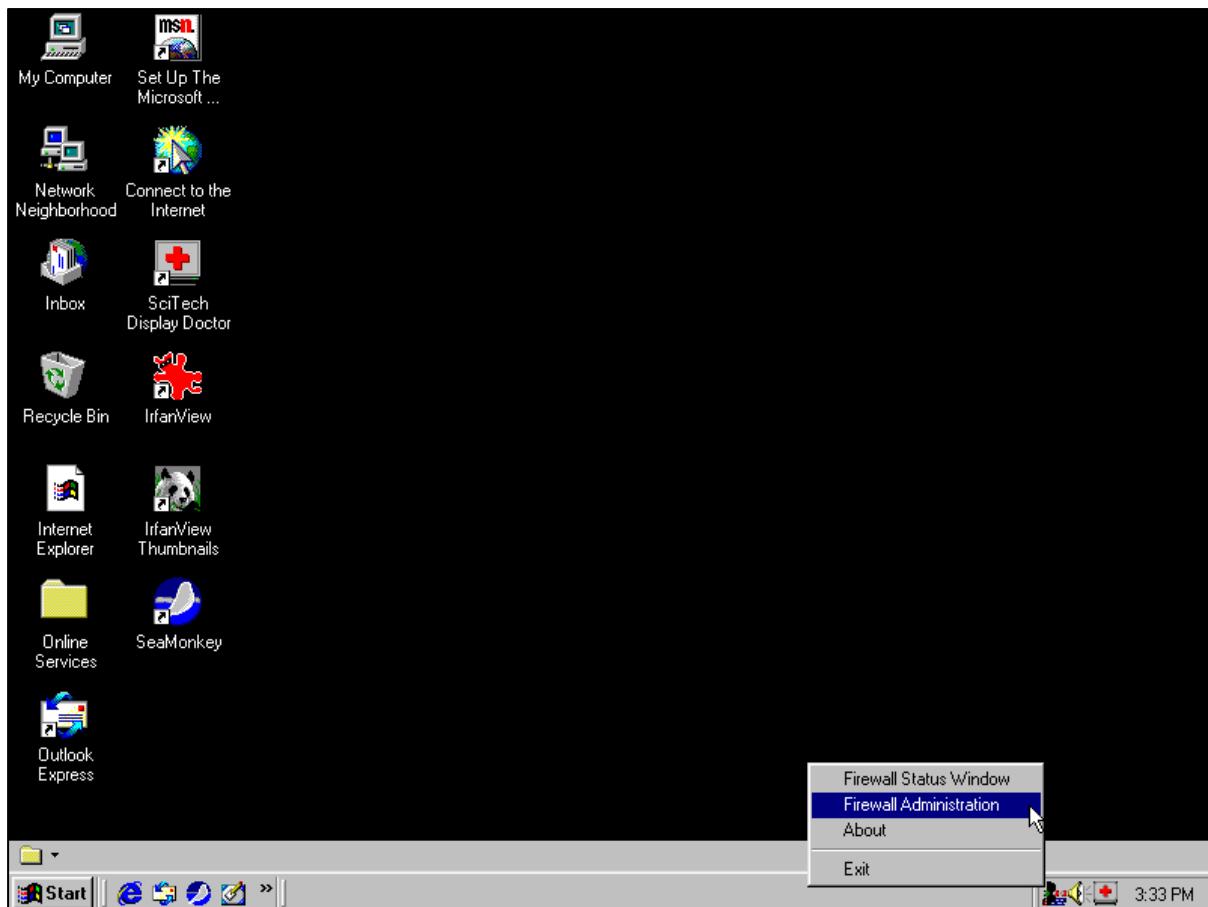
Tiny Personal Firewall will install the firewall service daemons before the windows login.

You will be met with a splash screen before the Windows Login.

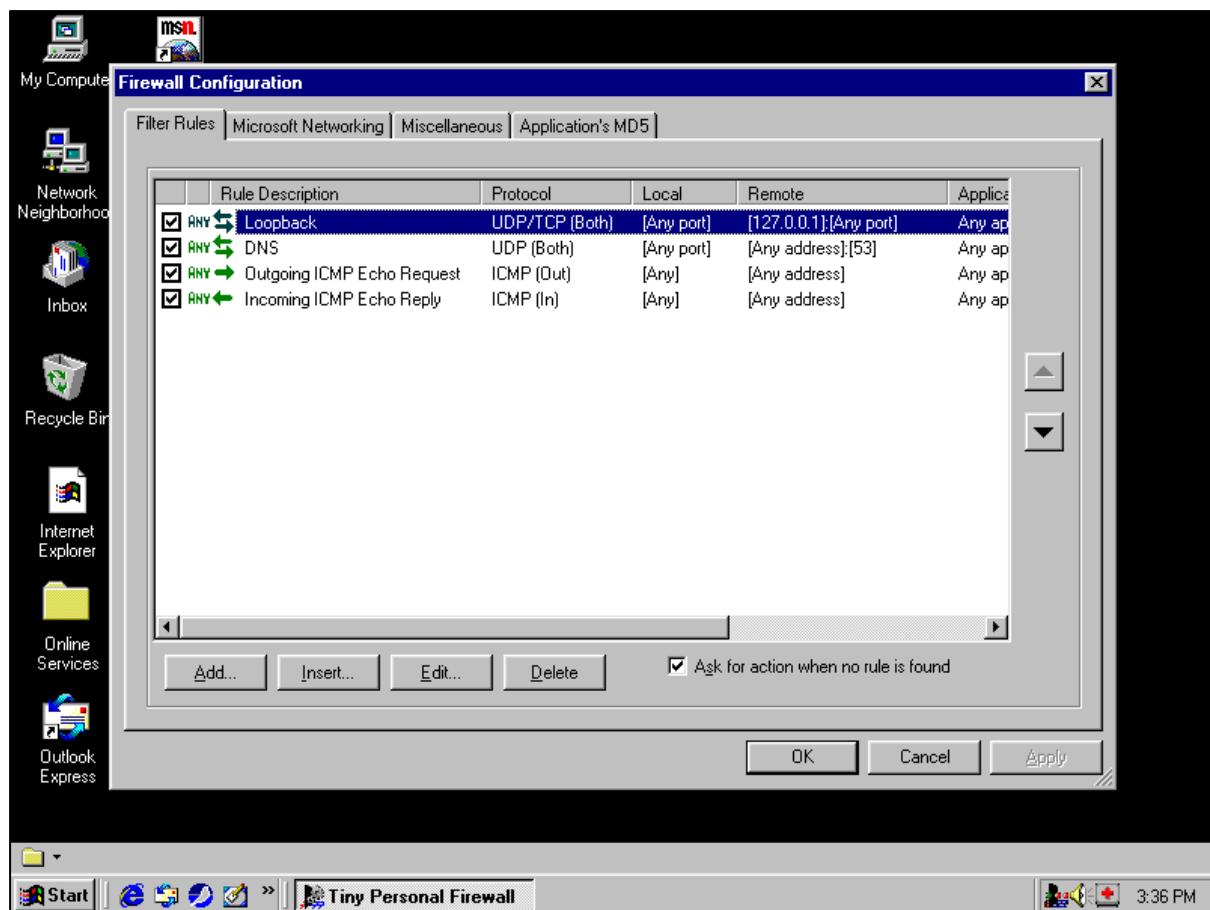
It is OK to keep the defaults when the firewall asks on the first start.



Open the Admin Panel to fine tune your firewall settings. More secure means more notifications to allow, disallow.



You can also fine tune the setting in the advanced configuration section.



Read the user guide if you have obtained a copy. The advanced configuration is not indifferent to other Firewalls.

There are many other tools and useful applications for essential Windows 95 tasks. I have just covered a few to get started.

Make sure you create a backup of the primary VirtualBox VHD file for windows 95. If you install an application, runtime distribution or Windows fix you will be able to roll it back if the system breaks.

### Install summary

At this point what I have installed is covered the following list:

- Setting up a VirtualBox guest
  - File transfers
  - 7-Zip
  - Folder2Iso
  - ImDisk Toolkit
  - HFS
- Setting up Windows 95

- Install Windows 95[B | C]
- Quick usability tweaks
- Explorer
- Fix Mouse – scroll
- Windows 95 Display drivers fix
- Windows 95 Upgrades
- Multiple Floppy drive issue
- Update Internet Explorer (IE 4.1 to IE5.5) Main Windows update path
- Runtime Libraries
  - DirectX
  - Microsoft Visual C(++) 6 Runtime Files
  - VisualBasic Runtime libraries
- Essential Applications
  - Archives
    - 7-Zip 9.20
  - Restore tools
    - ERU
    - CFGBack
  - System Clean-up
    - Windows CleanUp! 4.5.2
    - Microsoft RegClean
  - System tools
    - PrcView
    - Windows 95 Power Toys
    - TweakUI V.133
    - Command Prompt Here
  - Applications
    - IrfanView
    - Browsers
    - SeaMonkey V 1.09
    - Tiny Personal Firewall 2.0.15A

The base Windows 95 setup is complete at this point. You can make some of your own customizations from here including additional applications. When you are content with your final saved (Backed up) copy of Windows, move on to the Development Environments Setup.

---

## HFS over IP

For file transfers over HTTP TCP/IP you will need at minimum Internet Explorer or the SeaMonkey web browser installed and configured for internet access. On VirtualBox this requires installing the correct network card drivers and the Microsoft TCP/IP protocols as explained in the Windows 95 install section.

HFS can be downloaded from the applications home site:

<https://www.rejetto.com/hfs/?f=dl>

“hfs.exe” Version 2.3m

Install HFS as a portable application.

Note that I don't allow this application to have access from outside networks such as the internet due to potential security concerns.

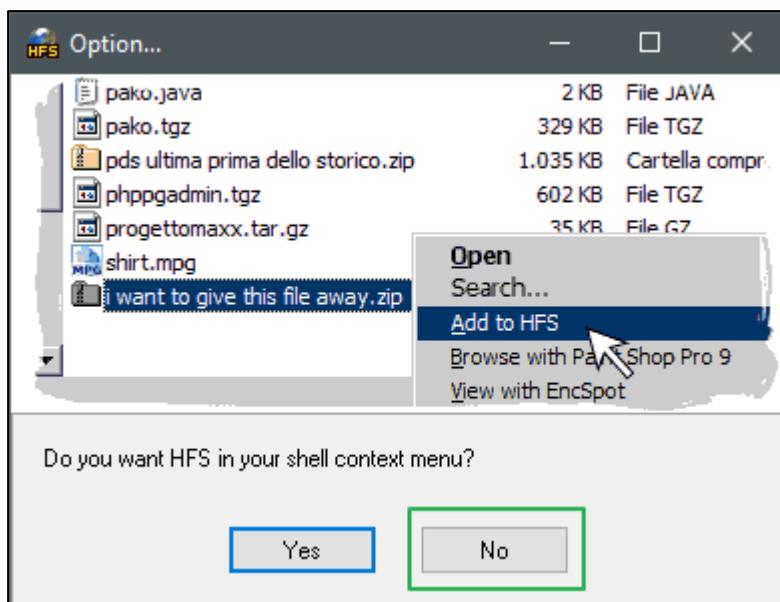
I typically have a dedicated directory for my portable applications aka applications that don't require an installer. Usually something like C:\Portable\Appname or C:\Portable\_Apps\Appname

Create a directory for HFS and copy the downloaded file to the directory, such as C:\Portable\HFS\hfs.exe

I usually create a shortcut to hfs.exe and place it near my VirtualBox shortcuts.

Start HFS.exe

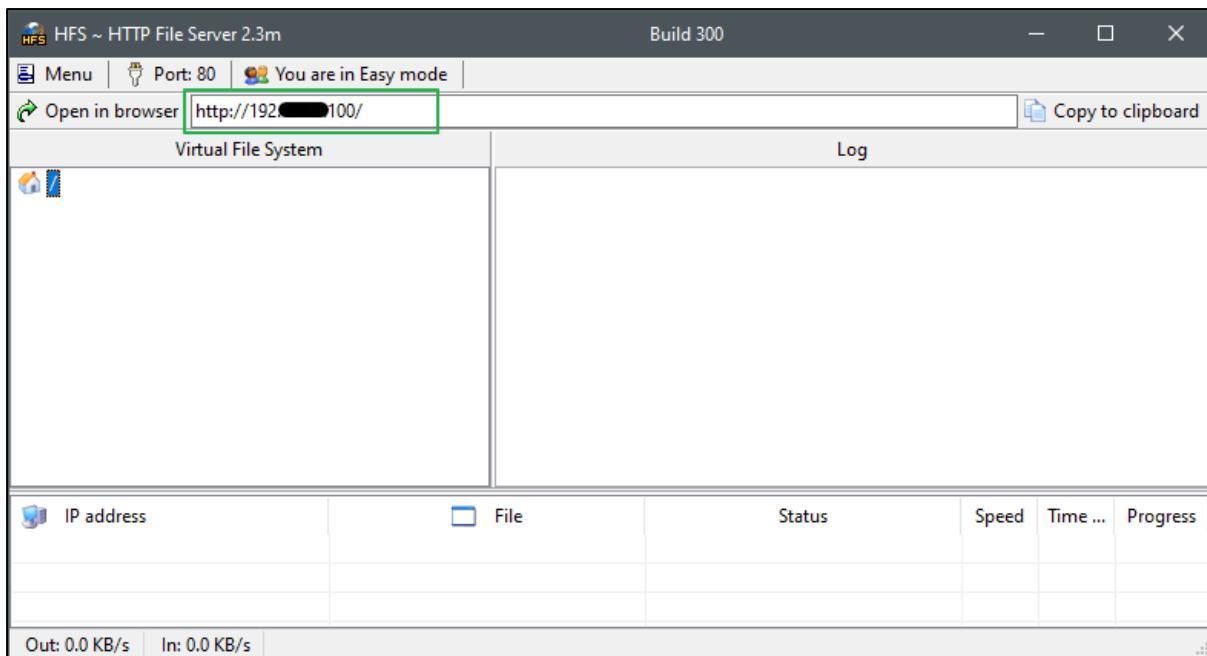
At first start you will be asked if you would like to be able to add files to the server from the Windows context menu. I usually select NO.



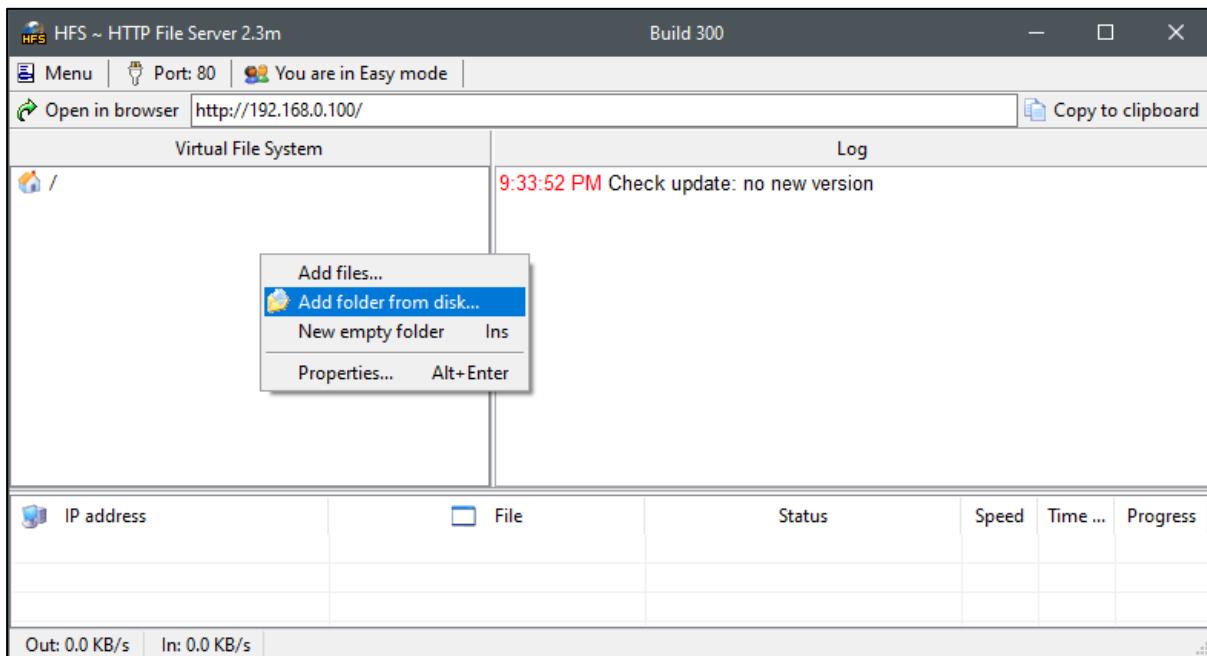
Next you will be met with the server control and configuration window.

Take note of the server IP address on the network.

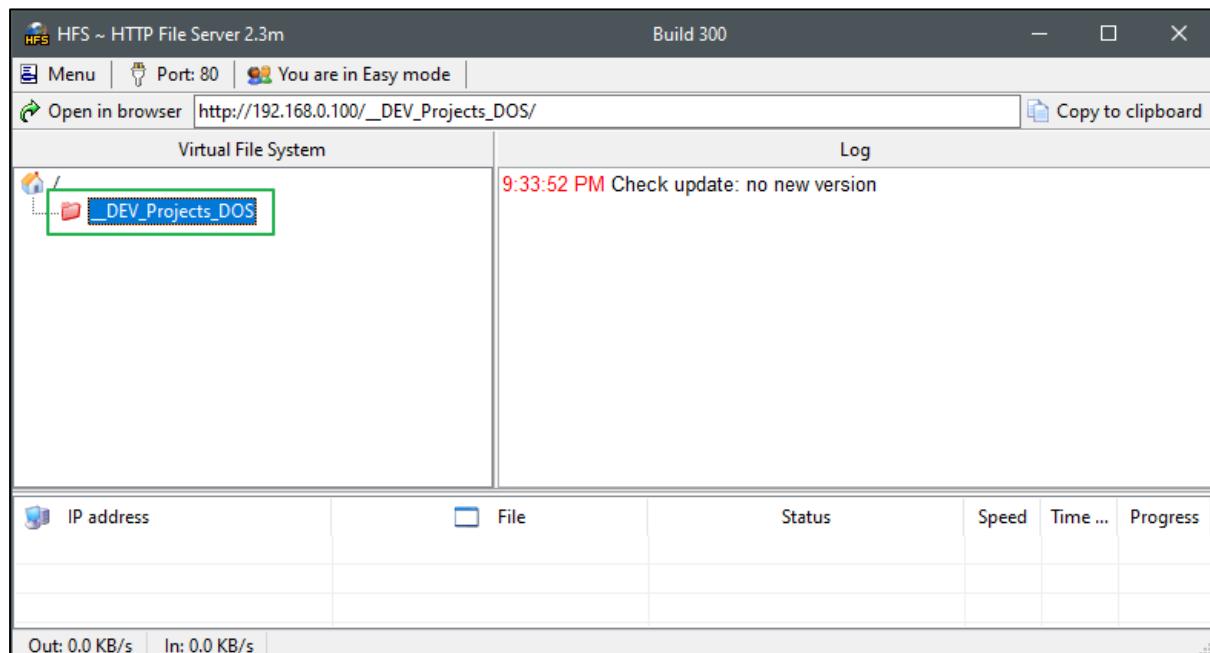
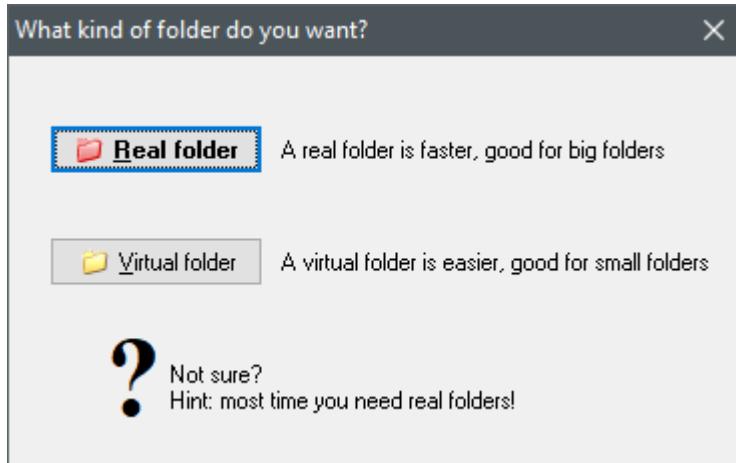
## A Beginners Guide To DOS Programming



Next right click on the panel next to the home icon and select a home directory for the files you wish to make available for HTTP sharing. This is just the root directory and you can add as many sub directories and files as you need at any time later.

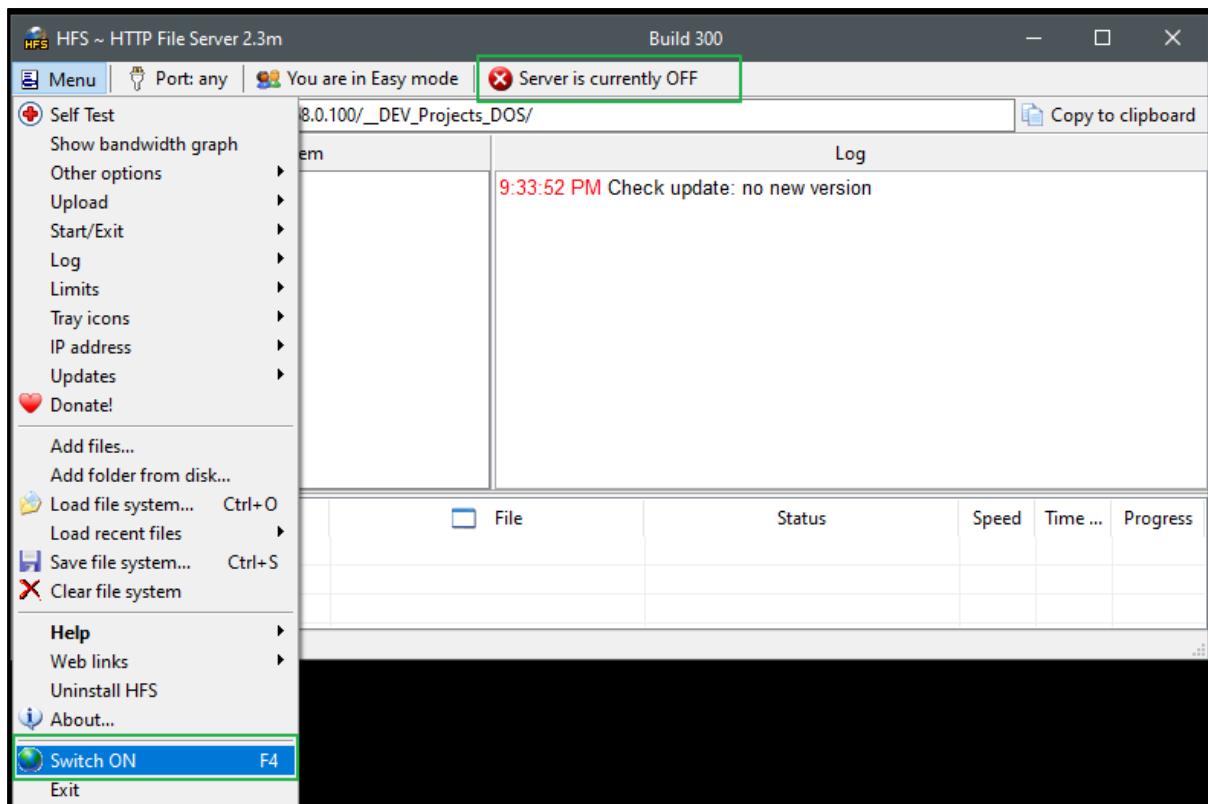


I typically use a real folder.



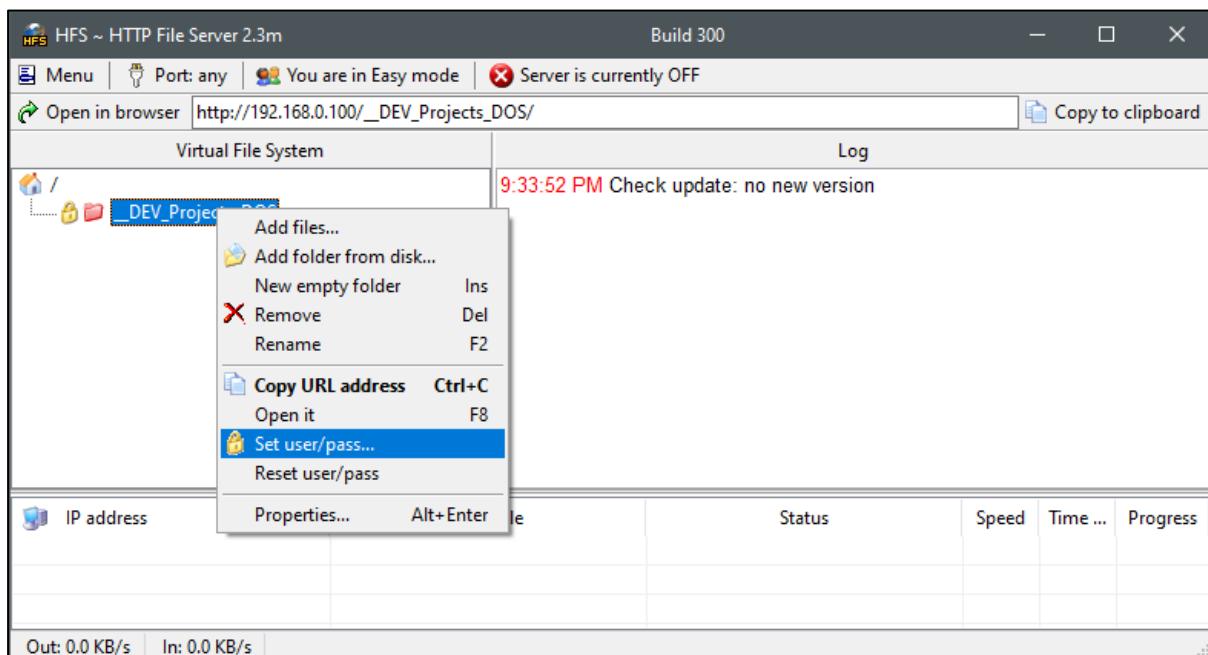
At this point any file or directory that is inside your selected directory will be available when you turn on the file server. The server is ON by default when first started. I suggest keeping it turned off unless you need to access it from another computer (FreeDOS or Windows 95 client in VirtualBox) in your local network.

## A Beginners Guide To DOS Programming

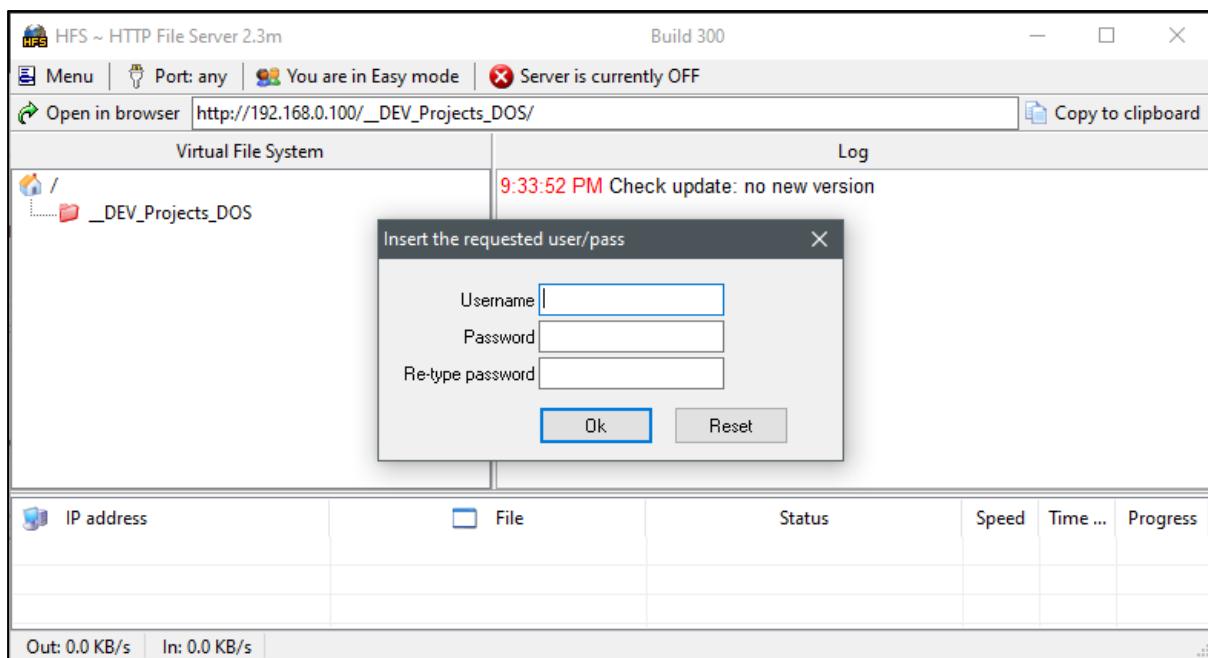


Next I would recommend setting a User Name and Password to access the server via HTTP.

Right click on the root share directory and select “Set user/pass...”

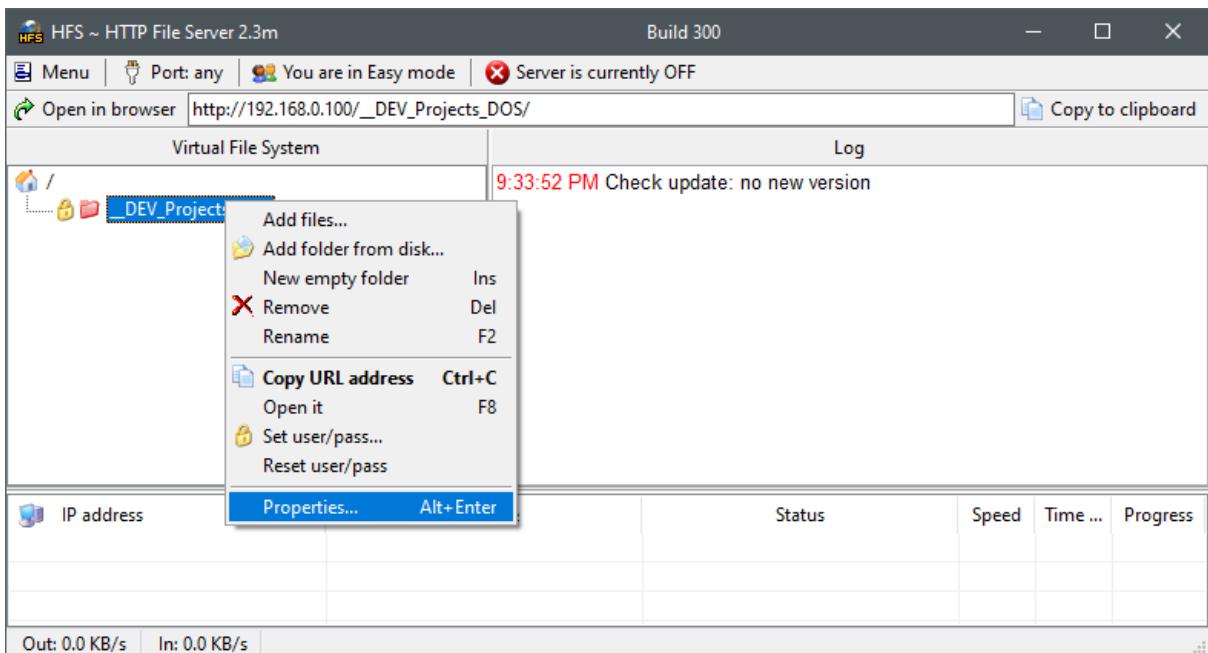


## A Beginners Guide To DOS Programming



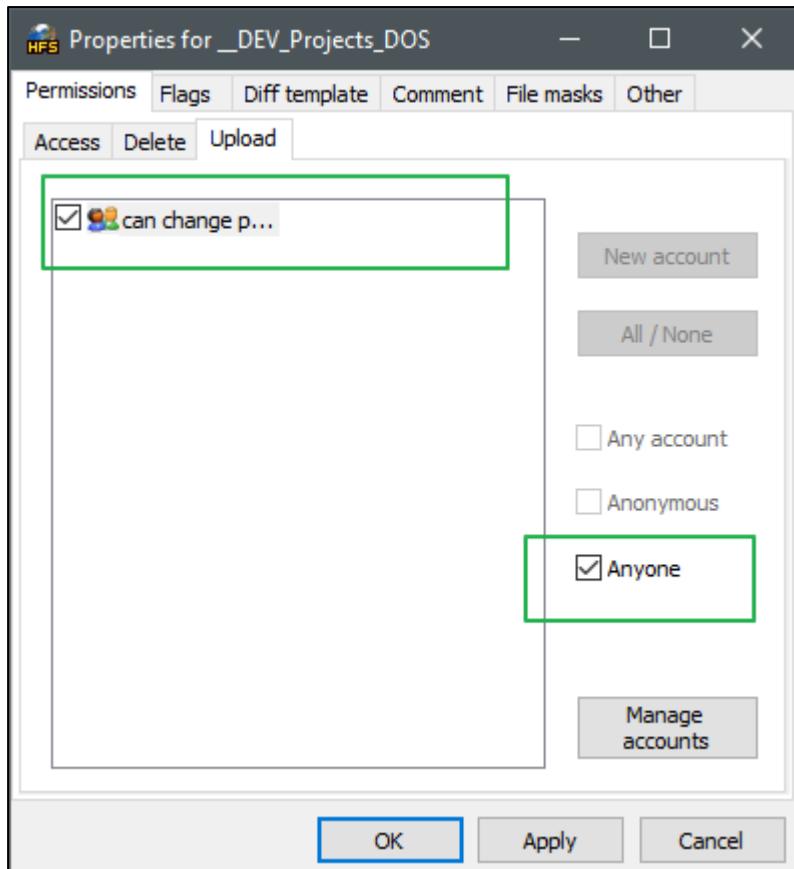
The lock will then show beside the file share directory.

Next become familiar with the directory and user share properties.

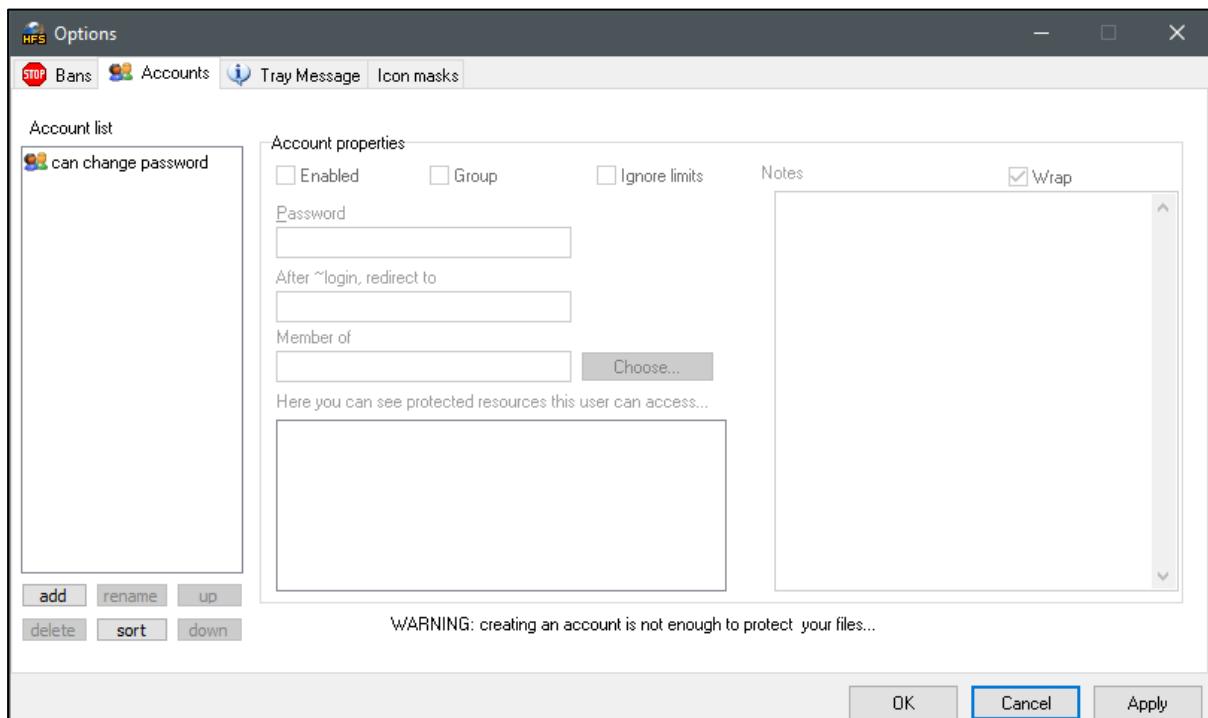


This is the root directory or Admin Account. You can add users and groups if you wish, but it is not required for simple file transfers. Become familiar with these setting just the same and also have a look at the “Manage accounts” button. These are the main server account and permission controls.

Select “can change password” (Admin account) and Check the box [/]Anyone to allow uploading files from DOS to the server.



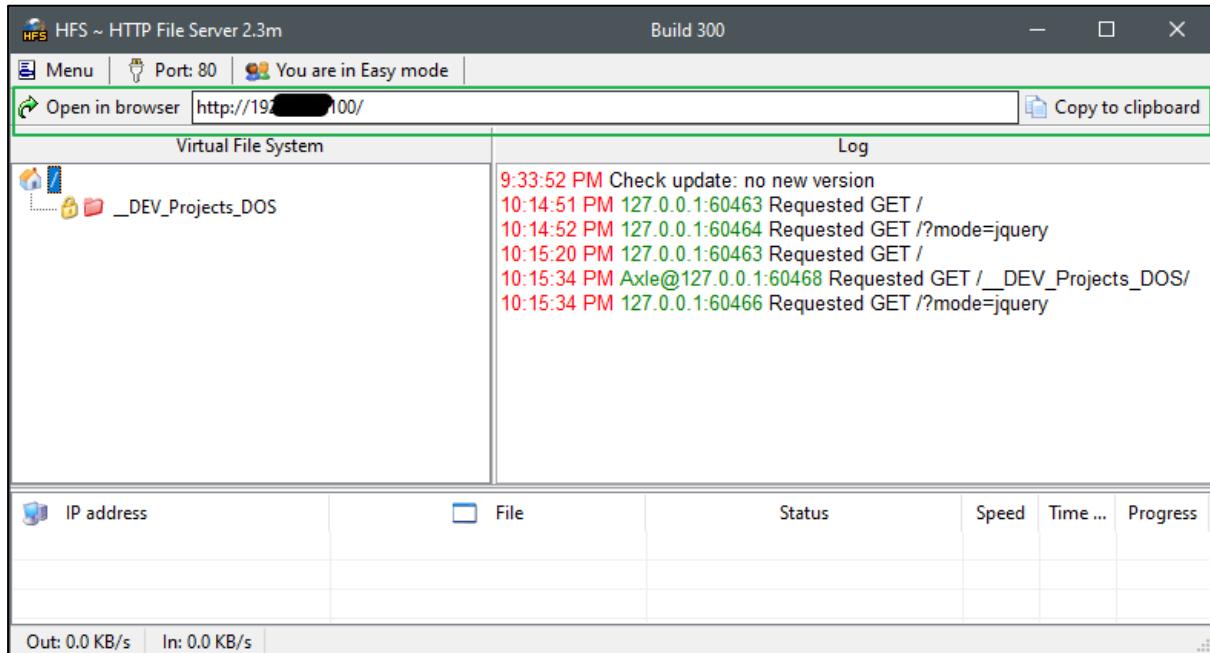
You can add Users and groups here later if you like.



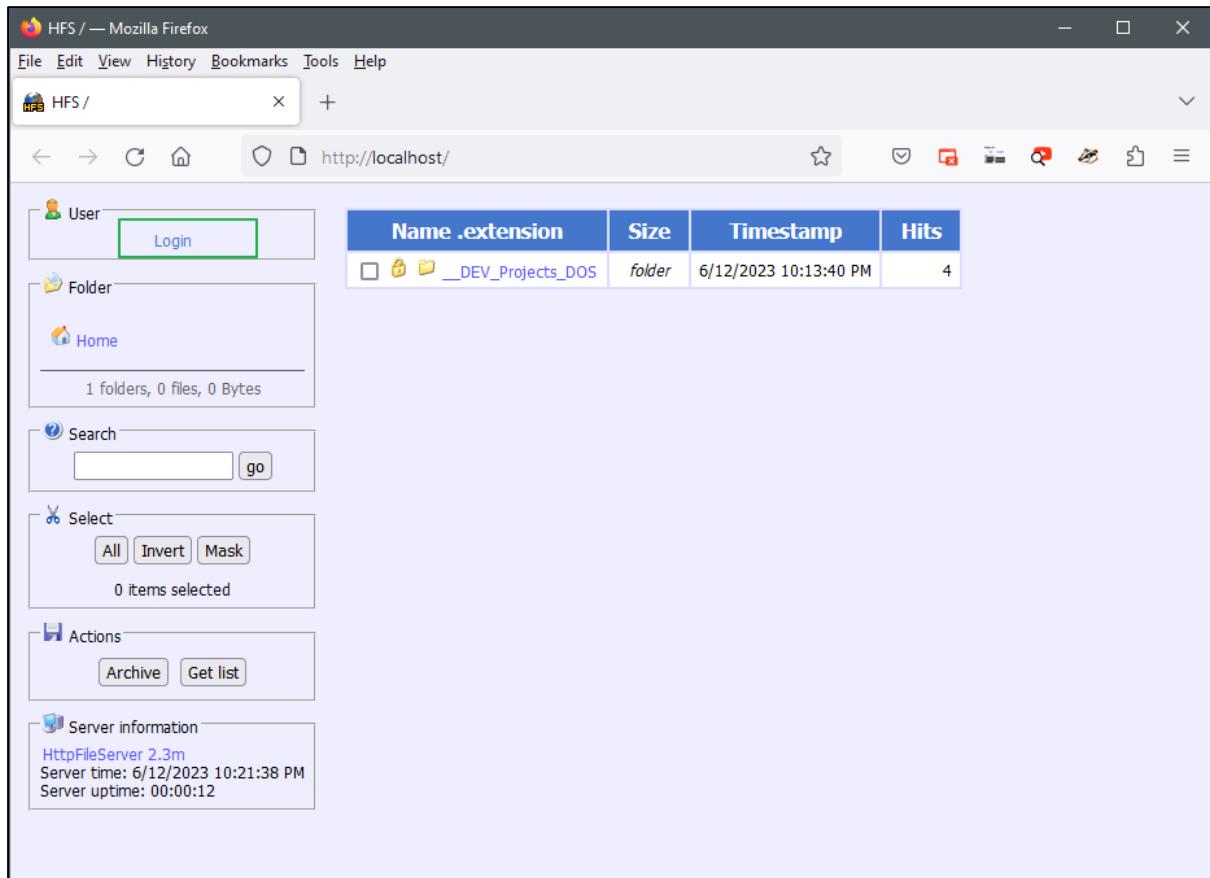
## A Beginners Guide To DOS Programming

Open the location of your server root directory and add a sub directory and a test text file.

Turn on the server and wait 5 to 10 seconds for the server to start up. Then open the browser to the local host 127.0.0.1 or IP address page.



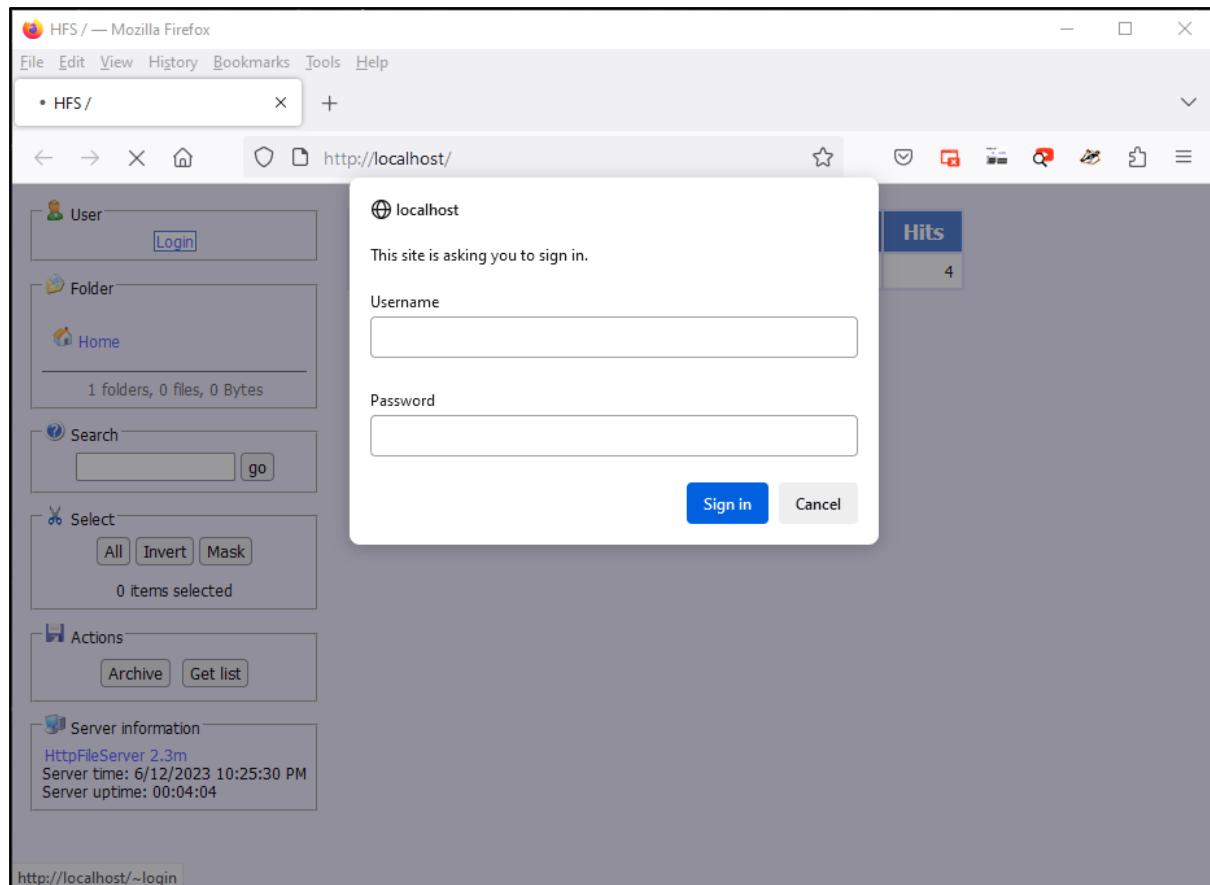
Click on the login link.



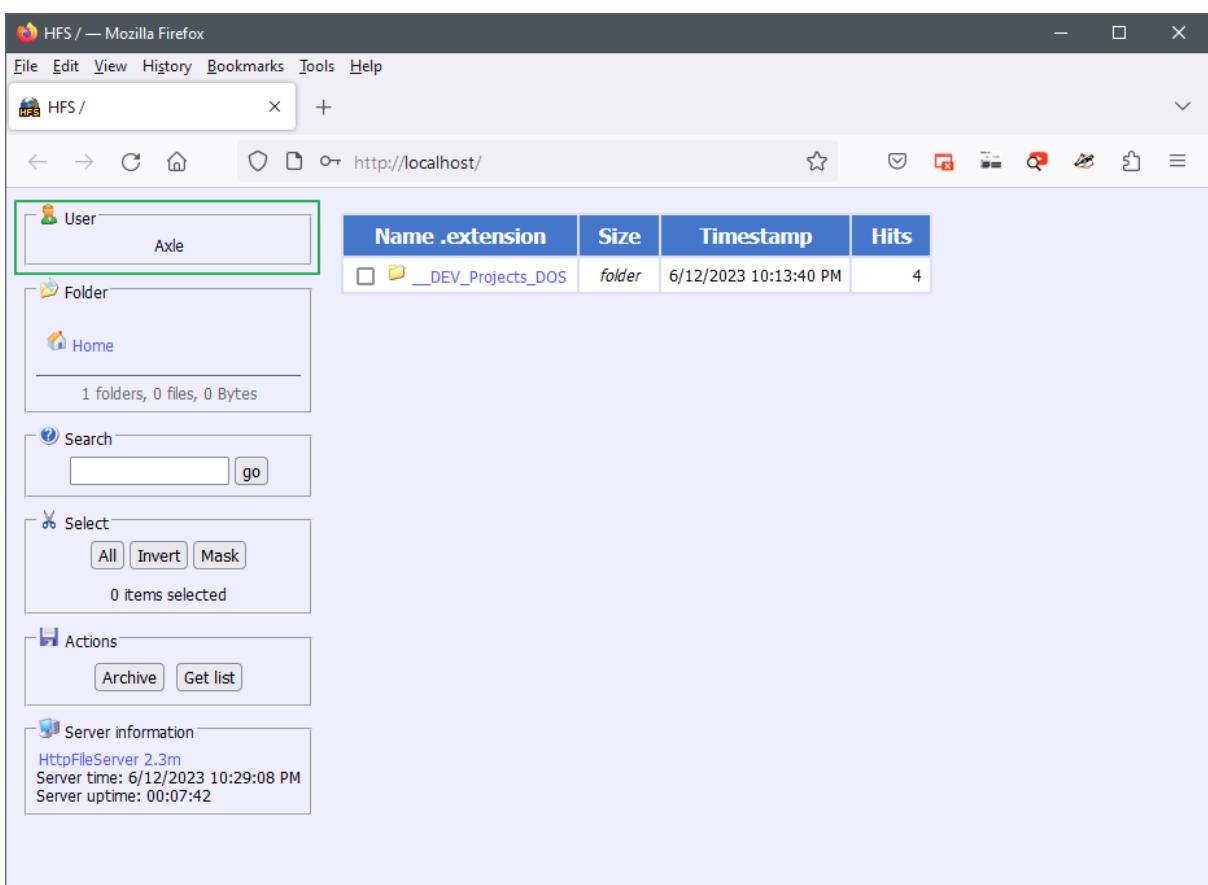
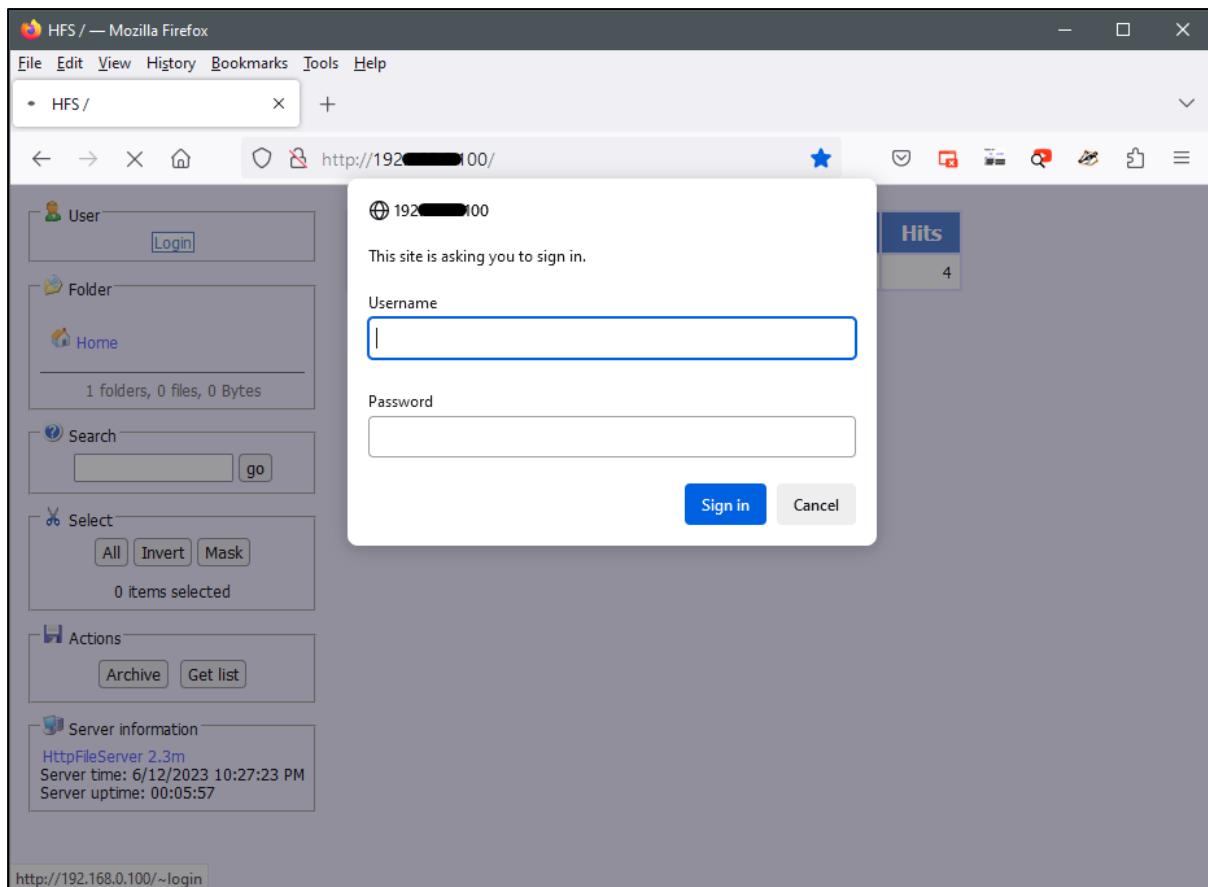
## A Beginners Guide To DOS Programming

If the login does not show, close the browser and then reopen the link from the server.

Enter the use name and password you created earlier.



## A Beginners Guide To DOS Programming



## A Beginners Guide To DOS Programming

Click on the root directory listed in the browser to view sub directories and files.

The screenshot shows a Mozilla Firefox browser window displaying a file listing for the directory `HFS / _DEV_Projects_DOS`. The address bar shows `http://localhost/_DEV_Projects_DOS/`. The page content includes:

- User:** Axe
- Folder:** [Up](#)
- Home:** [\\_DEV\\_Projects\\_DOS](#)
- Search:**  go
- Select:** All, Invert, Mask  
0 items selected
- Actions:** Archive, Get list
- Server information:** HttpFileServer 2.3m, Server time: 6/12/2023 10:30:19 PM, Server uptime: 00:08:53

A table lists the files and folders in the directory:

Name .extension	Size	Timestamp	Hits
Active	folder	18/08/2023 5:20:26 PM	0
Apps	folder	18/08/2023 8:29:57 AM	0
Examples	folder	18/08/2023 9:45:05 AM	4
Libs	folder	18/08/2023 8:49:55 AM	0
Proj	folder	18/08/2023 9:25:01 AM	0
Sort	folder	18/08/2023 8:40:59 AM	0
Test.txt	12B	6/12/2023 10:13:52 PM	0

The row for `Test.txt` is highlighted with a green border.

You should now be able to view and download your test text file form the server.

If you have enabled file uploads you should see the upload button in the Actions section.

## A Beginners Guide To DOS Programming

The screenshot shows a Mozilla Firefox window displaying a file browser interface. The address bar shows the URL [http://localhost/\\_DEV\\_Projects\\_DOS/](http://localhost/_DEV_Projects_DOS/). The main content area is a file listing table:

Name .extension	Size	Timestamp	Hits
Active	folder	18/08/2023 5:20:26 PM	0
Apps	folder	18/08/2023 8:29:57 AM	0
Examples	folder	18/08/2023 9:45:05 AM	4
Libs	folder	18/08/2023 8:49:55 AM	0
Proj	folder	18/08/2023 9:25:01 AM	0
Sort	folder	18/08/2023 8:40:59 AM	0
Test.txt	12B	6/12/2023 10:13:52 PM	2

The left sidebar contains the following sections:

- User: Axle
- Folder: Home (selected), Up, 6 folders, 1 files, 12 Bytes
- Search: Search bar, go button
- Select: All, Invert, Mask, 0 items selected
- Actions: Upload (highlighted with a green border), New folder, Comment, Archive, Get list
- Server information: HttpFileServer 2.3m, Server time: 6/12/2023 10:43:09 PM, Server uptime: 00:02:40

Upload will launch the file browser on your local computer and you can select the file to upload to the file server.

## A Beginners Guide To DOS Programming

The screenshot shows a Mozilla Firefox browser window displaying a file manager interface. The address bar shows the URL [http://localhost/\\_DEV\\_Projects\\_DOS/](http://localhost/_DEV_Projects_DOS/). The main content area is a file listing table:

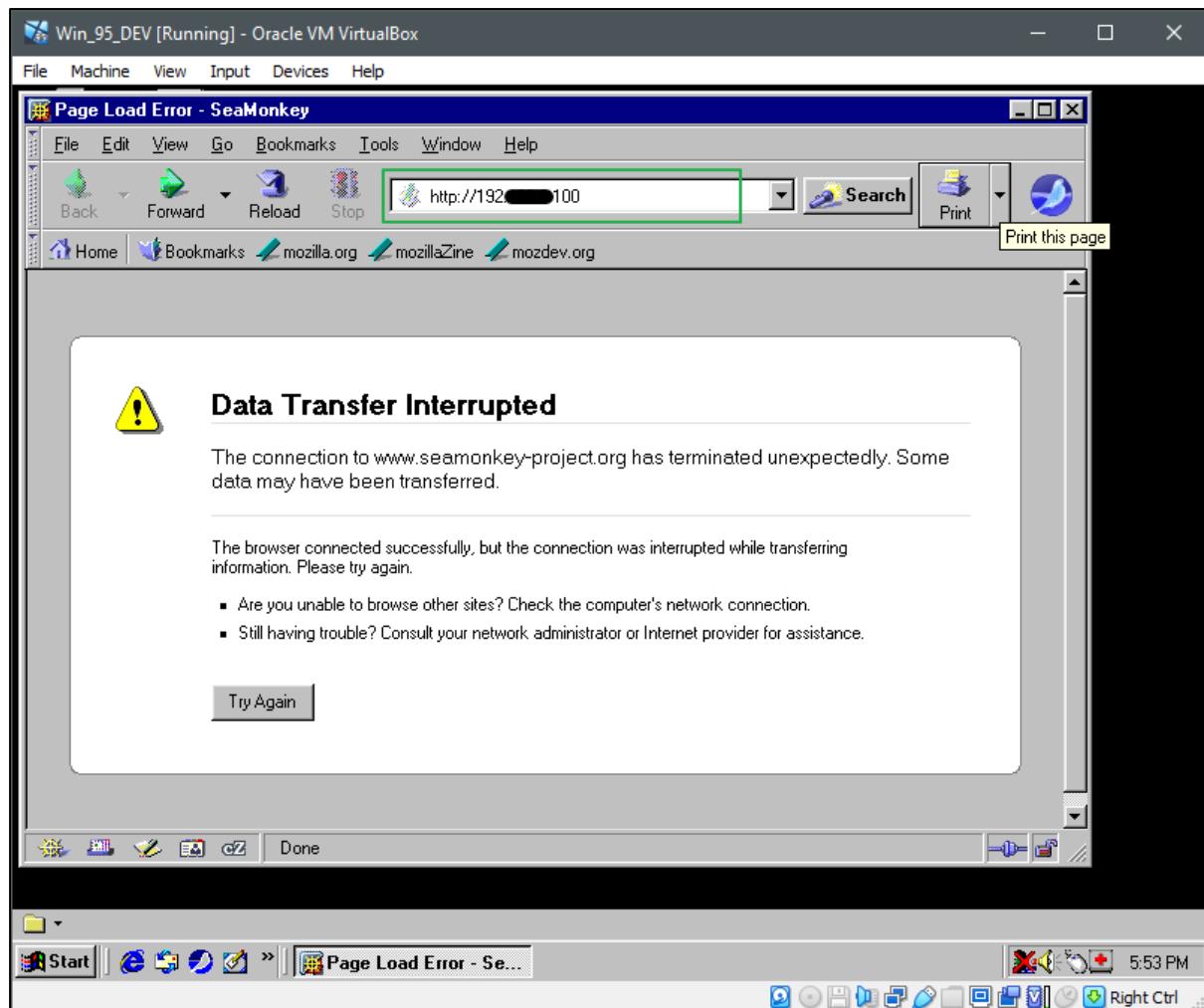
Name .extension	Size	Timestamp	Hits
Active	folder	18/08/2023 5:20:26 PM	0
Apps	folder	18/08/2023 8:29:57 AM	0
Examples	folder	18/08/2023 9:45:05 AM	4
Libs	folder	18/08/2023 8:49:55 AM	0
Proj	folder	18/08/2023 9:25:01 AM	0
Sort	folder	18/08/2023 8:40:59 AM	0
Test.txt	12B	6/12/2023 10:13:52 PM	2

The left sidebar contains navigation links: User (Axle), Folder (Up, Home, \_\_DEV\_Projects\_DOS, 6 folders, 1 files, 12 Bytes), Search (Search bar, go button), Select (All, Invert, Mask, 0 items selected), and Actions (New folder, Comment, Archive, Get list). A green box highlights the 'Upload' section, which includes a 'Browse...' button, a message 'No files selected.', and an 'Upload' button.

That's it for setting up the basic server and tests. Note that it is running in Admin mode which is fine for quick file transfers. If you wish to run the server for extended periods I would recommend reading the documentation and adding groups, user accounts and select file permission settings. This is enough for file transfers to and from your Windows 95 client in VirtualBox using IE or SeaMonkey web browser.

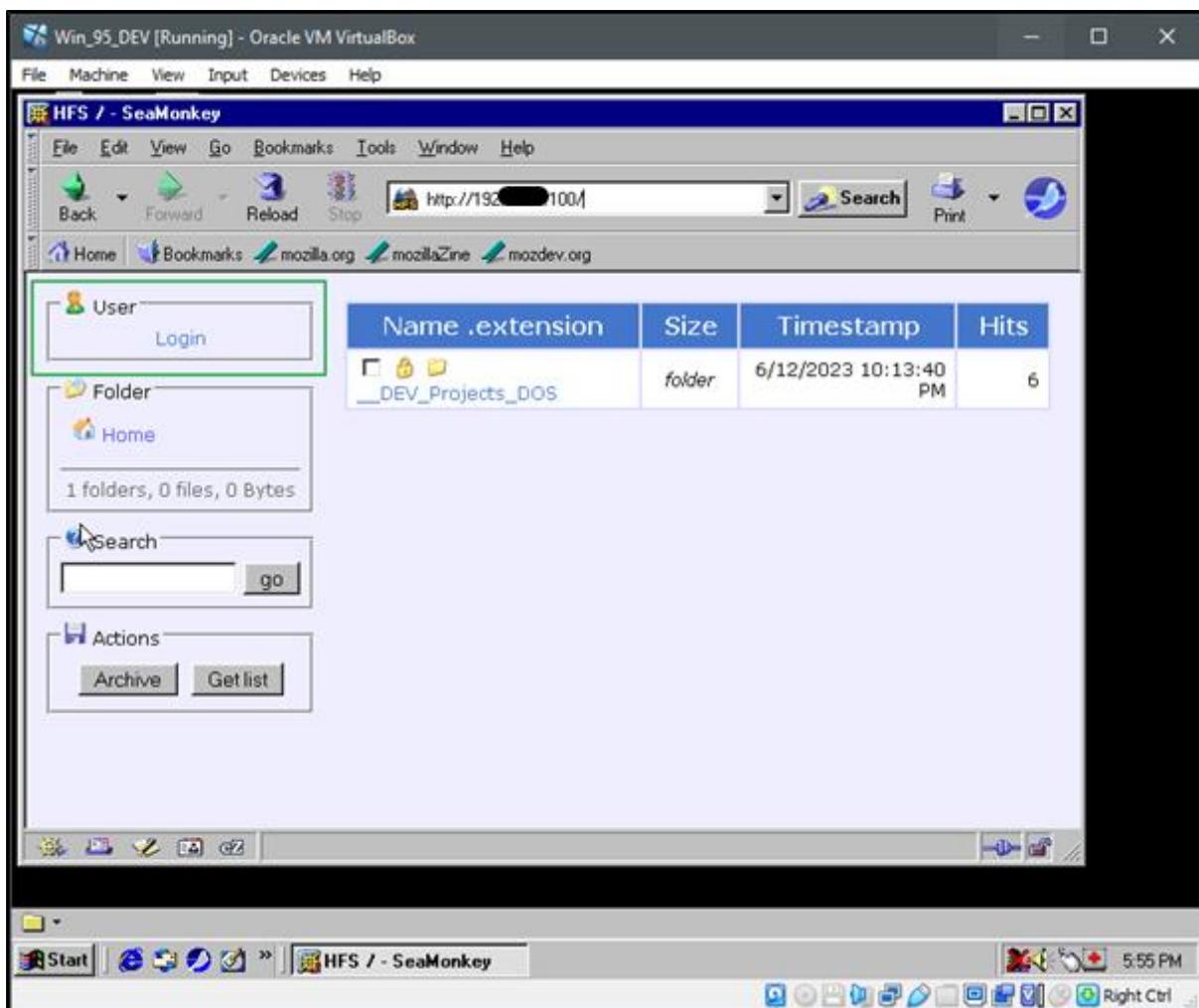
### Navigating with the browser in Windows 95.

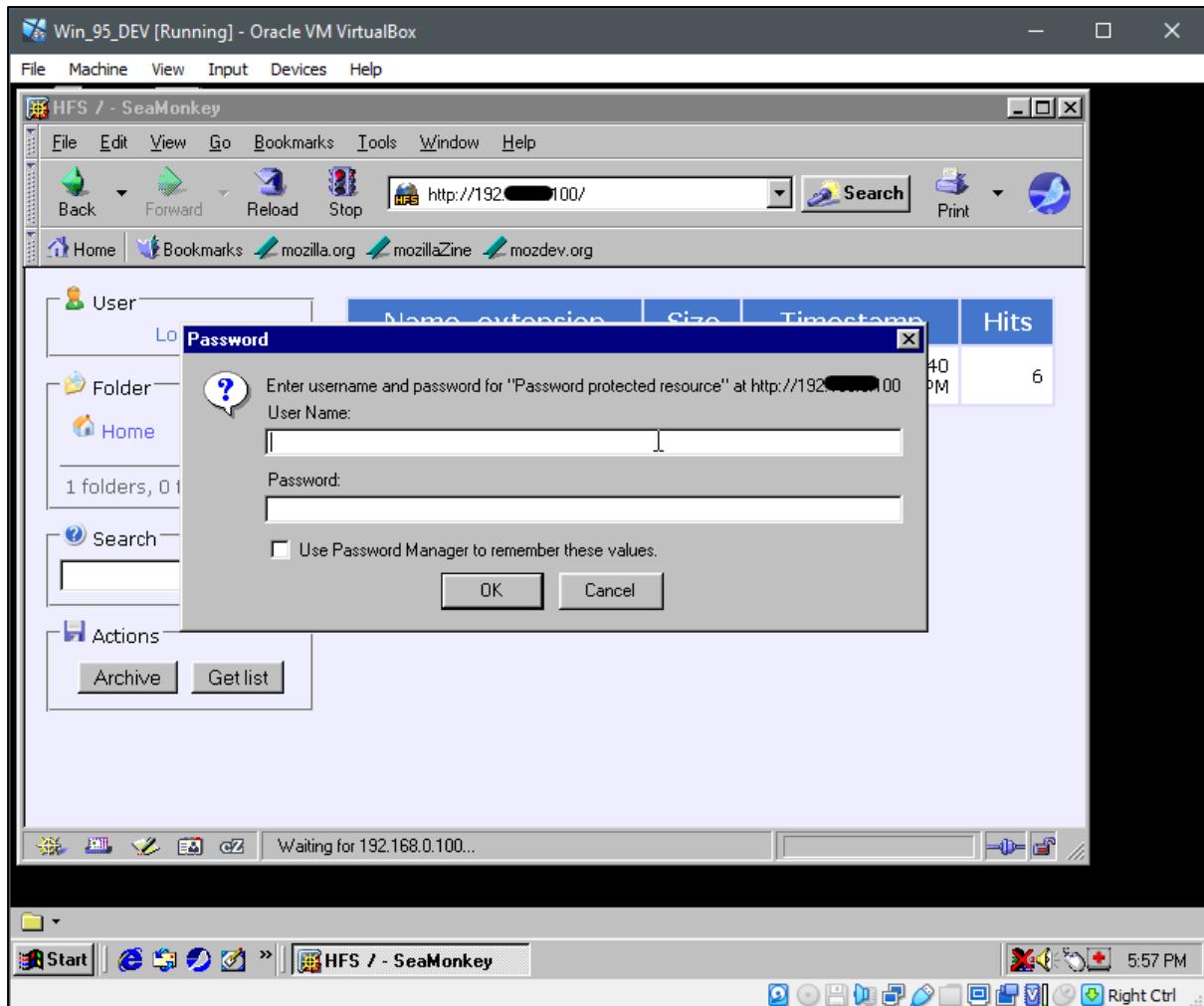
Open the web browser. And add the url of the HFS web server and then [Enter].



You are now at the HFS login page. Click on the login link and enter your user name and password.

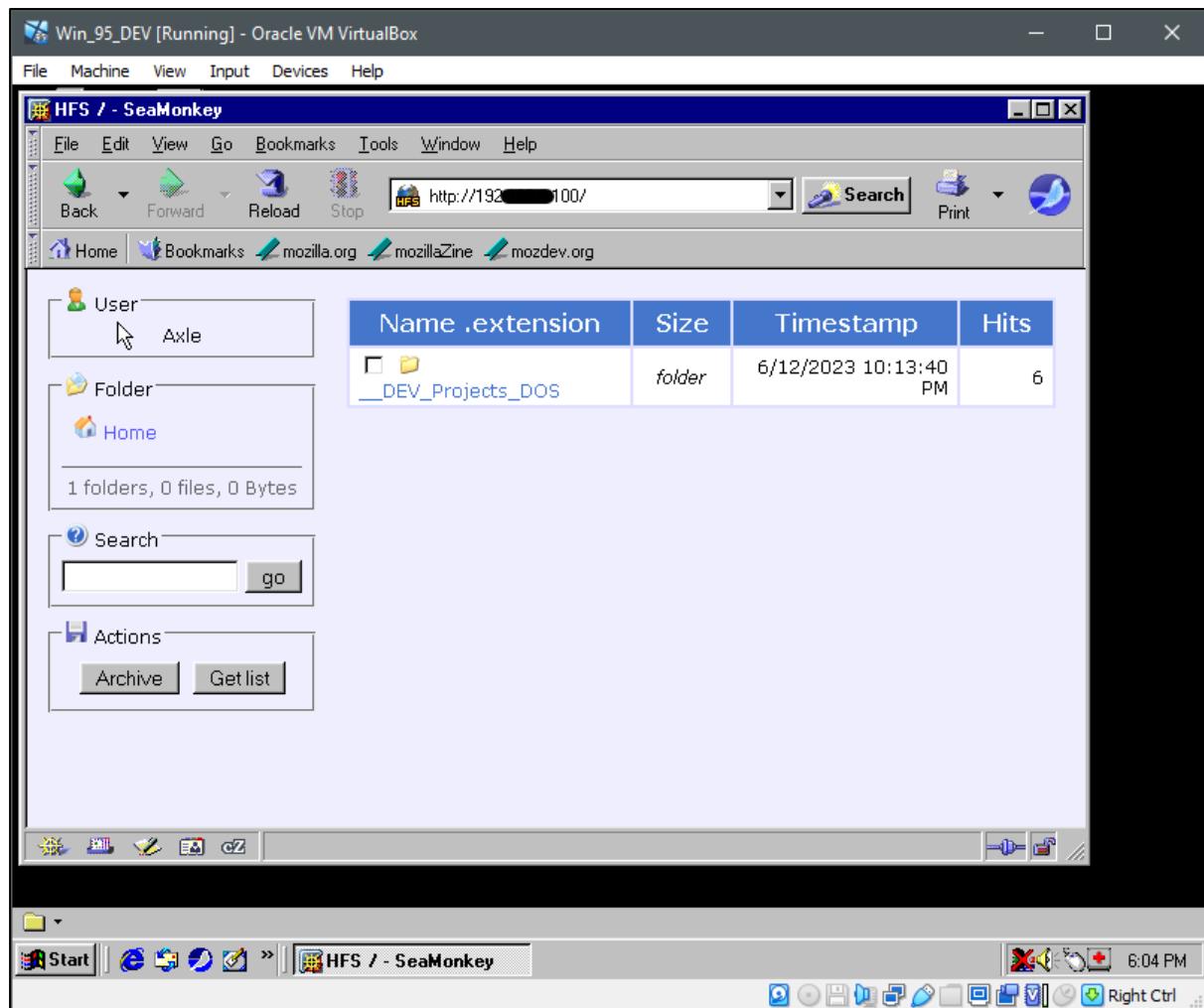
## A Beginners Guide To DOS Programming

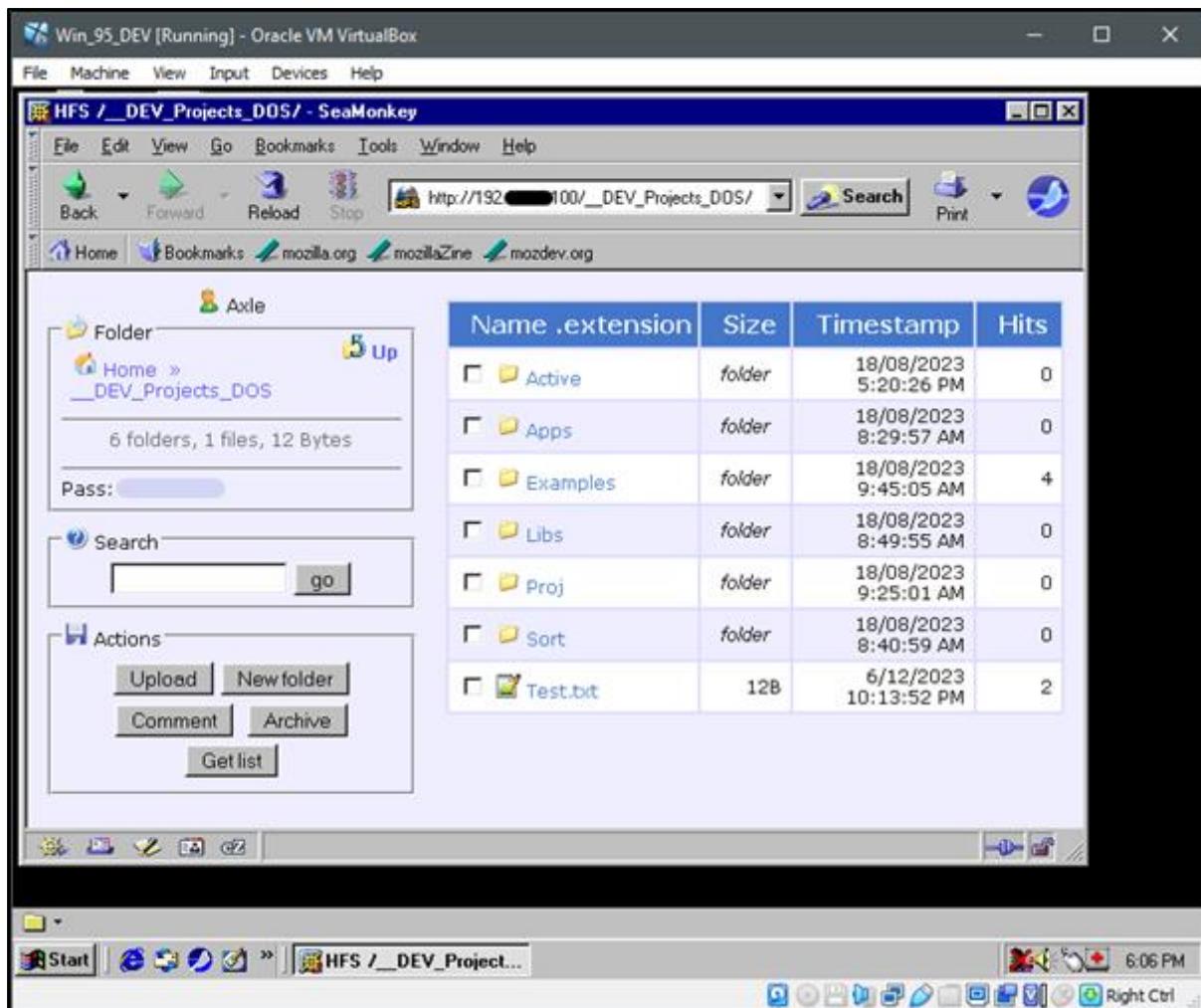




Once you have logged in, browsing the files as well as uploads and downloads are the same as for the original set up tests.

## A Beginners Guide To DOS Programming





That's it, you now have the ability to download from and upload files to your host file server from Windows 95.

## DOS – Windows

This chapter is for reference and understanding the different methods for accessing DOS under Windows 95. You do not have to implement any of the methods at this stage. You can come back and use this as a basic guide should you want to use the DOS development environments or access DOS in a different way to the guide. All Dos development environments will be set up using DOS Windowed or emulated mode.

Windows 95 uses MS-DOS 7.1. Unlike earlier versions of MS-DOS (V6.22) or FreeDOS, MS-DOS 7.1 is not a standalone pure OS as it is heavily integrated into the Windows OS. In essence Windows and DOS can switch back and forth between “Windows Mode” and “DOS Mode”. To make this concept even a little more complicated Windows can “Emulate” DOS in a virtual machine while remaining in

Windows (Windowed Command Console). In every case Windows 95 exercises some degree of control over the DOS environment and configuration.

We can launch a DOS context in a number of different ways which requires some amount of thought before launching a DOS application. As such I cannot predict or offer an example of every possibility and will offer a generalisation where you will need to fill in the gaps for your specific needs.

### DOS vs Dos Mode vs Windowed Mode

All 3 modes can run many DOS applications but each mode requires a slightly different method to set up the DOS environment and configurations. All 3 typical require an initial DOS environment to be set, as well as additional application specific environments to be set in a batch file for each application.

Booting directly to DOS Mode acts more like a traditional DOS Real Mode environment. We could switch off much of the Windows configurations and only run DOS "Without" Windows. This requires us to correctly set the MSDOS.SYS file (Hard coded) as well as supply the correct CONFIG.SYS and AUTOEXEC.BAT settings as well as balance the former with the environments set in batch files for each application requirement. If we only intended to use DOS this would be fine, but when accessing Windowed mode DOS or exiting Windows into DOS mode it becomes far more complex as we are setting the DOS configurations and environment from DOS as well as from Windows and you will encounter unintended conflicts.

It is safer, although requiring a little more effort in creating the configs for each application, to launch both "DOS Mode" or "Windowed Mode" directly from the Windows desktop. This removes the requirement for writing complex MSDOS.SYS, CONFIG.SYS and AUTOEXEC.BAT files.

Windowed Mode will run most application in most instances. Windows sets the initial DOS environment and we add additional environment settings via the applications batch file.

Going into DOS Mode from Windows is much the same except we are using DOS rather than a windowed emulator. As with windowed mode Windows sets the initial DOS environment and we add additional environment setting via the applications batch file.

Now, to add an additional layer of complexity Windows sets the initial DOS environment from its default windows settings regardless of how we launch a DOS instance. We are then required to place all of the DOS global settings as well as individual settings into the applications launching batch file.

We can take control over the initial Windows DOS configuration using Program Information Files (.PIF). The PIF file will set the initial DOS Configuration, global environment variables and Memory resources if required. If we edit the "Properties" of a DOS application or Batch file Windows will create a PIF file to launch the application with the settings you set in "Properties". PIF files are a little like shortcuts or links to a DOS applications that provide additional information. You can have multiple PIF files each linking to an individual DOS application, each with its own set of DOS configurations.

PIF, Batch files and the DOS application all work together as a combined unit when launching a DOS application. You can use a PIF alone to launch a DOS application, but I tend to use a batch file as well to keep some conformity with the traditional DOS environment. The pathway to launching a DOS application from DOS and Windows is as follows.

### Pure DOS:

1. Set DOS “global” configs and environment. (CONFIG.SYS, AUTOEXEC.BAT)
2. Use a batch file to launch the application. The batch file contains specific temporary (local) DOS settings for the application. These setting are applied on top of the global DOS settings. When the application closes the DOS context clears the temporary settings and reverts back to the initial DOS global settings.

### DOS MODE Terminal:

1. The PIF file sets the global DOS configuration (Similar to CONFIG, AUTOEXEC) except it is temporary and the settings are discarded after the DOS Mode ends. This runs the COMMAND.COM application (aka DOS) and all other applications are run from the command line.
2. Use a batch file to launch the application. The batch file contains specific temporary (local) DOS settings for the application. These setting are applied on top of the global DOS settings. When the application closes the DOS context clears the temporary settings and reverts back to the initial DOS global settings in DOS COMMAND.COM terminal.
3. When the Terminal is closed via Exit. The global settings of the PIF file are discarded and control is returned to Windows.

### DOS Mode Application:

1. The PIF file sets the global DOS configuration (Similar to CONFIG, AUTOEXEC) except it is temporary and the settings are discarded after the DOS Mode ends. This runs the application directly in an instance of COMMAND.COM.
2. The PIF is linked to the “batch file” to launch the application. The batch file contains specific temporary (local) DOS settings for the application. These setting are applied on top of the global DOS settings. When the application closes the DOS context clears the temporary settings and reverts back to the initial DOS global settings in DOS COMMAND.COM terminal.
3. When the application is closed via Exit it also automatically closes the COMMAND.COM instance. The global settings of the PIF file are discarded and control is returned to Windows.

It is also possible to use the PIF to link directly to the DOS application by adding the additional setting from the applications batch file to the end of the AUTOEXEC section of the PIF. I find that using the PIF only for the global DOS settings and using the batch file to launch the application keeps a degree of continuity with launching the application directly from the command prompt. It just means that we can launch the application easily from a DOS Mode context or directly from Windows via the batch file.

### DOS Windowed:

DOS Windowed (Emulated) is set up the same way as “DOS MODE Terminal” and “DOS Mode Application” except that it is running inside of a command emulator in Windows. Windows retains control over all configurations and memory use. Some older “Real Mode” DOS applications need direct access to the computer hardware and will not run correctly under Windows, so we need to run them in a more pure DOS environment outside of Windows.

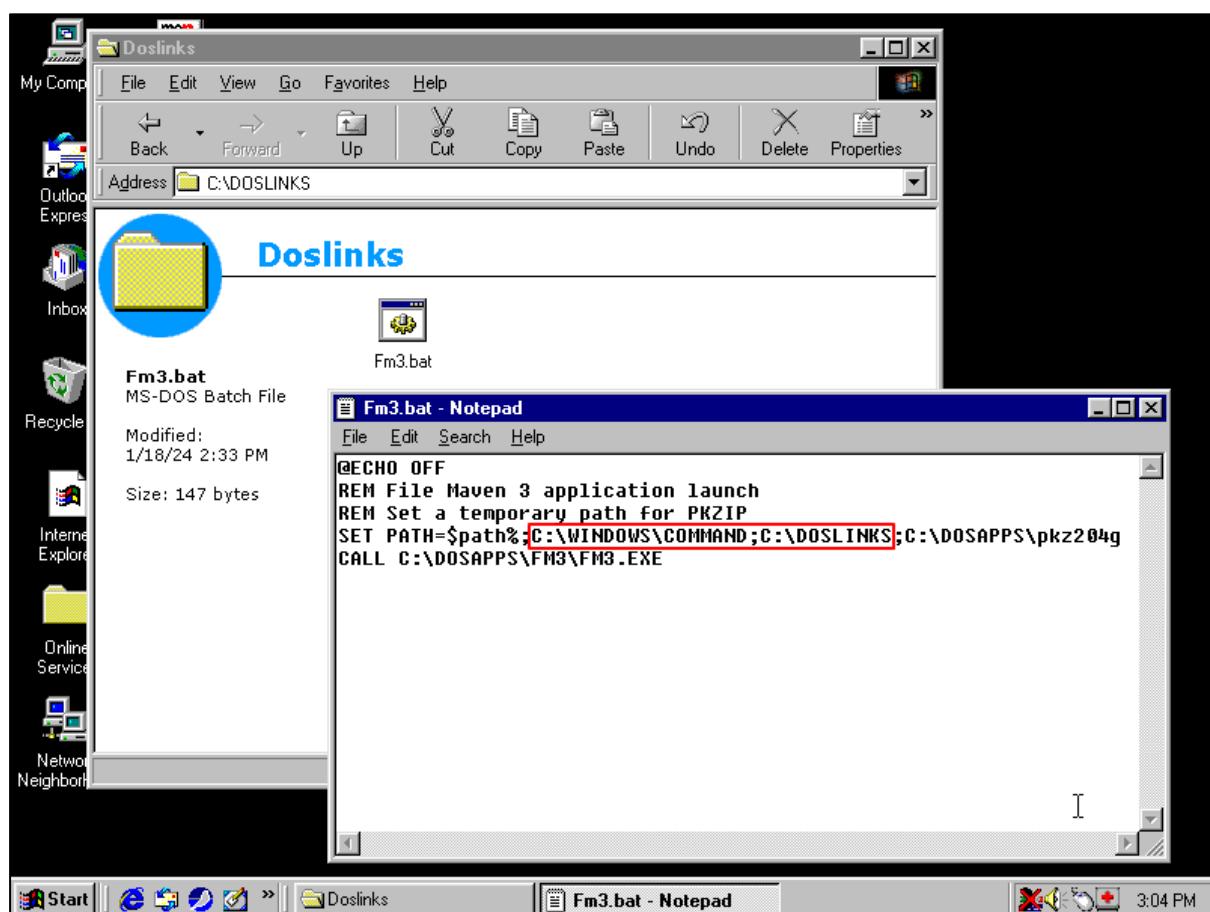
It all may appear complicated at first, but once you do it a few times and create a template for your initial PIF DOS environment it is as fast as creating any other Windows desktop shortcut.

An example of launching File Maven 3 via a batch file. The initial DOS environment is set by Windows and is missing some configurations for memory, sound and mouse as well as paths for the global DOS binaries. I have had to set the path for the DOS files when launching FM3.

The first example is using “DOS Mode”.

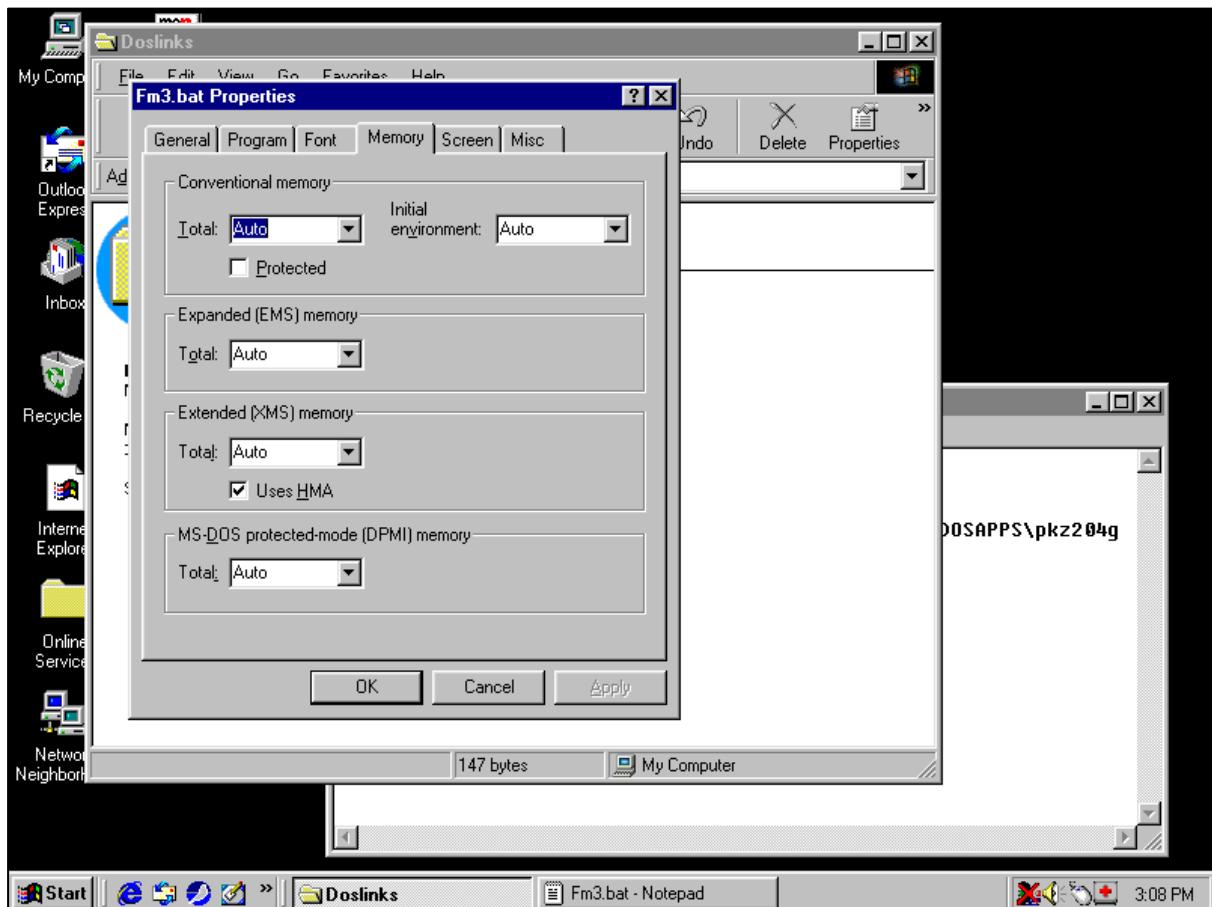
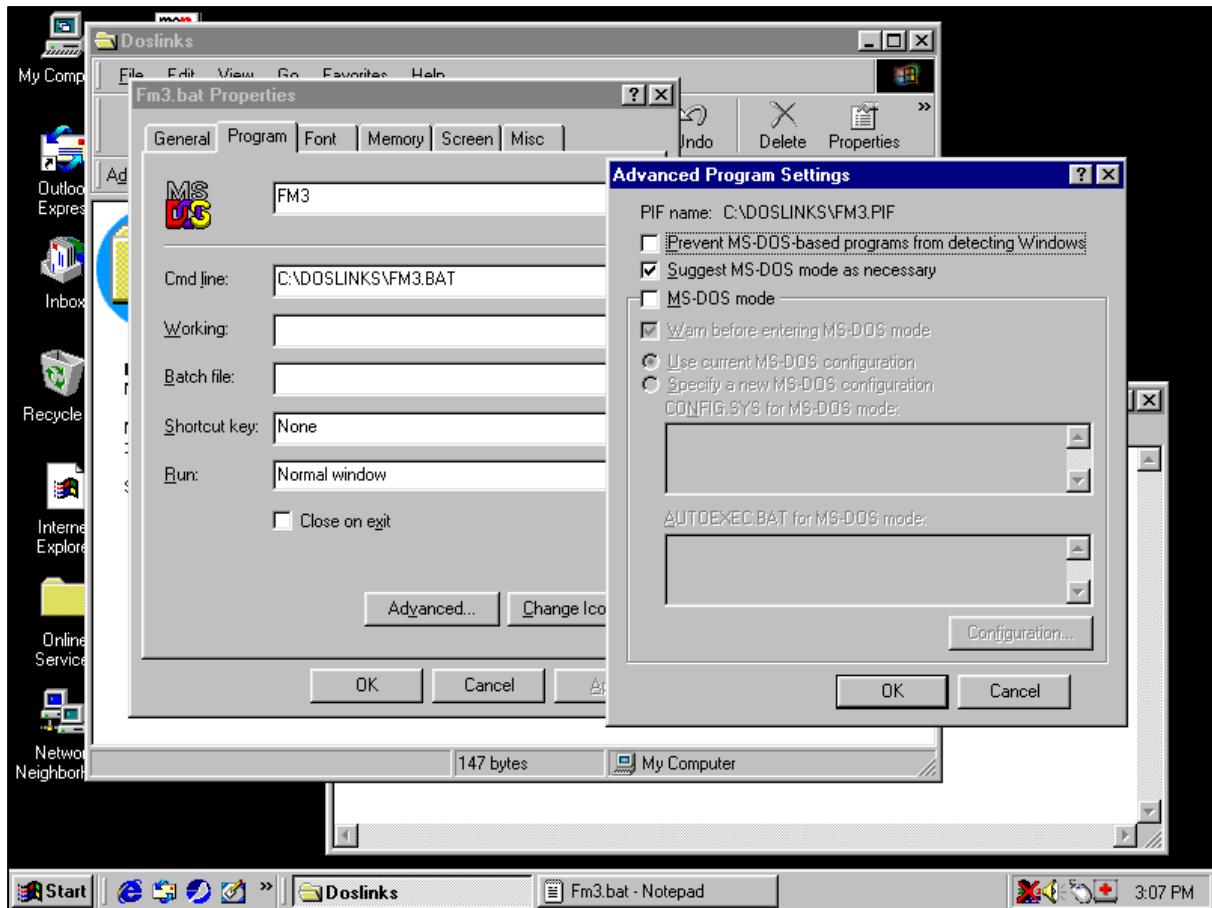
```
SET PATH=%path%;C:\WINDOWS\COMMAND;C:\DOSLINKS
```

The above line should be set as part of the global settings when DOS is launched, so we need to remove it from the batch file. When working in windowed mode the default \COMMAND directory is not used so you will need to have it in your batch files for windowed mode.



The Windows “Properties” for FM3.BAT only uses the very basic default DOS configs for the DOS application.

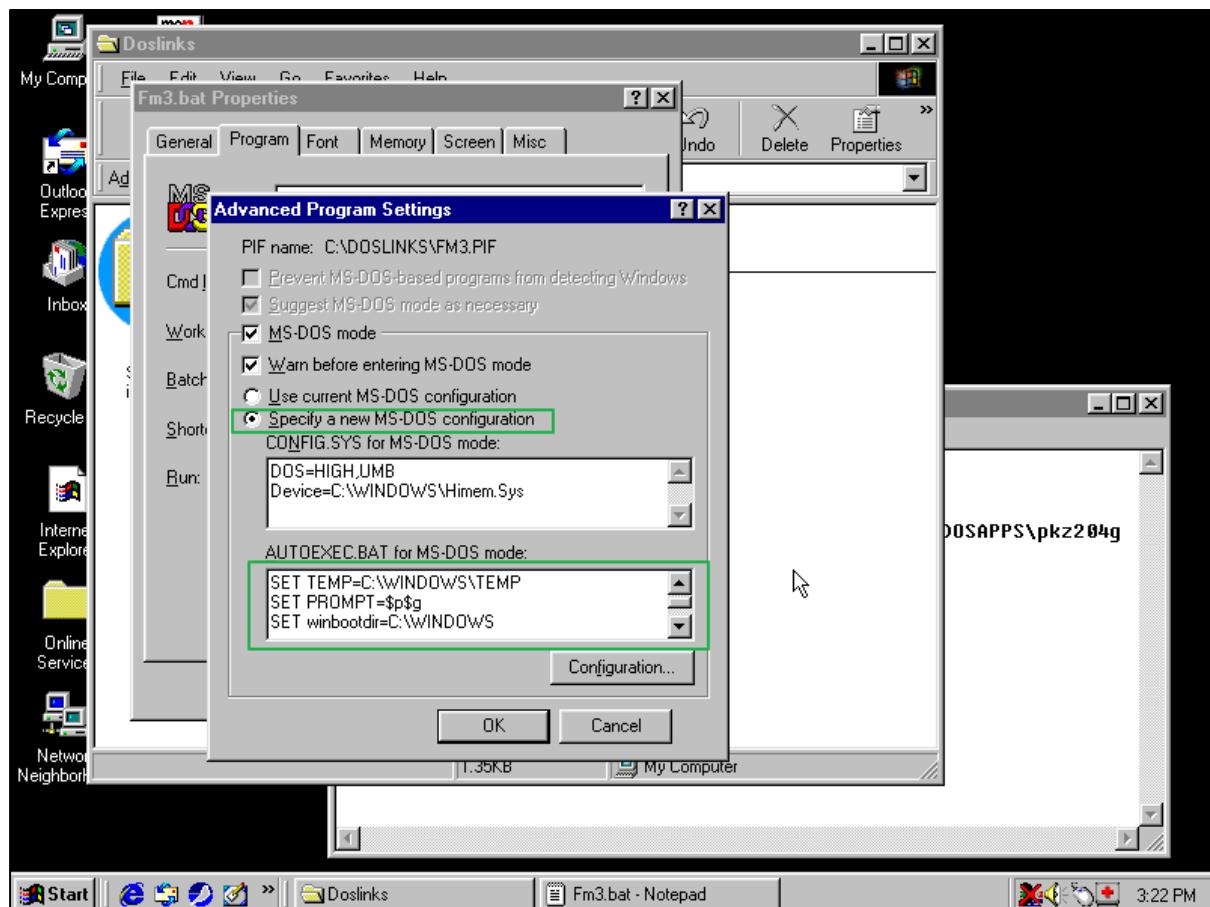
## A Beginners Guide To DOS Programming



Let's correct the initial DOS environment setting and create a PIF.

Note we are setting a "DOS Mode" environment.

From the Program tab select Advanced. Select "Specify a new MS-DOS configuration..." Windows will use its default setting which are insufficient. These default settings are what are used when launching any DOS applications without a PIF. Sometimes additional configs may be added depending upon previous PIFs that you have created.



The default for Windows:

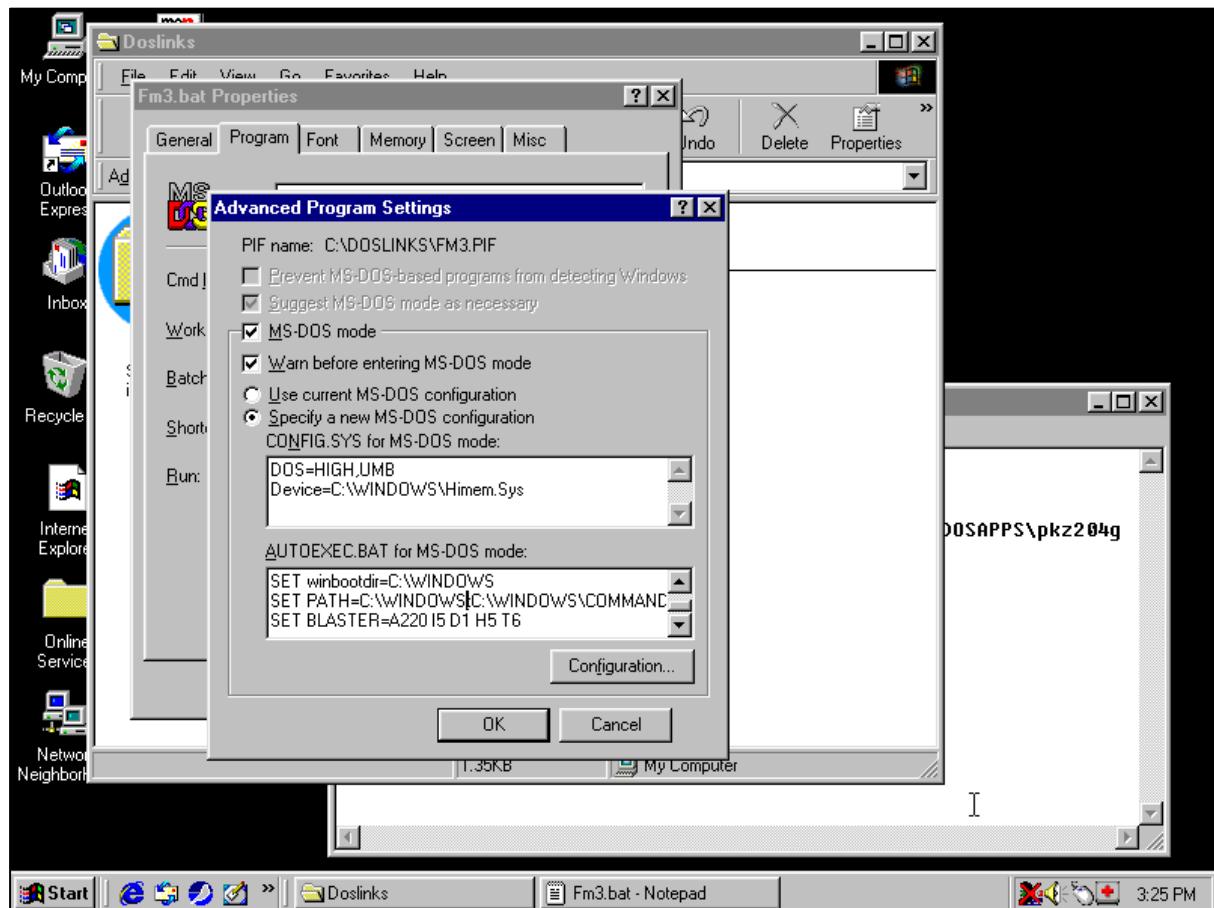
```
SET TMP=C:\WINDOWS\TEMP
SET TEMP=C:\WINDOWS\TEMP
SET PROMPT=$p$g
SET winbootdir=C:\WINDOWS
```

Add the following lines to include the basic DOS paths to the binaries.

Set the "Global" paths for the DOS environment. I have also added a directory for our batch files to launch applications. Don't worry about the SET BLASTER= line as I will cover this in a little more detail in the next chapter.

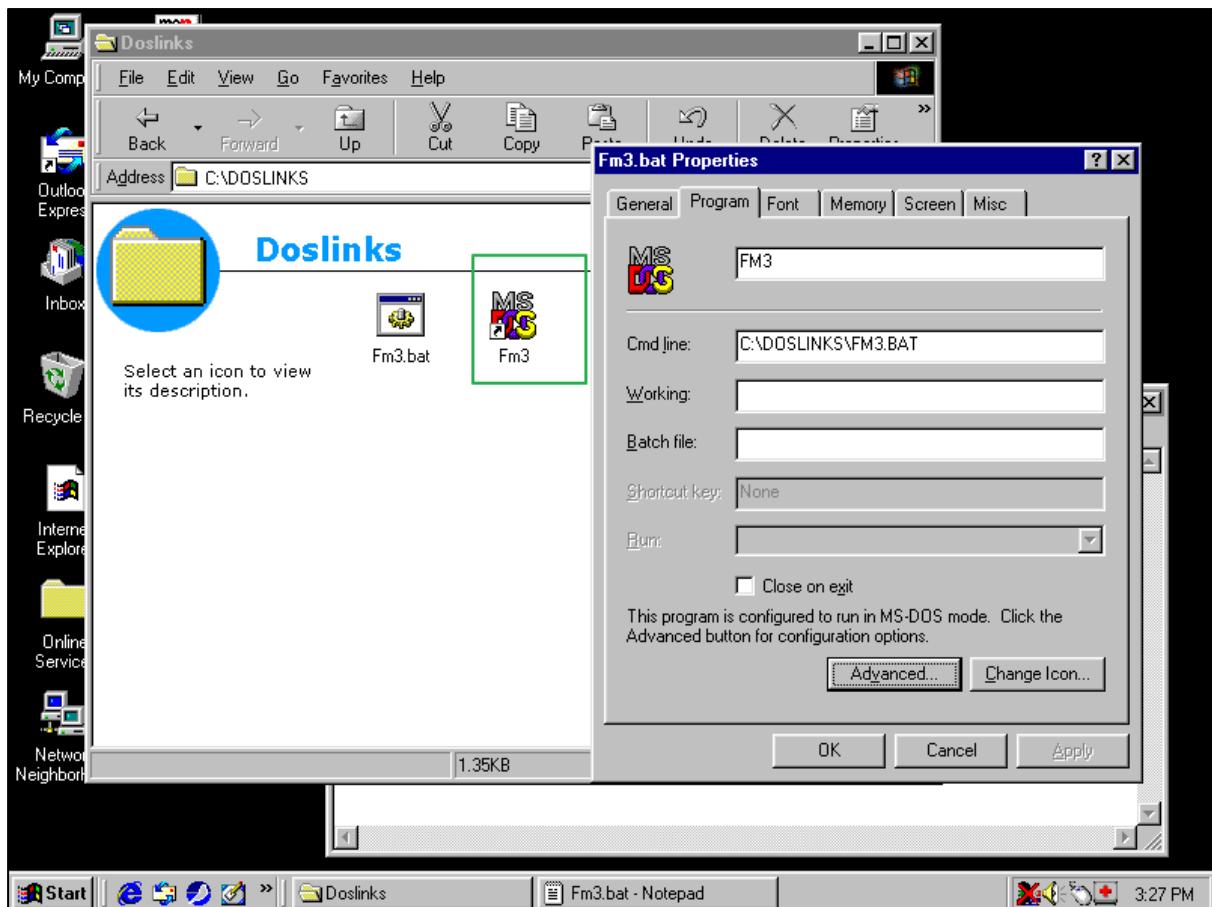
```
SET TMP=C:\WINDOWS\TEMP
```

```
SET TEMP=C:\WINDOWS\TEMP
SET PROMPT=$p$g
SET winbootdir=C:\WINDOWS
SET PATH=C:\WINDOWS;C:\WINDOWS\COMMAND;C:\DOSLINKS
REM CTMOUSE /R55
SET BLASTER=A220 I5 D1 H5 T6
```



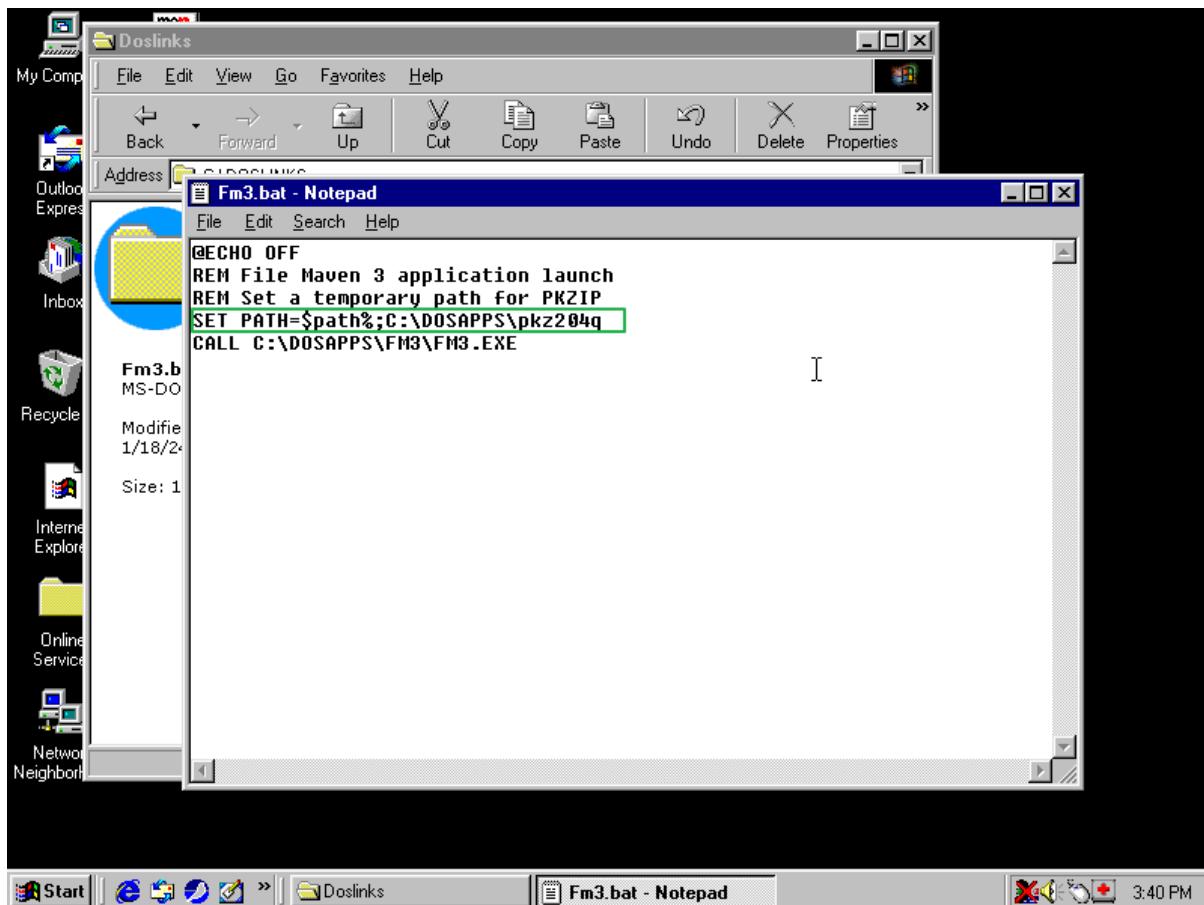
Select OK to save this config.

If you look back to the directory containing FM3.BAT you will notice the new configs have been created in an FM3.PIF file.



The FM3.BAT and FM3.PIF will show the same setting in "Properties". They are both linked as if they were the same file.

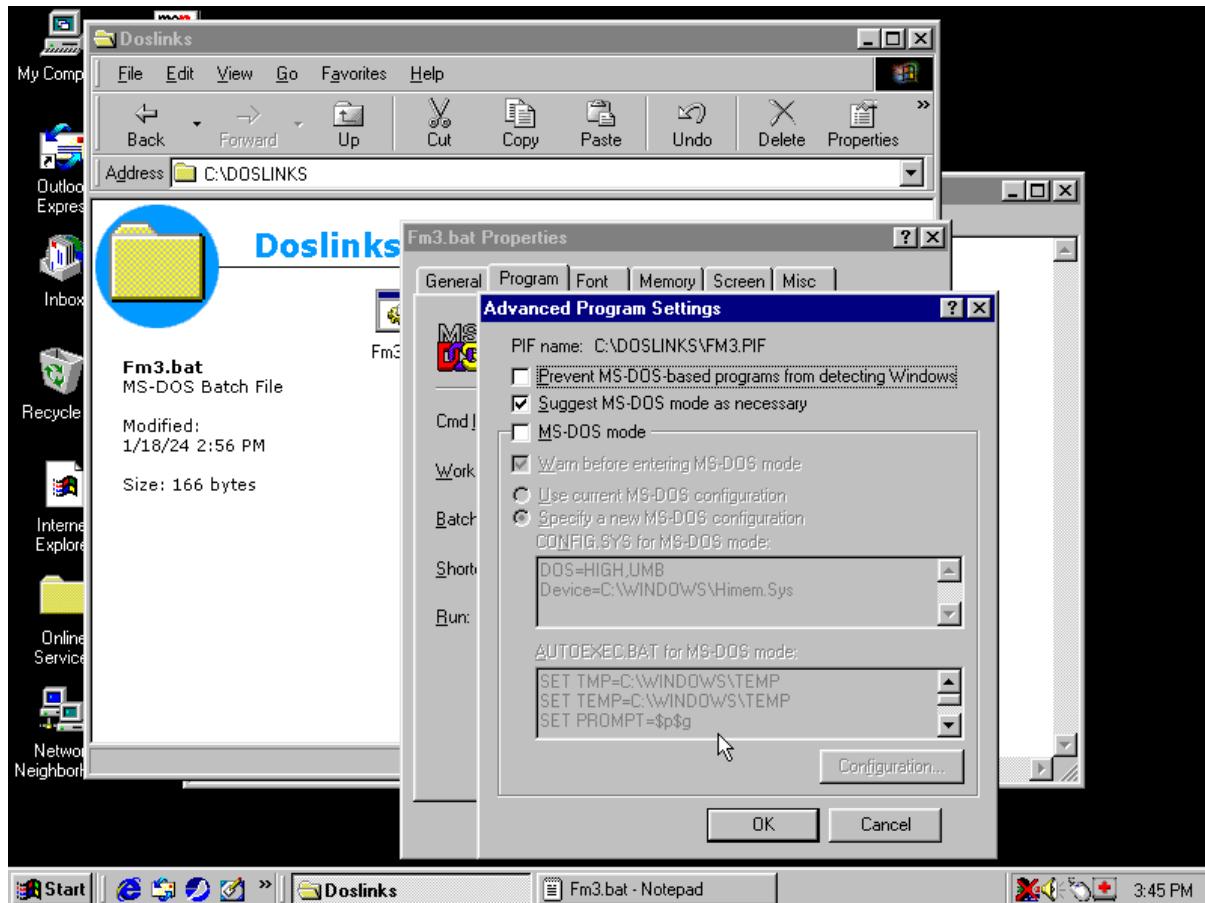
Remove the global paths from our batch file.



The following is an example of Windows emulated mode.

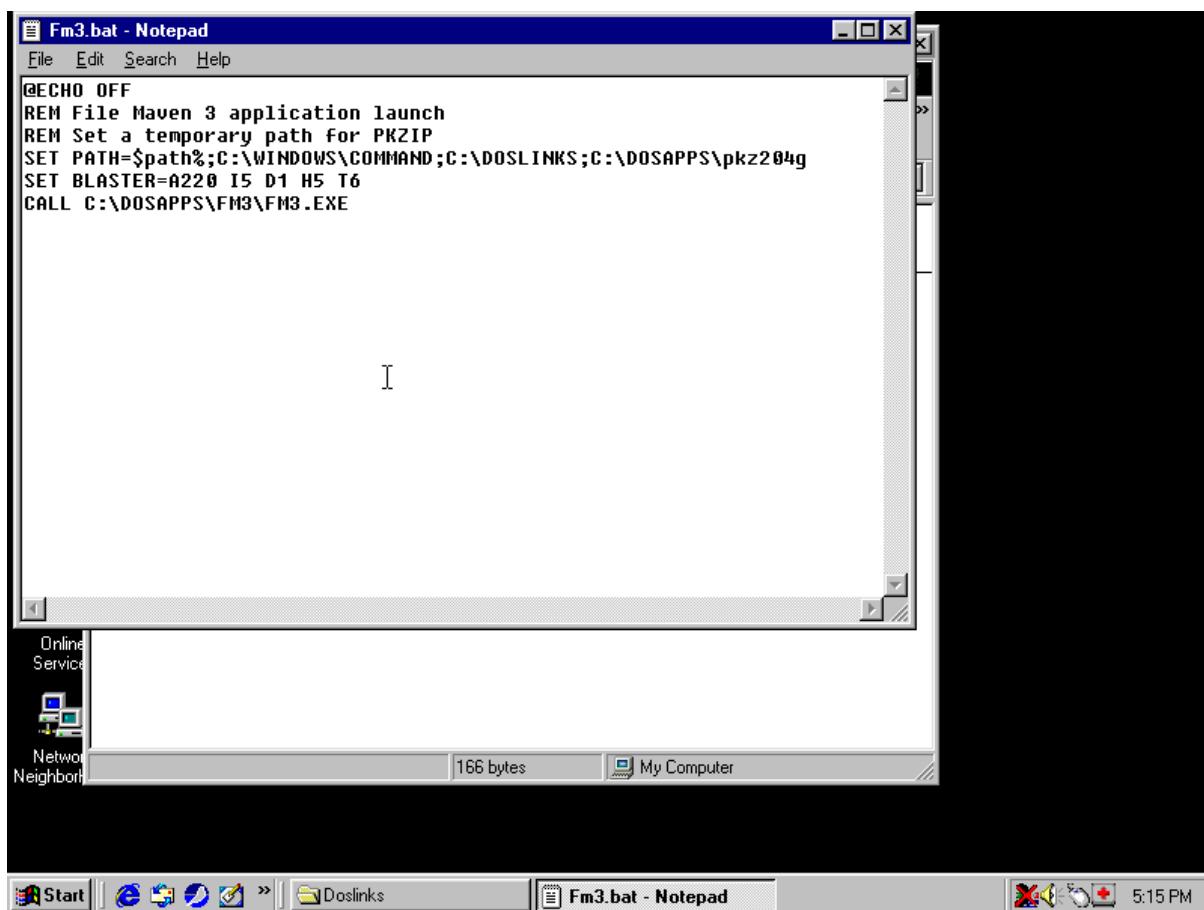
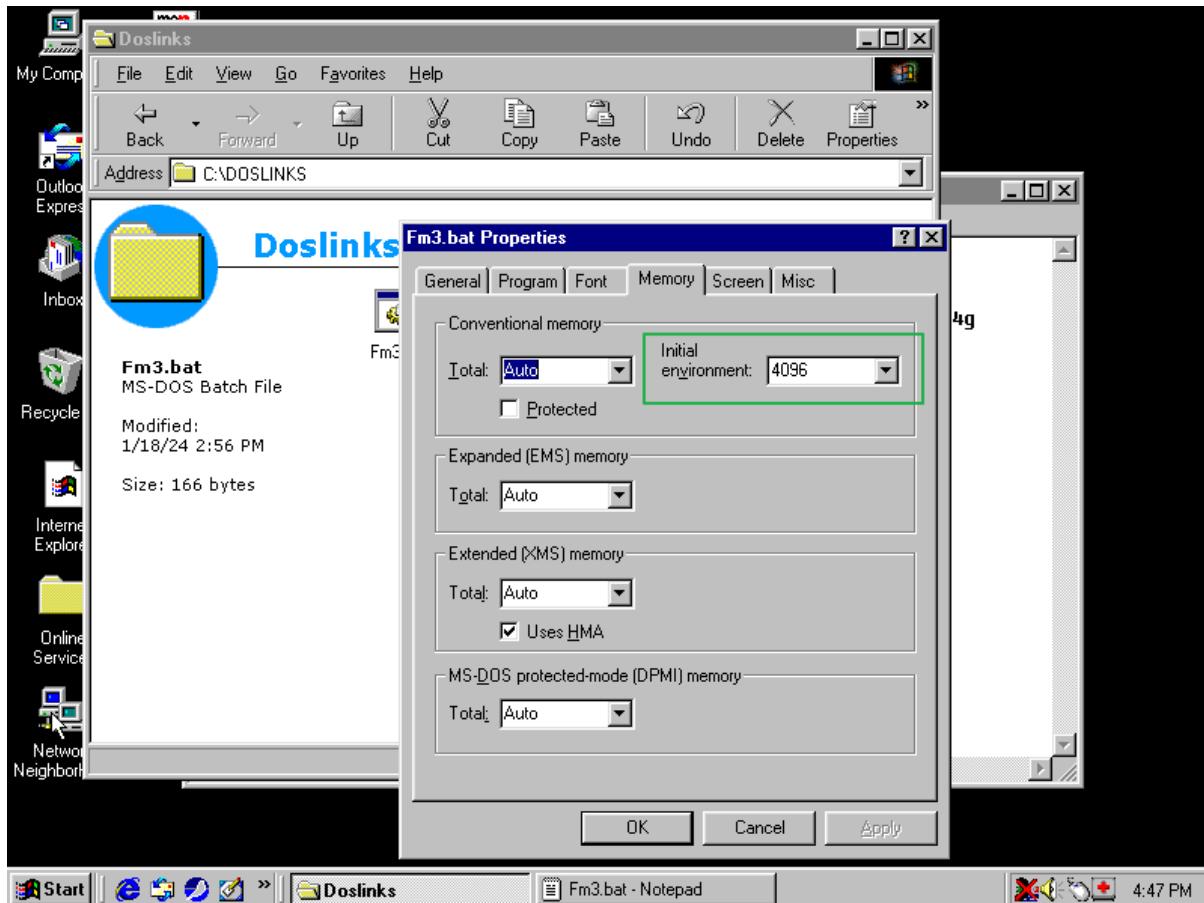
Sometimes we may need to set additional memory requirements for a single application. In this case we can include the additional lines in the CONFIG, or use the current MS-DOS configuration (generic) and set the memory in the PIF file and add the additional path environment to the application batch file.

An example of using the PIF to set the initial memory environment to allow an application more RAM space for variables. Setting 1024 or up to 4096 is a common requirement in some programming environments.



It is fine to leave the other memory setting as Auto unless your application has a specific need. Note Windows uses its own mouse drivers in emulated window mode so don't add the CTMOUSE /R55 anywhere.

## A Beginners Guide To DOS Programming



There are many ways to enter the DOS environment from Windows, but the above are a few common examples.

In the next chapter I will explain a little more about the different DOS modes under Windows, but I find it best to follow along the above method.

This guide is a supplementary to the FreeDOS guide on using development environments so the main focus is upon setting up the FreeDOS tools where they can be used with the convenience of being able to open additional documents such as programming guides while using the DOS tool sets. Some of the essential DOS techniques such as navigation and setting up batch files are assumed from Book 1.

---

## Booting to MS-DOS real mode.

This chapter is for reference and understanding the different methods for accessing DOS under Windows 95. You do not have to implement any of the methods at this stage. You can come back and use this as a basic guide should you want to use the DOS development environments or access DOS in a different way to the guide. All Dos development environments will be set up using DOS Windowed or emulated mode.

This is a continuation of “DOS – Windows”.

Booting into MS-DOS real mode can be a little difficult. We have at least 4 choices. I would recommend only using choice A. unless you plan on using MS-DOS as your primary OS (Not recommended).

- A. Use the “Restart in MS-DOS mode” from Windows. (Recommended)
- B. Use F8 at boot and select “6. Command prompt only”.
- C. Change BootGUI=1 to BootGUI=0. Use Win to start Windows.
- D. Make the system into a dual boot system using the Config.sys and Autoexec.bat with a menu to boot into DOS or Windows.

Each of the ways have their pros and cons, but A. is the most simple. The later B, C. and D. are a little more complex as windows is a multi-boot (DOS, Windows) system where the multi-boot parameters are controlled by the Windows start up files. This is a little more complex than a single DOS only boot.

<https://www.vogons.org/viewtopic.php?t=41865>

Note:

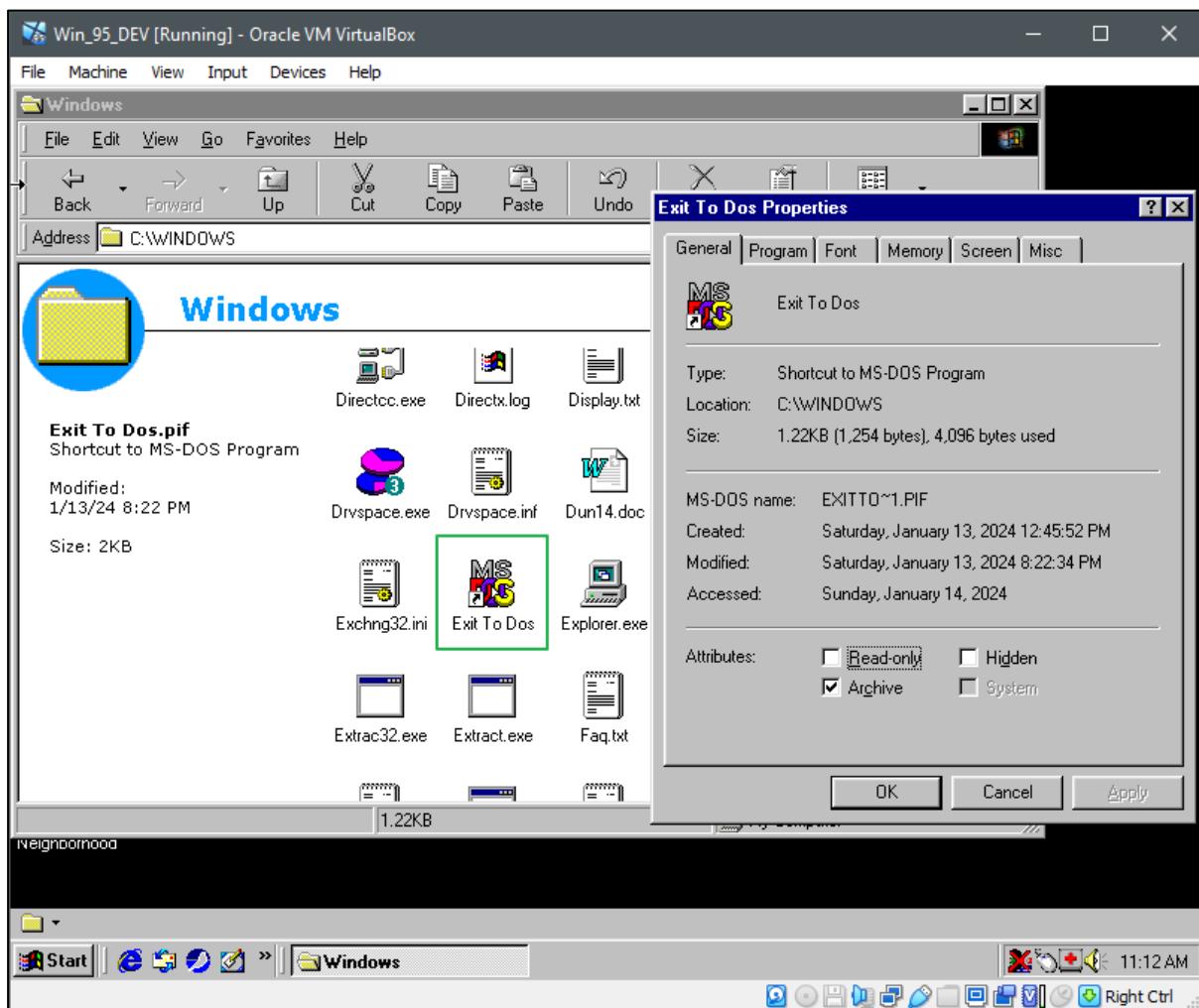
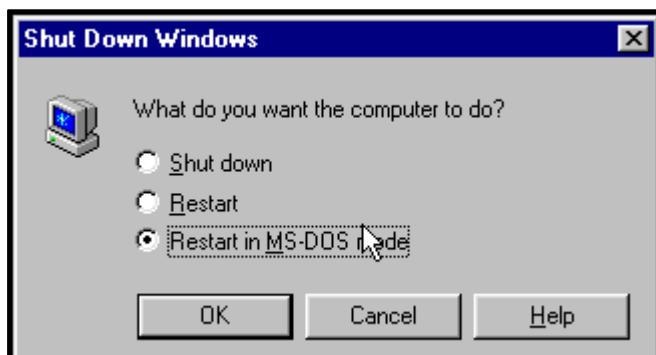
It is not uncommon for windows to stall or freeze when changing between Windows and DOS

modes. The only option here is the “Reset” the virtual machine (Cycle the power off and back on).

### A. Restart in MS-DOS mode

This is the most simple method and exits a running windows context to MS-DOS. You can later restart Windows by typing Exit or Win at the command prompt.

Windows 95 has an option in the shutdown menu named “Restart in MS-DOS mode”. This makes use of a Windows based MS-DOS Program Information File (.pif) that resides in the C:\WINDOWS directory named “Exit to DOS.pif”.

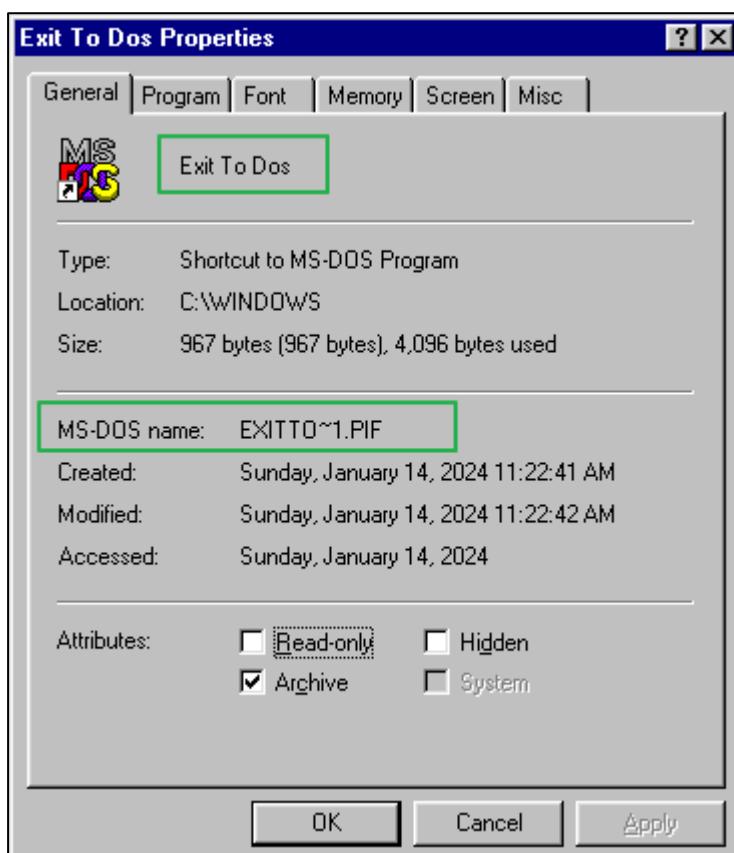


If this file does not exist you can select the “Shutdown -> Restart in MS-DOS mode” path and the file will be created. Windows will shut down to DOS mode and the DOS instance will fail with nothing but a blinking cursor on a black screen. Use “Ctrl + Alt + Del” to restart the machine or use the VirtualBox setting to reset/Restart the Windows 95 virtual machine. After the restart the “Exit to DOS.pif” should exist in the \WINDOWS directory.

If that fails you can create the file yourself by either copying another pif file such as “Dospromt.pif” and renaming it to “Exit to DOS.pif”. You will need to edit its properties to contain the information shown below. Or you can also create pifs by changing the properties of a blank batch file for example “dummy.bat”. It is easier to copy and rename one of the many \*.pifs that already exist.

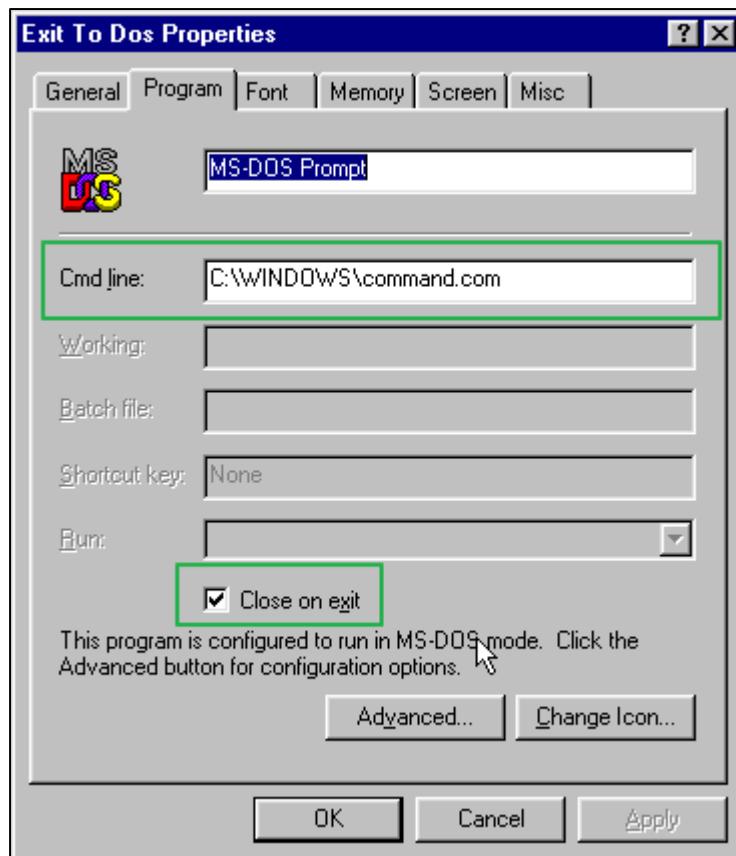
The “Exit to DOS.pif” only exists with that name in the Windows directory so that it aligns with the shortcut on the Windows shutdown menu. You can create a PIF file anywhere such as the desktop with any name.

Open the Properties dialog for the “Exit To DOS.pif”. Make sure the following settings are selected to be able to use the DOS mode. This assumes that the 2 system files in the C: root directory CONFIG.SYS and AUTOEXEC.BAT are empty (recommended).



In the program Tab the command line C :\WINDOWS\command . com is important as this is the application that will be run after Windows is shut down.

The Close on Exit is also important.

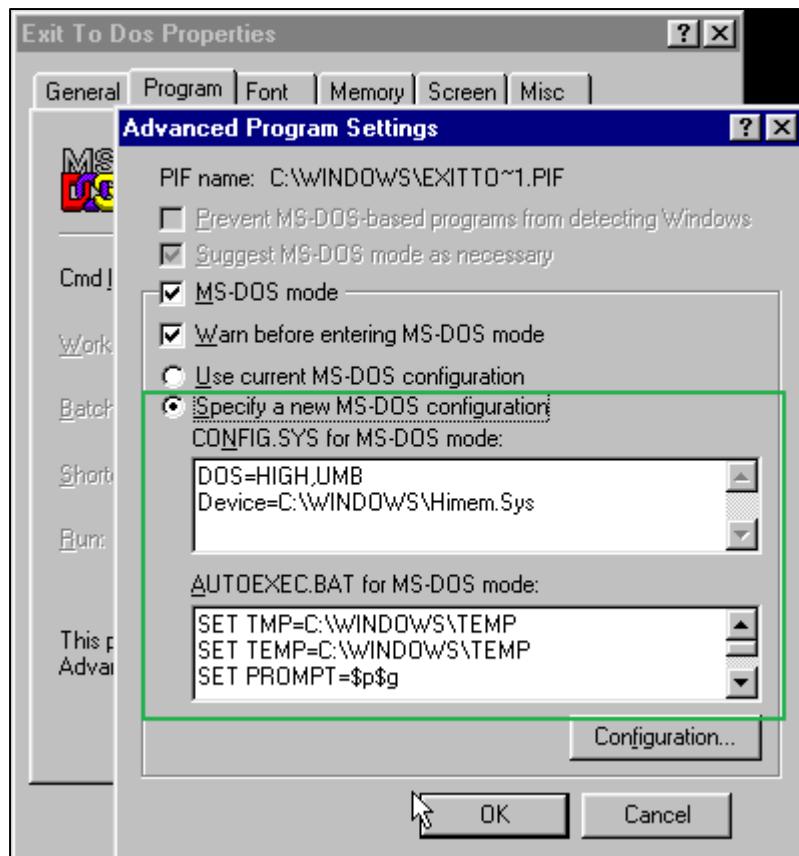


Next click on the “Advanced ...” Button. This is where we will need to set the DOS start up parameters to boot into DOS. DOS relies upon the CONFIG.SYS and AUTOEXEC.BAT files for the correct boot parameters. Windows also makes use of a few hidden files for this being a dual boot system controlled by Windows.

We have 2 options; 1. Make use of the default CONFIG.SYS and AUTOEXEC.BAT in the C: root directory (Not recommended), or 2. create a Windows based emulation of the 2 start-up files inside of the PIF. If you are running a genuine duel boot MS-DOS + Windows system (Difficult and not recommended) we can use option 1. For this part we will use option 2. as it does not require us to alter any of the default Windows start up files.

Warn before entering MS-DOS mode is optional but sometimes a warning is helpful to remind you to close all windows applications first.

The next is the important part to be able to use (Exit to) MS-DOS. Select the “Specify a new MS-DOS configuration”. The “Use current MS-DOS configuration” requires that we have already set up the default CONFIG.SYS, AUTOEXEC.BAT as well as other Windows boot parameters to behave as a proper duel boot system which I don’t recommend.



CONFIG.SYS	
<b>DOS=HIGH,UMB</b>	
<b>Device=C:\WINDOWS\Himem.Sys</b>	
AUOTEXEC.BAT	
<pre>SET TMP=C:\WINDOWS\TEMP SET TEMP=C:\WINDOWS\TEMP SET PROMPT=\$p\$g SET winbootdir=C:\WINDOWS SET PATH=C:\WINDOWS;C:\WINDOWS\COMMAND REM The following is an alternative that includes the path for REM application start up files. REM PATH=C:\WINDOWS;C:\WINDOWS\COMMAND;C:\DOSLINKS SET BLASTER=A220 I5 D1 H5 T6</pre>	

Windows automatically fills in the required minimum CONFIG and AUTOEXEC for the system. These are the setting used to re-boot into MS-DOS. The CONFIG and AUTOEXEC files in C:\ are ignored.

If we have a look at some typical DOS boot configurations you will find other lines added to the above such as sound card and mouse drivers. It is also common to include additional memory configurations. The above is a minimal setting to boot into DOS.

See the following section for adding Mouse and CD-ROM drivers for DOS.

Select [OK] to keep the custom settings. Select [Apply] if it is highlighted in black the select [OK] to close the properties dialog for the PIF file.

Make a copy of the “Exit To DOS.pif” and place it in a safe location such as a folder on the desktop or storage drive. We can use this as a template for more advanced MS-DOS configurations. The “Exit To DOS.pif” only has to reside in the \WINDOWS directory with that name to match with the Shutdown menu for windows. We can create a PIF file anywhere with any name and the correct parameters to restart in MS-DOS mode.

Try 2 tests:

Use “Shutdown -> Restart in MS-DOS mode” from the Windows start menu. This uses the “Exit To DOS.pif” in the WINDOWS directory. Once you are in the DOS command line type Exit or WIN to restart Windows and leave MS-DOS mode.

After Windows has restarted, this time use your copy of “Exit To DOS.pif” that you saved. You should exit windows to MS-DOS in the same way.

**NOTE:** Sometimes the DOS terminal can become unresponsive and you cannot type anything to the screen, This is a glitch that occurs between DOS and VirtualBox. Just reset the virtual machine and go back to DOS again.

### Note!

This is slightly different to how we set up the PIF file when running the command line in Windows protected mode (Windowed while Windows is still running).

Take note of the difference between DOS Real mode 16-bit, DOS Protected mode 32-bit and Windows Protected mode 32-bit (with Emulated DOS 16/32).

### Create a custom PIF with drivers and Memory tweaks.

#### Audio Drivers

Not required in DOS Windowed mode.

Typically the Audio Drivers will already be set from the Windows defaults. If the following line does not exist you can copy it into the AUTOEXEC.BAT file.

SET BLASTER=A220 I5 D1 H5 T6

I5 is the hardware interrupt and can sometimes be shared by other devices such as printers. I5 and I7 are common interrupt request values for audio cards.

### Mouse Drivers

Not required when using DOS windowed mode.

You will need a copy of CuteMouse (CTMOUSE). The drivers installed in Windows will not work with DOS Mode. You can download it from <https://cutemouse.sourceforge.net/>

You only need the ctmouse.exe in the C:\WINDOWS\COMMAND\ directory.

#### DOS Mouse support

Download the CUTEMOUSE drivers from:

<https://cutemouse.sourceforge.net/>

Alternatively you can copy the binary from FreeDOS.

You only need CTMOUSE.COM to make use of the mouse in DOS.

Unpack the downloaded archive and copy the following files to your Windows 95 drive.

cutemouse21b4\bin\ctmouse.exe

to

C:\WINDOWS\COMMAND\ctmouse.exe

Optional

Copy the full unpacked directory to C:\DRIVERS1\CTMOUSE\\*.\* for reference.

Make a new copy of the “Exit To DOS.pif” on your desktop. Rename it to something suitable such as CUST\_DOS.pif or DEV\_DOS.pif. You can actually create as many alternative DOS environments as you want as long as you give each \*.pif a unique name.

Open the Properties dialog and add the following lines to the CONFIG and AUTOEXEC sections.

We will allocate some more BUFFER space and allow more open FILES. I am also going to give the COMMAND SHELL some additional space for environment variables with the /E:[Size] switch. The initial environment variable memory allocation can be anything between 256 and 4096. 1024 is common but I find that for some development environments I need to use 4096.

I have also included the start up for the mouse driver. The /R55 switch sets the mouse x,y pixel sensitivity percentage for the mouse.

The following settings are very generic and will cover a lot of DOS applications. You will need to be aware of additional memory settings (EMS, XMS, DPMI, etc.) when required. There is no “One size fits all” config file as every system and application has different requirements. The following is generic and will cover most ground to get started.

Add the following lines to CONFIG

BUFFERS=40

FILES=40

REM STACKS=9,256

SHELL= C:\WINDOWS\COMMAND\COMMAND.com C:\WINDOWS\COMMAND\ /E:4096 /P

and the following to AUTOEXEC

CTMOUSE /R55

### CONFIG.SYS

```
DOS=HIGH,UMB
Device=C:\WINDOWS\Himem.Sys
BUFFERS=40
FILES=40
REM STACKS=9,256
LASTDRIVE=Z
SHELL= C:\WINDOWS\COMMAND\COMMAND.com C:\WINDOWS\COMMAND\ /E:4096 /P
```

### AUOTEXEC.BAT

```
SET TMP=C:\WINDOWS\TEMP
SET TEMP=C:\WINDOWS\TEMP
SET PROMPT=$p$g
SET winbootdir=C:\WINDOWS
SET PATH=C:\WINDOWS;C:\WINDOWS\COMMAND
REM application start up files.
REM SET PATH=C:\WINDOWS;C:\WINDOWS\COMMAND;C:\DOSLINKS
SET BLASTER=A220 I5 D1 H5 T6
CTMOUSE /R55
```

### From Book - 1

#### CuteMouse

CuteMouse is the default mouse driver for FreeDOS. Specified in FDAUTO and loaded with default setting at boot time. Sometimes the default mouse settings can be too fast, jumpy to control depending upon the screen mode TEXT or VESA.

You can alter this by adding or changing some extra switch options to the mouse start up in the FreeDOS FDCONFIG.BAT file.

From the CTMOUSE .TXT shipped with the drivers in C:\FREEDOS\CTMOUSE or from <http://help.fdos.org/en/hhstndrd/ctmouse.htm> we will find the /R switch.

*/R[h[v]] - horizontal/vertical resolution (h,v=0-9; missing option R, no arguments or 0 as argument after R mean auto resolution, missing*

We can set the mouse resolution by adding /R11, /R19, /R46 etc. where the first value is the horizontal and the second value is the vertical.

FDAUTO.BAT contains a number of different places where CTMOUSE is called so you will need to add the /Rnn switch to all of them for consistency. I have found that /R33 has offered a reasonable balance on my system but each system is different and some amount depends upon how you have set the speed settings for the mouse on your host system. Try a few /Rnn settings with a reboot after each and test it in a graphical interface such as Costa.

#### FDAUTO.BAT (Partial)

```
...
goto NoLFN
:UseLFN
LH %DOSDIR%\BIN\DOSLFN.COM
```

```
set DIRCMD=%DIRCMD% /LFN  
:NoLFN  
  
CTMOUSE /R33  
goto InitCDROM  
  
:Support386Low  
FDAPM APMDO  
CTMOUSE /R33  
  
:InitCDROM  
if not exist %dosdir%\bin\cdrom.bat FINAL  
echo.  
...
```

### CD-ROM Drivers

Not required in DOS Windowed mode.

There are many CD-ROM drivers available for MS-DOS. I find the simplest path is to copy the CD-ROM drivers from the Windows 98 Boot disk.

You will need to copy the following files from the Windows 98 boot floppy drive to C:\WINDOWS\COMMAND\\*.\* directory.

- oakcdrom.sys
- btdosm.sys
- flashpt.sys
- btcdrrom.sys
- aspi2dos.sys
- aspi8dos.sys
- aspi4dos.sys
- aspi8u2.sys
- aspicd.sys
- Mscdex.exe (This is the same as what already exist in Windows 95)

Add the following lines to the CONFIG and AUTOEXEC sections of the pif. Be sure to use the full path as the system PATH environment is not set until the AUTOEXEC file is processed.

#### CONFIG:

DEVICE= C:\WINDOWS\COMMAND\oakcdrom.sys /D:oemcd001

DEVICE= C:\WINDOWS\COMMAND\btdosm.sys

DEVICE= C:\WINDOWS\COMMAND\flashpt.sys

## A Beginners Guide To DOS Programming

```
DEVICE= C:\WINDOWS\COMMAND\btcdrrom.sys /D:oemcd001
DEVICE= C:\WINDOWS\COMMAND\aspi2dos.sys
DEVICE= C:\WINDOWS\COMMAND\aspi8dos.sys
DEVICE= C:\WINDOWS\COMMAND\aspi4dos.sys
DEVICE= C:\WINDOWS\COMMAND\aspi8u2.sys
DEVICE= C:\WINDOWS\COMMAND\aspicd.sys /D:oemcd001
```

### AUTOEXEC:

```
LH MSCDEX.EXE /D:oemcd001 /L:D
```

### CONFIG.SYS:

```
DEVICE= C:\WINDOWS\COMMAND\oakcdrom.sys /D:oemcd001
DEVICE= C:\WINDOWS\COMMAND\btdosm.sys
DEVICE= C:\WINDOWS\COMMAND\flashpt.sys
DEVICE= C:\WINDOWS\COMMAND\btcdrrom.sys /D:oemcd001
DEVICE= C:\WINDOWS\COMMAND\aspi2dos.sys
DEVICE= C:\WINDOWS\COMMAND\aspi8dos.sys
DEVICE= C:\WINDOWS\COMMAND\aspi4dos.sys
DEVICE= C:\WINDOWS\COMMAND\aspi8u2.sys
DEVICE= C:\WINDOWS\COMMAND\aspicd.sys /D:oemcd001

DOS=HIGH,UMB
Device=C:\WINDOWS\Himem.Sys
BUFFERS=40
FILES=40
REM STACKS=9,256
LASTDRIVE=Z
SHELL= C:\WINDOWS\COMMAND\COMMAND.com C:\WINDOWS\COMMAND\ /E:4096 /P
```

### AUOTEXEC.BAT:

```
SET TMP=C:\WINDOWS\TEMP
SET TEMP=C:\WINDOWS\TEMP
SET PROMPT=$p$g
SET winbootdir=C:\WINDOWS
SET PATH=C:\WINDOWS;C:\WINDOWS\COMMAND
REM application start up files.
REM SET PATH=C:\WINDOWS;C:\WINDOWS\COMMAND;C:\DOSLINKS
SET BLASTER=A220 I5 D1 H5 T6
LH MSCDEX.EXE /D:oemcd001 /L:D
CTMOUSE /R55
```

At this point you can boot into MS-DOS from Windows 95 with mouse, CD-ROM support and some memory and buffer enhancements.

**Hint:**

You can find a copy of the above config lines in the Windows 98 Boot Floppy disk CONFIG.SYS and AUTOEXEC.BAT files

### B. Use F8 at boot and select “6. Command prompt only”.

**Not recommended!**

**NOTE!** If the CONFIG.SYS and AUTOEXEC.BAT files have any entries set they will be processed at boot time (Longer boot time) and used as part of the default (Global) configuration and environment when using the console emulator while running Windows. You will need to take this into consideration when setting additional local environment settings in batch files when running DOS applications. Windows typically creates a PIF with the memory and environment variables set which can lead to unintended configuration conflicts with the CONFIG and AUTOEXEC settings.

**NOTE!** To revert from this setup may require deleting all lines in the CONFIG.SYS and AUTOEXEC.BAT files and leaving them blank to be able to boot directly into Windows again.

This is due to windows duel boot writing to the hard drive Master Boot Record (MBR). Even though we change the boot settings in Windows the master boot record takes precedence and the boot fails. This MBR boot can be cleared but is beyond the scope of this guide.

After Windows reboots successfully 2 times the MBR will be cleared and you should be able to restore the AUTOEXEC.BAT and CONFIG.SYS files again and boot normally to windows as long as you leave BootGUI=1.

It is far safer and more convenient to use option A. above and leave the C:\CONFIG.SYS;AUTOEXEC.BAT file empty.

Will only boot with generic windows environments if no CONFIG or AUTOEXEC set or if the following configs are set they will be used. Use Exit or Win to boot into windows.

Alternatively Windows will be the default OS to boot into.

Use the following CONFIG and AUTOEXEC as a start template for your MS-DOS boot. This will be the same as used in C. and D. below. It is essentially a copy of the custom config and autoexec used in the pif file with the edition of ECHO OFF .

#### CONFIG.SYS:

```
DEVICE= C:\WINDOWS\COMMAND\oakcdrom.sys /D:oemcd001
DEVICE= C:\WINDOWS\COMMAND\btdosm.sys
DEVICE= C:\WINDOWS\COMMAND\flashpt.sys
DEVICE= C:\WINDOWS\COMMAND\btcdrom.sys /D:oemcd001
DEVICE= C:\WINDOWS\COMMAND\aspi2dos.sys
DEVICE= C:\WINDOWS\COMMAND\aspi8dos.sys
DEVICE= C:\WINDOWS\COMMAND\aspi4dos.sys
DEVICE= C:\WINDOWS\COMMAND\aspi8u2.sys
```

```
DEVICE= C:\WINDOWS\COMMAND\aspicd.sys /D:oemcd001
DOS=HIGH,UMB
Device=C:\WINDOWS\Himem.Sys
BUFFERS=40
FILES=40
REM STACKS=9,256
LASTDRIVE=Z
SHELL=C:\WINDOWS\COMMAND.com C:\WINDOWS\COMMAND\ /E:4096 /P
```

AUOTEXEC.BAT:

```
ECHO OFF
SET TMP=C:\WINDOWS\TEMP
SET TEMP=C:\WINDOWS\TEMP
SET PROMPT=$p$g
SET winbootdir=C:\WINDOWS
SET PATH=C:\WINDOWS;C:\WINDOWS\COMMAND
REM application start up files.
REM SET PATH=C:\WINDOWS;C:\WINDOWS\COMMAND;C:\DOSLINKS
SET BLASTER=A220 I5 D1 H5 T6
LH MSCDEX.EXE /D:oemcd001 /L:D
CTMOUSE /R55
```

### C. Change BootGUI=1 to BootGUI=0. Use Win to start Windows.

and

E. Make the system into a dual boot system using the Config.sys and Autoexec.bat with a menu to boot into DOS or Windows.

**Not recommended!**

This will make windows always boot to DOS as the first OS and requires the correct config files.

**NOTE!** If the CONFIG.SYS and AUTOEXEC.BAT files have any entries set they will be processed at boot time (Longer boot time) and used as part of the default (Global) configuration and environment when using the console emulator while running Windows. You will need to take this into consideration when setting additional local environment settings in batch files when running DOS applications. Windows typically creates a PIF with the memory and environment variables set which can lead to unintended configuration conflicts with the CONFIG and AUTOEXEC settings.

**NOTE!** To revert from this setup may require deleting all lines in the CONFIG.SYS and AUTOEXEC.BAT files and leaving them blank to be able to boot directly into Windows again.  
This is due to windows duel boot writing to the hard drive Master Boot Record (MBR). Even though

we change the boot settings in Windows the master boot record takes precedence and the boot fails. This MBR boot can be cleared but is beyond the scope of this guide.

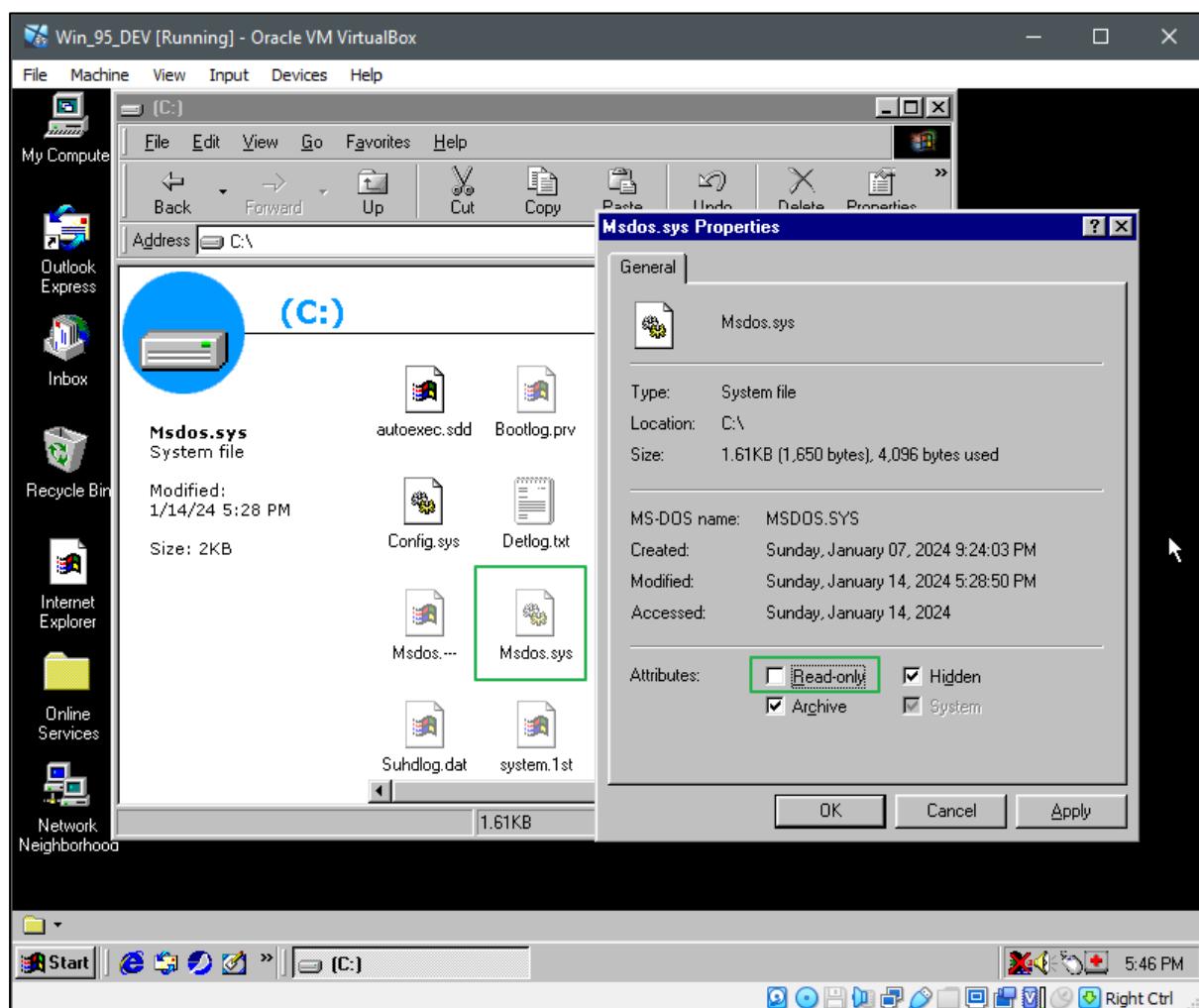
After Windows reboots successfully 2 times the MBR will be cleared and you should be able to restore the AUTOEXEC.BAT and CONFIG.SYS files again and boot normally to windows as long as you leave BootGUI=1.

It is far safer and more convenient to use option A. above and leave the C:\CONFIG.SYS;AUTOEXEC.BAT file empty.

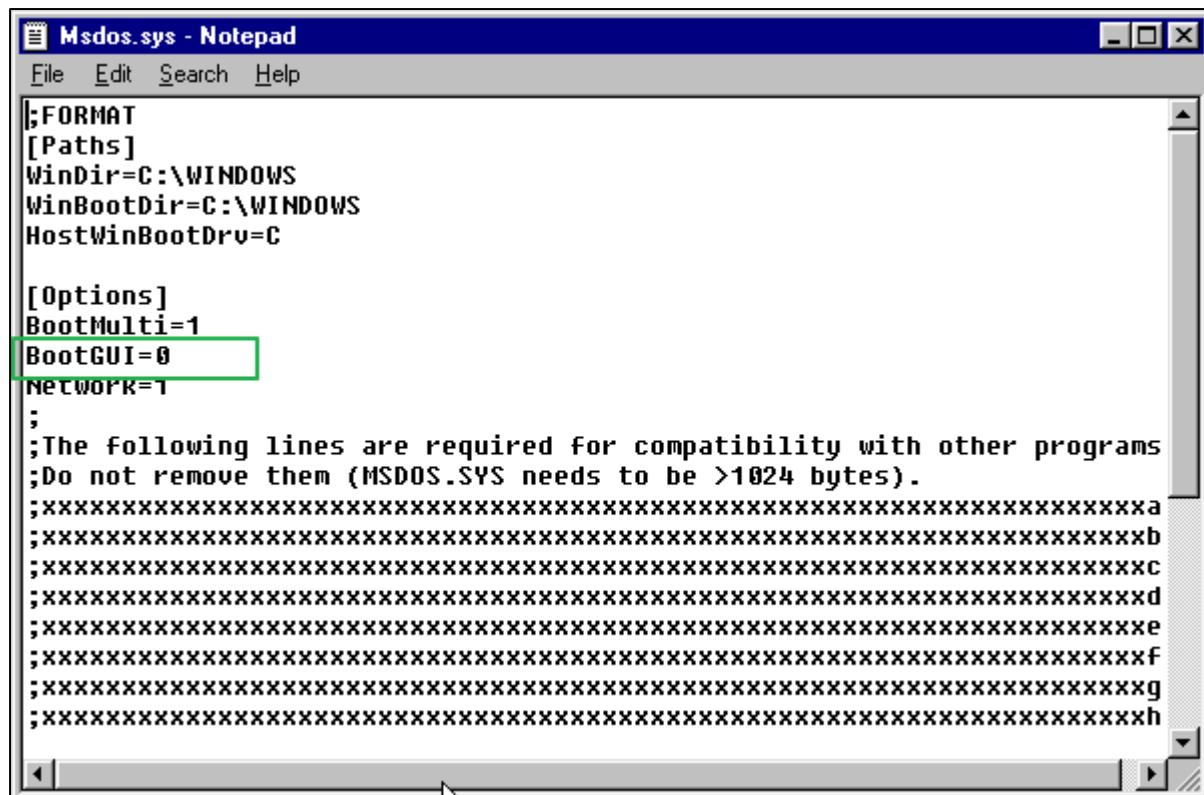
You can also create a boot menu as part of the CONFIG.SYS or AUTOEXEC.BAT file to choose between DOS and windows as the OS. Take note of the SHELL= environment as it will need to be ignored for the Windows part of the menu selection.

As with above you can start Windows from the console by typing Exit or Win.

Select Properties for C:\Msdos.sys and switch off the read only attribute and then close the Properties dialog with [OK].



Open Msdos.sys in a text editor and change the line BootGUI=1 to BootGUI=0



```
Msdos.sys - Notepad
File Edit Search Help

;FORMAT
[Paths]
WinDir=C:\WINDOWS
WinBootDir=C:\WINDOWS
HostWinBootDrv=C

[Options]
BootMulti=1
BootGUI=0
NETWORK=1
;
;The following lines are required for compatibility with other programs
;Do not remove them (MSDOS.SYS needs to be >1024 bytes).
;xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxa
;xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxb
;xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxc
;xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxd
;xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxe
;xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxf
;xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxg
;xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxh
```

Save the Msdos.sys file and close the editor.

Open the Properties dialog again and switch the [/] Read Only attribute back on.

If you want to turn your system back into booting to Windows as the default change back to BootGUI=1 (Note that you may need to clear the contents of CONFIG.SYS and AUTOEXEC.BAT to get a successful automatic Windows ("BootGUI=1") boot).

Use the following CONFIG and AUTOEXEC as a start template for your MS-DOS boot.

This will be the same as used in B. above. It is essentially a copy of the custom config and autoexec used in the pif file with the edition of ECHO OFF.

#### CONFIG.SYS:

```
DEVICE= C:\WINDOWS\COMMAND\oakcdrom.sys /D:oemcd001
DEVICE= C:\WINDOWS\COMMAND\btdosm.sys
DEVICE= C:\WINDOWS\COMMAND\flashpt.sys
DEVICE= C:\WINDOWS\COMMAND\btcdrom.sys /D:oemcd001
DEVICE= C:\WINDOWS\COMMAND\aspi2dos.sys
DEVICE= C:\WINDOWS\COMMAND\aspi8dos.sys
DEVICE= C:\WINDOWS\COMMAND\aspi4dos.sys
DEVICE= C:\WINDOWS\COMMAND\aspi8u2.sys
DEVICE= C:\WINDOWS\COMMAND\aspicd.sys /D:oemcd0011

DOS=HIGH,UMB
Device=C:\WINDOWS\Himem.Sys
```

```
BUFFERS=40
FILES=40
REM STACKS=9,256
LASTDRIVE=Z
SHELL=C:\WINDOWS\COMMAND.com C:\WINDOWS\COMMAND\ /E:4096 /P
```

#### AUOTEXEC.BAT:

```
ECHO OFF
SET TMP=C:\WINDOWS\TEMP
SET TEMP=C:\WINDOWS\TEMP
SET PROMPT=$p$g
SET winbootdir=C:\WINDOWS
SET PATH=C:\WINDOWS;C:\WINDOWS\COMMAND
REM application start up files.
REM SET PATH=C:\WINDOWS;C:\WINDOWS\COMMAND;C:\DOSLINKS
SET BLASTER=A220 I5 D1 H5 T6
LH MSCDEX.EXE /D:oemcd001 /L:D
CTMOUSE /R55
```

There are many different ways to set up a DOS and dual boot system. You can get some hints from my custom selection menu in the FreeDOS FDAUTO.BAT sections in book 1. You will also find many explainers on DOS configurations as well as the unofficial documentation on using CONFIG.SYS and AUTOEXEC.BAT. Just keep in mind that you are using a hybrid Windows dual boot system controlled by Windows which is different to a single DOS OS installation such as FreeDOS. I recommend leaving the C:\ CONFIG.SYS and AUTOEXEC.BAT files empty and make use of the PIF files from Windows.

---

## DOS navigation

This section relates to navigating when in DOS Mode or DOS Real Mode. You can also use these tools in the Windows command prompt as well.

## DOS Directories

Windows 95 only sets a single DOS directory **C:\WINDOWS\COMMAND**

This is somewhat equivalent to the FreeDOS \BIN directory. The Windows DOS \COMMAND contains only the very basic of DOS command line tools. If you need additional binary tools then they will be

placed in the \COMMAND directory. Any application file that would normally go into the FreeDOS\BIN directory will also be placed in the \WINDOWS\COMMAND directory.

The Windows default path environment only contains a reference to the DOS binaries in C:\WINDOWS\COMMAND so you will need to set up some additional DOS directory structures.

For convenience I try to follow a similar structure to FreeDOS, although there is no strict standard directory structure in DOS so you can organize the DOS directories in any way that you choose.

For DOS applications I use C:\DOSAPPS

For development tools I use C:\DEVEL the same as FreeDOS.

For application start up batch files I use C:\DOSLINKS which is the equivalent to C:\FREEDOS\LINKS.

You will need to add C:\DOSLINKS to the DOS path environment when start DOS. The PATH environment in your “Exit To DSO.pif” will contain the line:

**PATH=C:\WINDOWS;C:\WINDOWS\COMMAND;C:\DOSLINKS**

This will need to be in any PIF that launches the DOS command line environment. Alternatively you can place the paths in a batch file if they are not in the PIF file.

Because we are working between 3 different system environment we will need to create some additional start up files compared to pure DOS. Some care is required when doing this as we will have Batch files to launch the application as well as PIF files. We can launch a batch file from DOS Real Mode, DOS emulated Mode or from windows which will switch to DOS emulated mode. When launching from Windows we will need to have a batch file and its associated PIF available to “SET” the correct environment variables. This is quite complex at first as we have multiple ways in which we can launch a DOS application. The important part is to ensure that we have the correct environment variables set in either the batch file or the PIF. If the PIF launches the batch file we only need to set the environment variables in one of the 2 files.

I typically set the additional paths in the batch file that launches the application. In DOS Real Mode I set the memory requirements in the PIF that launches the initial command prompt. When calling an application launch batch file from windows “Windowed Mode” I set the additional memory requirements for DOS in the PIF file and paths in the batch file.

After a few examples when setting up the development environments it will become a little clearer :)

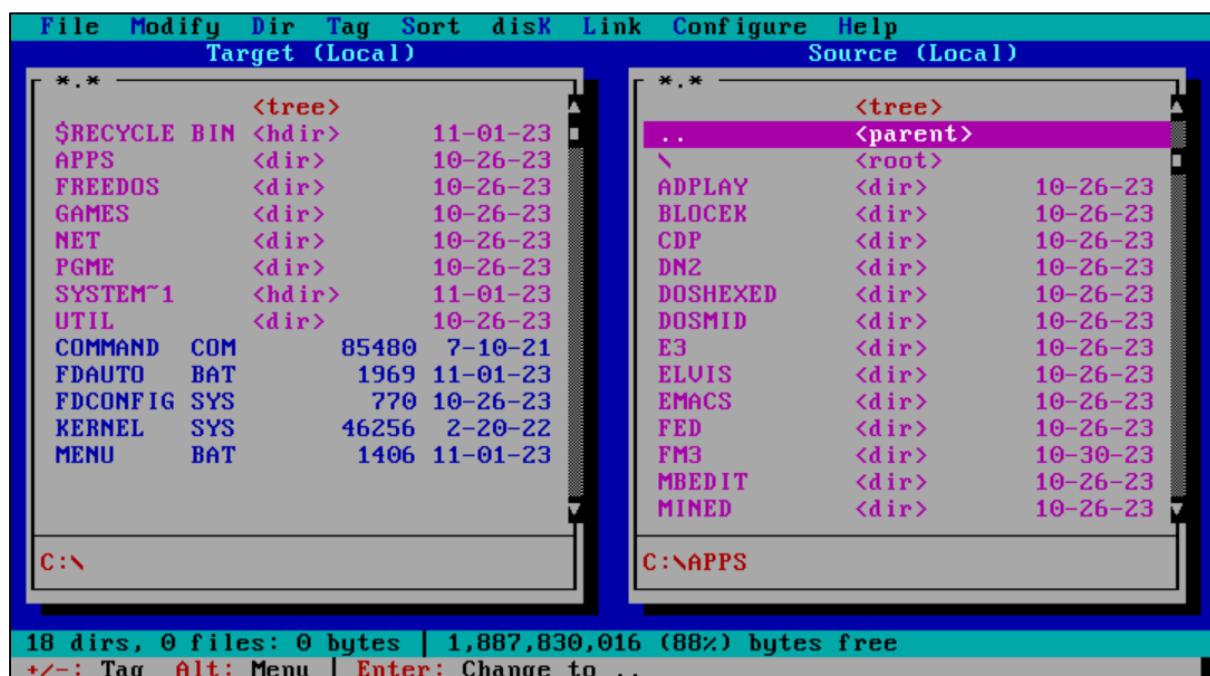
### Essential navigation

This section applies more to Real Mode DOS rather than the dos emulated window.

If you are content navigating from the DOS command prompt you can skip this step if you wish but I prefer an easy TUI or GUI based file manager for navigation. We have a GUI file Manager (Explorer) available in Windows so there is no real need for a DOS file manager, but on occasions I do find it convenient to have. There are many good file managers available for DOS, each with its own degree complexity and features. File Maven 3 is one of the easiest 2 pane file managers with the best mix of feature and ease of use. FMAV35A has been my go to file manager for DOS for as long as I can

remember. You will need to download 2 separate applications to get the full benefit of File Maven. The first is the File Maven (DOS) 3.5a (Free) "fmav35a.zip" application from Briggs Soft at the following link.

<https://www.briggsoft.com/fmdos.htm>



Although this is not essential for using File Maven the PKZIP application is required for viewing and unpacking archive files. You will find many copies of the "**pkz204g.exe**" download across the internet. It is a commercial product released under the description of shareware meaning it is a trial that does not expire.

We only need the Unpack utility for File Maven.

As an alternative InfoZIP makes a zip extraction utility that is API compatible with PKZIP. You will need to change the Info-Zip file name from "unzip.exe" to "pkunzip.exe". I have not tested this method at the time of creating this guide.

Zip utility for DOS "zip232x.zip"

<https://www.sac.sk/files.php?d=7&l>

UNZip utility for DOS (386) "unz552x3.exe"

<https://www.sac.sk/files.php?d=7&l>

Dos real mode 286 "unz550x.exe"

<https://www.sac.sk/files.php?d=7&l>

More information.

<https://www.btrr-software.de/freesoft/arc1.htm>

<https://www.computerhope.com/software/pkutil.htm>

Other newer versions for DOS unzip utilities may work, but I have not tested them.

Note that I have placed PKZIP path in the FM3.BAT file instead of the DOS start up Configuration files.

File Maven 3 Can be used as a portable application from any location including a floppy drive. This makes it a very useful navigation tool when encountering DOS environments where a TUI file manager is not available.

We are going to install File Maven 3 as part of the system so that it is available at all times.

Unpack the “fmav35a.zip” archive as well as the “pkz204g.exe” using 7-Zip. Transfer the File Maven and PKZIP files to your virtual DOS guest drive or an attached storage drive (preferred) using one of the above methods. Take note of the path to .\fmav35a\install\FM3.exe

From the DOS command prompt (welcome screen) navigate to the location of .\fmav35a\install\FM3.exe

You will need to make use of the basic DOS commands CD and DIR

[http://wiki.freedos.org/wiki/index.php/Dos\\_commands](http://wiki.freedos.org/wiki/index.php/Dos_commands)

[Drivename] (D:), CD [Directoryname] (CD install), CD .. (Up one directory level)

DIR [/p|/w|a] (to list the current directory contents.)

Be aware of DOS 8.3 file naming FILEMA~1(8.3) = FileMaven3(LFN). 8.3 naming has a maximum of 8 characters. I would stay within the DOS 8 character file names where possible as some legacy applications will fail when attempting to list long file names.

When you have confirmed you are in the \FMAV35A directory type INSTALL.EXE into the command line followed by [Enter].

Type C:\DOSAPPS\FM3 into the destination Directory followed by Enter. Type Y to install FM3.

After the install the current working directory will be C:\DOSAPPS\FM3. Type FM3 then Enter to start the file manager. You now have an easy 2 pane file manager to navigate and organize your DOS install.

I recommend making a backup of the FM3 directory and files. The new install creates a fresh FM.CFG file which can only be created using the installer. You can use a copy of the fresh FM3 directory as a portable app by copying it to any location. You can also use the files or just the FM.CFG if you make changes to your installed File Manager 3 and wish to restore it to the original settings.

Next use the arrow keys to move focus to the second file pane, select <tree> Enter. Use the TAB key to select the drive containing the original File Maven Installer then Enter to select the drive. Navigate to the File Maven directory (or alternatively create a New BackUp directory using ALT Dir, Make dir).

When the second pane is in the directory you wish to copy the FM3 directory too, use the arrow keys to shift focus back to the FM3 directory on the C drive. Navigate up one level using <parent> and highlight the FM3 directory.

Next press the Alt and navigate to File -> Copy then Enter. Select yes to include subdirectories, OK and then OK to confirm the destination.

The copy of the FM3 directory is now visible in the right pane. This is the basics for moving and copying files with FM3.

Note That Windows 95 does not have the “DOS Mode” mouse drivers installed by default, so you will need to make sure CTMOUSE is available and set in the DOS configs. Windows uses its own mouse drivers when in Windowed mode. Note that DOS mouse has no scroll wheel so you will need to use the scroll bar at the right.

In windows 95 the Mouse driver “CTMOUSE.EXE” must be set in the batch file or PIF AUTOEXEC section.

The default mouse settings can be a little sensitive. I explain how to change the mouse sensitivity in a later section “CuteMouse”.

The mouse drivers are only required in “DOS Mode”.

FM3 uses and equivalent the DOS command MORE for viewing files and I would leave this as the default. For editing I would recommend using MS-DOS EDIT.COM, FreeDOS “EDIT.EXE” or FED.

Select Alt + C to open the Configure menu. Select “Editor” from the menu and type in EDIT.EXE and OK/Enter. You could also use FED or any other editor. Note that FM3 only allows 13 characters for the editor path\filename. FED does not exist in the system path and will need to have a batch file created before it will work in FM3. The following section “Application launch BAT” will show you how to do that.

Basic batch launch without PKZIP.

```
@ECHO OFF  
REM File Maven 3 application launch.  
CD \  
CALL C:\DOSAPPS\FM3\FM3.EXE
```

If you have created the batch files for FM3 and for FED we can complete the FM3 setup by including the PKZIP files for unpacking archives. You can use the installer and add PKZIP as a system wide application if you want, but I don’t recommend it as FreeDOS already has a number of viable open source zip file and archive utilities available.

Unpack “pkz204g.exe” on your host machine using 7-Zip or similar. Copy the directory and all included files to your FreeDOS system drive and place the directory in C:\DOSAPPS\pkz204g\\*.\*

Create a “ZIP” file from the original unpacked “pkz204g” directory including the files “pkz204g.zip”.

Create a new temporary directory C:\TEST and copy the original “pkz204g.zip” from the host into C:\TEST\pkz204g.zip. We will use this as a test for the ZIP file unpacking in File Maven 3.

Navigate to C:\DOSLINKS and open FM3.BAT in a text editor (See: “Application launch BAT”).

Add “SET PATH=%path%;C:\DOSAPPS\pkz204g” before the call to FM3.EXE

### FM3.BAT

```
@ECHO OFF
REM File Maven 3 application launch.
REM Set a temporary path for PKZIP
REM PATH=%path%;C:\WINDOWS;C:\WINDOWS\COMMAND;C:\DOSLINKS
REM CTMOUSE /R55
SET PATH=%path%;C:\DOSAPPS\pkz204g
REM Launch File Maven 3
CALL C:\DOSAPPS\FM3\FM3.EXE
```

This will make PKUNZIP available to File Maven 3 but PKZIP will not be seen by the system otherwise.

Exit File Maven 3 if it is open including any parent instances and restart FM3.EXE. Restarting FM3 from FM3.BAT will invoke the new environment path to the PKZIP files.

From FM3 navigate to the directory C:\TEMP and select PKZ204G.ZIP and press Enter to view the contents of the zip file. If PKZIP is correctly recognized you will now have the archive opened in the lower part of the FM3 panel to select (Tag) and unpack files from the archive.

Select Alt+T to tag all files for extraction, then Alt+E to extract all tagged files to the TEST directory.

Alternatively press Alt+X to extract all directories and files including the original paths under the archive name. This is generally the safest method to unpack a full archive.

Press the Esc (Escape) key to close the file archive viewer and return to FM3 navigation. You can now see and navigate to the extracted files.

You can now delete the \TEST\\*.\* directory as we no longer need it.

That concludes the File Maven 3 setup tasks. You can customize colours and other features if you want, but I typically find the default configuration works well and is easy to read.

### Application launch BAT

See the section in Book 1 for more information on creating batch files.

## Installing Development tools

<https://www.ibiblio.org/pub/micro/pc-stuff/freedos/files/repositories/1.3/pkg-html/index.html>

Setting up DOS application development tools can feel like a daunting task to the uninitiated and does come with its difficulties so I have created the following section as quick guide to get through some of the initial hurdles.

Why write software for DOS? Although DOS is considered by many as an archaic OS that is no longer used it does still have a significant following. Be it for running your old games or setting up historical computer systems for a collection display is still widely used. DOS is also used in quite a number of modern hardware implementations where a small and light computer system is required.

Another aspect of DOS programming is it's low level use and access to a less complicated hardware design making it also useful as a learning tool for people who may wish to go into computer engineering or work with small modern "System On Board" SOB or "System On Chip" SOC microcomputers. DOS can easily be set up in a virtual computer hardware environment making experimentation less risky than working with physical devices.

Another aspect is the immense archive of literature, source code and other resources available for DOS making for easier access for novices to experiment and understand different aspects of the hardware and software coding.

I am not going to go into detail about DOS programming in this guide beyond a few "Hello World!" applications to test your development environment setup. The main focus is to get you started with a few select programming tools and environments so you can jump straight into learning to code on DOS with tools that will work.

The 2 main languages that I will focus upon are C and BASIC. I am also going to include a novel JavaScript interpreter as well as some debugging and utility tools.

I may include a short section on the Flat Assembler FASM in a future revision as well as some Windows 9x development tools.

Setting up the DOS development environments in Windows 95 is very similar to the guides used in the FreeDOS guide with the addition of the PIF shortcuts.

### NOTE:

We could easily copy and paste the C:\DEVEL directory from FreeDOS with all of the development tools directly to the C: drive of the Windows 95 Install.

I have copied the complete C:\DEVEL directory to my Windows 95 C:\DEVEL drive for this guide. I have also copied the batch files used to launch the development tools from FreeDOS C:\FREEDOS\LINKS to Windows 95 C:\DOSLINKS.

Most of the batch files in both the \DOSLINKS and \DEVEL directories will need to be modified for the Windows MS-DOS environment.

The following install and setup guides are a copy of the FreeDOS install guides with some additional notes for the Windows environment including the batch file modifications and PIF file configs.

Keep in mind that this guide will focus upon working in a Win32 environment using the DOS emulator COMMAND.COM in "Windowed Mode". You can [re]boot the Windows 95 system into real mode DOS if needed but I don't recommend it. When running in real DOS V7.1 mode using the development tools are much the same as for FreeDOS and will making use of the AUTOEXEC.BAT

and CONFIG.SYS files depending upon how we arrive at the command prompt as well as the same batch files used to launch applications.

So the development tools setup will be much the same as for Real Mode DOS “COMMAND.COM”, and when working in Windows will have some additional Windows 32 shortcuts to setup the memory requirements for the COMMAND.COM emulator. We also have the convenience of being able to use some Windows 32-bit applications such as Notepad++ with our DOS programming environments when working in the Windows context.

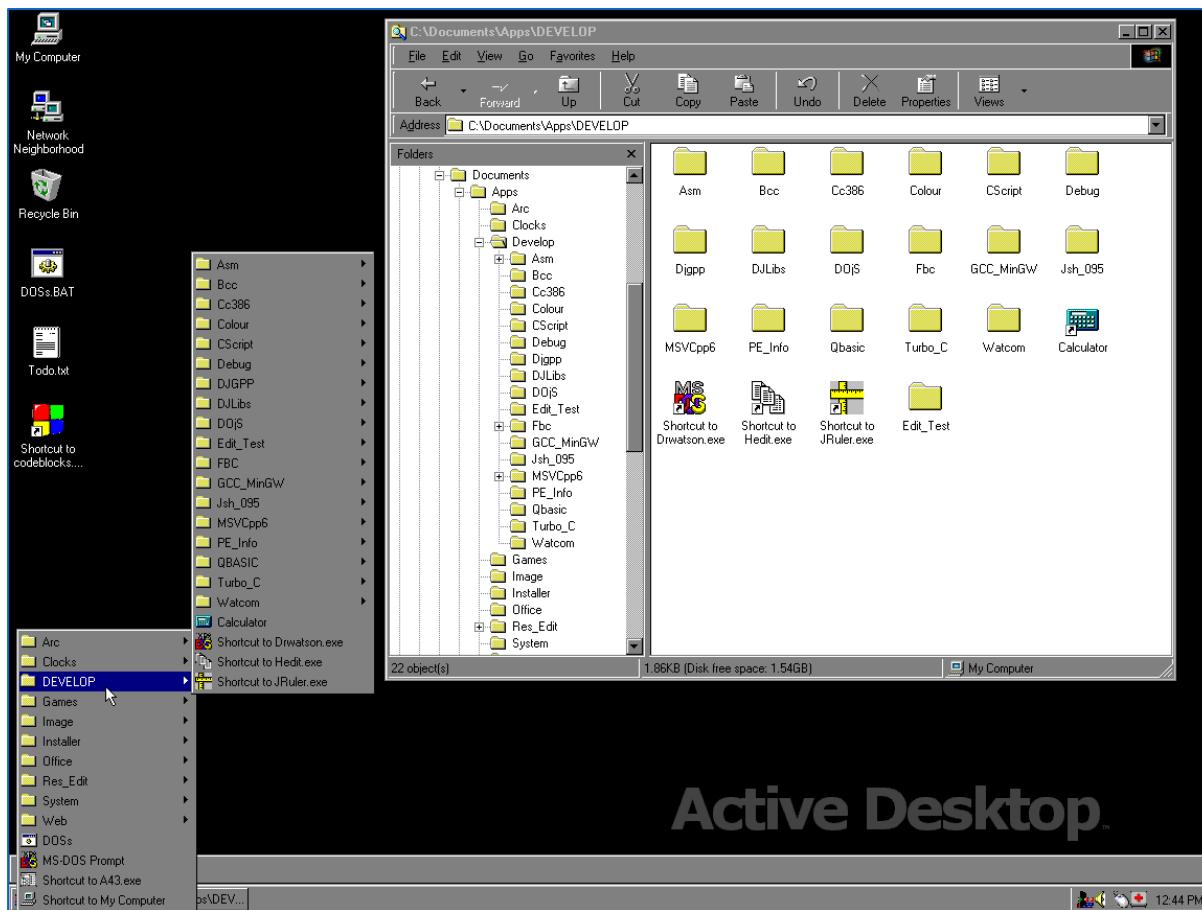
Also note that you can use applications such as Costa Desktop and FileMaven 3 when booting in DOS 7.1 real mode.

Please keep in mind that the 8.3 (8 Character) naming convention still applies when working under Windows 95. Windows 95 will work fine with file and directory names longer than 8 characters, but the underlying DOS architecture will truncate long names to the 8.3 format, so keep any DOS related names below the 8 character limit.

I typically use a separate directory from the C:\DOSAPPS for development tasks for better organisation. I will use C:\DEVEL (The default used by FreeDOS; Recommended) in the guide, but you can use any suitable name that you like.

Always keep a note, or have access to the environment paths in each development environment as you will need to know them when setting up your IDEs; An image capture, copy of the directory structure on the host machine or in a note file that is easily viewable.

You can create shortcuts for the DOS launch batch file or PIF on the Windows 95 desktop for each development application or tool. The shortcut will link to the batchname.BAT file or batchname.PIF file. I would also recommend creating directories in your Advanced Launcher (ALaunch) quick launch directory and placing the shortcuts there.

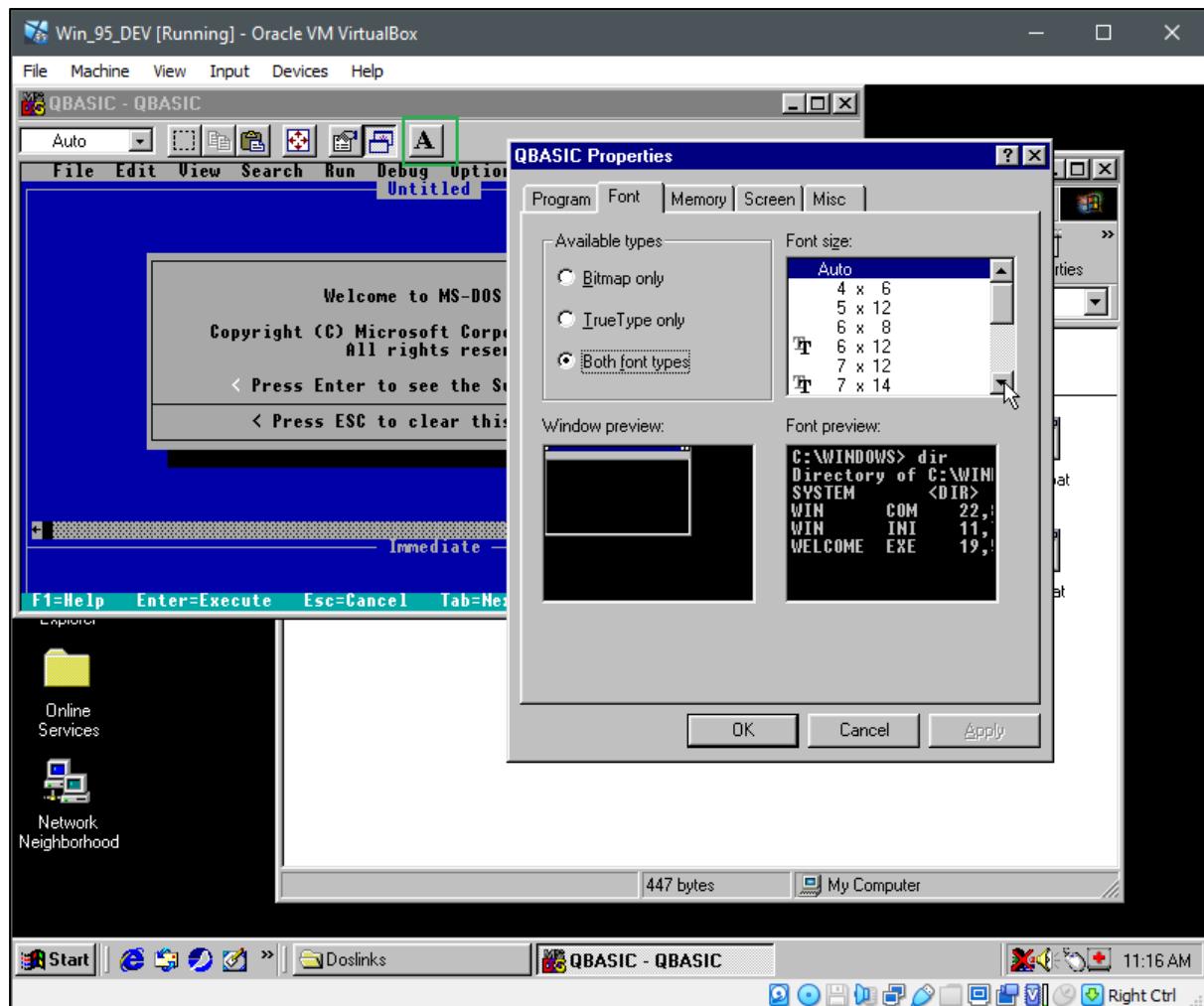


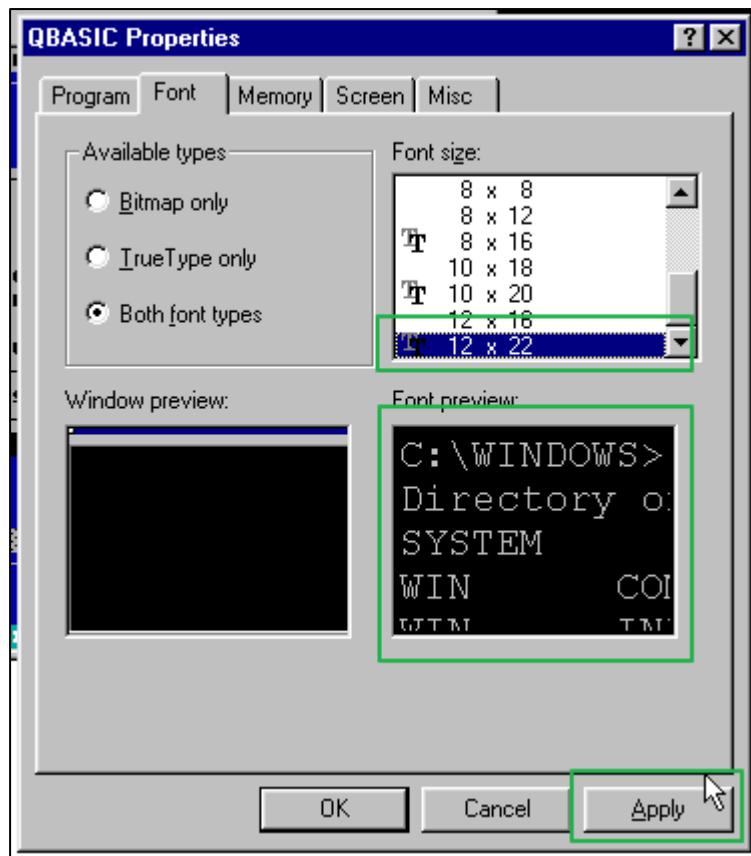
### Hint

Sometimes the DOS “Windowed” screen can appear too small. This can make it difficult to work with text editors in the development IDEs.

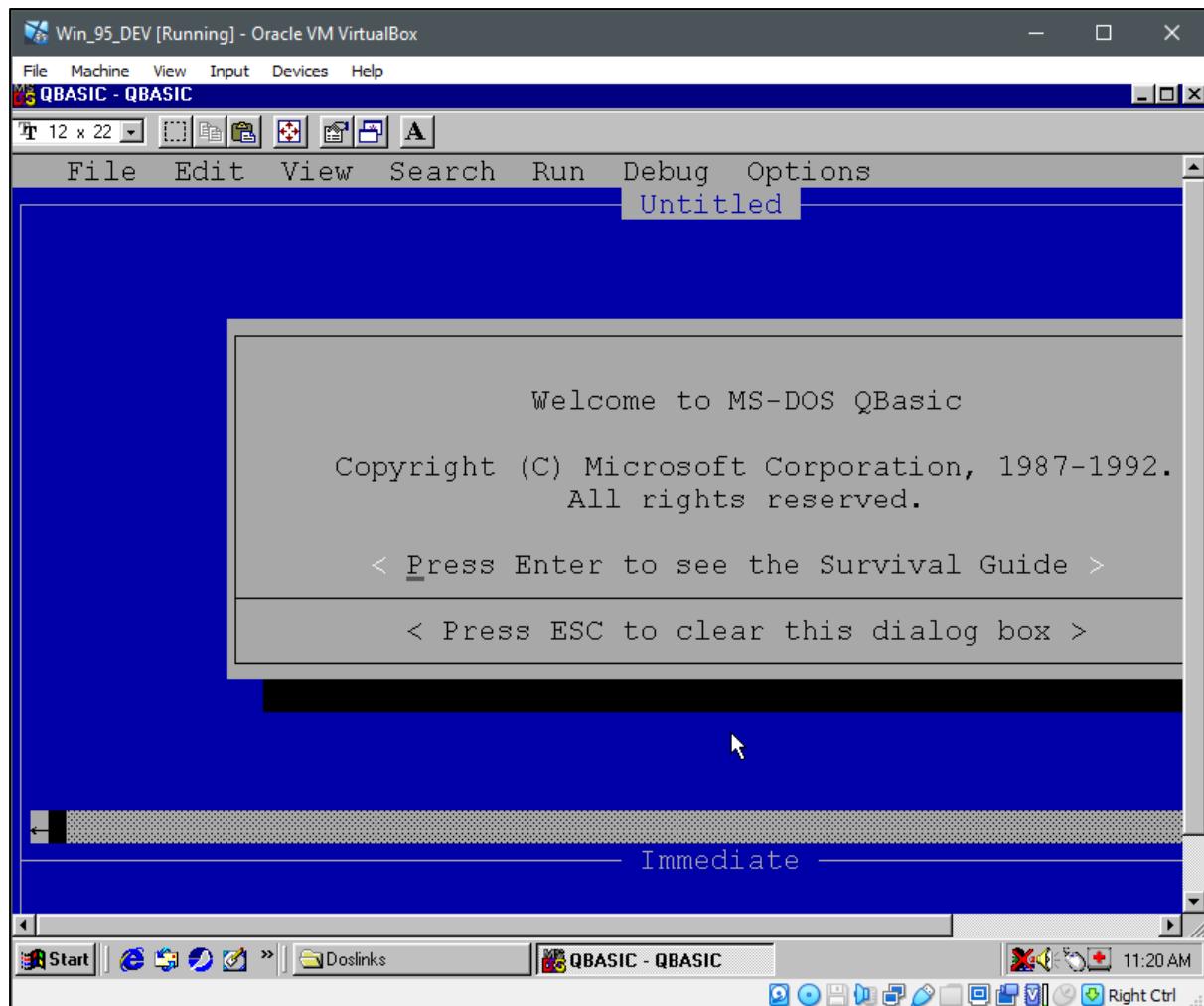
Either from the properties of the launching batch file or PIF, or directly from the Text option in the Quick launch bar of an open DOS windows, change the text size to that which is more suitable

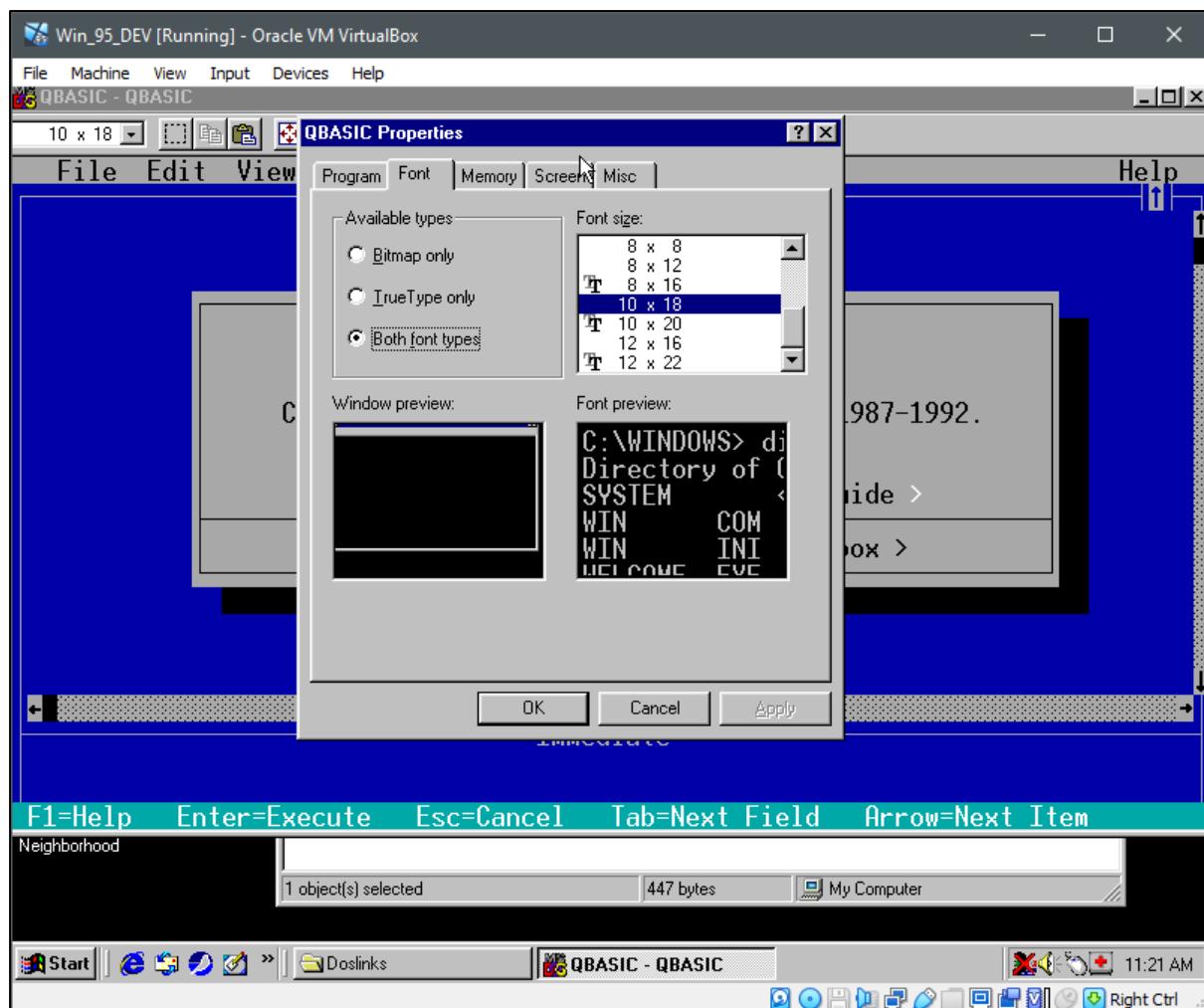
## A Beginners Guide To DOS Programming





If using VirtualBox just be sure that your font size is not too large for your current screen size. I can usually find a happy medium that is readable and fits on the screen.





This will also create a PIF file for the application or the batch file that launched the application.

## MS QuickBASIC

Microsoft QuickBASIC (also QB) is an Integrated Development Environment (IDE). It comes in an interpreted form as well as a compiled format. Although it is a proprietary product, MS allows its use for educational and historical purposes. Please read the original and more recent license agreements before distributing any application made with QuickBASIC. Please note the difference between QBASIC and QuickBASIC as they are not the same product.

QBASIC and QuickBASIC comes in 3 main releases; The version 1.1 which is an interpreted script that can only run from the IDE, Version 4.5 which comes as both interpreted (basic edition) and compiled (Pro edition) and Version 7.1 (QBX) which has additional database extensions. QuickBASIC V4.5 and 7.1 were replaced with VisualBASIC for DOS and Windows. QBASIC and QuickBASIC were derived from the earlier GW-BASIC for DOS and the original variants of MS BASIC that were shipped with some of the first personal microcomputers.

Why bother with a legacy language like BASIC? Isn't it dead? Yes and no. There has been a revival of legacy games that have been modernised for newer computer environments. Understanding BASIC affords us the opportunity to transcribe that source code into a modern context. FreeBASIC is an example of a tool capable of converting traditional BASIC into a modern C language context. The other important note is that many popular modern programming languages draw heavily upon the constructs of BASIC and C language. If you can code in Python then it is likely you will have little difficulty coding in BASIC and vice versa. I often transcribe between BASIC and Python "On the fly".

The QBASIC interpreter is the most simple to use of the 3 products and offers an easy way into understanding traditional BASIC language programming. FreeBASIC is a modern implementation of the BASIC language that is capable of translating and compiling legacy BASIC source code. FreeBASIC can run a number of legacy BASIC languages as well as make use of modern C language constructs and libraries.

<https://en.wikipedia.org/wiki/QBasic>

<https://en.wikipedia.org/wiki/QuickBASIC>

If you want to look at a more serious BASIC development environment to distribute software I would consider FreeBASIC over the legacy MS versions.

### QBASIC V1.1

QB was never released as a standalone product and was only release as part of MS-DOS 5.0 and Windows 95/98 install CDs.

You will need 4 files from the any of the install CDs. Only qbasic.exe and qbasic.hlp are required, but I do recommend also having the MS-DOS Command line help files available for reference.

- **qbasic.exe**
- **qbasic.hlp**
- **help.com**
- **help.help**

They will most often be found on the CD under .\other\oldmsdos, or .\tools\oldmsdos.

Copy the 4 files into a directory with an appropriate name such as QB Or QBASIC. Note that "QUICKBASIC" (V4.5) has 9 characters and the maximum 8.3 is an 8 letter limit. So take care with the naming conventions between QBASIC and QuickBASIC.

Copy your .\QBAS directory to you Windows 95 install drive. You can use any directory you want such as C:\APPS or C:\PORTABLE, but I typically create a separate directory for development tools called **C:\DEVEL** to keep your work organised and separated from general applications. C:\DEVEL is the default directory used by FreeDOS so I will use the same convention with Windows 95.

Now that you have the files copied to C:\DEVEL\QBAS create a batch file to launch both "qbasic.exe" and "help.com", then copy the batch file to C:\DOSLINKS.

The following is the most basic Batch file to launch a DOS application. If you need access to the DOS binaries then remove the REM from SET PATH=%path%;C:\WINDOWS\COMMAND;C:\DOSLINKS.

### QBAS.BAT (Basic)

```
@ECHO OFF
REM Launcher for QBAISICv1.1
REM DOS paths if required by the application.
REM SET PATH=%path%;C:\WINDOWS\COMMAND;C:\DOSLINKS
REM CTMOUSE /R55
SET PATH=%path%;C:\DEVEL\QBAS
REM Change current working directory
CD \DEVEL\QBAS
REM File Maven 3 application launch.
CALL C:\DEVEL\QBAS\QBASIC.EXE %1 %2 %3 %4 %5 %6 %7 %8 %9
```

### QBAS.BAT (EXE error check)

```
@ECHO OFF
REM Launcher for QBAISICv1.1
IF EXIST C:\DEVEL\QBAS\QBASIC.EXE GOTO QBAS
GOTO NOFOUND
GOTO END
:QBAS
REM You can add environment paths or change working directory here.
REM DOS paths if required by the application.
REM SET PATH=%path%;C:\WINDOWS\COMMAND;C:\DOSLINKS
REM CTMOUSE /R55
SET PATH=%path%;C:\DEVEL\QBAS
REM Change current working directory
CD \DEVEL\QBAS
REM Launch QBAS
CALL C:\DEVEL\QBAS\QBASIC.EXE %1 %2 %3 %4 %5 %6 %7 %8 %9
GOTO :END
:NOFOUND
ECHO QBAISIC not found!
ECHO Press any key to end...
PAUSE
:END
```

QBASIC doesn't require any special DOS environments to run. You can create a PIF file if needed but creating a windows shortcut on the desktop to the QBAS.BAT file is fine. You can create a new directory such as QBASIC and place the shortcuts there.

A Shortcut to a DOS application or batch file is a PIF file.

You can create a shortcut directly to "help.com". It does not require any additional batch file, PIF or DOS configuration.

Run the QBASIC.BAT

Create a hello world script in the QBASIC 1.1 editor.

Select File -> New

Select File -> Save As

File Name: [HELLO.BAS]

**NOTE**

QBASIC file Open and File save navigation can take a little time to get used to. You will need to carefully TAB to each navigation panel, and then use the up/down arrow keys to select before pressing Enter.

QBASIC splits the screen between the main source document and any SUB-ROUTINES or FUNCTIONS.

Give the source an internal name, date, creator, copyright etc.

**REM Hello world!**

From the menu select “Edit -> New SUB...”

Name: [**MYPROC**]

[OK]

Note that you now in a different editing tab to the main document.

Place the following code between SUB MYPROC and END SUB

```
SUB MPROC
  CLS
  PRINT "Hello world!"
  PRINT "Press any key to continue..."
  SLEEP
END SUB
```

Select “File -> Save”

Select “View -> SUBS...”

Select the main document “HELLO.BAS”

Add the following line after the REM to call the subroutine.

```
REM Hello world!
MPROC
```

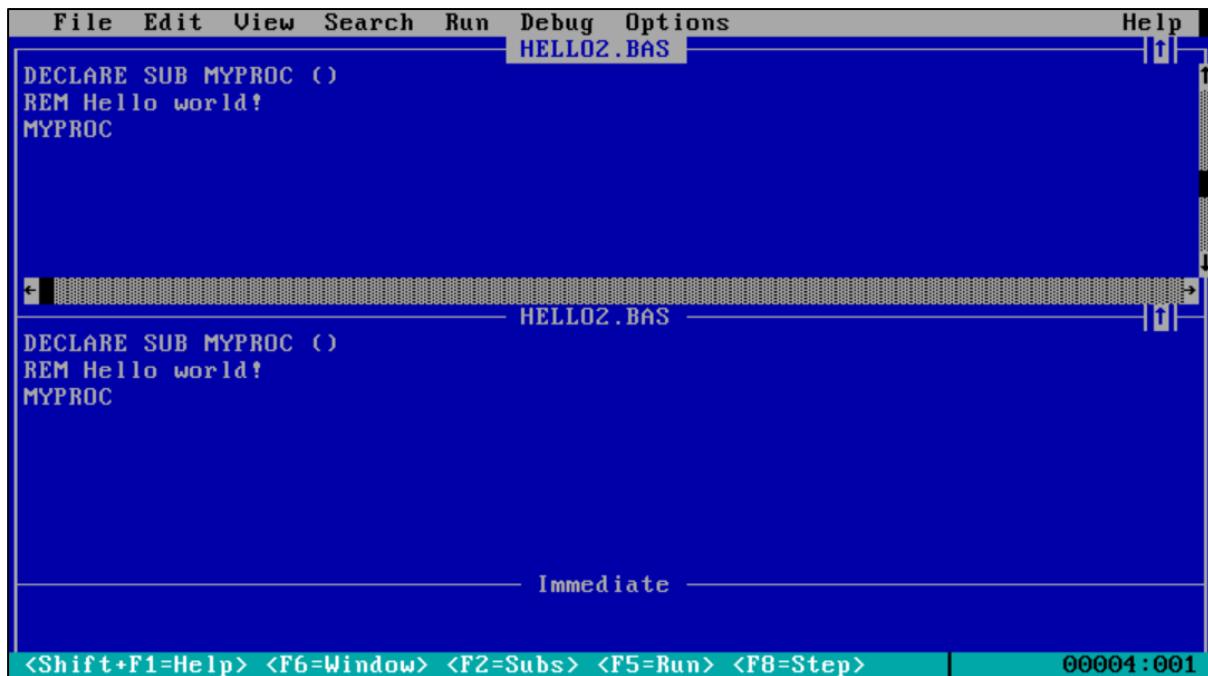
Select “Run -> Start”

You will see the Hello world! Console screen. Press a key to continue and end the application.

When you return to the main source screen in the editor you will notice an extra line **DECLARE SUB MYPROC ()**. All SUBs and Functions must be declared at the top of the page before any other lines in the application. The QBASIC application has automatically corrected this for you this time.

Next select “View -> Split”

You will now see both parts of the source file. You can change between screens using F6 or close the split screen by selecting “View -> Split”.



The following is what the source code looks like in a text editor.

```
HELLO.BAS  
DECLARE SUB MPROC ()  
REM Hello world!  
MPROC  
  
SUB MPROC  
CLS  
PRINT "Hello world!"  
PRINT "Press any key to continue..."  
SLEEP  
END SUB
```

You can access the BASIC help with [Alt] + H

<https://www.qbasic.net/en/top-ten-downloads/>

Online help

<https://hwiegman.home.xs4all.nl/qb-man/index.html>

QuickBASIC V4.5

**Original 5 install disk images.**

“003495\_microsoft\_quickbasic.7z”

<https://archive.org/details/003495-MicrosoftQuickbasic45>

**As files (unpacked disk images and 2 floppy disk images).**

<https://winworldpc.com/product/quickbasic/45>

“Microsoft QuickBASIC Compiler 4.50 (3.5).7z”

**As single copy paste directory.**

<https://www.qbasic.net/en/qbasic-downloads/compiler/qbasic-compiler.htm>

“qb45.zip”

NOTE: That the QB.INI file is set with the path C:\LIB etc. This needs to be checked/corrected.

The correct paths can be set in the Menu “Options” Set Paths.

This also needs to be corrected in FreeDOS\_1.3 install.

**PDF Manual (Microsoft\_QuickBASIC\_4.5\_2nd\_Edition\_Manual.pdf)**

[https://archive.org/details/Microsoft\\_QuickBASIC\\_4.5\\_2nd\\_Edition\\_Manual](https://archive.org/details/Microsoft_QuickBASIC_4.5_2nd_Edition_Manual)

You can choose between any of the downloadable install options above. Each will ultimately end with the directory structure as the archive downloaded from qbasic dot net on your DOS hard drive.

Either Install QuickBASIC by using the 5 floppy images mounted in VirtualBox (Recommended), copy the unpacked floppy images to a suitable location in your DOS drive with both directories combined and run the installer or copy the complete unpacked ready to go directory. I would recommend either the 5 mounted .img install path, or copy the premade directory.

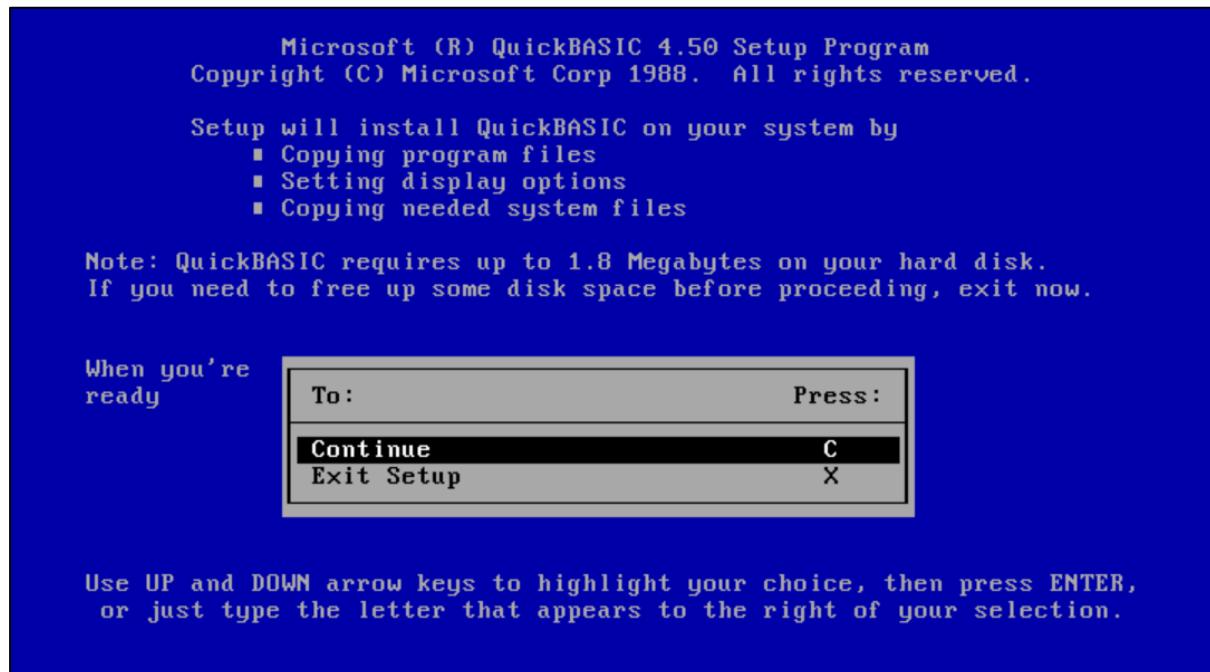
In Windows 95 you should be able to use the emulated DOS windows for the installs. If that fails use the “Exit To DOS.PIF” pathway. Ensure that the correct DOS PATHs are set as well as the CD-ROM drivers.

**Install (003495\_microsoft\_quickbasic.7z) Full install set.**

Mount “setup.img” in VirtualBox and navigate to the floppy drive.

Read the “READTHIS.NOW” text document. For a full list of files see “PACKING.LST” on Utilities 2 disk image as well as “README.DOC” which contains additional technical information.

Open a command prompt and type A:\ SETUP (SETUP.EXE) and Enter to begin the install.



I would choose “Run Full Setup” so that we can choose the install location for QuickBASIC. If you are unsure about the directory structure you can use “qb45.zip” from qbasic.net as a guide.

Most development environments will follow a similar directory structure. Some compiler languages will also have a BIN directory. In this case the binary files are in the root directory.

- .\QB45 (root + binaries)
- .\QB45\INC (Header files for LIB and uncompiled library source files)
- .\QB45\LIB (pre-compiled library files)
- .\QB45\HLP (Help files)

The “Run Easy Setup” will just install to a default location, usually C:\QB45 as well as default options which we do not want. We want to place our install in C:\DEVEL\QB45\\*.\*

When you're ready

Setup Main Menu  
~~~~~  
Easy Setup automatically installs QuickBASIC with all the options set in a way that should work for you.  
Full Setup lets you change the standard options.

| To:                                            | Press: |
|------------------------------------------------|--------|
| Run Easy Setup                                 | E      |
| Run Full Setup                                 | F      |
| Exit                                           | X      |
| Exit and Run "QB Express" (a training program) | T      |

Use UP and DOWN arrow keys to highlight your choice, then press ENTER, or just type the letter that appears to the right of your selection.

In the next screen change the “Set Paths/Directories for files” options to your C:\DEVEL\QB45 path.

When you're ready

| To:                                     | Press: |
|-----------------------------------------|--------|
| Set Paths/Directories for Files         | D      |
| Set Color and Display Options           | C      |
| Show Options                            | S      |
| Install QuickBASIC With Current Options | I      |
| Return to Setup Main Menu               | R      |

Use UP and DOWN arrow keys to highlight your choice, then press ENTER, or just type the letter that appears to the right of your selection.

Make sure you select the correct directory name of your development directory if it is different to C:\DEVEL. I don't recommend using the default directory.

Full Setup: Set Paths/Directories for Files - Page 1 of 4

Set EXE Path:

C:\QB45

This is the directory to which Setup should copy executable programs  
(for example, QB.EXE, BC.EXE, LINK.EXE, LIB.EXE).

Press the TAB key if you want to make changes  
or select from the menu below.

When you're  
ready

| To:                       | Press: |
|---------------------------|--------|
| Continue                  | C      |
| Return to Full Setup Menu | R      |

Full Setup: Set Paths/Directories for Files - Page 1 of 4

Set EXE Path:

C:\DEVEL\QB45

This is the directory to which Setup should copy executable programs  
(for example, QB.EXE, BC.EXE, LINK.EXE, LIB.EXE).

Press the TAB key if you want to make changes  
or select from the menu below.

When you're  
ready

| To:                       | Press: |
|---------------------------|--------|
| Continue                  | C      |
| Return to Full Setup Menu | R      |

Use TAB to select C Continue.

In the next screen create a path to for our "Include" Files. Name it "INC" as a subdirectory of \QB45 so you have C:\DEVEL\QB45\INC

## A Beginners Guide To DOS Programming

Full Setup: Set Paths/Directories for Files - Page 2 of 4

Set INCLUDE Path:

C:\DEU\QB45\INC

This is the directory to which Setup should copy include files and sample programs (for example, QB.BI, TORUS.BAS, QCARDS.BAS).

Press the TAB key if you want to make changes or select from the menu below.

When you're ready

| To:                       | Press: |
|---------------------------|--------|
| Continue                  | C      |
| Back Up One Screen        | B      |
| Return to Full Setup Menu | R      |

Select C Continue.

Change the library \LIB path as we did above for \INC.

Full Setup: Set Paths/Directories for Files - Page 3 of 4

Set LIB Path:

C:\DEU\QB45\LIB

This is the directory to which Setup should copy all the libraries (for example, BQLB45.LIB, BRUN45.LIB, BCOM45.LIB).

Press the TAB key if you want to make changes or select from the menu below.

When you're ready

| To:                       | Press: |
|---------------------------|--------|
| Continue                  | C      |
| Back Up One Screen        | B      |
| Return to Full Setup Menu | R      |

Select C Continue and set the path for the help files .\QB45\HLP.

Full Setup: Set Paths/Directories for Files - Page 4 of 4

Set HELP Path:

C:\DEV\QB45\HLP

This is the directory to which Setup should copy the help files used by QB's On-Line Help System (for example, QB45QCK.HLP).

Press the TAB key if you want to make changes or select from the menu below.

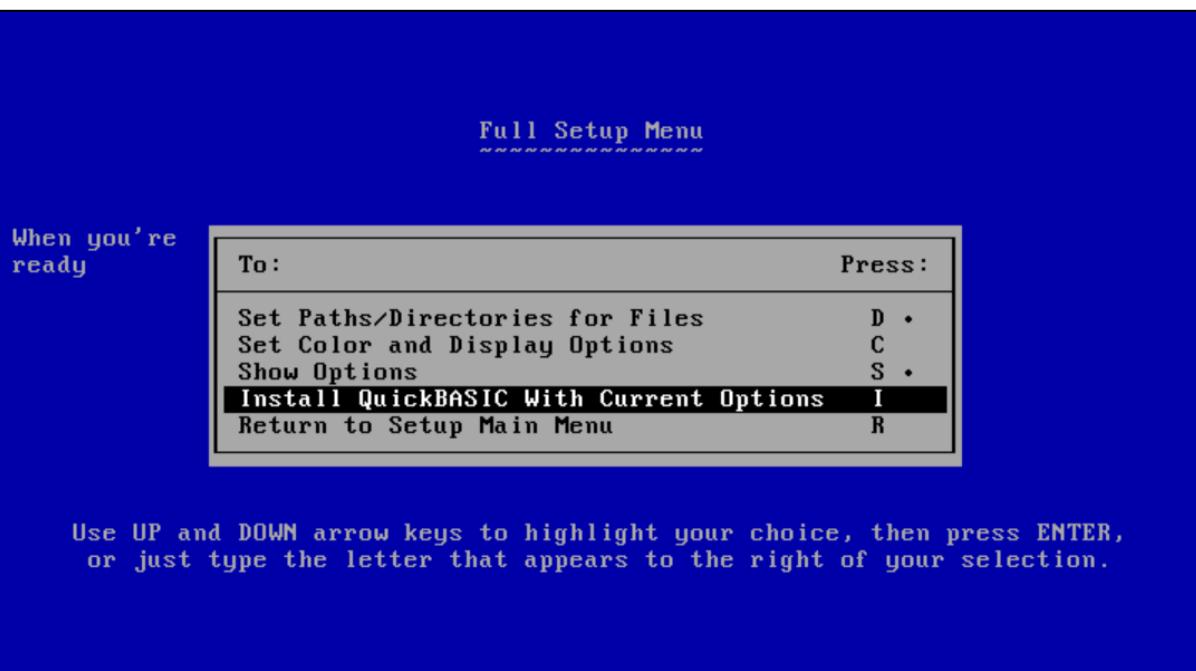
When you're ready

| To:                       | Press: |
|---------------------------|--------|
| Return to Full Setup Menu | R      |
| Back Up One Screen        | B      |

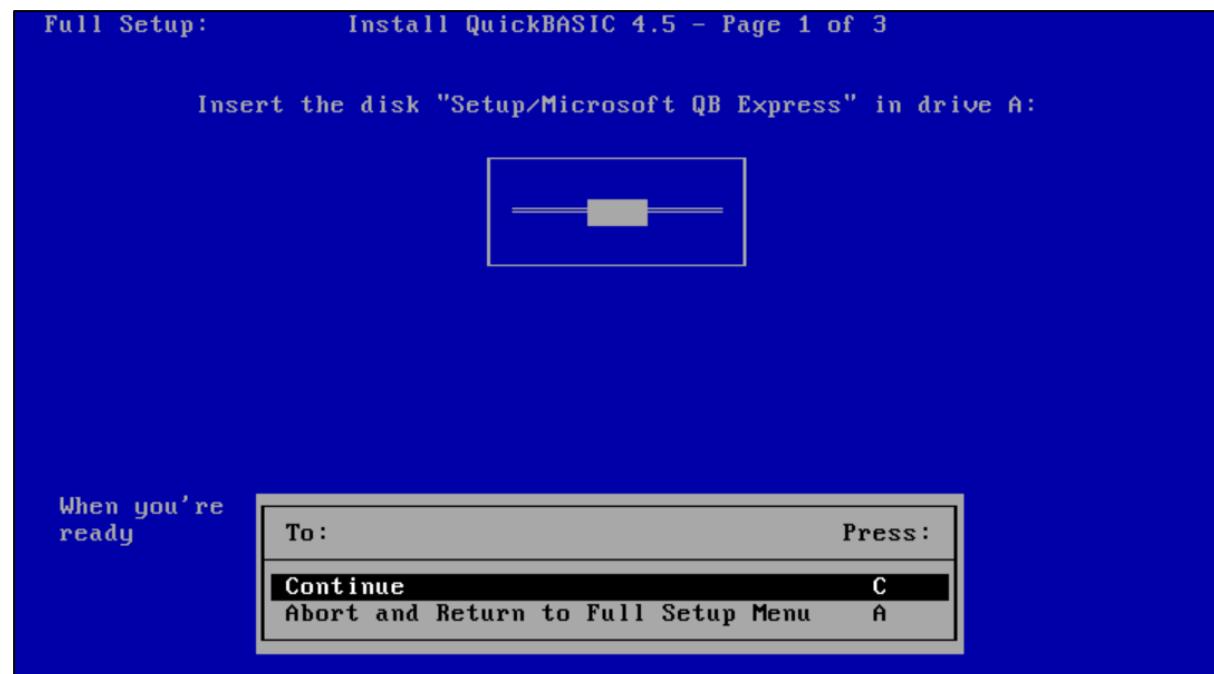
Select R "Return to the Full Setup Menu" as we are done setting the custom install paths.

The default "Set Color and Display Options" are usually fine to leave as they are and can be altered at any time later. The "Show Options" will let you view and check the current install options before proceeding with the install.

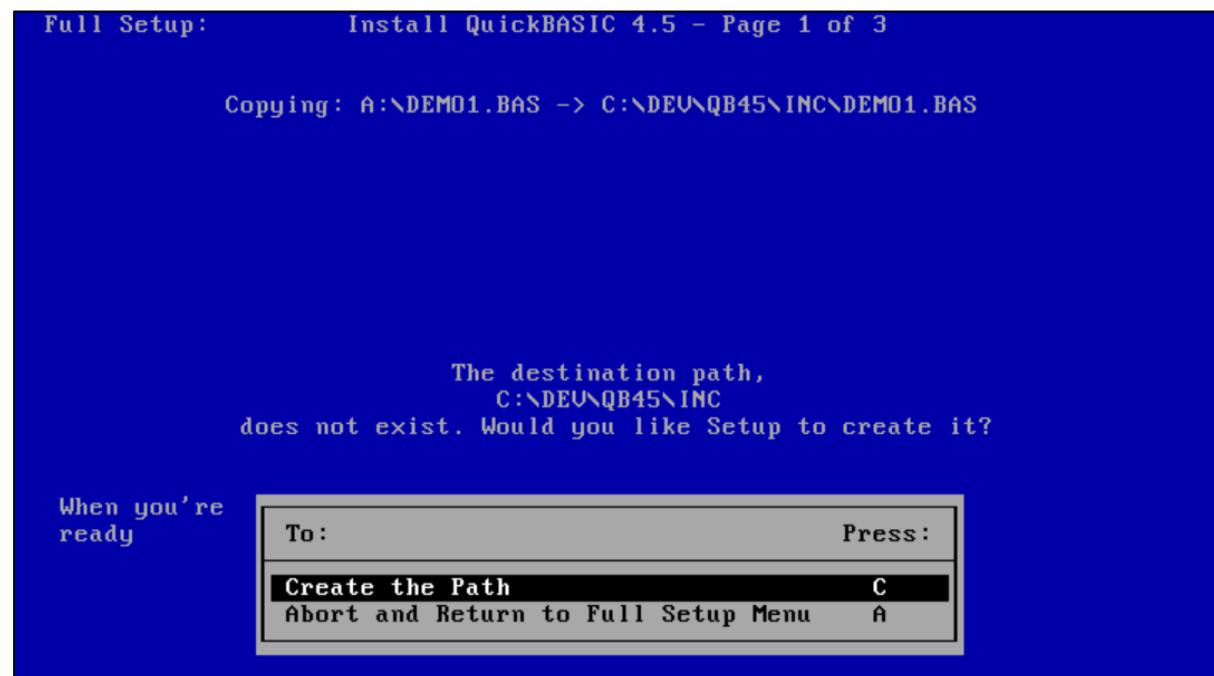
Next select "Install QuicBASIC With Current Options".



The next step relates to the current “setup.img” disk so it OK to just continue.



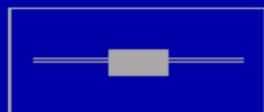
Select “Create the Path” for all of the next screens.



Next mount the “program.img” and select Continue.

Full Setup:      Install QuickBASIC 4.5 - Page 2 of 3

Insert the disk "Program/Microsoft QB Advisor" in drive A:



When you're  
ready

|                                     |        |
|-------------------------------------|--------|
| To:                                 | Press: |
| Continue                            | C      |
| Abort and Return to Full Setup Menu | A      |

Select "Create the Path" and continue.

Full Setup:      Install QuickBASIC 4.5 - Page 2 of 3

Copying: A:\QB45QCK.HLP -> C:\DEU\QB45\HLP\QB45QCK.HLP

The destination path,  
C:\DEU\QB45\HLP  
does not exist. Would you like Setup to create it?

When you're  
ready

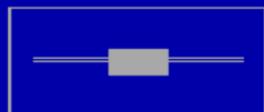
|                                     |        |
|-------------------------------------|--------|
| To:                                 | Press: |
| Create the Path                     | C      |
| Abort and Return to Full Setup Menu | A      |

Next mount the "qb\_advisor.img" and continue.

Full Setup: Install QuickBASIC 4.5 - Page 2 of 3

Copying: A:\QB45ADUR.HLP -> C:\DEU\QB45\HLP\QB45ADUR.HLP

Insert the disk "Program/Microsoft QB Advisor" in drive A:



When you're ready

| To:                                 | Press: |
|-------------------------------------|--------|
| Continue                            | C      |
| Abort and Return to Full Setup Menu | A      |

Select "Create the Path" to continue.

Full Setup: Install QuickBASIC 4.5 - Page 2 of 3

Copying: A:\ADUR\_EX\CALL\_EX.BAS -> C:\DEU\QB45\INC\ADUR\_EX\CALL\_EX.BAS

The destination path,  
C:\DEU\QB45\INC\ADUR\_EX  
does not exist. Would you like Setup to create it?

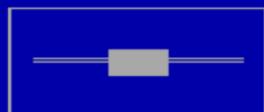
When you're ready

| To:                                 | Press: |
|-------------------------------------|--------|
| Create the Path                     | C      |
| Abort and Return to Full Setup Menu | A      |

Next mount the utilities\_1.img and select Continue.

Full Setup: Install QuickBASIC 4.5 - Page 3 of 3

Insert the disk "Utilities" in drive A:



When you're ready

| To:                                 | Press: |
|-------------------------------------|--------|
| Continue                            | C      |
| Abort and Return to Full Setup Menu | A      |

Select "Create the Path" to continue.

Full Setup: Install QuickBASIC 4.5 - Page 3 of 3

Copying: A:\BQLB45.LIB -> C:\DEU\QB45\LIB\BQLB45.LIB

The destination path,  
C:\DEU\QB45\LIB  
does not exist. Would you like Setup to create it?

When you're ready

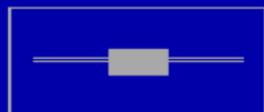
| To:                                 | Press: |
|-------------------------------------|--------|
| Create the Path                     | C      |
| Abort and Return to Full Setup Menu | A      |

Next mount the utilities\_2.img and select Continue.

Full Setup:                    Install QuickBASIC 4.5 - Page 3 of 3

Copying: A:\BASIC45.LIB -> C:\DEV\QB45\LIB\BASIC45.LIB

Insert the disk "Utilities" in drive A:



When you're ready

|                                     |        |
|-------------------------------------|--------|
| To:                                 | Press: |
| Continue                            | C      |
| Abort and Return to Full Setup Menu | A      |

The next screen shows that you have completed the install. After this we will next to choose how to start the application. Don't set the path environment in AUTOEXEC.BAT just yet.

QuickBASIC Successfully Installed

Microsoft QuickBASIC 4.5 is now installed on drive C: in directory  
C:\DEV\QB45.

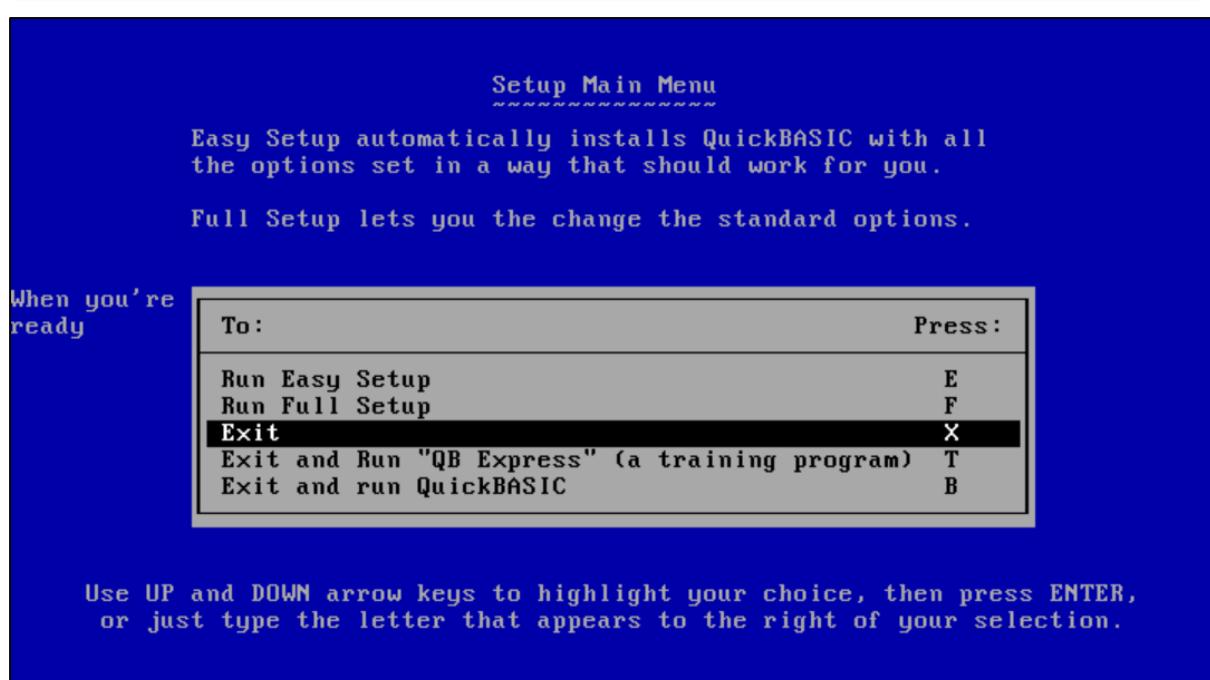
To enter QuickBASIC from DOS, type "QB" from the  
C:\DEV\QB45 directory.

For best results, make sure this directory is in your PATH environment  
variable defined in AUTOEXEC.BAT. See your DOS manual for details.

When you're ready

|          |        |
|----------|--------|
| To:      | Press: |
| Continue | C      |

Exit the installer and unmount the floppy disk image from VirtualBox.



Note that the "Paths" can be altered after the full install if needed.

QB45 appears to run and compile without setting the environment path in the AUTOEXEC.BAT or the launch batch file but I would consider SETting the path statement just the same.

See the next section below "Launching QB45" for how to create the batch files for QB.EXE

#### **Install (qb45.zip) Copy paste.**

This is a copy of the full install from the previous full 5 disk setup. Simply unpack the qb45.zip file and copy the.\QB45\\*.\* directory to a suitable location on your DOS drive. I typically use C:\DEVEL\QB45\\*.\*

Be sure to Use the Menu "Options" "Set Paths..." to set the correct path locations for QB45 library files.

#### **Launching QB45**

We can launch the application directly but I would recommend setting the environment path and creating a batch file to lunch the QB.EXE

We can do this in 2 different ways as is described in the last image capture from the full install above. Either by SETting the path environment in the AUTOEXEC.BAT file (Not recommended in Windows 95), or by placing the path in our batch file to launch the application. The former is fine if you only have one or two developments environments but can become complex and prone to name and path conflicts when multiple development environments are installed. I am preferable to SETting the paths in the launching batch file for each separate development environment. This is also the recommended Windows 95 method for this guide.

To use the AUTOEXEC.BAT (FDAUTO.BAT). **Not recommended in Windows 95.**

We can Place the additional path after the pre-existing SET

PATH=C:\WINDOWS;C:\WINDOWS\COMMAND;C:\SOSLINKS;C:\DEVEL\QB45

Or we can place the following in the appropriate single line, usually near the end of the file. I would suggest after the :END label in FDAUTO.BAT

```
:END  
SET PATH=%path%;C:\DEVEL\QB45
```

Note when using AUTEXEC.BAT or any other batch file, command.com has a 127 character limit on the path length. If you need long paths consider setting the PATH variable in CONFIG.SYS instead as it not affected by the 127 character limit.

<https://www.robvanderwoude.com/path.php>

Next you can create a batch file as is described in “**Application launch BAT**” and place it in the **C:\DOSLINKS** directory.

### QB45.BAT

```
@ECHO OFF  
REM Launcher for QB45  
REM SET PATH=%path%;C:\WINDOWS\COMMAND;C:\DOSLINKS  
REM CTMOUSE /R55  
SET PATH=%path%;C:\DEVEL\QB45  
CD \DEVEL\QB45  
CALL C:\DEVEL\QB45\QB.EXE %1 %2 %3 %4 %5 %6 %7 %8 %9
```

Or a more complete version of the batch file with an error check.

### QB45.BAT

```
@ECHO OFF  
REM Launcher for QuickBAISICv45  
IF EXIST C:\DEVEL\QB45\QB.EXE GOTO QB  
GOTO NOFOUND  
GOTO END  
:QB  
REM You can add environment paths or change working directory here.  
REM SET PATH=%path%;C:\WINDOWS\COMMAND;C:\DOSLINKS  
REM CTMOUSE /R55  
SET PATH=%path%;C:\DEVEL\QB45  
CD \DEVEL\QB45  
REM Launch QB45  
CALL C:\DEVEL\QB45\QB.EXE %1 %2 %3 %4 %5 %6 %7 %8 %9  
GOTO :END  
:NOFOUND  
ECHO QUICKBAISIC not found!  
ECHO Press any key to end...  
PAUSE  
:END  
CLS
```

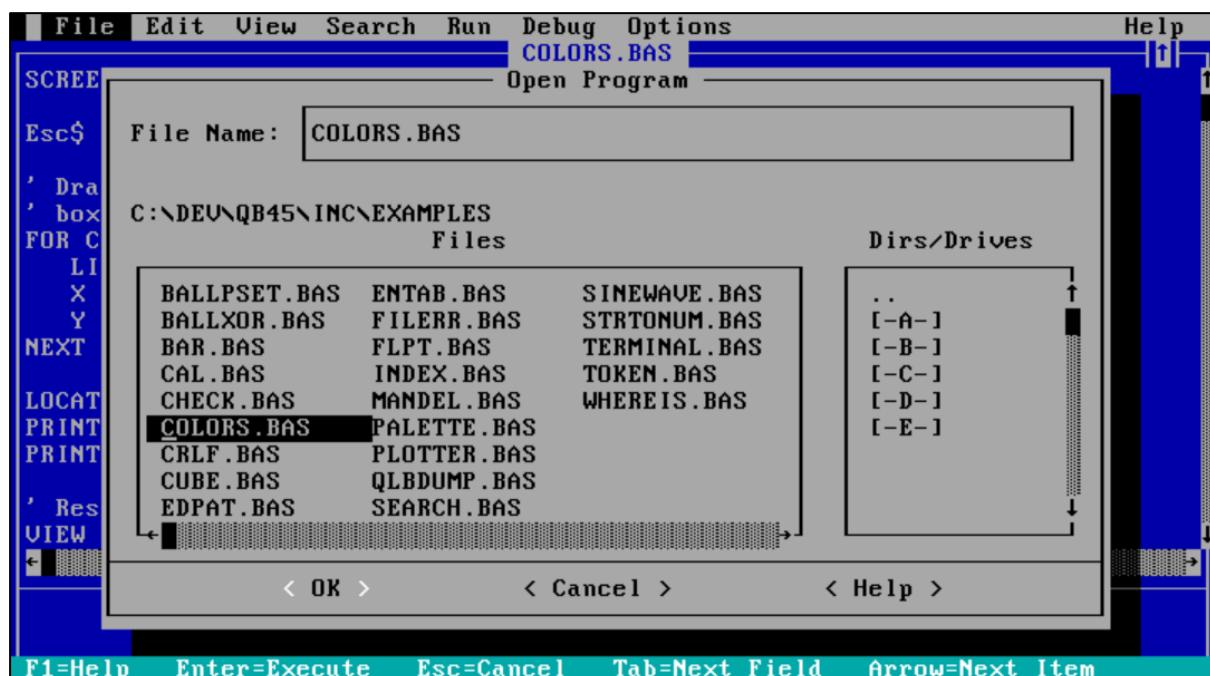
Create a shortcut to the QB45.BAT on your desktop or in the ALaunch directory. A Shortcut to a DOS application or batch file is a PIF file. Adjust the font size in the batch file properties or Allow more space in “Initial environment” for variables if needed.

A PIF file will be created only if you need to change the “Properties” for the batch file. If we use our shortcut to the batch file it will make use of the same name PIF if it has been created so you do not need to change the shortcut from the batch file to the PIF.

### Run a test application.

Open the QuickBASIC IDE using the batch file. Use the ALT menu options “File -> Open Program...” and then using the TAB and navigation keys, navigate to \INC\EXAMPLES.

Select COLORS.BAS and then OK to load the source in the IDE.



Next select “Run -> Start” to run and test the source code in interpreted mode.

The screenshot shows a DOS terminal window with a menu bar at the top. The menu bar includes File, Edit, View, Search, Run, Debug, Options, and Help. A context menu is open over some code, showing options: Start (Shift+F5), Restart, Continue (F5), and Make EXE File... The main window displays a program in BASIC. The code uses the Esc\$ character constant to represent the escape key. It draws three boxes: a green one in the top-left, a red one in the middle-left, and a yellow one in the bottom-right. The background is black. The code also includes instructions to press ESC to end the program and any other key to continue.

```
SCREEN 1
Esc$ = CHR$(27)

' Draw three boxes and pair
' box with a different color
FOR ColorVal = 1 TO 3
    LINE (X, Y)-STEP(60, 50), ColorVal, BF
    X = X + 61
    Y = Y + 51
NEXT ColorVal

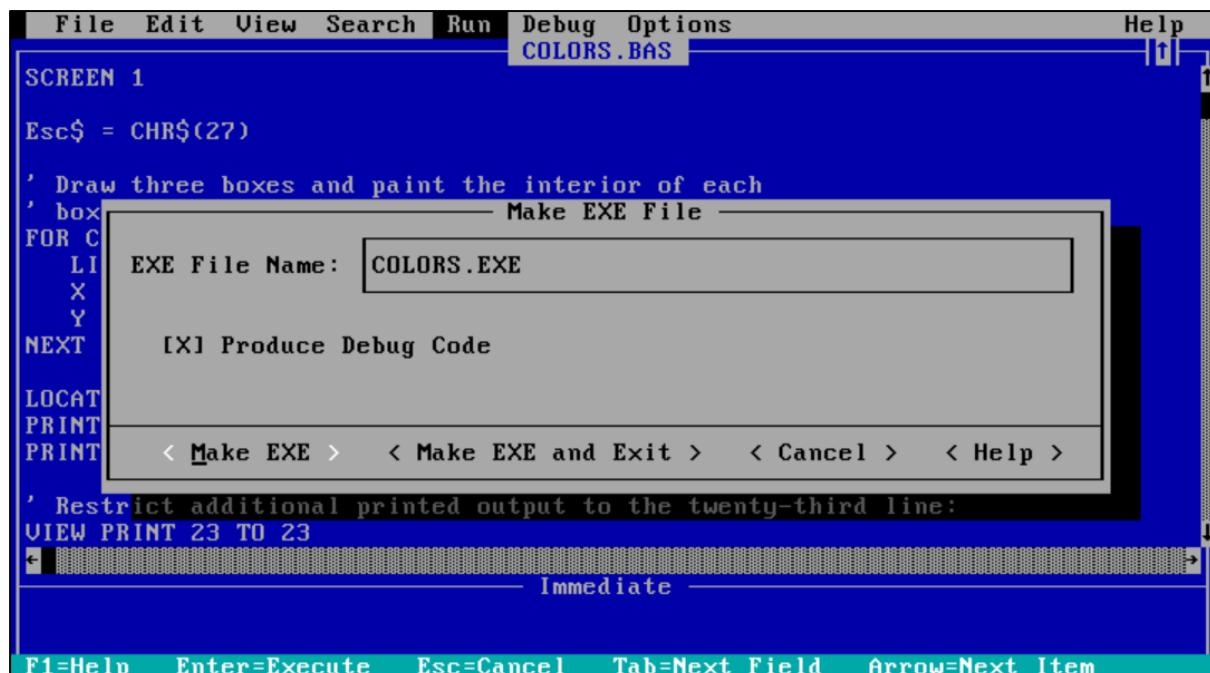
LOCATE 21, 1
PRINT "Press ESC to end."
PRINT "Press any other key to continue."

' Restrict additional printed output to the twenty-third line:
VIEW PRINT 23 TO 23
```

Output:



Next select “Run -> Make EXE File. We will compile the code to a distributable executable file.



Exit the application and navigate to QB45 root directory C:\DEVEL\QB45 and you will find the created COLOR.EXE file.

Note that all compiled executables (OBJ, EXE) will be created in the .\QB45 directory. It is up to you to “Back Up”, “Move” or “Delete” these files. Take note of the original binary files used by QB45 so that you don’t accidentally delete them. Also if you create different projects, be careful and watch for name conflicts.

That's it you are ready to learn and create DOS applications using MS QuickBASIC 4.5.

### QuickBASIC Extended 7.1 (QBX)

Quickbasic Extended 7.1 PDS (Professional Development System) aka QBX: The last version of QuickBasic; it has a few slight improvements on QB 4.5 and a lot of database commands (powerful built-in ISAM database routines) plus some extras.

I am not going to show the install process as it is relatively similar to the QB45 install. I recommend using the “Microsoft BASIC Professional Development System 7.1 (3.5).7z” with the 8 original setup disk images.

Unpacked original 8 setup floppy disk images and a copy of the DOS installed application directories.

<https://www.qbasic.net/en/qbasic-downloads/compiler/qbasic-compiler.htm>

“pds71.zip”

“pds71\_floppy.zip”

Minimal copy and paste of the install directory.

<https://archive.org/details/qb71zip>

“qb71.zip”

Original 8 setup floppy disk images.

<https://winworldpc.com/product/microsoft-basic/pds-71>

“Microsoft BASIC Professional Development System 7.1 (3.5).7z”

QB71 manual

<https://winworldpc.com/product/microsoft-basic/pds-71>

“Microsoft BASIC Professional Development System 7.x Manuals (1990).7z”

---

## Visual BASIC V1.0 for DOS

Microsoft Visual Basic 1.0 for DOS is unique in that it was the only version released for something other than Microsoft Windows. VB for DOS uses the text video mode, but provides forms and UI controls similar to Windows. This product replaces Microsoft BASIC Professional Development System 7.1.

Unpacked floppy images.

<https://www.qbasic.net/en/qbasic-downloads/compiler/qbasic-compiler.htm>

“vbdos.zip”

“vbdos\_floppy.zip”

Original install floppy disk images. 2 – 7 images.

<https://winworldpc.com/product/microsoft-visual-bas/10-for-dos>

“Microsoft Visual Basic 1.0 Standard for MS-DOS (1992) (3.5-1.44mb).7z”

“Microsoft Visual Basic 1.0 Professional for MS-DOS (1992) (3.5-1.44mb).7z”

Set of 7 virtual floppy drives VDF

<https://archive.org/details/ms-vbdos10>

“ms-vbdos10.zip”

In Windows 95 you should be able to use the emulated DOS windows for the installs. If that fails use the “Exit To DOS.PIF” pathway. Ensure that the correct DOS PATHs are set as well as the CD-ROM drivers.

### Install instructions – standard edition.

I will be using the 2 disk Standard Edition from WinWorld PC “Microsoft Visual Basic 1.0 Standard for MS-DOS (1992) (3.5-1.44mb).7z”

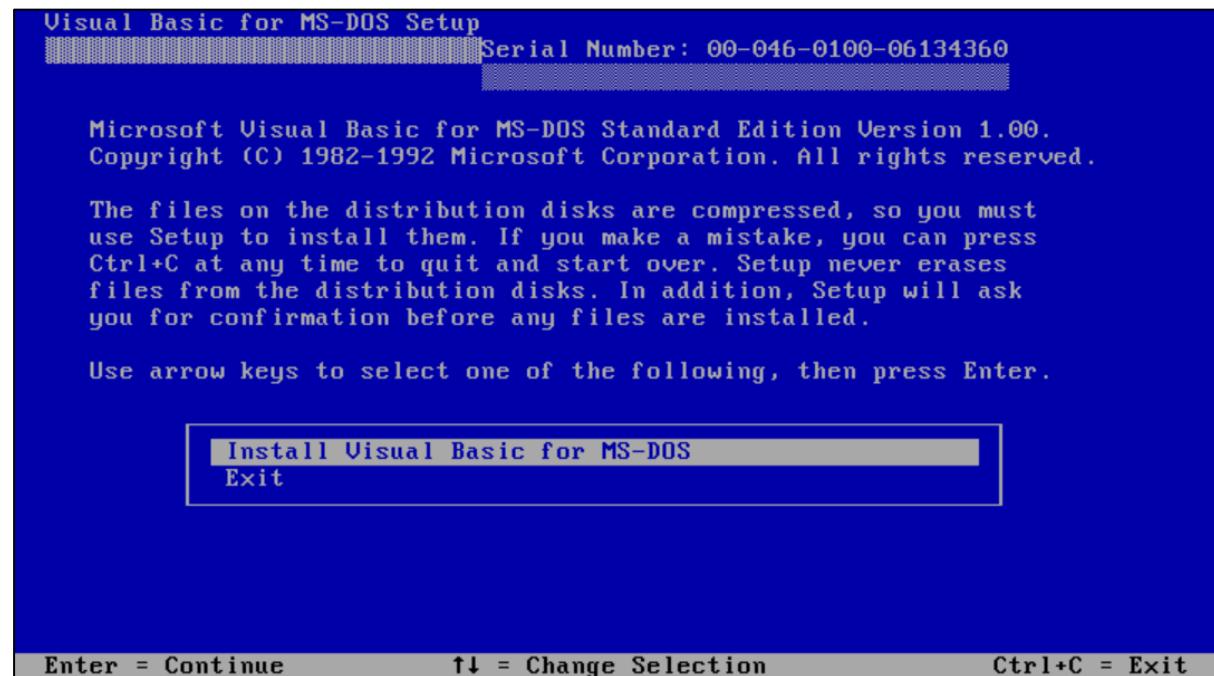
Note:

I am uncertain of the license restriction for use of the Standard and Professional edition. It is up to you to check and purchase a legacy key if it is required.

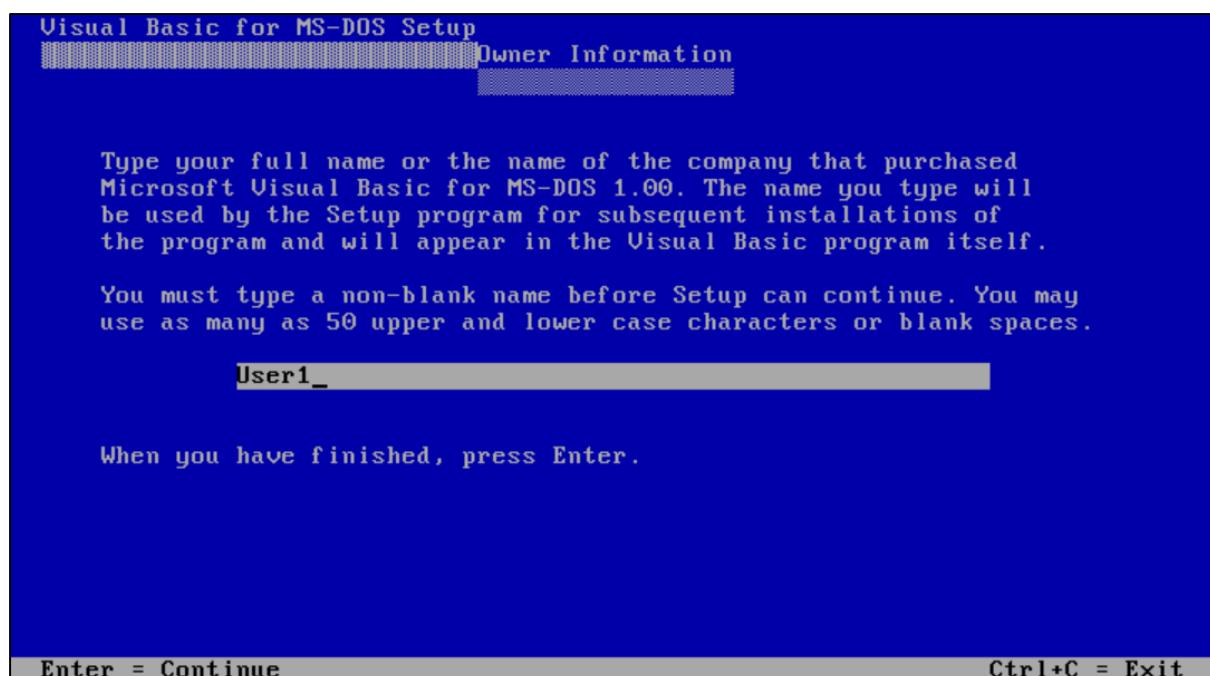
Unpack “Microsoft Visual Basic 1.0 Standard for MS-DOS (1992) (3.5-1.44mb).7z” and mount disk01.img in VirtualBox.

From your Windows 94 command prompt, navigate to drive A: and type SETUP[.EXE].

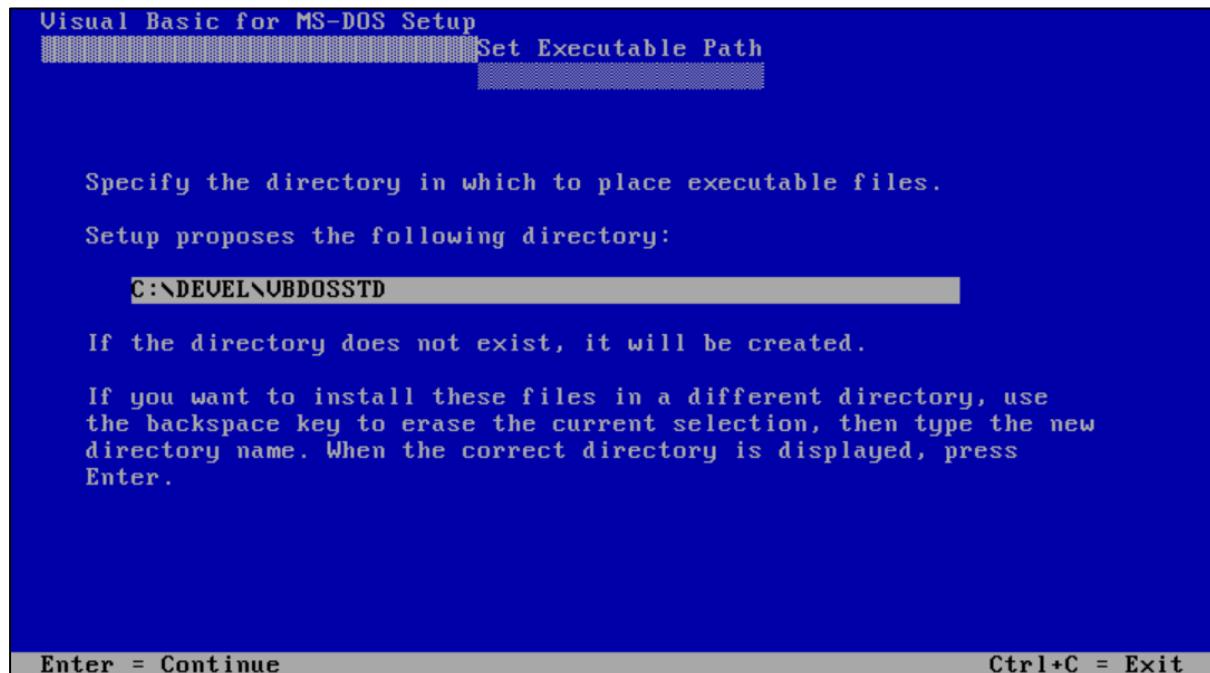
Select “Install Visual Basic for MS-DOS”.



Enter the registered name of your copy of Visual Basic for MS-DOS, and verify the name.



Next select the install directory. I am using C:\DEVEL\\*.\* for this guide which is the FreeDOS 1.3 default (This is the same for Windows 95). Although the default directory is C:\VBDS I have changed this to VBDSSTD to so I will have a different directory name in case I choose to install the Pro edition later.



Enter the include path C:\DEVEL\VBDSSTD\INC

Visual Basic for MS-DOS Setup

Set Include Path

Specify the directory in which to place include and source files.

Setup proposes the following directory:

C:\DEVEL\VBDOSSTD\INC\_

If the directory does not exist, it will be created.

If you want to install these files in a different directory, use the backspace key to erase the current selection, then type the new directory name. When the correct directory is displayed, press Enter.

Enter = Continue

Ctrl+C = Exit

Enter the library path C:\DEVEL\VBDOSSTD\LIB

Visual Basic for MS-DOS Setup

Set Library Path

Specify the directory in which to place library files.

Setup proposes the following directory:

C:\DEVEL\VBDOSSTD\LIB\_

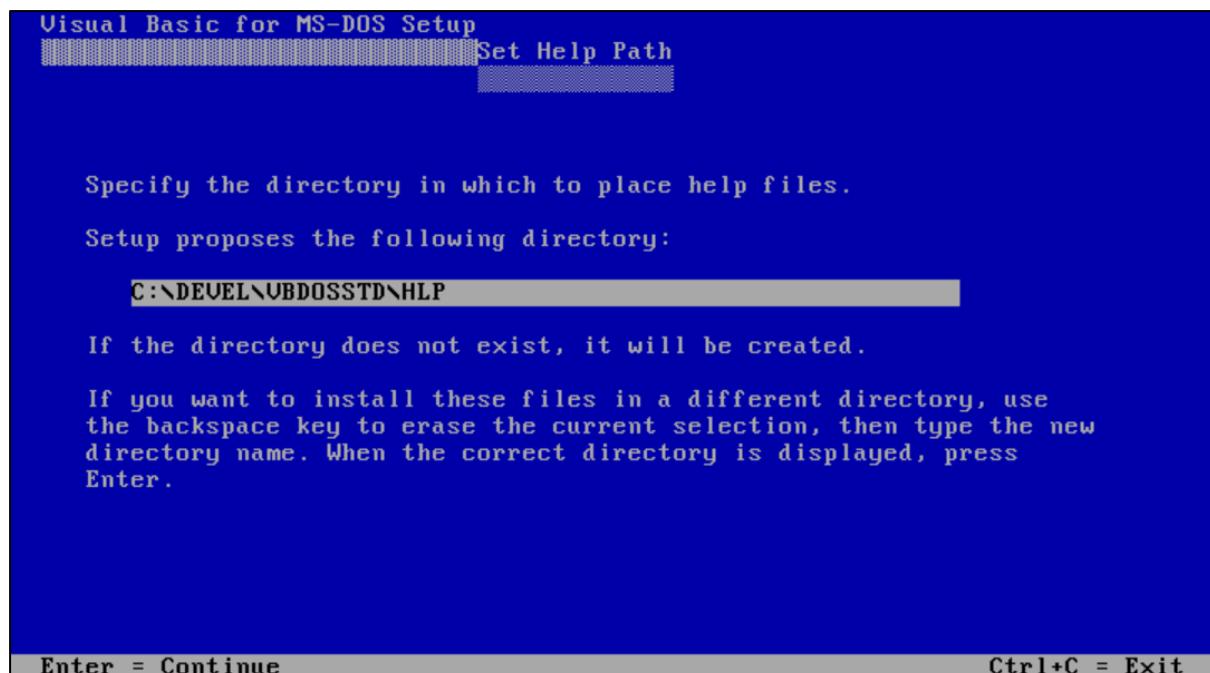
If the directory does not exist, it will be created.

If you want to install these files in a different directory, use the backspace key to erase the current selection, then type the new directory name. When the correct directory is displayed, press Enter.

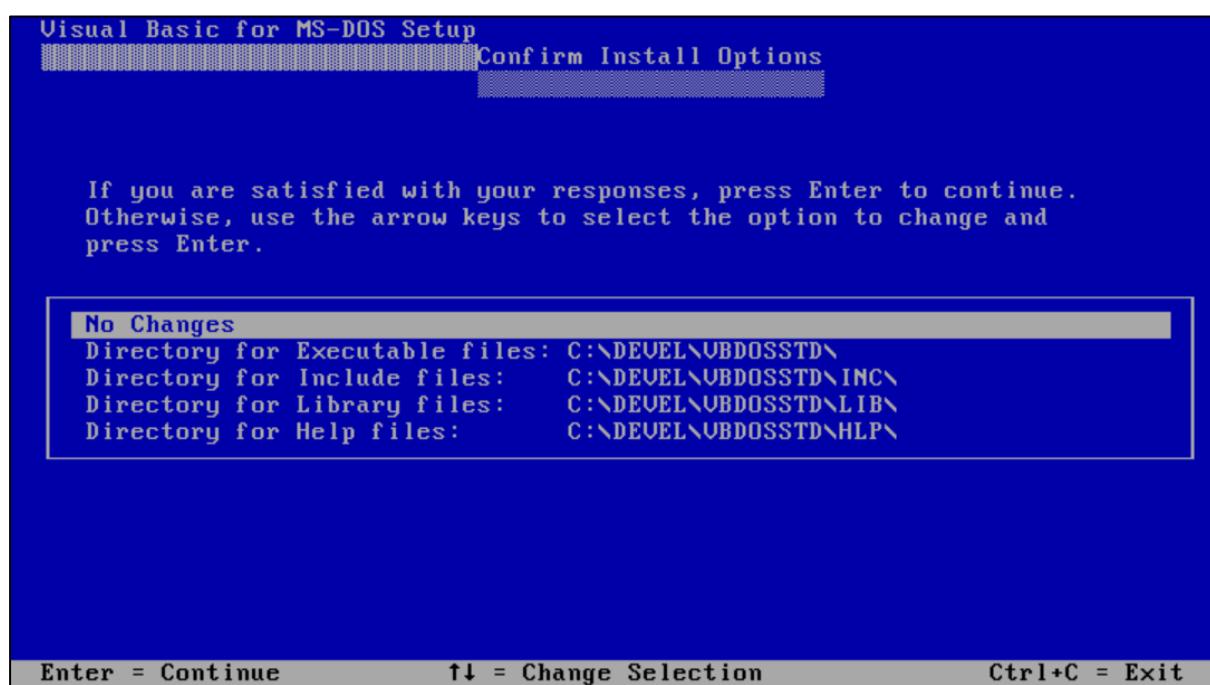
Enter = Continue

Ctrl+C = Exit

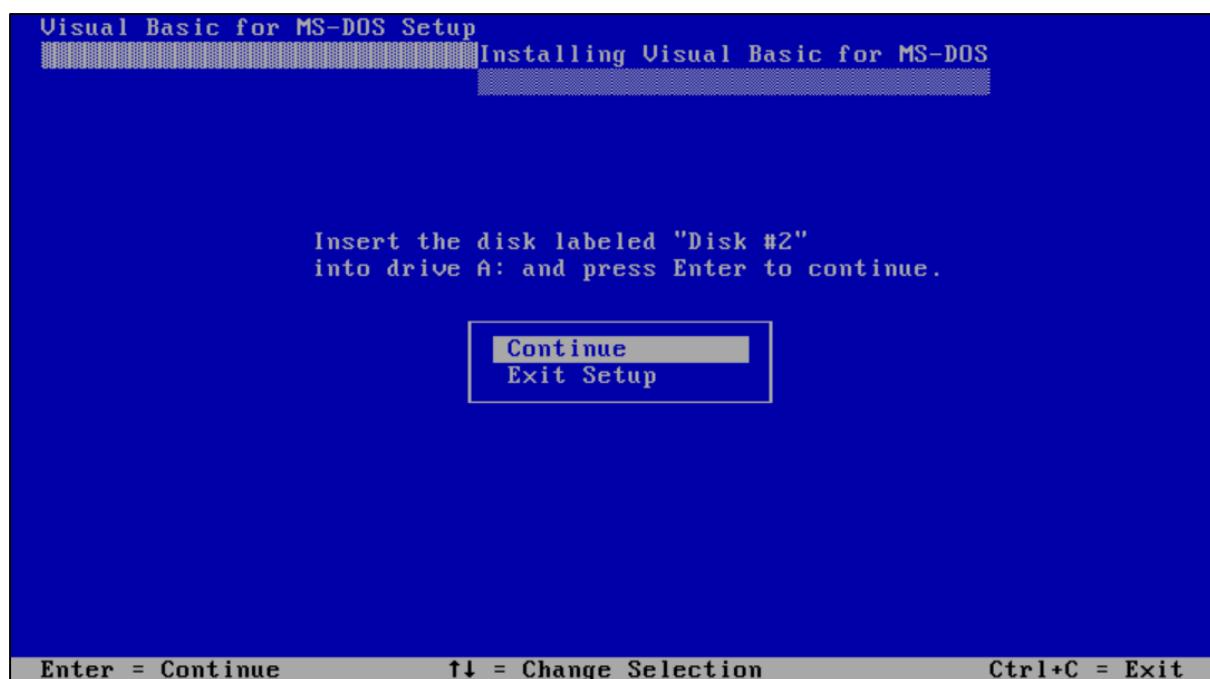
Enter the help path C:\DEVEL\VBDOSSTD\HLP



Check that the install paths are correct and press Enter to continue the install.

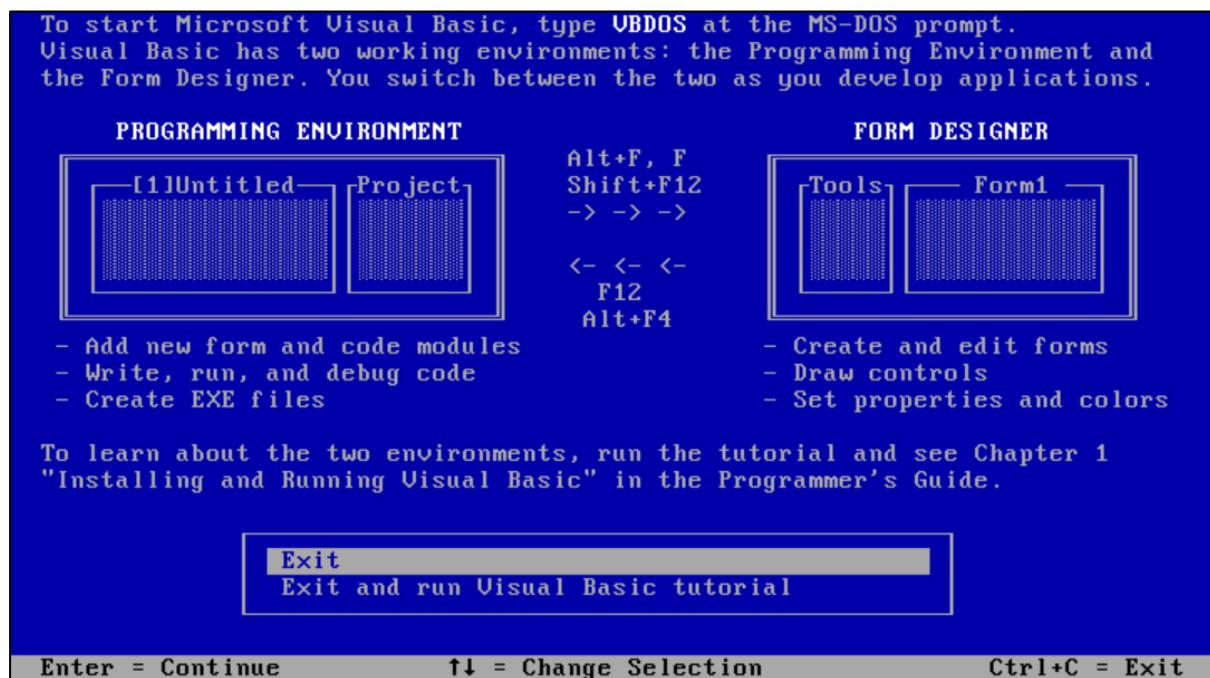


Mount the second disk in VirtualBox and enter to continue the file install.



Visual Basic for MS-DOS is now installed. You can exit the installer or continue to the tutorial.

I am going to "Exit" the installer.



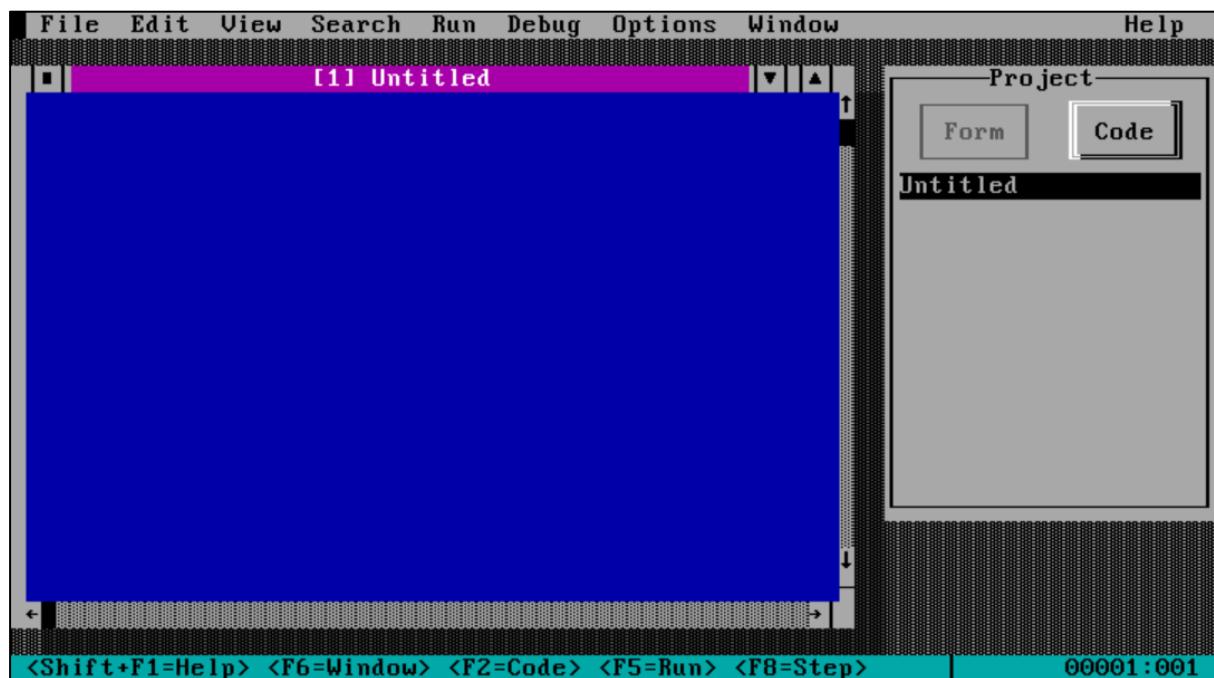
If you are installing to FreeDOS or Windows 95 MS-DOS then the AUTOEXEC.BAT and other launch batch files will not be created.

Create a basic VBDOSSTD.BAT file from the guide "**Application launch BAT**" and place it in the \DOSLINKS directory.

### VBDOSSTD.BAT

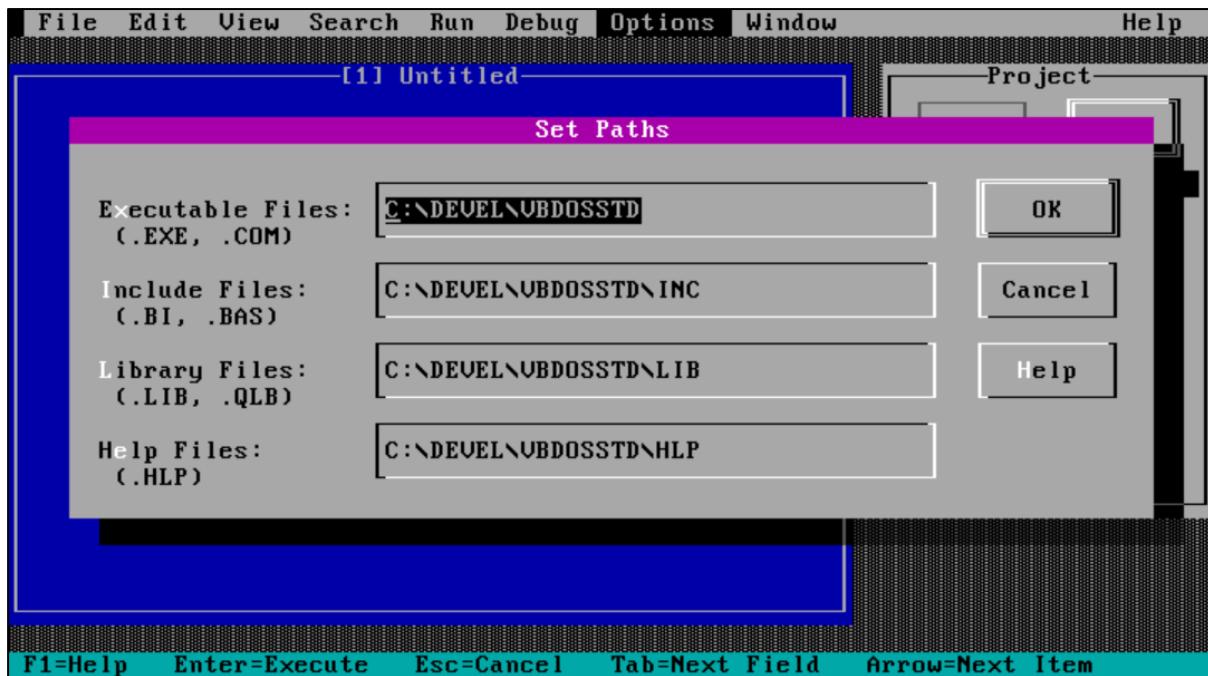
```
@ECHO OFF  
REM Launch for Visual basic for MS-DOS Standard edition.  
CLS  
REM SET PATH=%path%;C:\WINDOWS\COMMAND;C:\DOSLINKS  
REM CTMOUSE /R55  
SET PATH=%path%;C:\DEVEL\VBDOSSTD  
CD \DEVEL\VBDOSSTD  
CALL VBDOS.EXE  
CLS
```

Run VBDOSSTD.BAT



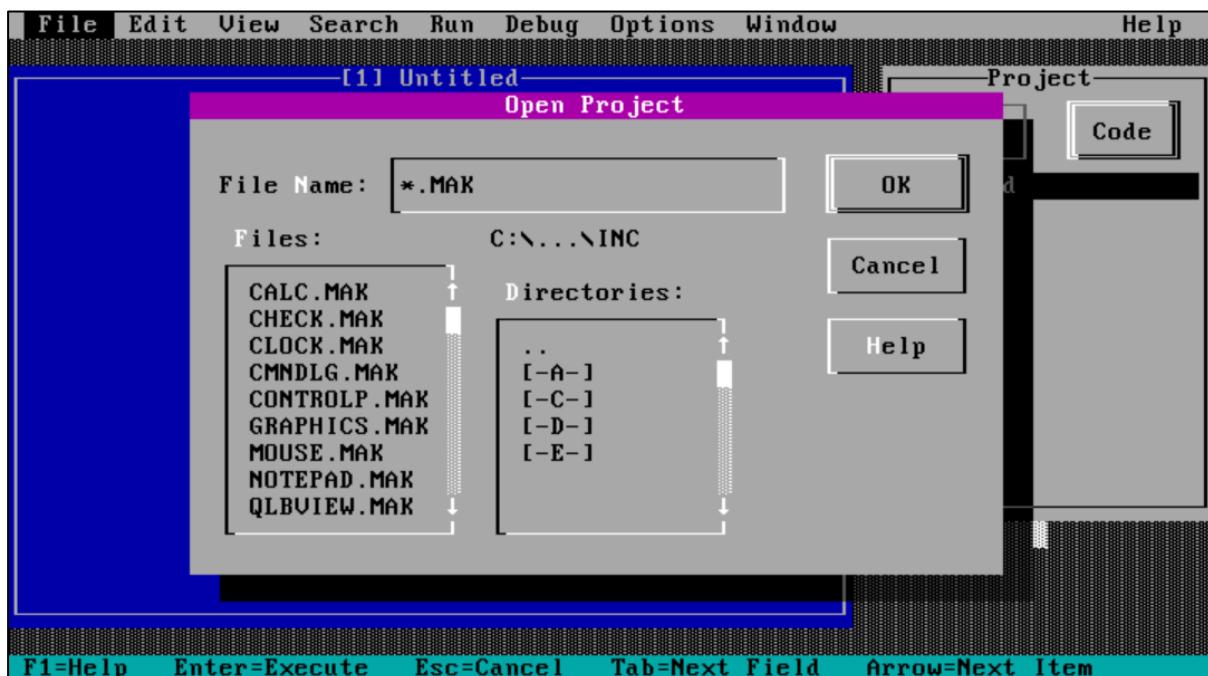
Next double check the paths for the VBDOS config file by selecting Alt + O.

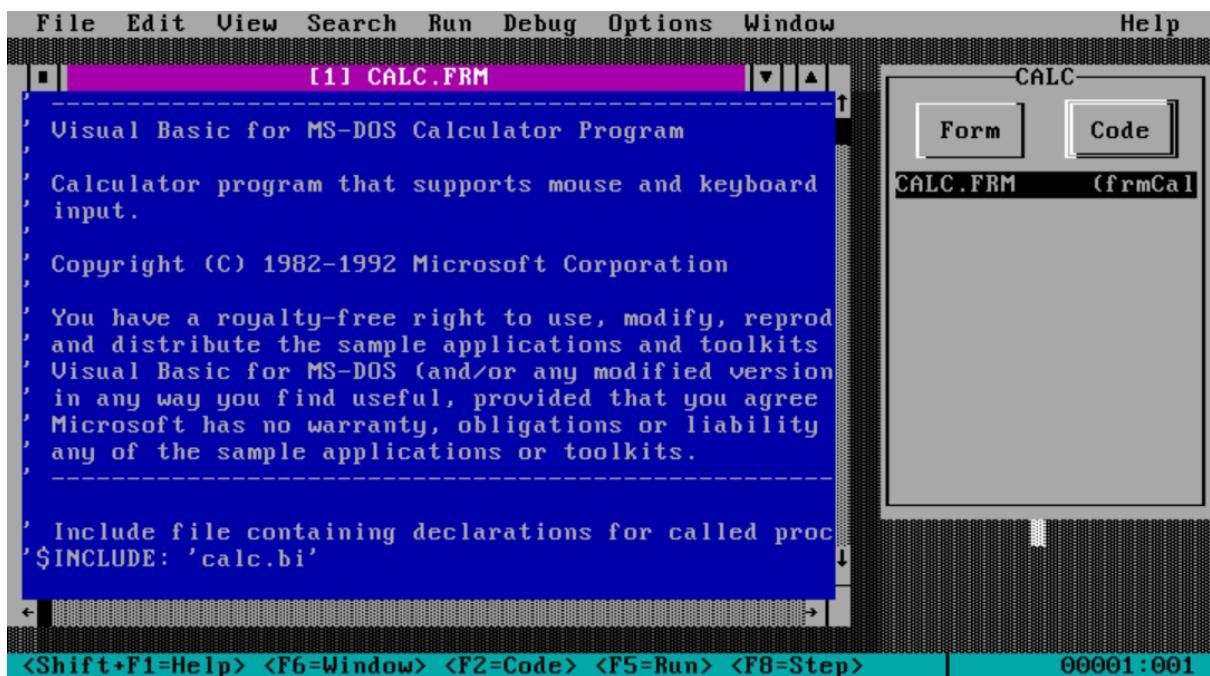
Under “Options -> Set Paths ...” check that the paths are correct. This was set during the install and can always be manually altered at a later time if needed.



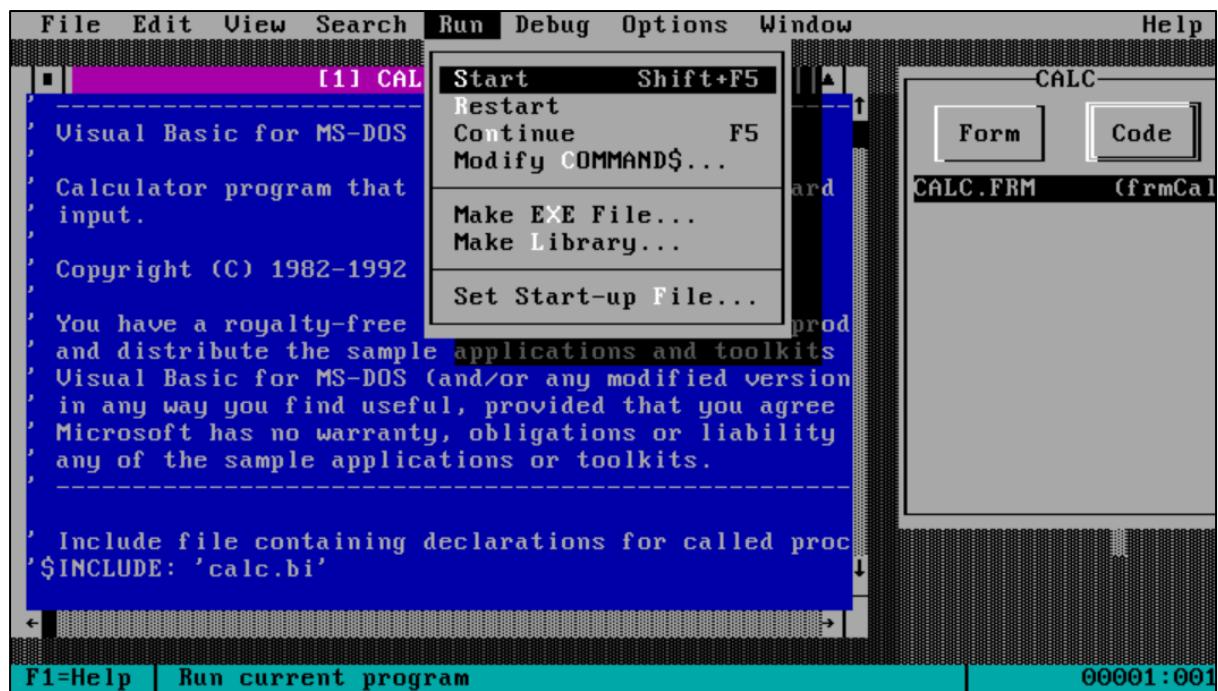
Next load the CALC.MAK example application to test the environment.

"File -> Open Project..." This will be found in the \INC directory.



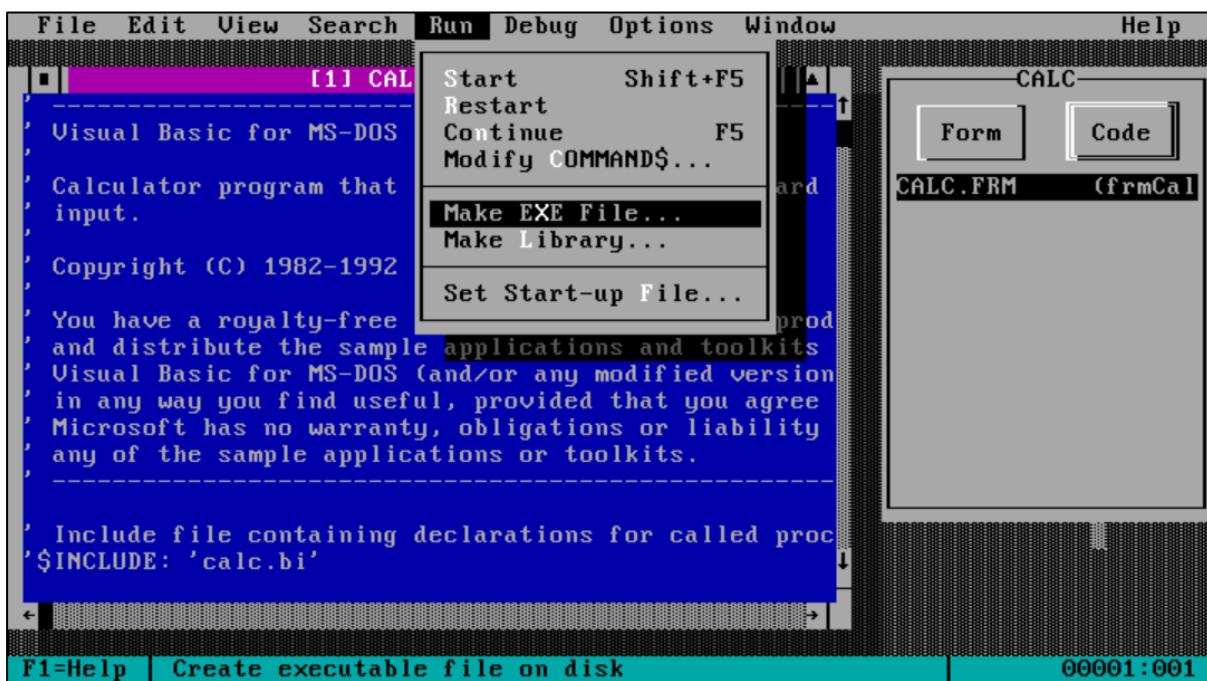


Next select Alt + R "Run -> Start" to test the calculator examples.

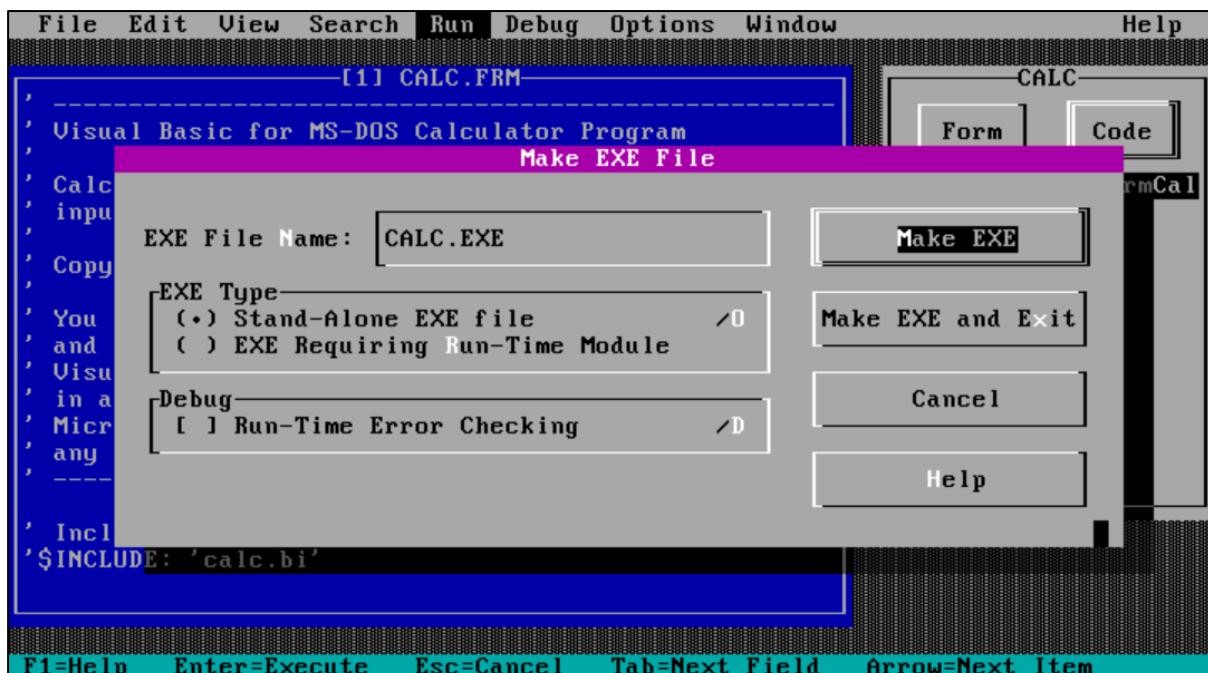




Next select Alt + R "Run -> Make EXE File..." to compile the source to an executable file.

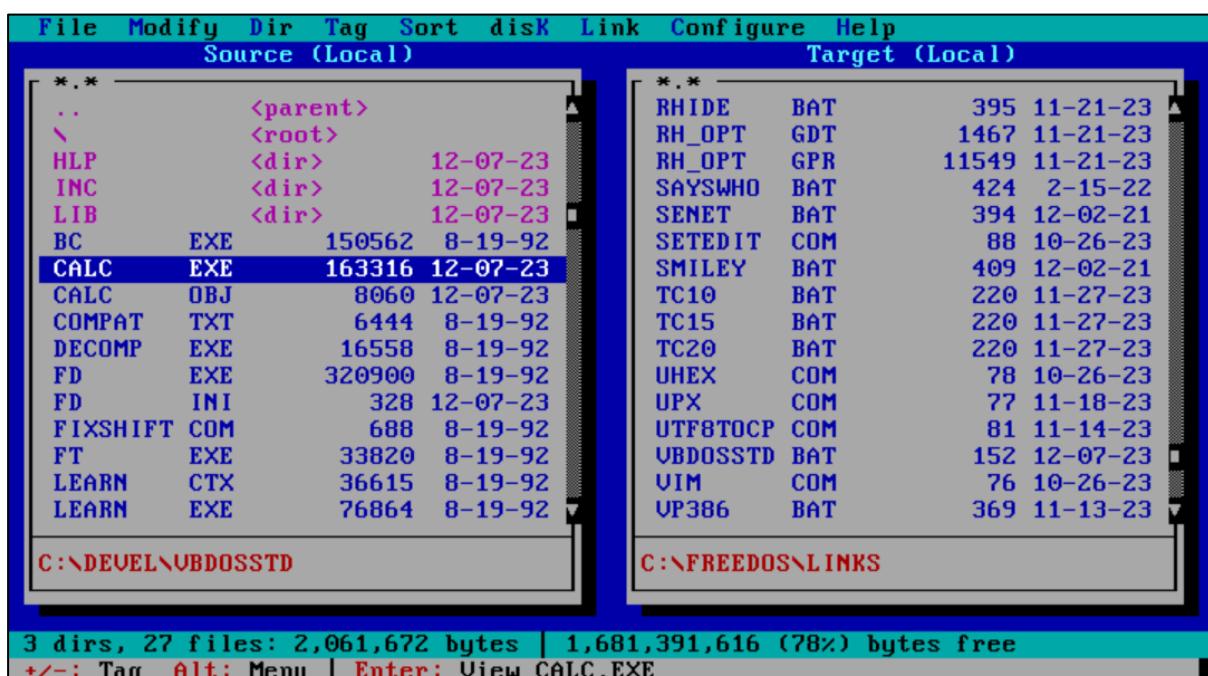


I have selected a "Stand-Alone" EXE so the runtime library will be static compiled into the executable file.

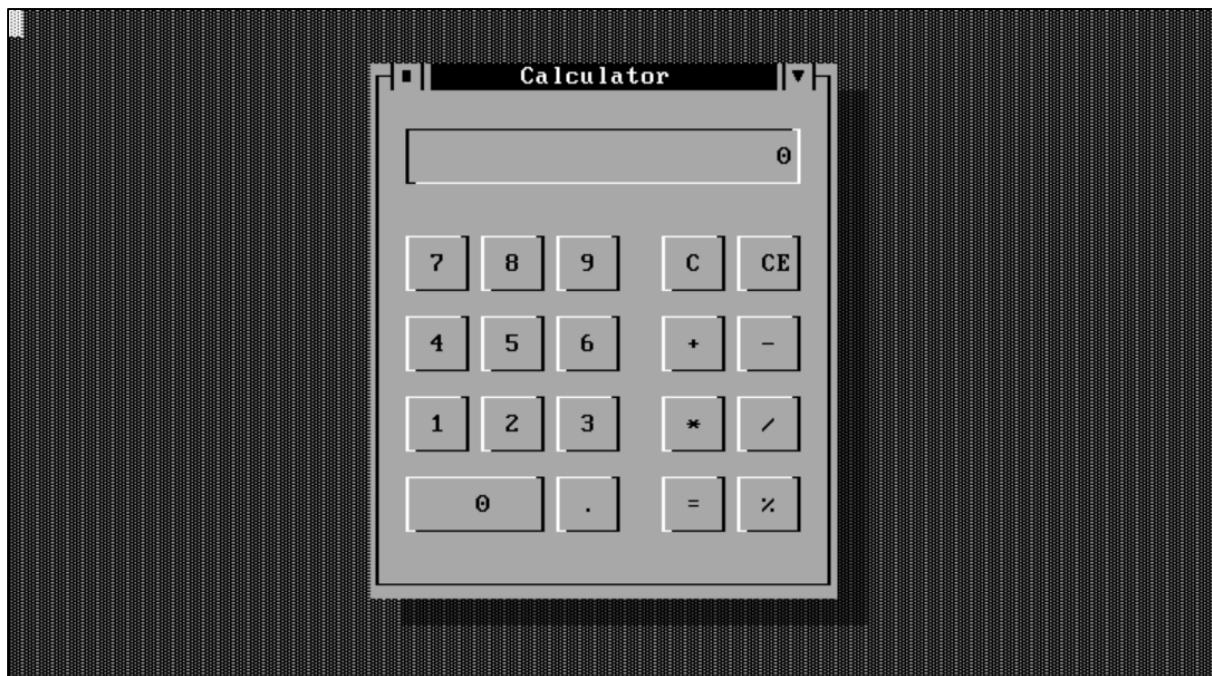


Exit VBDOS or open the MS-DOS Shel and navigate to the VBDOSSTD root directory  
C:\DEVEL\VBDOSSTD

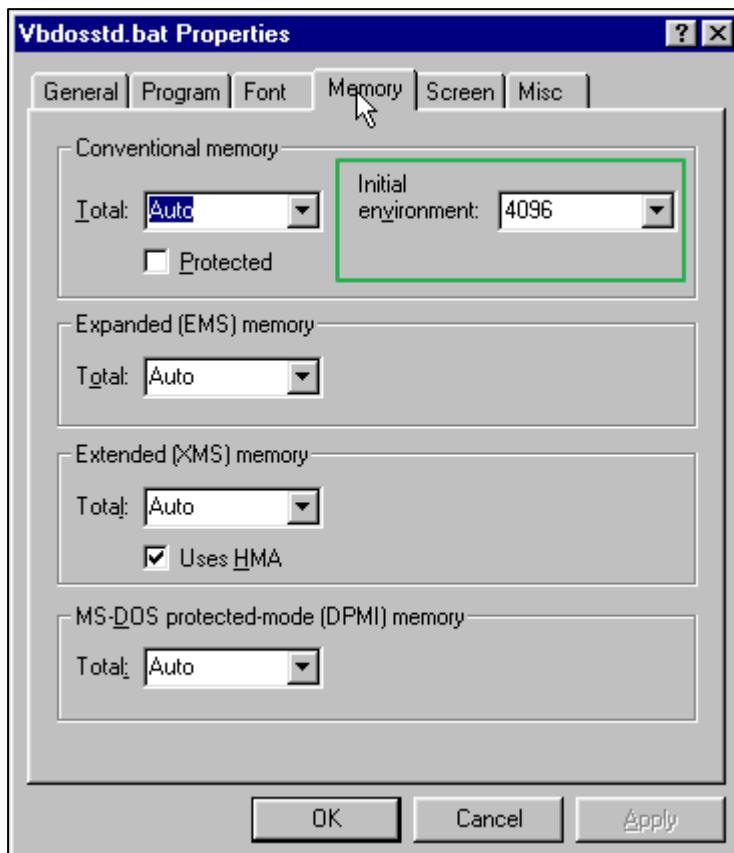
Here you will find the output executable that was compiled from your project. Please be careful with Project and EXE names to avoid name conflicts.



From here you can test the example calculator application as a standalone executable.



Create a shortcut on the Windows desktop or the ALaunch menu directory to the VBDOSSSTD.BAT in the \DOSLINKS directory. A Shortcut to a DOS application or batch file is a PIF file. A PIF file is not required but will be generated if you change the screen Font setting or “Initial environment” setting in the “Properties” dialog or create a shortcut. I would recommend setting the initial environment setting to 4096.



Visual Basic for DOS is now installed and ready for use.

---

## DJGPP (C/C++)

The following is duplicated from the FreeDOS install guide. As we cannot use the FDIMPLES install CD in Windows 95 the most simple pathway here is to copy the entire DJGPP directory from the FreeDOS \DEVEL\DJGPP directory to the Windows 95 C:\DEVEL\DJGPP directory. Alternatively you can construct the DJGPP directory by downloading the files from the DJGPP website zip picker.

DJGPP is a complete 32-bit C/C++ development system for Intel 80386 (and higher) PCs running DOS. It includes ports of many GNU development utilities. The development tools require an 80386 or newer computer to run, as do the programs they produce. In most cases, the programs it produces can be sold commercially without license or royalties.

### IA16GNU

The IA16 compiler can produce real mode x86 16-bit computer code. Creating 16-bit real mode applications is somewhat more advanced than the x80386 protected mode 32-bit DOS applications. If you wish to experiment with x86 real mode for x86 and x286 than it may be better to start with Borland Turbo C++ v 1.0. As such I won't have any focus upon IA16GNU in this guide.

Having a C language compiler is a useful asset when writing applications for DOS. Many libraries for other languages such as FreeBASIC are compiled using a C compiler. Even though FreeBASIC contains many of the popular header files (include files) you will often need to compile the binary archive (a .a file) of the library yourself from the source.

For windows 95 either copy the entire directory from the FreeDOS install, reconstruct the directory structure from the FreeDOS repository or download the zip files from the DJGPP home site.

Typically additional library packages will need to be installed manually.

There is also an older package manager project called pakke but has stalled at the alpha stage in 2003.

### FreeDOS Bonus CD install

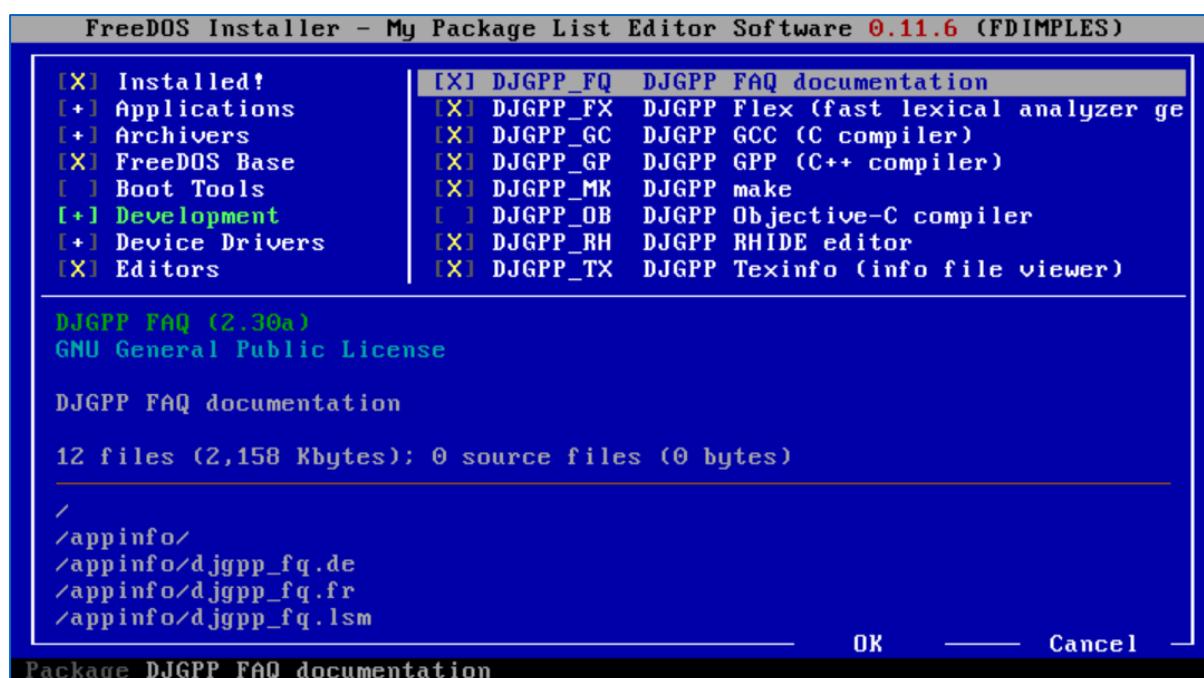
#### **ONLY FOR FREEDOS!**

You can run the install in FreeDOS and then copy the directory structure to the Windows 95 drive.

Mount the FD13BNS.ISO in VirtualBox and then run FDIMPLES.

Navigate to the [ ]Development section and select the following packages from the list.

- [X]DJGPP
- [X]DJGPP\_BN
- [X]DJGPP\_BS
- [X]DJGPP\_DB
- [X]DJGPP\_FQ
- [X]DJGPP\_FX
- [X]DJGPP\_GC
- [X]DJGPP\_GP
- [X]DJGPP\_MK
- [X]DJGPP\_RH
- [X]DJGPP\_TX
- [X]INSIGHT (Optional)
- [X]UPX (Optional)



Once you have selected the above select OK and [Enter] to install the Base DJGPP environment.

You will now have the above files installed into the C:\DEVEL directory.

If you go to the repository <https://www.ibiblio.org/pub/micro/pc-stuff/freedos/files/repositories/1.3/pkg-html/group-devel.html>

You will also find IDebug which is not included on the Bonus CD. This will need to be downloaded and installed to C:\DEVEL\IDEBUG or C:\FREEDOS\BIN manually.

Note that FreeDOS also has its own built in debugger in the .\FREEDOS\BIN directory.

You can download all of the individual files from the repository if you wish and do a manual install but I would strongly recommend using the built in package managers. See the section on manual DJGPP install.

Note that the DJGPP documentation recommends installing to C:\DJGPP. I have followed the FreeDOS 1.3 install directory of C:\DEVEL\DJGPP. As long as the correct path and environment variables are set this shouldn't be an issue.

**Do not** attempt to launch or use DJGPP at this stage. Please scroll down to the section “**Setting up DJGPP IDEs**”

### Manual DJGPP Install from repository

Windows 95 (MS-DOS 7.1)

You can manually download the DJGPP files from the FreeDOS package repository at  
<https://www.ibiblio.org/pub/micro/pc-stuff/freedos/files/repositories/1.3/pkg-html/group-devel.html>

Download the following from the file list. The names will reflect the names found on the FD Bonus CD.

DJGPP binutils

DJGPP Bison

djgpp

djgpp faq

DJGPP Flex

djgpp gcc

djgpp gdb

djgpp gpp

DJGPP make

djgpp rhide

DJGPP Texinfo

Extract the contents of the zip files to a single directory alongside of the zip files keeping the directory and path names from the zip file. djgpp.zip contains 2 directories DEVEL and APPINFO, so all files from the zip archives will be unpacked to

.\Download\_files\DEVEL\\*.\*

and

.\Download\_files\APPINFO\\*.\*

After you have unpacked the entire DJGPP zip file to the 2 directories, copy .\DEVEL\\*.\* to the root directory of your Windows 95 system drive C:\DEVEL\DJGPP\\*.\*

The APPINFO directory isn't required and can be deleted. If you want you can copy .\APPINFO to the DJGPP directory C:\DEVEL\DJGPP\APPINFO\\*.\*

This will now have the manually installed DJGPP files in the same locations as is done by FDIMPLES from the Bonus CD.

### Setting up from the DJGPP website

FreeDOS and Windows 95 (MS-DOS 7.1)

The DJGPP home site has a package picker or zip file picker. From here you can select the required files for your DJGPP installation as separate archives. The home site also contains a full list of additional utilities and libraries for DJGPP programming. If you have used any of the above methods to install DJGPP you may need to come to this repository for additional libraries.

<https://www.delorie.com/djgpp/zip-picker.html>

Pick the download mirror. I would recommend a http site.

Next select "Build and run programs with DJGPP"

Select the DJGPP base components.

[/]C

[/]C++

[/]Bison

[/]Flex

Select RHIDE as the IDE.

I would select "Yes" to the GNU debugger.

Extra Stuff is optional, but if you are considering Game, Graphics or TUI console programming I would also select the 3 Toolkits.

[www.delorie.com/djgpp/zip-picker.html](http://www.delorie.com/djgpp/zip-picker.html)

[search](#)

### DJGPP Zip File Picker

This page helps you decide which ZIP files you need to download based on what kinds of tasks you are trying to do.

Note that this service covers those packages that are most popular. Not all DJGPP packages are included. If you want to download more than this service covers, simply visit the FTP site and choose the extra packages you want (they're in the *v2\** directories). Look for files called *00\_index.txt* in each directory to get descriptions of all the packages in that directory.

#### FTP Site

Select a suitable FTP site:

US, New Hampshire (<http://www.delorie.com>)

#### Basic Functionality

Pick one of the following:

Build and run programs with DJGPP

Which operating system will you be using?

Windows 2000/XP

Do you want to be able to read the on-line documentation? (note: rhide and emacs include online help viewers)

Yes

Which programming languages will you be using?

- C
- C++
- Objective C
- Assembler
- Bison (yacc clone)
- Flex (lex clone)

#### Integrated Development Environments and Tools

Which IDE(s) would you like?

RHIDE, similar to Borland's IDE, including a built-in editor and debugger.

Emacs, a very powerful and complex text editor with lots of built-in functions (available for Unix and NT also).

Emacs, but without all the extra functionality that I'll probably never use (saves 4.1Mb).

Would you like gdb, the text-mode GNU debugger? You don't need it if you get RHIDE.

Yes

#### Extra Stuff

Please check off each extra thing that you want.

- "Pakke" DJGPP Installer System
- Sources for everything you download
- Extra documentation formats (texi, dvi, html, ps)

##### Toolkits

- Allegro - game graphics/sound/keyboard
- GRX - Graphics Library (points, lines, etc, includes BCCtoGRX for porting Borland Graphics programs)
- Unix Curses Emulator

[webmaster](#)

[Copyright © 2005](#)

[delorie software](#) [privacy](#)

Updated May 2005

Finally click "Tell me which files I need"

You will be presented with a list of base files for DJGPP to download plus the install instructions.

Download the files and take notice of the batch file lines to set up the paths and environment variables.

Basic paths and variables format. Be sure to use the correct path to your DJGPP directory.  
**Don't actually use this we will cover that in the IDE section.**

```
@echo off  
set PATH=c:\djgpp\bin;%PATH%  
set DJGPP=c:\djgpp\djgpp.env  
chdir c:\djgpp\mystuff (or any other directory)  
command
```

Note, because these archives contain a wide mix of programming binary and source files and they may get tagged as unsafe. This is a generic warning because of the extent of the code routines in the libraries. They are safe to download.

Although the DJGPP documentation says to place DJGPP directory in the system root, I have found it OK to place it in C:\DEVEL\DJGPP\ the same as is done in FreeDOS.

Next follow the link at the top of the page to “Index Of” of the file repository.

<http://www.delorie.com/pub/djgpp/current/>

This is where you will find the additional library files and other binary dependencies. They usually come in the form of nameS.zip or NameB.zip to denote source or binary. In most cases you will be looking for the precompiled binary of the library with the b postfix.

Choosing libraries can be a little complicated as you will also need to also know the libraries dependencies. Library dependencies are not listed in the repository so you will need to check the original developer sites, or check through forums and guides on the particular library to ascertain what collection of files to download. Some libraries may share dependencies with other libraries and only need to be installed the one time. It is worth keeping your own log/list of the installed components for future reference.

The following is the same for Windows 95 (MS-DOS 7.1) often Windows 95 offers a more convenient platform for compiling libraries for C or FreeBASIC. We can move the compiled libraries from our Windows 95 drive to the correct location in our FreeDOS drive.

FreeBASIC for DOS libraries are compiled in DJGPP as well as the FreeBASIC Compiler itself. The library binaries used in FreeBASIC are taken directly from the compatible DJGPP version that FreeBASIC is built with. If you use FreeBASIC you will likely need DJGPP to compile some libraries. many of the precompiled library files can be copied directly from the DJGPP repository into FreeBASIC. In many cases the Modified header/Include files will already exist in the FreeBASIC distribution. If you encounter a “Linker cannot find libxyz” check if the binary exists in FreeBASIC LIB directory, if not go to DJGPP repository and see if the precompiled library file is available, if not you may need to compile the library from source in DJGPP first and then copy the LIB file to FreeBASIC.

Please take note of the incompatible threads library used between DJGPP and FreeBASIC. The details are found at <https://www.freebasic.net/wiki/DevBuildDos>  
“pthread.h”

```
#include <sys/socket.h> /* for sockaddr /  
#include <sys/wtime.h> /* for struct timespec */  
#include <sys/select.h>
```

socket.h, wtime.h and select.h are not compatible with FreeBASIC and need to be commented out. This applies to compiling the FreeBASIC compiler itself and I am uncertain if it also applies to libraries compiled for FreeBASIC. I have been compiling library files for FreeBASIC with the above 3 includes commented out without failure so far.

Also note that if you use the modified pthread.h file in DJGPP for compiling FreeBASIC you will need to use the unmodified pthread.h for compiling C source code or libraries. I have two separate DJGPP installs to keep my C build and FreeBASIC build separate. One has the modified pthread.h and the other uses the standard pthread.h.

Make sure to read the “**00\_index.txt**” in each directory as it lists the zip file names and descriptions.

In the directory **/pub/djgpp/current** you will find the descriptions of each archive in the repository to aid navigation.

### 00README.TXT

The subdirectories are organized as follows:

For DJGPP Version 2.01:

|        |                                         |
|--------|-----------------------------------------|
| v2     | DJGPP Version 2.XX                      |
| v2gnu  | GNU programs built with/for djgpp V2    |
| v2tk   | Toolkits for DJGPP V2 (libraries & etc) |
| v2apps | Programs built for DJGPP V2             |
| v2misc | Other DJGPP V2 stuff                    |

The following link has a longer list of the base development files required.

[http://www.delorie.com/digpp/v2faq/faq4\\_4.html](http://www.delorie.com/digpp/v2faq/faq4_4.html)

Unpack all of the zip archives into a common directory such as DJGPP. Keep the directory structure of the internal files.

| Name     | Size       | Packed Size | Modified         | Created |
|----------|------------|-------------|------------------|---------|
| share    | 2 530 145  | 658 097     | 2021-01-16 23:23 |         |
| manifest | 5 115      | 1 537       | 2021-01-16 23:23 |         |
| lib      | 15 737     | 5 864       | 2021-01-16 23:23 |         |
| gnu      | 1 746 860  | 490 171     | 2021-01-16 23:23 |         |
| bin      | 10 682 368 | 5 127 394   | 2021-01-16 23:23 |         |

Also copy “copying.dj” and “readme.1st” to the DJGPP directory.

After unpacking you should have a directory DJGPP\\*.\* that looks like the following.

| Name        | Date modified       | Type                | Size  |
|-------------|---------------------|---------------------|-------|
| allegro     | 7/09/2007 11:36 PM  | File folder         |       |
| bin         | 20/11/2023 6:52 PM  | File folder         |       |
| contrib     | 20/11/2023 6:52 PM  | File folder         |       |
| FAQ         | 20/11/2023 6:49 PM  | File folder         |       |
| gnu         | 20/11/2023 6:51 PM  | File folder         |       |
| include     | 20/11/2023 6:51 PM  | File folder         |       |
| info        | 20/11/2023 6:49 PM  | File folder         |       |
| lib         | 20/11/2023 6:51 PM  | File folder         |       |
| libexec     | 20/11/2023 6:50 PM  | File folder         |       |
| manifest    | 20/11/2023 6:52 PM  | File folder         |       |
| share       | 20/11/2023 6:52 PM  | File folder         |       |
| tmp         | 20/11/2023 6:49 PM  | File folder         |       |
| copying     | 19/07/2015 12:50 PM | File                | 18 KB |
| copying.dj  | 19/07/2015 12:57 PM | DJ File             | 3 KB  |
| copying.lib | 19/07/2015 12:50 PM | Object File Library | 26 KB |
| djgpp.env   | 30/08/2015 3:14 PM  | ENV File            | 4 KB  |
| readme.1st  | 4/05/2015 10:09 AM  | 1ST File            | 23 KB |

Copy your .\DJGPP\\*.\* environment to C:\DEVEL\DJGPP\\*.\* in your Windows 95 install.

Note when copying library files and additional dependencies follow the above method and merge the unpacked directories into the root of .\DJGPP in your DOS install.

This is similar to the previous installs from FreeDOS except for the additional 3 libraries etc.

### Setting up DJGPP IDEs

All of the batch files used in Windows 95 are much the same as for FreeDOS with the following changes.

The addition of the default COMMAND and BIN paths for Windows 95 as well as the CTMOUSE when in DOS-Mode.

```
REM SET PATH=%path%;C:\WINDOWS;C:\WINDOWS\COMMAND;C:\DOSLINKS
REM CTMOUSE /R55
And you need to change any instance of %dosdrv% to C:\%
%dosdrv%\DEVEL
to
C:\DEVEL
```

DJGPP comes with the default RHIDE if selected and can be found in the .\BIN\rhide.exe directory.

Before running RHIDE or using the compiler from the command line we must create a batch file to set the environment path and variables.

I will also use FED (Folding Editor) as the primary IDE. Note that I am using FED V2.2.4b in the development directory C:\DEVEL\DJGPP\FED224B\\*.\*

NOTE: I will explain the FED install in the section after RHIDE.

If you already have FED installed as part of the Windows 95 install you can use that but I would recommend a separate copy of FED in each development environment. If you encounter a name conflict with FED.BAT in your .\DOSLINKS directory then alter the name for each alternative FED.BAT that you use. For example for DJGPP rename the batch file to DJFED.BAT. Take note of the path and file names and adjust to suite your DOS environment.

Create the following 3 batch files to open a command.com instance to DJGPP, and RHIDE instance or a FED instance. Place all 3 into the .\DOSLINKS directory. It is also acceptable to place them directly in the .\DJGPP directory but you will have to navigate to that directory to launch the batch files.

### DJGPP.BAT (Command console Development instance)

```
@ECHO OFF
REM Use DJGPP from the command line.
REM You can place this file in the root directory of DJGPP or
REM in the C:\DOSLINKS directory (preferred).
REM Set the DOS path environments
SET PATH=%path%;C:\WINDOWS;C:\WINDOWS\COMMAND;C:\DOSLINKS
REM For DOS-MODE only.
REM CTMOUSE /R55
CLS
REM Make sure the correct path to your install is used.
set PATH=%path%;C:\DEVEL\DJGPP\BIN
set DJGPP=C:\DEVEL\DJGPP\DJGPP.ENV

REM CD Working directory of project or source code.
CD \DEVEL\DJGPP

REM Open a new instance of the command console using the temporary variable and
PATH
call COMMAND.COM
REM You can now run command line arguments against gcc

REM Clear the console after the child app closes
CLS
```

You can do a quick test of the functionality of the above batch file by running DJGPP.BAT and entering the following to the child command line instance.

```
GCC --version
```

```
GCC -v
```

If the GCC.exe is in the path then the version information will be displayed. If not, recheck the batch file and paths.

### RHIDE.BAT (RH IDE Development instance)

```
@ECHO OFF
REM RHIDE as the IDE for DJGPP.
REM You can place this file in the root directory of DJGPP or
REM in the C:\FREEDOS\LINKS directory preferred.
REM Set the DOS path environments
SET PATH=%path%;C:\WINDOWS;C:\WINDOWS\COMMAND;C:\DOSLINKS
REM For DOS-MODE only.
REM CTMOUSE /R55
CLS
REM Make sure the correct path to your install is used.
set PATH=%path%;C:\DEVEL\DJGPP\BIN
set DJGPP=C:\DEVEL\DJGPP\DJGPP.ENV

REM CD Working directory of project or source code.
CD \DEVEL\DJGPP

REM Open the RH IDE
call C:\DEVEL\DJGPP\BIN\RHIDE.EXE

REM Clear the console after the child app closes
CLS
```

### DJFED.BAT (FED Development instance)

```
@ECHO OFF
REM FED as the IDE for DJGPP.
REM You can place this file in the root directory of DJGPP or
REM in the C:\FREEDOS\LINKS directory preferred.
REM Set the DOS path environments
SET PATH=%path%;C:\WINDOWS;C:\WINDOWS\COMMAND;C:\DOSLINKS
REM For DOS-MODE only.
REM CTMOUSE /R55
CLS
REM Make sure the correct path to your install is used.
set PATH=%path%;C:\DEVEL\DJGPP;C:\DEVEL\DJGPP\BIN
set DJGPP=C:\DEVEL\DJGPP\DJGPP.ENV

REM CD Working directory of project or source code.
CD \DEVEL\DJGPP

REM Open the FED IDE
call C:\DEVEL\DJGPP\FED224B\FED.exe

REM Clear the console after the child app closes
CLS
```

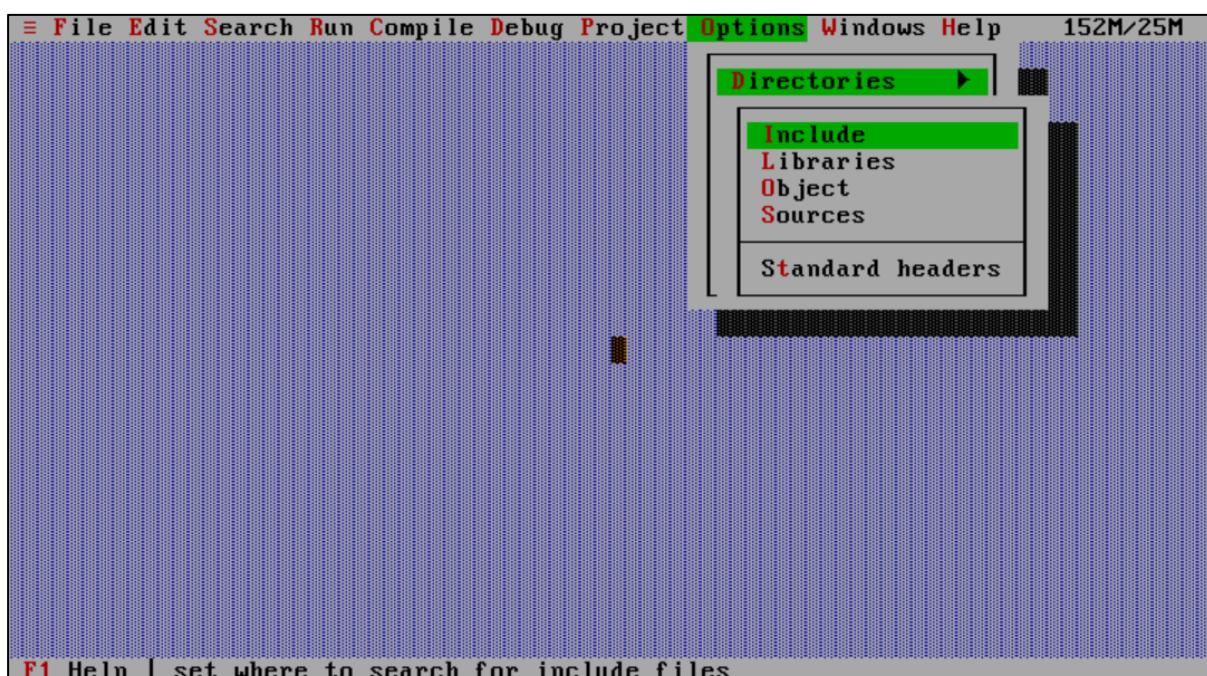
You can create a PIF shortcut to these 3 batch files and place them on your desktop, or in the ALaunch directories under \DJGPP.

Adjust your font size in the PIF “Properties” for readability and add a larger 4096 Initial Environment for variables.

## RHIDE

Launch RHIDE from the bat file.

Using “ALT + O” open the Option menu and select “Directories”.



You will need to add the correct full paths to Include, Libraries and Objects. Source is optional and relates to a code project settings.

C:\DEVEL\DEVEL\DJGPP\INCLUDE

C:\DEVEL\DEVEL\DJGPP\LIB

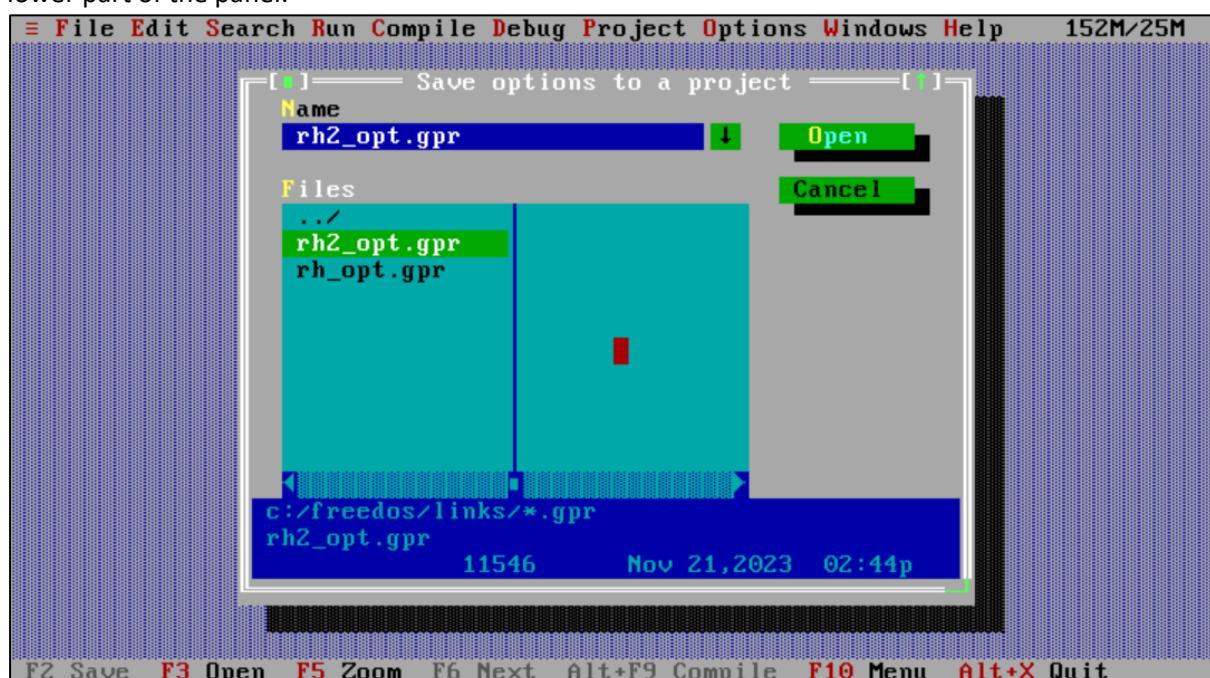
C:\DEVEL\DEVEL\DJGPP\LIB also for Objects but can be left blank at the start.

Objects are precompiled code with the .o or .a extension and will often exist in .\LIB  
The compiler will always look in the project source directory for Includes, Libs and Objects before checking the paths above.

Remember that the DJGPP already has the environment paths set in DJGPP.ENV



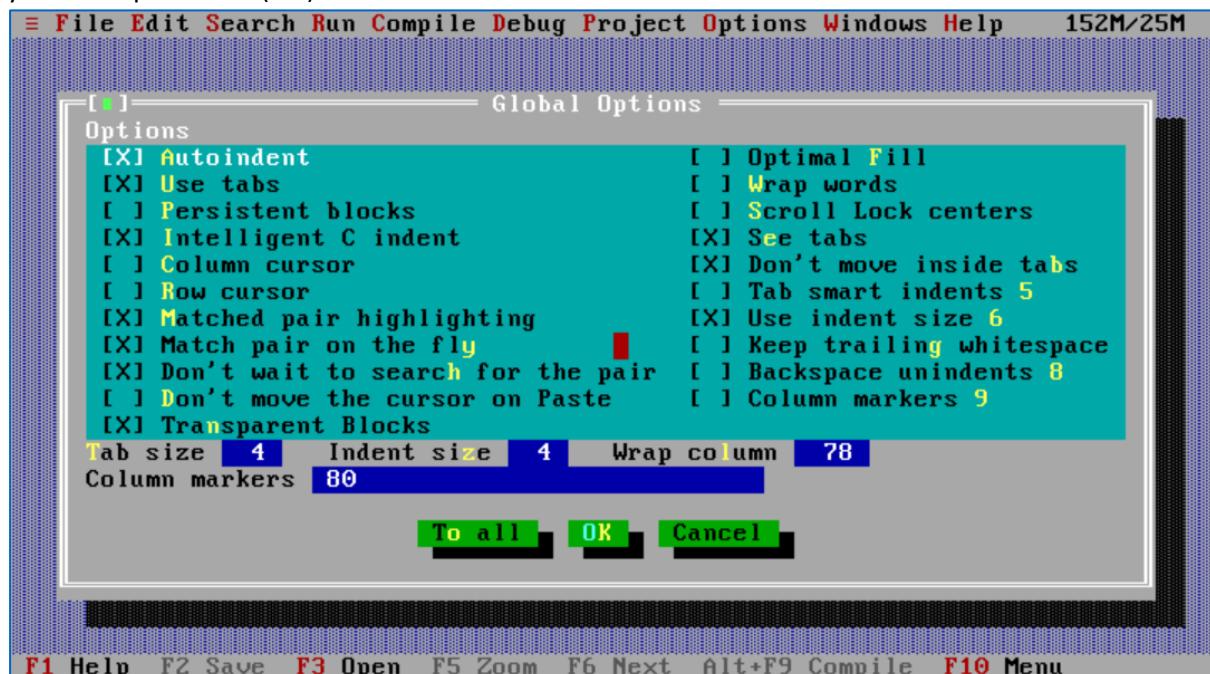
After setting your path or any other options I suggest saving a backup of the RHIDE configs with “Save Options”. Take care with the path in the file name. The current directory path is listed in the lower part of the panel.



<https://www.emse.fr/~boissier/enseignement/sdao/Rhide.htm>

Under “Options -> Compilers” you can set the specific compiler switches for your project such as warning levels debugging and optimisation for release code. This is project specific so use the GCC compiler switches required for the individual project.

I would also suggest checking the editor options under “Options -> Environment -> Editor” and set your TAB spaces to 4 (std) etc.



Create a project directory to test the IDE with a “Hello world!”. I will use  
C:\DEVEL\DJGPP\PROJECT\hello.c

Note that RHIDE uses Unix paths with forward slash / instead of the DOS backslash \  
So inside the editor file dialog you will use C:/DEVEL/DJGPP/PROJECT/hello.c  
This can be a point of confusion when working with GNU Unix tool sets.

From RHIDE create a “New” file and save it to the project directory.

Add your “Hello world” code and then save.

## A Beginners Guide To DOS Programming

The screenshot shows a DOS-based IDE interface. The top menu bar includes File, Edit, Search, Run, Compile, Debug, Project, Options, Windows, and Help. The status bar at the bottom indicates memory usage: 152M/25M. The main code editor pane displays the following C code:

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char **argv)
{
    system("CLS");
    printf("Hello world!\n");
    system("PAUSE");
    return 0;
}
```

The message window at the bottom is titled "Message Window". It displays the output of the compilation process:

```
= Message Window =
Compiling: hello.c
no errors
```

The keyboard shortcut bar at the bottom includes F2 Save, F3 Open, F5 Zoom, F6 Next, Alt+F9 Compile, F10 Menu, and Alt+X Quit.

Next select “Compile -> Compile”. The message window will appear in the lower pane.

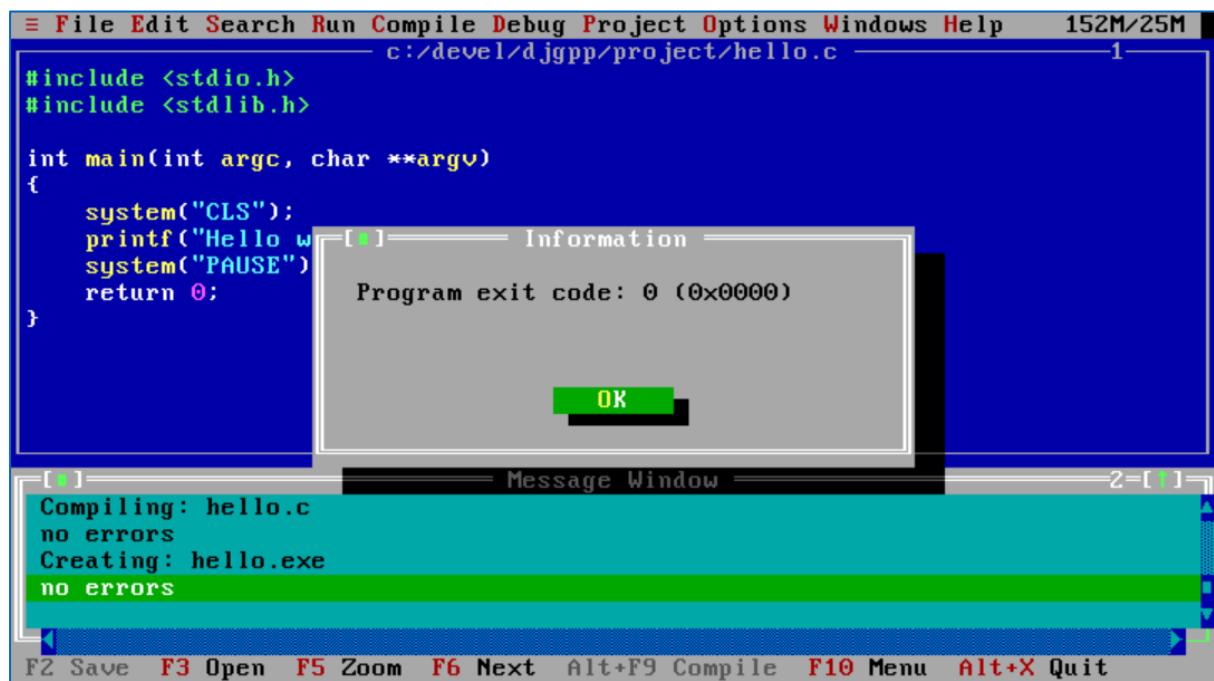
The screenshot shows the same DOS-based IDE interface after selecting "Compile -> Compile". The message window now displays the compilation results:

```
= Message Window =
Compiling: hello.c
no errors
```

The keyboard shortcut bar at the bottom includes Enter, Jump to source, Alt+F9 Compile, Del Delete, Ctrl+Del Delete all.

Next select “Run -> Run” and you should see the Hello world application in the console. Note: Run will also compile the source code.

```
Hello world!  
Press any key to continue . . .
```



Note that RHIDE defaults to outputting the hello.EXE executable to the DJGPP root directory. You will need to add additional output commands to the compiler linker for an alternative output directory.

That's it you have set up the basic to use RHIDE with DJGPP.

### FED

Fed is a little more simplistic as an IDE compared to RHIDE, but comes with the advantage of being able to customise your tool sets. That means that there is a little more effort in the setup phase of

using FED as you will need to set up the quick tools menu as well as create a few convenience batch files to compile the source and run the application. It also allows you to use make files with convenience if that is your preference.

I am using a copy of FED version 2.24 from BTTR software “fed224b.zip”. <http://www.bttr-software.de/products/fed/>

Being open source there are a number of similar variants. V 2.24b is usually aligned with the version used in the FreeDOS repository.

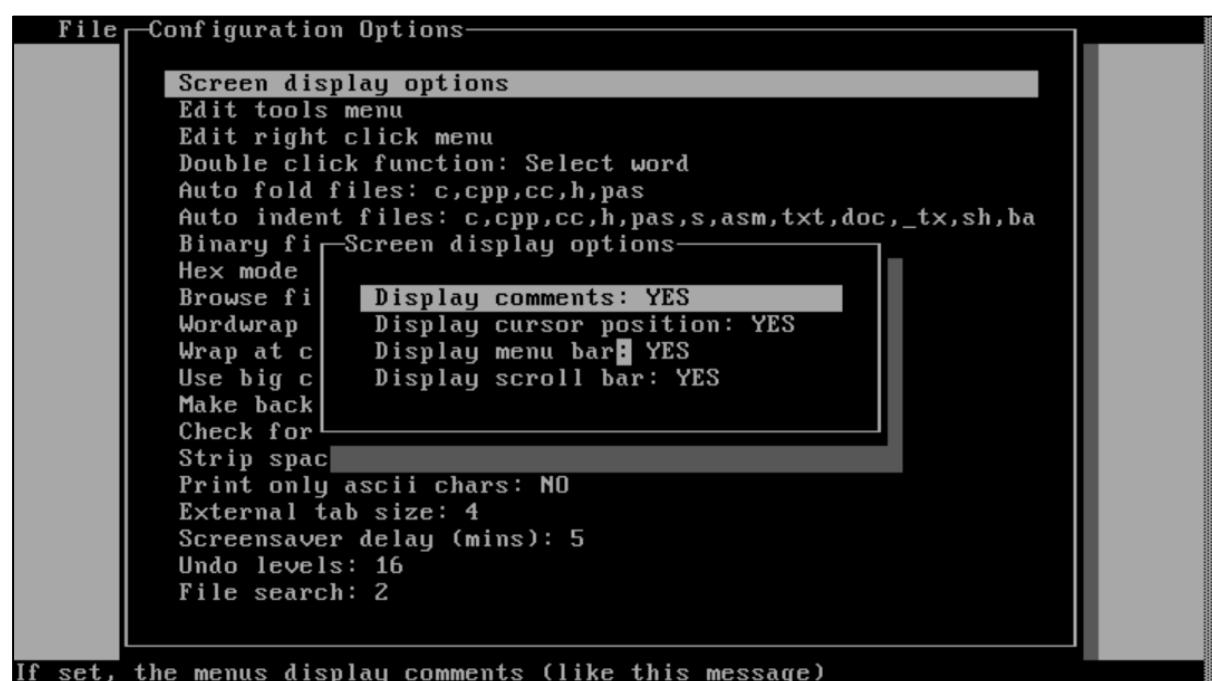
You can use the FED version that exists in the .\[DOS]APPS directory in FreeDOS if you wish but I prefer to use a separate copy within the Development directory.

Copy C:\[DOS]APPS\FED to C:\DEVEL\DJGPP\FED224 or from the downloaded archive above if you haven't already done so. The directory name FED224 correlates with the batch files created earlier and allows us to discern between the FED in .\[DOS]APPS and the copy in our dev environments. If you have multiple environments using FED you may have to alter the names of the batch files in .\[DOS]LINKS to correspond.

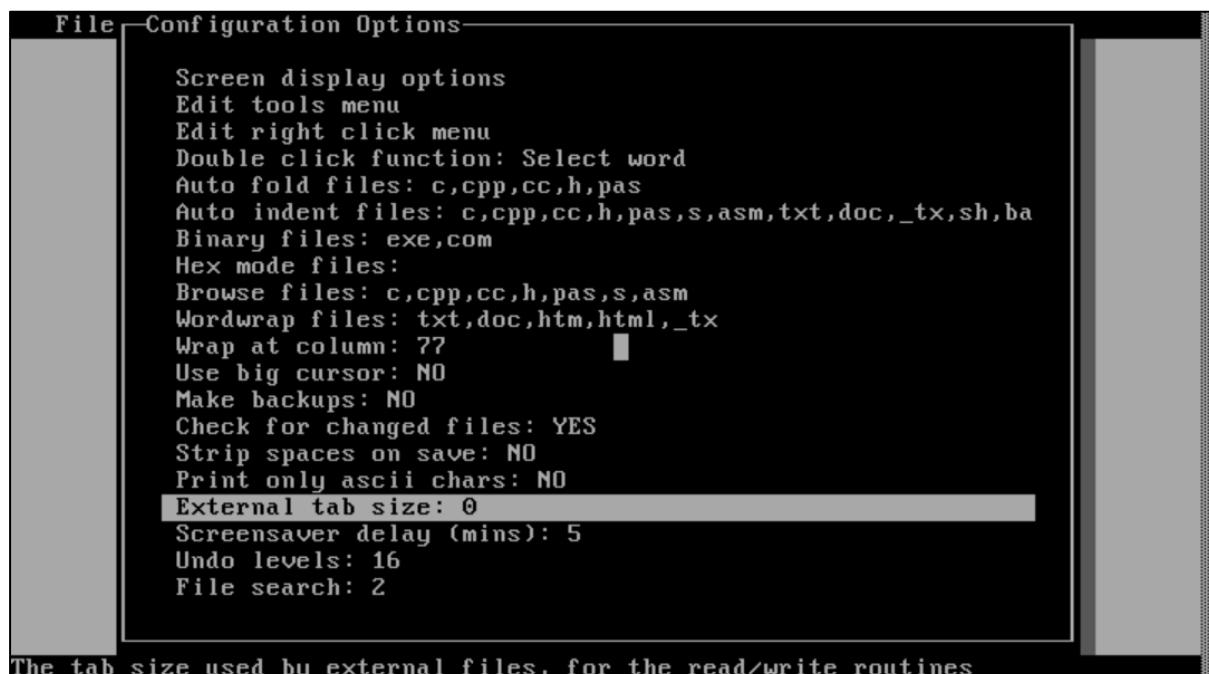
Open FED224 using the batch file in .\[DOS]LINKS or from the command line with DJFED.BAT or create a Windows 95 desktop shortcut (PIF) to the batch file.

You will need to hold down the ALT key to show the menu if you have not changed the default settings.

Select Alt + C to open the Config menu and select Options. You will have a long list of different configuration options for the way the editor behaves. I suggest changing the “Screen display option” to something suitable. I typically change all to YES to add in the TUI navigation.



Next turn off (NO) Strip spaces on save, and set External tab size to 0. The external tab size other than zero will alter the indents when opening other source documents.



The tab size used by external files, for the read/write routines

You can spend time trying other options if you wish, but these will get you at a point where you can start coding.

Note that the option "Auto fold files: c, cpp etc. will open the file with all blocks folded for these file extensions (Alt + Misc). If you prefer to open the source code with all lines fully expanded, remove the file extensions from the config.

Next in Options -> Config change the "Tab size" to 4(std).



Alter the tab size

Don't forget to "Save Config" before exiting FED.

To set up FED to compile and run source code we will need to add some custom Tools entries as well as creating a couple of batch files. There are many different modifications to the way this can be done including additional batch file for different purposes, but I am just going to show the Compile and Run batch files to keep it simple.

You will need an extra command line application names "STRINGS.COM" for one of the batch file tasks.

Go to the following link and download "strings.zip" or "string25.zip" from the section "STRINGS — Enhanced SET."

<https://www.btrr-software.de/freesoftware/batch1.htm>

DOS and FreeDOS does not have a built in string manipulation library as is found in later version of Windows command console CMD.EXE so we need to use an external command line tool for this.

I will also use the STRINGS.COM method in Windows 95.

STRINGS.COM can be difficult to find on the internet so I will place a copy of this with the source code for this guide.

<http://www.pl.exim.org/packages/coast/msdos/batutil/> "string25.zip"

<http://www.manmrk.net/tutorials/batch/index.htm> "string25.zip"

<http://www.lanet.lv/simtel.net/msdos/batchutl-pre.html> "string25.zip"

You will need to have STRINGS.COM placed either in the system path C:\WINDOWS\COMMAND\STRINGS.COM or in the directory with the batch files for running commands on the DJ GCC compiler.

The following 2 batch files are created to work in the current working directory or system path. If you wish to use them from an alternative location you will need to adjust some of the path parameters for both the COMP and RUN batch files as well as the call to the batch files from FED.

The 2 batch files perform similar tasks. They take the source code file name from FED %f (including the fully qualified path), truncate the file extension 'C' and add 'EXE' in its place. HELLO.C becomes HELLO.EXE

If an optional file name is input at the FED tools menu then the source code file name is replaced with the optional file name.

Please note: If using these 2 batch files for source that has a different file extension be sure to change the token lengths to match \$LEFT. For example .C is 1 and .CPP will require a value of 3. I should get rid of the magic number and use a configurable variable at the top of the page for readability.

In COMP.BAT both the source code path and file name as well as the modified path and output file name are used at line 73 to invoke the compiler. You can alter the compiler and linker arguments here to create alternative versions of the batch file; for example debug compile, or release compile.

call GCC -v -Wall %cfile% -o %xfile% (GCC.exe verbose, Warn ALL, source file, compile to object, output to exe name).

In the RUN.BAT this line is used to run the compiled exe using the modified file name sent from FED.

Create or copy the following 2 batch files to the DJGPP root; C:\DEVEL\DJGPP\\*.\*

[Test the compile and run batch files on the new Windows 95 VM]

### COMP.BAT

```
@echo OFF
REM This is a generic command line runner for FED and DJGPP.
REM With small modification the compiler command line call can be used with
REM FED for other compilers or languages.
REM Change "GCC.EXE -v -Wall %cfile% -o %xfile%" switches to suite your
REM project requirements.
REM For Script engines you will need to use:
REM Interpreter.exe/com -[switches] %cfile%
REM (Other compilers) Change str=EXE to EXE/COM etc.
REM For Script engines you will need to
REM SEE: Accompanying RUN.BAT for FED
REM Requires STRINGS.COM (Version 2.5) Copyright (c) 1991, 1992 Douglas Boling
REM ftp.sunet.se/pub/simtelnet/msdos/batchutl/string25.zip
set cfile=%1
set efile=%2
REM If %2 (%p) is empty, use source file name.
IF [%2]==[] goto :SOURCENAME
REM Else use custom file name from %2 (%p).
IF NOT [%2]==[] goto :CUSTNAME

goto :ERROR

:SOURCENAME
REM Use %f path\source.ext and change to path\%1.exe
STRINGS plen =LENGTH %cfile%
REM .BAS = 3, .C = 1
STRINGS plen =SUB %plen%, 1
STRINGS xfile =LEFT %cfile%, %plen%

REM Add (concatenate) EXE to the path\Name._____
set str=EXE
set xfile=%xfile%%str%

REM GCC parameters
REM call GCC.EXE -v -Wall %cfile% -o %xfile%
call GCC -v -Wall %cfile% -o %xfile%
goto :END

:CUSTNAME
REM Build our custom path\filename.EXE/COM
REM ECHO Use %f path\source.ext and change to path\%2
REM Token index number starts at 1
Set index=1
REM Set delimiter "\"
set character=\

:LOOP
```

## A Beginners Guide To DOS Programming

```
REM Loop through each token with "\" delimiter
STRINGS token = PARSE %cfile%, %index%, %character%

REM Look ahead to see if the next token is empty string
STRINGS lkahdidx = ADD %index%, 1
STRINGS lkahead = PARSE %cfile%, %lkahdidx%, %character%
IF [%lkahead%] == [] goto :FINISH

set xfile=%xfile%%token%%character%
REM Increment the index number
STRINGS index = ADD %index%, 1

REM Safety stop catch endless loop on error.
IF %index% == 10 goto :ERROR

goto :LOOP

:FINISH
REM Add file name.exe/com
REM Build the path\filename.EXE
set xfile=%xfile%%efile%

REM Call FBC with custom file name.EXE/COM
REM call FBC.EXE -v -Wall %cfile% -o %xfile%

call GCC -v -Wall %cfile% -o %xfile%
goto :END

:ERROR
ECHO Unknown Error!
goto :END

:END
REM Release the temp variables
set cfile=
set efile=
set plen=
set str=
set xfile=

REM You could place a PAUSE here followed by a CLS to keep the console
REM window open to view the output of the compiler.
REM FED already adds a pause after invoking the command line.
REM PAUSE
REM CLS
```

### RUN.BAT

```
@echo OFF
REM This is a generic command line runner for FED and DJGPP.
REM With small modification the compiler command line call can be used with
REM FED for other compilers or languages.
REM (Other compilers) Change str=EXE to EXE/COM etc.
REM SEE: Accompanying COMP.BAT for FED
REM Requires STRINGS.COM (Version 2.5) Copyright (c) 1991, 1992 Douglas Boling
REM ftp.sunet.se/pub/simtelnet/msdos/batchut/string25.zip
set cfile=%1
set efile=%2
```

```
REM If %2 (%p) is empty, use source file name.  
IF [%2]==[] goto :SOURCENAME  
REM Else use custom file name from %2 (%p).  
IF NOT [%2]==[] goto :CUSTNAME  
  
goto :ERROR  
  
:SOURCENAME  
REM Use %f path\source.ext and change to path\%1.exe  
STRINGS plen =LENGTH %cfile%  
REM .BAS = 3, .C = 1  
STRINGS plen =SUB %plen%, 1  
STRINGS xfile =LEFT %cfile%, %plen%  
  
REM Add (concatenate) EXE/COM to the path\Name._____  
set str=EXE  
set xfile=%xfile%%str%  
  
REM Run the executable  
call %xfile%  
goto :END  
  
:CUSTNAME  
REM Build our custom path\filename.EXE/COM  
REM ECHO Use %f path\source.ext and change to path\%2  
REM Token index number starts at 1  
Set index=1  
REM Set delimiter "\."  
set character=\  
  
:LOOP  
REM Loop through each token with "\" delimiter  
STRINGS token = PARSE %cfile%, %index%, %character%  
  
REM Look ahead to see if the next token is empty string  
STRINGS lkahdidx = ADD %index%, 1  
STRINGS lkahead = PARSE %cfile%, %lkahdidx%, %character%  
IF [%lkahead%] == [] goto :FINISH  
  
set xfile=%xfile%%token%  
REM Increment the index number  
STRINGS index = ADD %index%, 1  
  
REM Safety stop catch endless loop on error.  
IF %index% == 10 goto :ERROR  
  
goto :LOOP  
  
:FINISH  
REM Add file name.exe/com  
REM Build the path\filename.EXE  
set xfile=%xfile%%efile%  
  
REM Call the custom EXE NAME  
call %xfile%  
goto :END  
  
:ERROR  
ECHO Unknown Error!
```

```
goto :END

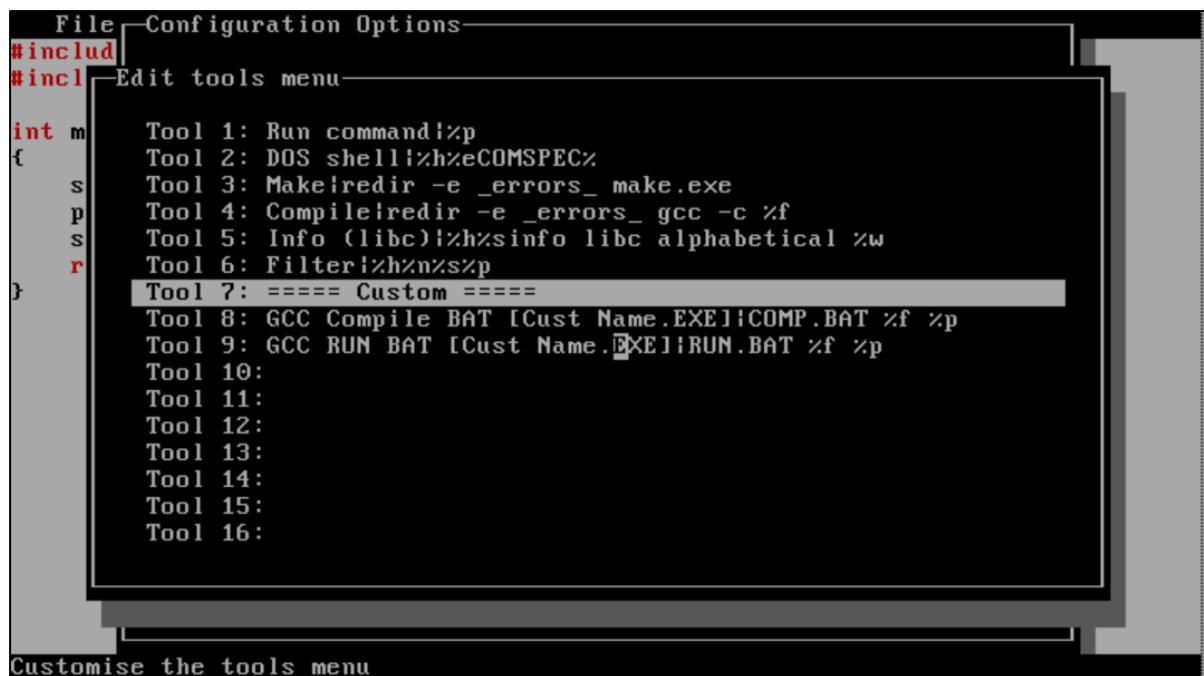
:END
REM Release the temp variables
set cfile=
set efile=
set plen=
set str=
set xfile=

REM You could place a PAUSE here followed by a CLS to keep the console
REM window open to view the output of the compiler.
REM FED already adds a pause after invoking the command line.
REM PAUSE
REM CLS
```

Once the 2 batch files are in place, open FED and add the following to the tools menu so we can call the 2 batch files from FED.



Open from the menu “Config ->Options -> Edit tools menu” and add the following 2 lines.



Customise the tools menu

Tool6===== Custom =====

Tool7=GCC Compile BAT [Cust Name.EXE] | COMP.BAT %f %p

Tool8=GCC RUN BAT [Cust Name.EXE] | RUN.BAT %f %p

The first part before ‘|’ is the tool description. The section after the ‘|’ is the command line argument **COMP.BAT SourceName UserInput** where f% is the internal FED variable containing the full qualified path and source file name in the editor window and p% is a variable to take a custom output name from the user. If p% is left blank the batch file routines ignore the empty p% variable and use the f%. If p% is given a value then the batch file will substitute the custom name in place of the original source name and the file name.ext component of f% is ignored.

To remove a line, just leave it blank.

Hint you can copy the FED.CFG to other instances of FED to keep your FED customizations.

The FED website also makes available some alternative pre made syntax highlighting schemes FED.SYN. It is a good idea to keep a backup of your FED configs in case of mistakes or for recovering your settings.

Open FED for your DJGPP environment and create a “Hello World!” HELLO.C source file.

The screenshot shows a DOS terminal window with a menu bar at the top. The menu bar includes File, Edit, Search, Misc, Tools, Config, and Help. Below the menu bar is a command line interface. The command line displays the path "c:\develop\djgpp\project\hello.c" and shows three items in a dropdown menu: "c:\develop\djgpp\", "c:\develop\djgpp\project\hello.c", and "c:\develop\djgpp\project\hello.exe". The bottom of the screen has a status bar with the text "Open a file, or switch to it if it is already open".

```
File      Edit      Search      Misc      Tools      Config      Help
Open (ctrl+enter for binary mode)
c:\develop\djgpp\project\hello.c
c:\develop\djgpp\project\hello.c
c:\develop\djgpp\project\hello.exe
Open a file, or switch to it if it is already open
```

The screenshot also shows a code editor window below the terminal. The code editor has its own menu bar and displays the following C code:

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char **argv)
{
    system("CLS");
    printf("Hello world!\n");
    system("PAUSE");
    return 0;
}
```

The status bar at the bottom of the code editor window shows the path "- a-d c:\develop\djgpp\project\hello.c - line 1 - col 1 - 0x23 (35)".

From the menu open “Tools -> ... ” and select your menu entry to COMPILE the source code followed by you menu entry to RUN the source code. I would also test both menu entries using the custom name for the output file. You will find that if you compile with a custom name the variable p% will be auto filled when using the Run option.

This screenshot shows a DOS application window with a menu bar and a code editor area. The menu bar includes File, Edit, Search, Misc, Tools, Config, and Help. A context menu is open over some code, showing options like Run command, DOS shell, Make, Compile, Info (libc), Filter, and a section for Custom tools. The code in the editor is a standard Hello World program.

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char **argv)
{
    system("CLS");
    printf("Hello world!\n");
    system("PAUSE");
    return 0;
}
```

Run external tool

This screenshot shows a DOS application window with a menu bar and a code editor area. The menu bar includes File, Edit, Search, Misc, Tools, Config, and Help. A command line window is open at the bottom, showing the text "GCC Compile BAT [Cust Name.EXE]". The code in the editor is a standard Hello World program.

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char **argv)
{
    system("CLS");
    printf("Hello world!\n");
    system("PAUSE");
    return 0;
}
```

Run external tool

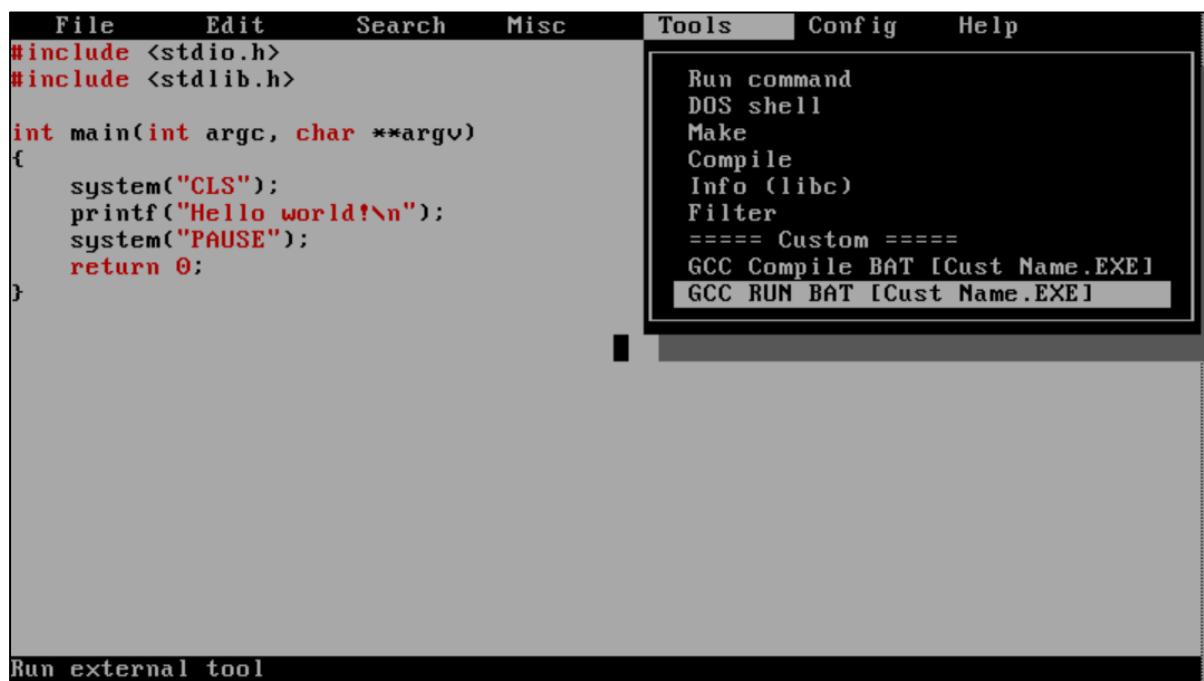
```

COLLECT_GCC_OPTIONS=' -v' '-Wall' '-o' 'c:\devel\djgpp\project\hello.EXE' '-mtune
=pentium' '-march=pentium'
c:/devel/djgpp/bin/as.exe -v -o c:/devel/djgpp/tmp/ccGLFMZF.o c:/devel/djgpp/tm
p/ccUdGWXC.s
GNU assembler version 2.21.1 (djgpp) using BFD version (GNU Binutils) 2.21.1
COMPILER_PATH=c:/devel/djgpp/bin/../../libexec/gcc/djgpp/4.71/:c:/devel/djgpp/bin/..
./libexec/gcc/:c:/devel/djgpp/bin/
LIBRARY_PATH=c:/devel/djgpp/bin/../../lib/gcc/djgpp/4.71/:c:/devel/djgpp/bin/..../lib
/gcc/:c:/devel/djgpp/lib/:c:/devel/djgpp/lib/:c:/devel/djgpp/bin/..../lib/gcc/djgp
p/4.71/.....
COLLECT_GCC_OPTIONS=' -v' '-Wall' '-o' 'c:\devel\djgpp\project\hello.EXE' '-mtune
=pentium' '-march=pentium'
c:/devel/djgpp/bin/..../libexec/gcc/djgpp/4.71/collect2.exe -o c:/devel/djgpp\pro
ject\hello.EXE c:/devel/djgpp/lib/crt0.o -Lc:/devel/djgpp/bin/..../lib/gcc/djgpp/4
.71 -Lc:/devel/djgpp/bin/..../lib/gcc -Lc:/devel/djgpp/lib -Lc:/devel/djgpp/lib -L
c:/devel/djgpp/bin/..../lib/gcc/djgpp/4.71/..... c:/devel/djgpp/tmp/ccGLFMZF.o
-lgcc -lc -lgcc
COLLECT_GCC_OPTIONS=' -v' '-Wall' '-o' 'c:\devel\djgpp\project\hello.EXE' '-mtune
=pentium' '-march=pentium'
c:/devel/djgpp/bin/stubify.exe -v c:/devel/djgpp\project\hello.EXE
stubify for djgpp V2.X executables, Copyright (C) 1995 DJ Delorie
stubify: c:/devel/djgpp\project\hello.EXE -> c:/devel/djgpp\project\hello.000 ->
c:/devel/djgpp\project\hello.exe

<press a key>_

```

Now select run the output executable.



The screenshot shows a DOS-based development environment. At the top is a menu bar with File, Edit, Search, Misc, Tools, Config, and Help. Below the menu is a code editor window containing a C program:

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char **argv)
{
    system("CLS");
    printf("Hello world!\n");
    system("PAUSE");
    return 0;
}
```

Below the code editor is a terminal window titled "GCC RUN BAT [Cust Name.EXE]". It displays the output of the program:

```
Hello world!
Press any key to continue . . .
```

Using the alternative output name (Make sure to add the extension “world.EXE”)...

```
File      Edit      Search      Misc      Tools      Config      Help
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char **argv)
{
    system("CLS");
    printf("Hello world!\n");
    system("PAUSE");
    return 0;
}

GCC Compile BAT [Cust Name.EXE]-
world.exe

Run external tool

COLLECT_GCC_OPTIONS=' -v' '-Wall' '-o' 'c:\devel\djgpp\project\world' '-mtune=pentium' '-march=pentium'
c:/devel/djgpp/bin/as.exe -v -o c:/devel/djgpp/tmp/cch54k8k.o c:/devel/djgpp/tmp/ccfTyrn4.s
GNU assembler version 2.21.1 (djgpp) using BFD version (GNU Binutils) 2.21.1
COMPILER_PATH=c:/devel/djgpp/bin/../../libexec/gcc/djgpp/4.71/:c:/devel/djgpp/bin/../../libexec/gcc/:c:/devel/djgpp/bin/
LIBRARY_PATH=c:/devel/djgpp/bin/../../lib/gcc/djgpp/4.71/:c:/devel/djgpp/bin/../../lib/gcc/:c:/devel/djgpp/lib/:c:/devel/djgpp/lib/:c:/devel/djgpp/bin/../../lib/gcc/djgpp/4.71/../../..
COLLECT_GCC_OPTIONS=' -v' '-Wall' '-o' 'c:\devel\djgpp\project\world' '-mtune=pentium' '-march=pentium'
c:/devel/djgpp/bin/../../libexec/gcc/djgpp/4.71/collect2.exe -o c:/devel/djgpp\project\world c:/devel/djgpp/lib/crt0.o -Lc:/devel/djgpp/bin/../../lib/gcc/djgpp/4.71 -Lc:/devel/djgpp/bin/../../lib/gcc -Lc:/devel/djgpp/lib -Lc:/devel/djgpp/lib -Lc:/devel/djgpp/bin/../../lib/gcc/djgpp/4.71/.... c:/devel/djgpp/tmp/cch54k8k.o -lgcc -lc -lgcc
COLLECT_GCC_OPTIONS=' -v' '-Wall' '-o' 'c:\devel\djgpp\project\world' '-mtune=pentium' '-march=pentium'
c:/devel/djgpp/bin/stubify.exe -v c:/devel/djgpp\project\world
stubify for djgpp V2.X executables, Copyright (C) 1995 DJ Delorie
stubify: c:/devel/djgpp\project\world -> c:/devel/djgpp\project\world.000 -> c:/devel\djgpp\project\world.exe

<press a key>
```

When we "run" the compiled exe the custom name already exists...

The screenshot shows a DOS-based development environment. At the top is a menu bar with File, Edit, Search, Misc, Tools, Config, and Help. Below the menu is a code editor window containing C code for a 'Hello World' application. The code includes #include directives for stdio.h and stdlib.h, a main function that clears the screen, prints 'Hello world!', and pauses before exiting. In the bottom right corner of the code editor, there is a status bar with the text 'GCC RUN BAT [Cust Name.EXE]'. A terminal window is open below the code editor, showing the output of the program: 'Hello world!' followed by a prompt 'Press any key to continue . . .'. The entire interface is set against a dark background.

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char **argv)
{
    system("CLS");
    printf("Hello world!\n");
    system("PAUSE");
    return 0;
}
```

Run external tool

```
Hello world!
Press any key to continue . . .
```

If you had any problems with compiling check though the GCC output for hints and recheck the 2 batch files for possible syntax errors using debug prints.

```
ECHO Debug Pause
ECHO %1
ECHO %cfile%
ECHO %efile%
REM etc.
PAUSE
```

There are many valid ways to set up your development environment. The above is just one method to get you started creating with DJGPP using FED as your IDE. Take care on large projects as you may need to invoke more memory in your FDCONFIG.SYS or FDAUTO.BAT file. Take note of !BUFFERS=, !FILES=, !STACKS=, !FCBS= etc. NOTE: The ‘!’ means that it is “forced or mandatory” to be loaded.

When including libraries remember that DJGPP keeps its own DJGPP.ENV file for library paths, but you may still have to add additional paths and linker options in the batch file when using specific libraries. Follow the instructions from the library as well as the manual for GCC.

It is worthwhile creating the DJGPP subdirectories for ALaunch and copying the shortcuts (PIF) there for your menu.

### Using Libraries

The DJGPP contains many additional libraries and there are many more in 3<sup>rd</sup> part repositories.

You will find some common source and pre-compiled LIB files at:

<https://www.delorie.com/pub/digpp/current/v2tk/>

For example if you go to the DJGPP repository and look under Tool Kits we can find the Public domain curses library pdcur39a.zip. ‘a’ stands for library “Archive” file or just LIB file.

Be aware of the 8.3 file naming convention as many libraries are created for cross compiling under newer operating systems where long file names are not an issue. For example “libpd़curses.a” will be truncated to “libpd़c~1.a”. You may need to change some of the Linker names to reflect this such as lpdc~1 in the compiler setting or alternatively shorten the library file name and #include “name.h” to match.

Be aware of this 8.3 file naming when bringing any library into the DOS environment.

---

### Open Watcom

I may consider including the Open Watcom compiler in future revisions.

---

### Turbo C (1.0, 1.5, 2.0)

Note that I am specifically referring to Turbo C, not C++.

Turbo C is an Integrated Development Environment and compiler for the C programming language from Borland. First introduced in 1987, it was noted for its integrated development environment, small size, fast compile speed, comprehensive manuals and low price. After 2.0 it this product was

replaced with Borland Turbo C++. It competed against, Microsoft Quick C and was generally considered faster and more successful.

Embarcadero, formerly Borland has altered the licence conditions for Turbo C as a free download to allow for educational and historical use in 2006. Unfortunately the links can be difficult to find. The following is one of the museum software site that keep old versions, but I can't seem to find the updated licence file.

<http://cc.embarcadero.com/Default.aspx>

### Embarcadero TurboC licence

ID: 25636, Turbo C 2.01

<http://cc.embarcadero.com/item/25636>

These historical files are provided to the Borland community free of charge. They may be downloaded and used "as is" for personal use only. No developer support is provided. Each individual product contains copyright notices that are still in force. These files may not be made available via the Internet or any hard copy media (e.g. diskette, CDROM). We make no claims about Year 2000 compatibility for our antique software. If you have technical questions, you should ask the questions on our Internet newsgroups (there may be someone who remembers these old tools).

ID: 26014, Turbo C++ 1.01

<http://cc.embarcadero.com/item/26014>

These historical files are provided to the Borland community free of charge. They may be downloaded and used "as is" for personal use only. No developer support is provided. Each individual product contains copyright notices that are still in force. These files may not be made available via the Internet or any hard copy media (e.g. diskette, CDROM). We make no claims about Year 2000 compatibility for our antique software. If you have technical questions, you should ask the questions on the threaded conversions for this antique software or ask our Internet newsgroups (there may be someone who remembers these old tools).

Alternatively you can download the 3 editions from WinWorld.

<https://winworldpc.com/product/borland-turbo-c/1x>

I would not recommend using Turbo C for production use as it is old and not C standards compliant. Please see the open source compiler sets such as GCC (DJGPP), Open Watcom and some of the others that are currently maintained.

The only reason that I am including the Turbo C compilers is that they had a significant following in the day and many of the available libraries and tutorials were centred around turbo C. This can make it easier to test some libraries in the original context before obtaining or modifying libraries for use in DJGPP etc.

Turbo C V1.0 and v1.5 produce native "Real Mode" x286 code, unlike many newer DOS compilers that are 32-bit protected mode. This allows us to experiment with real mode286 source in a native environment before attempting to step through the nuances of setting up the DJ IA16GNU compiler

and altering old source code to work in newer compilers. Turbo C will compile and run very small, tight real mode 286 “Out of the box” so to speak.

I often resource old turbo C source code archives and examples. I can test them in Turbo C to make sure they are working, and then go about transcribing the working code to a 32-bit context such as DJGPP. One example of this is the well-known Borland Graphics Interface BGI that was distributed with Turbo C V2.0. Many variants of the library have been recreated such as WinBGI, OpenBGI, ZBGI and the more modern SDL\_bgi. DOS development environments also have the GRX graphics library which has a mostly compatible BGI API. Note that all of these libraries are very different under the hood compared to the original BGI and will often require some modification of the source code to compile. Being able to test an old graphics application in its original Borland Turbo C context gives us the opportunity to make sure the code was bug free and functional, as well as allowing us to see the underlying routines in the original context before transcribing the code to a different library context.

I am only going to show the setup for Turbo C V1.0 as the other 2 are very similar. I keep all 3 versions available for testing old code. Version 1.5 and 2.0 both have an automated installer whereas V1.0 does not have an installer. Version 1.0 needs to be manually copied to the correct directory structure which can be difficult to identify. I have offered some guides to this directory structure and file placement below. If you feel that it is too complicated you can copy all files from the drive images to the root TC10 directory and TC1.0 will work fine. Just keep in mind the config files when using the subdirectories INCLUDE and LIB.

For Windows 95 it is perfectly acceptable to copy the TC development directories from FreeDOS to your Windows 95 C:\DEVEL\TCnn\ directory.

All of the batch files used in Windows 95 are much the same as for FreeDOS with the following changes.

The addition of the default COMMAND and BIN paths for Windows 95 as well as the CTMOUSE when in DOS-Mode.

```
REM SET PATH=%path%;C:\WINDOWS\COMMAND;C:\DOSLINKS
REM CTMOUSE /R55
And you need to change any instance of %dosdrv% to C:\
%dosdrv%\DEVEL
to
C:\DEVEL
```

Turbo C versions 1.5 and 2.0 come with an automated installer.

Note I have not used V2.01 due to a legacy bug in the library code.

Download the version/s required from the following link:

<https://winworldpc.com/product/borland-turbo-c/1x>

“Borland Turbo C 1.0 (5.25).7z”

“Borland Turbo C 1.0 Manuals.7z”

...

“Borland Turbo C 1.5 (5.25).7z”

“Borland Turbo C 1.5 Manual Additions.7z”

...

“Borland Turbo C 2.0 (1988) (3.5-720k).7z”

“Borland Turbo C 2.0 Manual.7z”

**NOTE:** Before installing Turbo C please make a backup of your FDAUTO>BAT and FDCONFIG.SYS files as installers can make changes to these files.

Each of the 3 downloads will contain a set of floppy install disk images. You will need to open the .img files with something like 7-Zip to establish which one contains the Installer and help documents. IDE.img contains the README file with the information for installing Turbo C as well as the order of disks and the file contents.

“Turbo\_C\_Users\_Guide\_1987.pdf” pg. 14 also has some descriptions of how to install and set up Turbo C V1.0

You may find it more convenient to unpack the floppy disk images and arrange the file and directory structures on you host machine rather than doing so in the DOS command line.

You can also download and view the directory structure of an already organized TC10 install directory “tc.zip”. Note that is slightly modified from the original install disks with the addition of some other library and binary utilities..

<https://eecs.wsu.edu/~cs150/prog/tcininstall.htm>

There are a number of acceptable ways to set up Turbo C as you will see in the illustrated directory trees and the guides are a little ambiguous. If you encounter a compiler error or more likely a linker error due to a LIB object not found then recheck the location of the library against the paths in the TC config files.

#### Turbo C V1.0 disk image labels:

- Disk 1 - IDE.img
- Disk 2 - Command Line Utilities.img
- Disk 3 - Headers, Libraries, Examples.img
- Disk 4 - Libraries, Examples.img

#### Turbo C V1.0 recommended directory structure:

- C:\DEVEL\TC10\\*.\*
- C:\DEVEL\TC10\INCLUDE\\*.\*
- C:\DEVEL\TC10\INCLUDE\SYS\\*.\*
- C:\DEVEL\TC10\LIB\\*.\*

NOTE: The directory name is only to distinguish between the 3 versions mentioned in this guide TC10, TC15 and TC20.

The Turbo C documentation states to place the files in the following places:

Note that some files on the disk are example projects with .C, .LIB and .OBJ extensions.

TC10 (root)

TC.EXE

TCC.EXE

TLINK.EXE

All program files, .c source files and project files.

\INCLUDE

All header files on disk 3.

INCLUDE\SYS

\LIB

all the library and start-up files

\*.LIB

C0x.OBJ

LIB\STARTUP

You can also get some hints from the Turbo C 1.5 Disk01.img INSTALL.BAT and INSTALLH.BAT

Just note that it contains 5 organised disks rather than the 4 mixed disks in V1.0

The following is my install of Turbo C V1.0 from the 4 install disk images. This is essentially following the instructions from the README and Manual. I have left the additional executables. .ASM file in the root instead of a .\UTILITY directory as some have used.

### TC10 from “Borland Turbo C 1.0 (5.25).7z”

```
\---TC10
|   BAR.C
|   BUILD-C0.BAT
|   C0.ASM
|   CNVTCFG.EXE
```

```
| CPP.EXE  
| FILECOMP.C  
| GETOPT.C  
| HELLO.C  
| MAIN.C  
| MAKE.EXE  
| MATHERR.C  
| PBAR.PRO  
| README  
| README.COM  
| RULES.ASI  
| SETARGV.ASM  
| SETENV.PASM  
| TC.EXE  
| TC10.BAT  
| TCC.EXE  
| TCCCONFIG.TC  
| TCHELP.TCH  
| TCINST.COM  
| TLINK.EXE  
| TOUCH.COM  
  
+---INCLUDE  
| | ALLOC.H  
| | ASSERT.H  
| | BIOS.H  
| | CONIO.H  
| | CTYPE.H  
| | DIR.H  
| | DOS.H  
| | ERRNO.H  
| | FCNTL.H  
| | FLOAT.H  
| | IO.H  
| | LIMITS.H  
| | MATH.H  
| | MEM.H  
| | PROCESS.H  
| | SETJMP.H  
| | SHARE.H  
| | SIGNAL.H  
| | STDARG.H  
| | STDDEF.H  
| | STDIO.H  
| | STDLIB.H  
| | STRING.H  
| | TIME.H  
| | VALUES.H  
| |  
| \---SYS  
| | STAT.H
```

```
|  
+---LIB  
|   COC.OBJ  
|   COH.OBJ  
|   COL.OBJ  
|   COM.OBJ  
|   COS.OBJ  
|   COT.OBJ  
|   CC.LIB  
|   CH.LIB  
|   CL.LIB  
|   CM.LIB  
|   CPINIT.OBJ  
|   CS.LIB  
|   EMU.LIB  
|   FP87.LIB  
|   MATHC.LIB  
|   MATHH.LIB  
|   MATHL.LIB  
|   MATHM.LIB  
|   MATHS.LIB  
|   MCMVSMEM.OBJ  
  
+---OUT  
|   HELLO.EXE  
|   HELLO.OBJ  
  
\---PROJ  
    MCALC.C  
    MCALC.DOC  
    MCALC.H  
    MCALC.PRJ  
    MCDISPLY.C  
    MCINPUT.C  
    MCMVSMEM.C  
    MCOMMAND.C  
    MCPARSER.C  
    MCUTIL.C
```

You can also use the following as a reference. It may not be exactly the same as the instructions from the original floppy disks or contain exactly the same files but offers a reasonable guide.

### TC10 directory tree from “tc.zip”

<https://eecs.wsu.edu/~cs150/prog/tcinstall.htm>

```
\---TC10  
    |   BGIDEMO.EXE  
    |   BGIDEMO.OBJ  
    |   TC.EXE  
    |   TC1p.html
```

```
| TCCONFIG.TC
| TCHELP.TCH
|
+---BGI&CHR
| ATT.BGI
| BGI.ARC
| BGIDEMO.C
| BGIDEMO.EXE
| BGIOBJ.EXE
| CGA.BGI
| EGAVGA.BGI
| GOTH.CHR
| HERC.BGI
| IBM8514.BGI
| LITT.CHR
| PC3270.BGI
| SANS.CHR
| TC1p.html
| TRIP.CHR
|
+---INCLUDE
| | ALLOC.H
| | ASSERT.H
| | BIOS.H
| | CONIO.H
| | CTYPE.H
| | DIR.H
| | DOS.H
| | EDITOR.H
| | ERRNO.H
| | EVAL.H
| | EXC.H
| | FCNTL.H
| | FLOAT.H
| | GRAPHICS.H
| | IO.H
| | LIMITS.H
| | MATH.H
| | MEM.H
| | PROCESS.H
| | SETJMP.H
| | SHARE.H
| | SIGNAL.H
| | STDARG.H
| | STDDEF.H
| | STDIO.H
| | STDLIB.H
| | STRING.H
| | TC1p.html
| | TIME.H
| | VALUES.H
```

```
| |
| \---SYS
|   STAT.H
|   TC1p.html
|   TIMEB.H
|   TYPES.H
|
+---LIB
|   COC.OBJ
|   COH.OBJ
|   COL.OBJ
|   COM.OBJ
|   COS.OBJ
|   COT.OBJ
|   CC.LIB
|   CGA.OBJ
|   CH.LIB
|   CL.BAK
|   CL.LIB
|   CM.LIB
|   CS.LIB
|   EGAVGA.OBJ
|   EMU.LIB
|   EXC.OBJ
|   FP87.LIB
|   GRAPHICS.LIB
|   HERC.OBJ
|   LIBTMPAA.AAA
|   MATHC.LIB
|   MATHH.LIB
|   MATHL.LIB
|   MATHM.LIB
|   MATHS.LIB
|   TC1p.html
|
\---STARTUP
  BUILD-C0.BAT
  C0.ASM
  EMUVARS.ASI
  MAIN.C
  RULES.ASI
  SETARGV.ASM
  SETENV.PASM
  STARTUP.ARC
  TC1p.html
  WILDARGS.OBJ
|
\---UTILITY
  CINSTXFR.EXE
  CPP.EXE
  GREP.COM
```

```
MAKE.EXE  
OBJXREF.COM  
TC1p.html  
TCC.EXE  
TCCCONFIG.EXE  
TCINST.EXE  
TLIB.EXE  
TLINK.EXE  
TOUCH.COM  
UNPACK.COM
```

Once you have created the directory structure with the files from the 4 floppy disk images, copy the directory to your DOS drive as C:\DEVEL\TC10\\*.\*

If you choose to also install Turbo C 1.5 and 2.0 then use the same naming convention TC15 and TC20.

TC.EXE is the IDE version, and TCC.EXE is the command line version.

TCCCONFIG.TC is not created by default.

Check that your FDCONFIG.SYS file has the following 2 lines. They are usually set by default in FD V1.3. **In Windows 95 these lines will be added to the “Exit To DOS.PIF” for DOS Mode only. When accessing TC in windowed mode this is not required.**

```
!BUFFERS=20
```

```
!FILES=40
```

Create a batch file to launch TC1.x (or 2.0) and place it in .\FREEDOS\LINKS.

Note that I am setting the environment variables and paths as would be found in the config file.

Don't forget to name the TC directory according to each version.

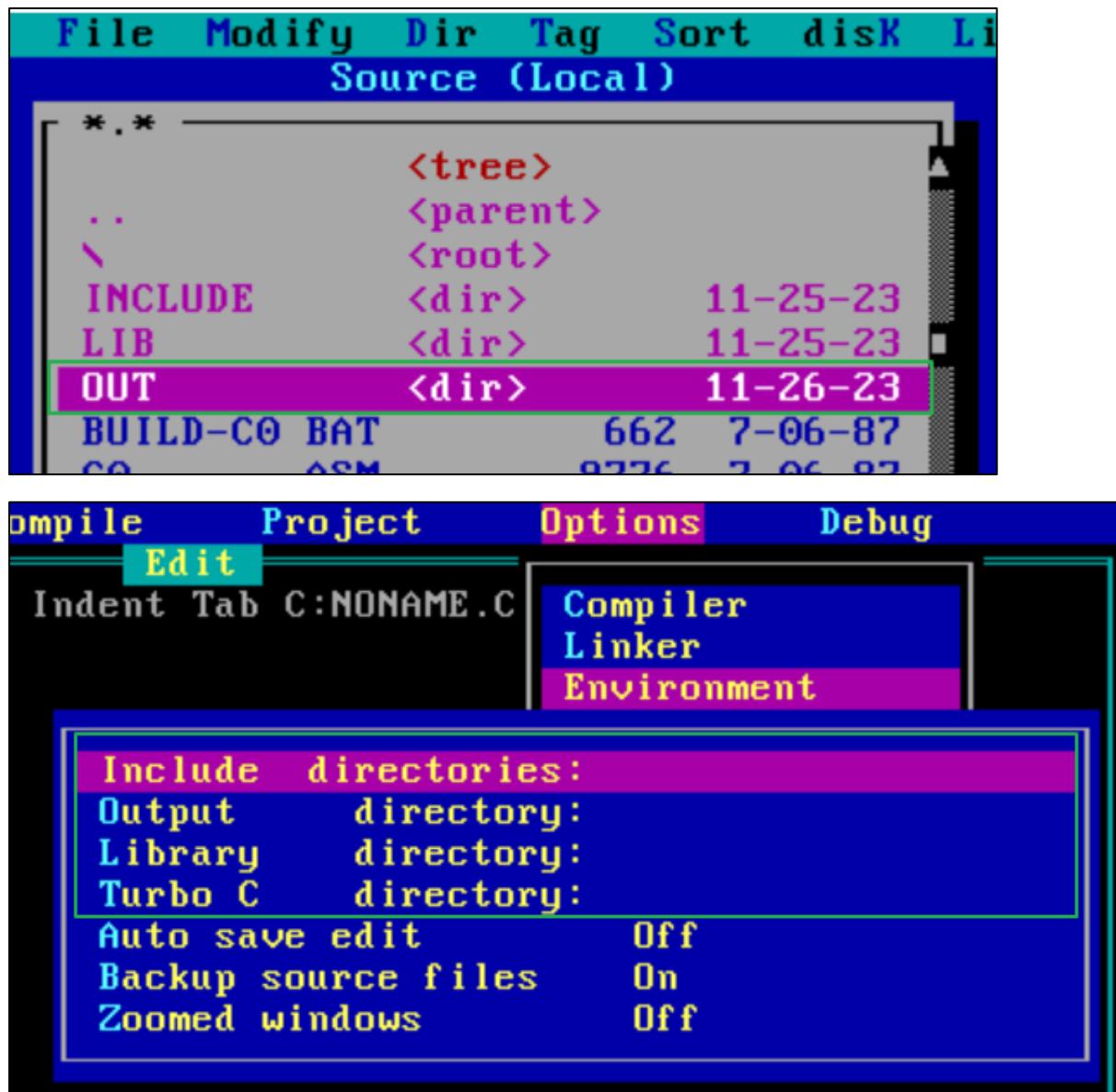
```
TC10.BAT
@ECHO OFF
REM Set the DOS path environments if needed.
REM SET PATH=%path%;C:\WINDOWS;C:\WINDOWS\COMMAND;C:\DOSLINKS
REM For DOS-MODE only.
REM CTMOUSE /R55
CLS
SET PATH=%path%;C:\DEVEL\TC10;C:\DEVEL\TC10\INCLUDE;C:\DEVEL\TC10\LIB
SET INCLUDE=C:\DEVEL\TC10\INCLUDE
SET LIB=C:\DEVEL\TC10\LIB
SET CLASSPATH=C:\DEVEL\TC10\LIB
CD \DEVEL\TC10
call C:\DEVEL\TC10\TC.EXE
CLS
```

For Windows 95 you can create a shortcut PIF file to the Batch file and place it on your desktop or in the ALaunch menu directory.

Launch TC.exe from the batch file TC.BAT or the Windows PIF shortcut.

Use "Alt + O" to open "Options -> Environment".

Fill in the 4 paths for TC as is found in the batch file above. Output is a directory of your choice to place the final compiles executables of your project. Make sure that the named output directory exists at the entered location before attempting to compile executables. If no Output directory is entered the executables are created in the TC root directory.



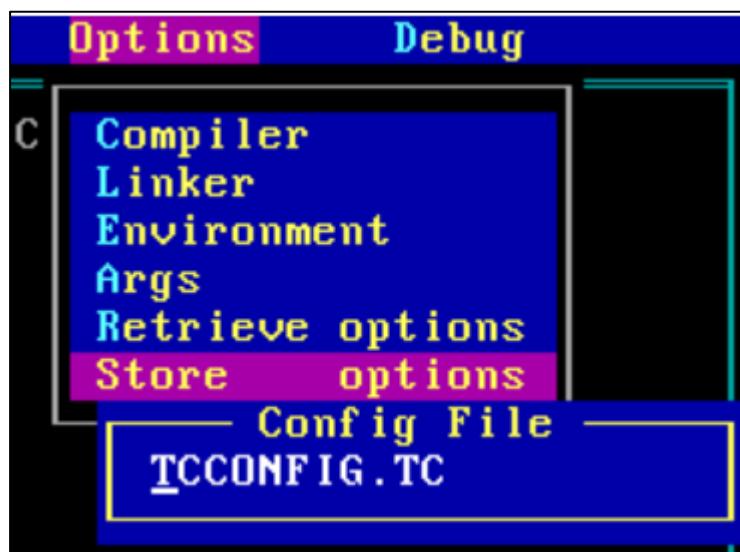
```

Include directories: C:\DEVEL\NTC10\INC
Output directory: C:\DEVEL\NTC10\OUT
Library directory: C:\DEVEL\NTC10\LIB
Turbo C directory: C:\DEVEL\NTC10
Auto save edit      Off
Backup source files On
Zoomed windows      Off

```

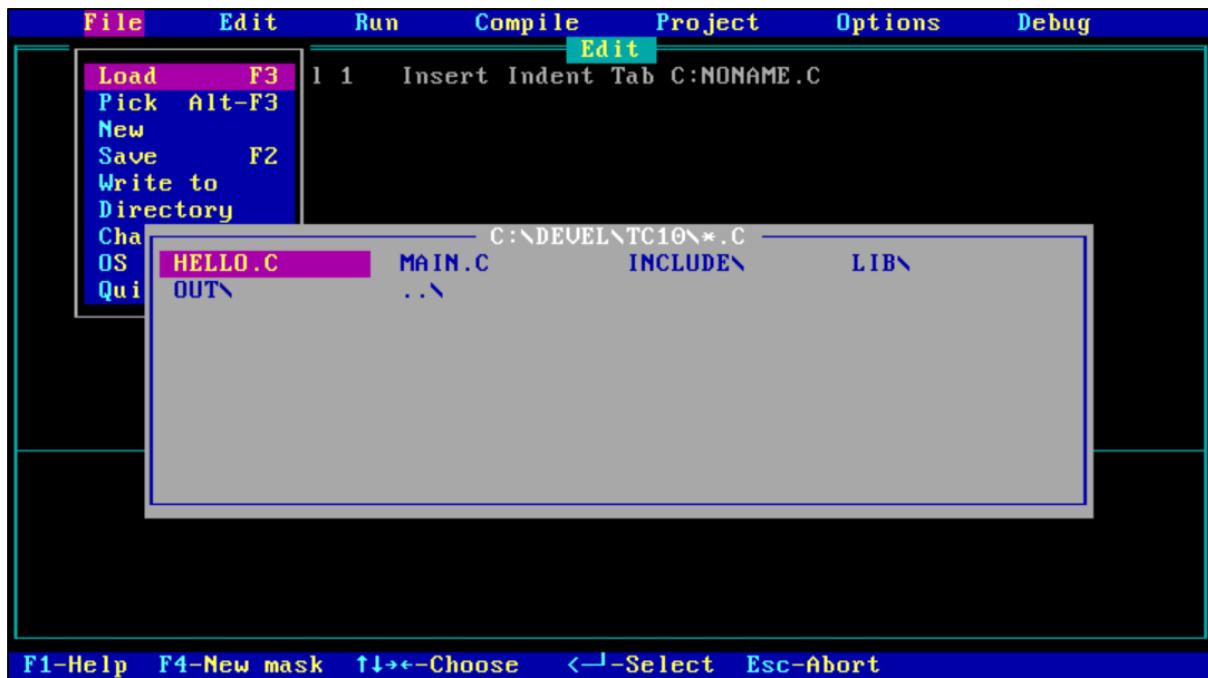
Also take note of the TAB size (4 recommended) and other editor options.

Next select “Store Options” and enter to use the default TCCONFIG.TC file. You can create additional config files with a different name for individual projects if needed. This will save the configs for next time you start the TC IDE.



Next use “Alt + F” to open “File -> Load” and select the HELLO.C example in the root directory.

## A Beginners Guide To DOS Programming



F1-Help F4-New mask ↑↓↔←Choose <→-Select Esc-Abort

This screenshot shows the same DOS-based IDE as the previous one, but now displaying a code editor with a C program. The menu bar and project tree are identical. The code editor window shows the following C code:

```
Line 1 Col 1 Insert Indent Tab C:HELLO.C
/* HELLO.C -- Hello, world */

#include <stdio.h>

main()
{
    printf("Hello, world\n");
}
```

The code editor has the title 'C:\DEUVEL\TC10\\*.\*'. Below the code editor is a message area labeled 'Message'.

Next use "Alt +C" to select "Compile -> Make EXE file, and then Enter.

## A Beginners Guide To DOS Programming

The screenshot shows a DOS-based IDE interface. The menu bar includes File, Edit, Run, Compile, Project, Options, and Debug. The Compile menu is currently open, displaying options: Compile to OBJ C:HELLO.OBJ, Make EXE file C:HELLO.EXE, Link EXE file, Build all, and Primary C file:.

```
Line 1 Col 1 Inse
/*
    HELLO.C -- Hello, wor
#include <stdio.h>
main()
{
    printf("Hello, world\n");
}
```

Message

F1-Help F5-Zoom F6-Edit F9-Make F10-Main Menu

The screenshot shows a DOS-based IDE interface. The menu bar includes File, Edit, Run, Compile, Project, Options, and Debug. The Edit menu is currently open, displaying options: Line 1 Col 1 Insert Indent Tab C:HELLO.C, and HELLO.C -- Hello, world \*/.

```
Line 1 Col 1 Insert Indent Tab C:HELLO.C
/*
    HELLO.C -- Hello, world */
#include <stdio.h>
main()
{
    printf("He
}
```

The message window displays the following text:

Making  
C:\DEVEL\TC10\OUT\HELLO.EXE  
is up to date.  
Success :

F1-Help F5-Zoom F6-Edit F9-Make F10-Main Menu

The screenshot shows the Turbo C IDE interface. The menu bar includes File, Edit, Run, Compile, Project, Options, and Debug, with Edit selected. The main window displays the source code for HELLO.C:

```
Line 1      Col 1  Insert Indent Tab C:HELLO.C
/*      HELLO.C -- Hello, world */

#include <stdio.h>

main()
{
    printf("Hello, world\n");
}
```

Below the code, a message window shows the compilation and linking process:

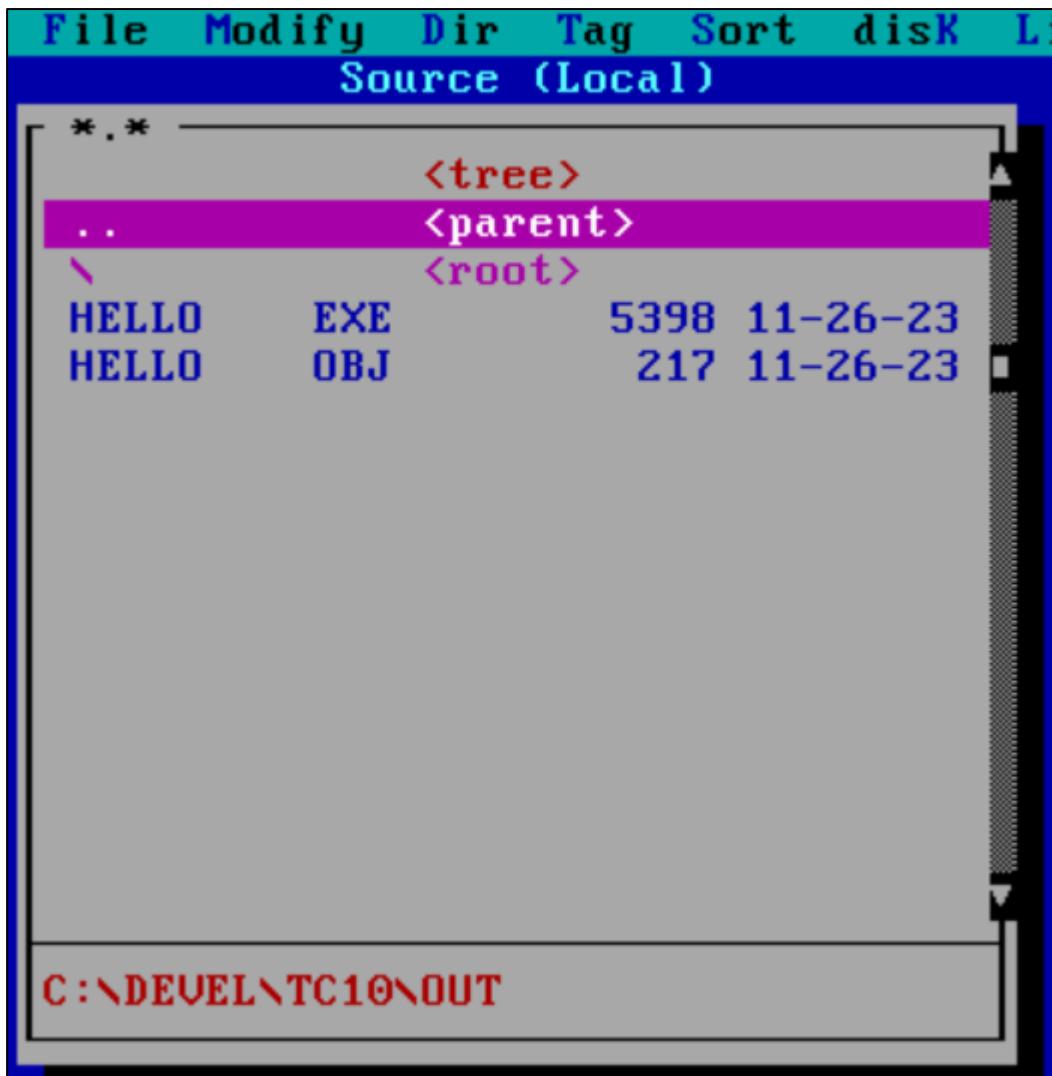
```
Message
Compiling C:\DEVEL\TC10\HELLO.C:
Linking C:\DEVEL\TC10\OUT\HELLO.EXE:
```

At the bottom, a status bar shows keyboard shortcuts: F1-Help, F5-Zoom, F6-Edit, F9-Make, F10-Main Menu.

Next select “Alt + R” to Run the compiled executable “HELLO.EXE”

The screenshot shows a DOS console window. The text "Hello, world" is displayed in yellow, followed by a prompt "Press any key to return to Turbo C . . ." in white.

Note: If we close TC.EXE (IDE) and navigate to .\TC10\OUT then run the HELLO.EXE the default console text colors will return. I am currently unsure why the console defaults to Yellow, but this is not reflected in the final executable.



```
Hello, world  
Press any key . . .
```

You will need to add extra linker options (Options -> Args) to link 3<sup>rd</sup> party libraries used in your project in much the same way as any other compiler.

You can also set up FED Folding Editor to use with Turbo C in much the same way as I did for DJGPP.

Turbo C 1.x did not ship the GBI graphics.h but if you look at the "tc.zip from <https://eeecs.wsu.edu/~cs150/prog/tcinstall.htm> you will see that it is included. You can use this as a guide for adding graphics or other libraries.

TC2.0 comes with BGI graphics.h included.

Please note that compilers are quite complex by design so I cannot give instructions for every possible way to implement the compiler settings or how to set up for different libraries and project. This is just a guide to get you to a position where you can compile basic source from the standard libraries. Take some time to read the documentation for the compiler as well as for libraries you wish to use.

Look at other public projects using Turbo C for hints and guides.

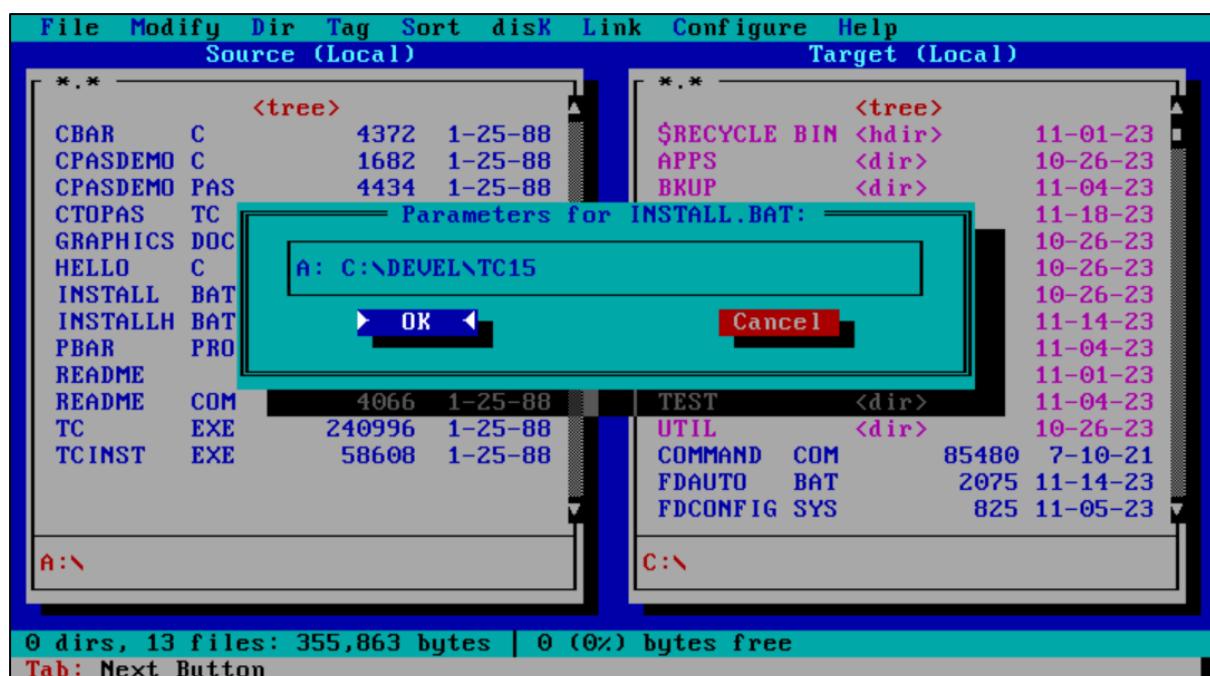
### TC15

Mount "Disck01.img".

Navigate to drive A: from the command line and run the following install line.

INSTALL.BAT A: C:\DEVEL\TC15

You can open a DOS windowed instance or use "Exit To DOS.PIF" to use DOS Mode.



```
A:>echo off
Making directories...
A:\INSTALLH.BAT =>> C:\DEVEL\TC15\INSTALLH.BAT
Copying files...
A:\README.COM =>> C:\DEVEL\TC15\README.COM
A:\TC.EXE =>> C:\DEVEL\TC15\TC.EXE
A:\INSTALL.BAT =>> C:\DEVEL\TC15\INSTALL.BAT
A:\INSTALLH.BAT =>> C:\DEVEL\TC15\INSTALLH.BAT
A:\TCINST.EXE =>> C:\DEVEL\TC15\TCINST.EXE
A:\GRAPHICS.DOC =>> C:\DEVEL\TC15\GRAPHICS.DOC
A:\HELLO.C =>> C:\DEVEL\TC15\HELLO.C
A:\CPASDEMO.PAS =>> C:\DEVEL\TC15\CPASDEMO.PAS
A:\CPASDEMO.C =>> C:\DEVEL\TC15\CPASDEMO.C
A:\CTOPAS.TC =>> C:\DEVEL\TC15\CTOPAS.TC
A:\CBAR.C =>> C:\DEVEL\TC15\CBAR.C
A:\PBAR.PRO =>> C:\DEVEL\TC15\PBAR.PRO
A:\README. =>> C:\DEVEL\TC15\README.

Please insert the Turbo C disk labeled COMMAND LINE/UTILITIES into drive A:
Press any key to continue . . .
```

Disk02.img

```
A:\CPASDEMO.PAS =>> C:\DEVEL\TC15\CPASDEMO.PAS
A:\CPASDEMO.C =>> C:\DEVEL\TC15\CPASDEMO.C
A:\CTOPAS.TC =>> C:\DEVEL\TC15\CTOPAS.TC
A:\CBAR.C =>> C:\DEVEL\TC15\CBAR.C
A:\PBAR.PRO =>> C:\DEVEL\TC15\PBAR.PRO
A:\README. =>> C:\DEVEL\TC15\README.

Please insert the Turbo C disk labeled COMMAND LINE/UTILITIES into drive A:
Press any key to continue . . .
A:\TCC.EXE =>> C:\DEVEL\TC15\TCC.EXE
A:\TLINK.EXE =>> C:\DEVEL\TC15\TLINK.EXE
A:\BGIOBJ.EXE =>> C:\DEVEL\TC15\BGIOBJ.EXE
A:\CPP.EXE =>> C:\DEVEL\TC15\CPP.EXE
A:\MAKE.EXE =>> C:\DEVEL\TC15\MAKE.EXE
A:\TCCCONFIG.EXE =>> C:\DEVEL\TC15\TCCCONFIG.EXE
A:\TLIB.EXE =>> C:\DEVEL\TC15\TLIB.EXE
A:\GREP.COM =>> C:\DEVEL\TC15\GREP.COM
A:\TOUCH.COM =>> C:\DEVEL\TC15\TOUCH.COM
A:\MAIN.C =>> C:\DEVEL\TC15\MAIN.C
A:\RULES.ASI =>> C:\DEVEL\TC15\LIB\RULES.ASI
A:\C0.ASM =>> C:\DEVEL\TC15\LIB\C0.ASM
A:\SETARGU.ASM =>> C:\DEVEL\TC15\LIB\SETARGU.ASM
A:\SETENUP.ASM =>> C:\DEVEL\TC15\LIB\SETENUP.ASM
A:\BUILD-C0.BAT =>> C:\DEVEL\TC15\LIB\BUILD-C0.BAT

Please insert the Turbo C disk labeled HEADER FILES/LIBRARIES into drive A:
Press any key to continue . . .
```

Dosk03.img

```
A:\MEM.H =>> C:\DEVEL\TC15\INCLUDE\MEM.H
A:\PROCESS.H =>> C:\DEVEL\TC15\INCLUDE\PROCESS.H
A:\SETJMP.H =>> C:\DEVEL\TC15\INCLUDE\SETJMP.H
A:\SHARE.H =>> C:\DEVEL\TC15\INCLUDE\SHARE.H
A:\SIGNAL.H =>> C:\DEVEL\TC15\INCLUDE\SIGNAL.H
A:\STDARG.H =>> C:\DEVEL\TC15\INCLUDE\STDARG.H
A:\STDDEF.H =>> C:\DEVEL\TC15\INCLUDE\STDDEF.H
A:\STDIO.H =>> C:\DEVEL\TC15\INCLUDE\STDIO.H
A:\STDLIB.H =>> C:\DEVEL\TC15\INCLUDE\STDLIB.H
A:\STRING.H =>> C:\DEVEL\TC15\INCLUDE\STRING.H
A:\TIME.H =>> C:\DEVEL\TC15\INCLUDE\TIME.H
A:\VALUES.H =>> C:\DEVEL\TC15\INCLUDE\VALUES.H
A:\SYS\STAT.H =>> C:\DEVEL\TC15\INCLUDE\SYS\STAT.H
A:\COT.OBJ =>> C:\DEVEL\TC15\LIB\COT.OBJ
A:\COS.OBJ =>> C:\DEVEL\TC15\LIB\COS.OBJ
A:\COL.OBJ =>> C:\DEVEL\TC15\LIB\COL.OBJ
A:\CS.LIB =>> C:\DEVEL\TC15\LIB\CS.LIB
A:\MATHS.LIB =>> C:\DEVEL\TC15\LIB\MATHS.LIB
A:\NCL.LIB =>> C:\DEVEL\TC15\LIB\NCL.LIB
A:\MATHL.LIB =>> C:\DEVEL\TC15\LIB\MATHL.LIB
A:\EMU.LIB =>> C:\DEVEL\TC15\LIB\EMU.LIB
A:\GRAPHICS.LIB =>> C:\DEVEL\TC15\LIB\GRAPHICS.LIB
A:\FP87.LIB =>> C:\DEVEL\TC15\LIB\FP87.LIB
Please insert the Turbo C disk labeled LIBRARIES into drive A:
Press any key to continue . . .
```

Disk04.img

```
A:\SYS\STAT.H =>> C:\DEVEL\TC15\INCLUDE\SYS\STAT.H
A:\COT.OBJ =>> C:\DEVEL\TC15\LIB\COT.OBJ
A:\COS.OBJ =>> C:\DEVEL\TC15\LIB\COS.OBJ
A:\COL.OBJ =>> C:\DEVEL\TC15\LIB\COL.OBJ
A:\CS.LIB =>> C:\DEVEL\TC15\LIB\CS.LIB
A:\MATHS.LIB =>> C:\DEVEL\TC15\LIB\MATHS.LIB
A:\NCL.LIB =>> C:\DEVEL\TC15\LIB\NCL.LIB
A:\MATHL.LIB =>> C:\DEVEL\TC15\LIB\MATHL.LIB
A:\EMU.LIB =>> C:\DEVEL\TC15\LIB\EMU.LIB
A:\GRAPHICS.LIB =>> C:\DEVEL\TC15\LIB\GRAPHICS.LIB
A:\FP87.LIB =>> C:\DEVEL\TC15\LIB\FP87.LIB
Please insert the Turbo C disk labeled LIBRARIES into drive A:
Press any key to continue . . .
A:\COC.OBJ =>> C:\DEVEL\TC15\LIB\COC.OBJ
A:\CC.LIB =>> C:\DEVEL\TC15\LIB\CC.LIB
A:\MATHC.LIB =>> C:\DEVEL\TC15\LIB\MATHC.LIB
A:\COM.OBJ =>> C:\DEVEL\TC15\LIB\COM.OBJ
A:\CM.LIB =>> C:\DEVEL\TC15\LIB\CM.LIB
A:\MATHM.LIB =>> C:\DEVEL\TC15\LIB\MATHM.LIB
A:\COH.OBJ =>> C:\DEVEL\TC15\LIB\COH.OBJ
A:\CH.LIB =>> C:\DEVEL\TC15\LIB\CH.LIB
A:\MATHH.LIB =>> C:\DEVEL\TC15\LIB\MATHH.LIB
ECHO is off
Please insert the Turbo C disk labeled EXAMPLES into drive A:
Press any key to continue . . .
```

Disk05.img

```
A:\CPINIT.OBJ =>> C:\DEVEL\TC15\CPINIT.OBJ
A:\CPINIT.LIB =>> C:\DEVEL\TC15\CPINIT.LIB
A:\ATT.BGI =>> C:\DEVEL\TC15\ATT.BGI
A:\CGA.BGI =>> C:\DEVEL\TC15\CGA.BGI
A:\EGAUGA.BGI =>> C:\DEVEL\TC15\EGAUGA.BGI
A:\HERC.BGI =>> C:\DEVEL\TC15\HERC.BGI
A:\IBM8514.BGI =>> C:\DEVEL\TC15\IBM8514.BGI
A:\PC3270.BGI =>> C:\DEVEL\TC15\PC3270.BGI
A:\GOTH.CHR =>> C:\DEVEL\TC15\GOTH.CHR
A:\LITT.CHR =>> C:\DEVEL\TC15\LITT.CHR
A:\SANS.CHR =>> C:\DEVEL\TC15\SANS.CHR
A:\TRIP.CHR =>> C:\DEVEL\TC15\TRIP.CHR
A:\MCALC.DOC =>> C:\DEVEL\TC15\MCALC.DOC
ECHO is off
You should now add C:\DEVEL\TC15 to your DOS PATH command so you can run
Turbo C from anywhere on your system. You should run TCINST to
install C:\DEVEL\TC15 as your Turbo C directory. Finally, you should
make certain that your root directory contains a file called CONFIG.SYS
which contains the following line somewhere in it:
ECHO is off
  FILES=20
ECHO is off
After you have done all this, Turbo C will be ready to use.

Press any key ..._
```

Check that you FDCONFIG.SYS file has the following 2 lines. They are usually set by default in FD V1.3. **In Windows 95 these lines will be added to the “Exit To DOS.PIF” for DOS Mode only. When accessing TC in windowed mode this is not required.**

```
!BUFFERS=20
!FILES=40
```

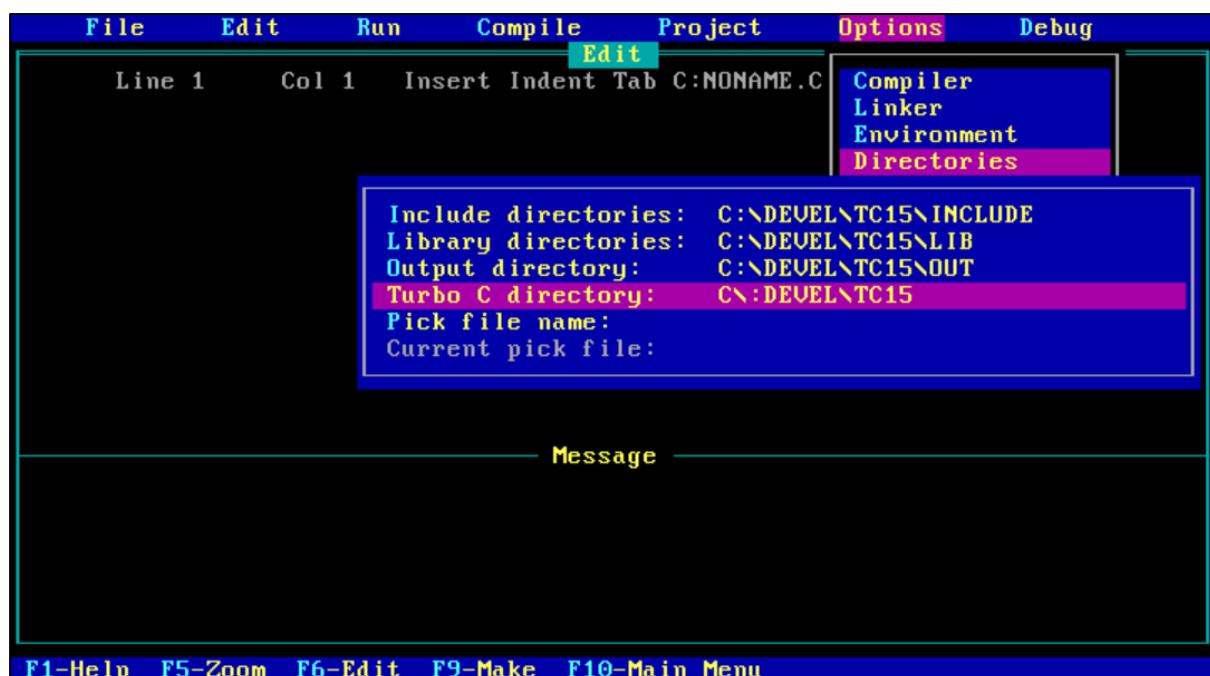
Create a batch file to launch V1.5 as is shown for TC10.BAT

For Windows 95 you can create a shortcut PIF file to the Batch file and place it on your desktop or in the ALaunch menu directory.

Set the Config paths from the TC.EXE IDE.

When the TC.EXE (1.5) IDE is launched for the first time, set the “Directories” paths to the correct locations as shown in the TC10 guide. Remember to create the Output directory on the drive if you set a path and name here.

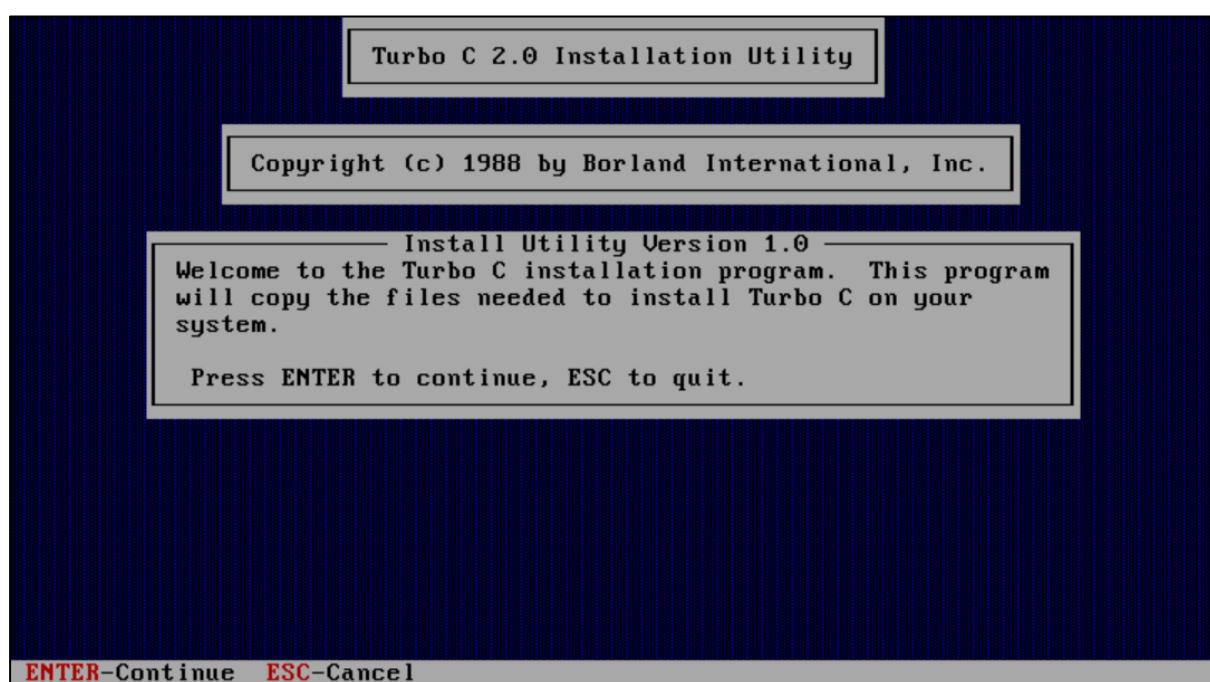
Also take note of the TAB size (4 recommended) and other editor options.

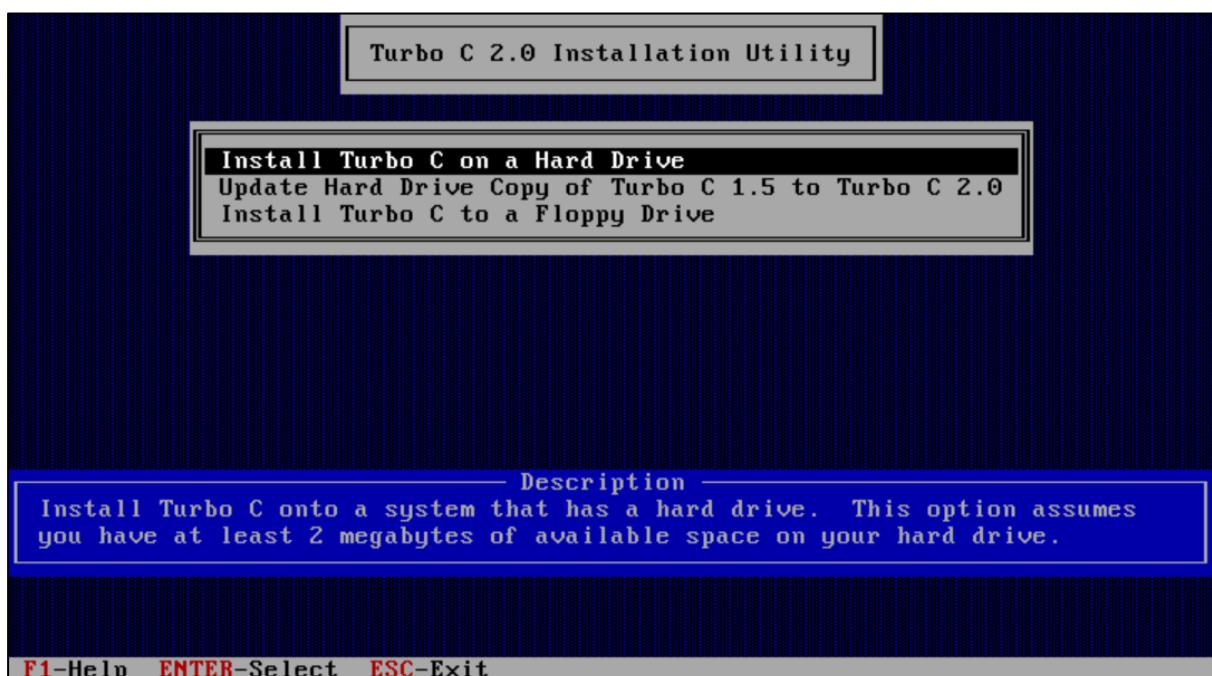


## TC20

Mount the “disk01.img”.

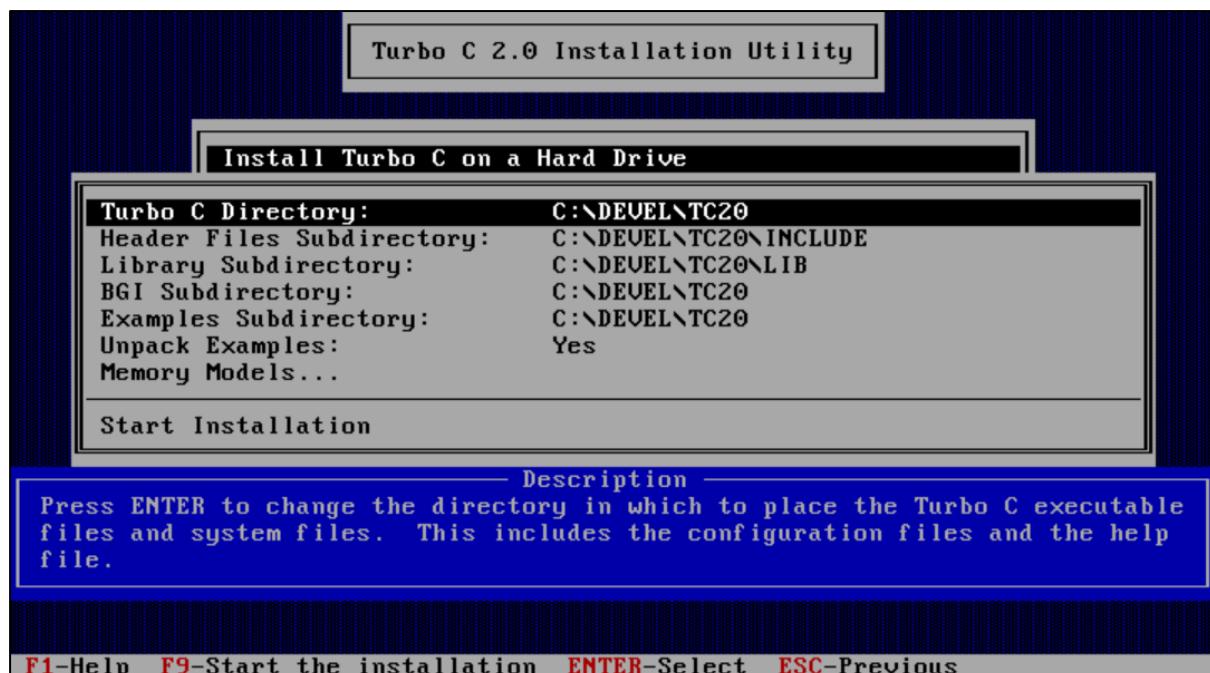
Navigate to A: and run the “INSTALL.EXE”.



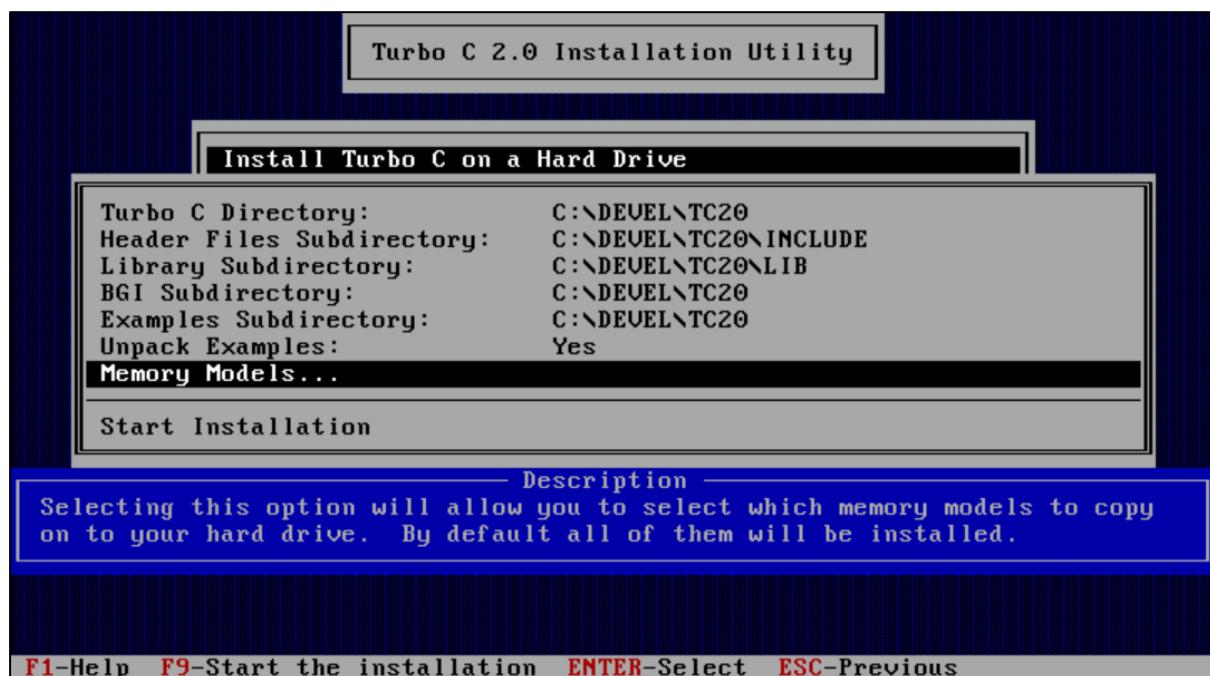


Enter the correct path for each. If you correct the "Turbo C Directory" the others are automatically updated.

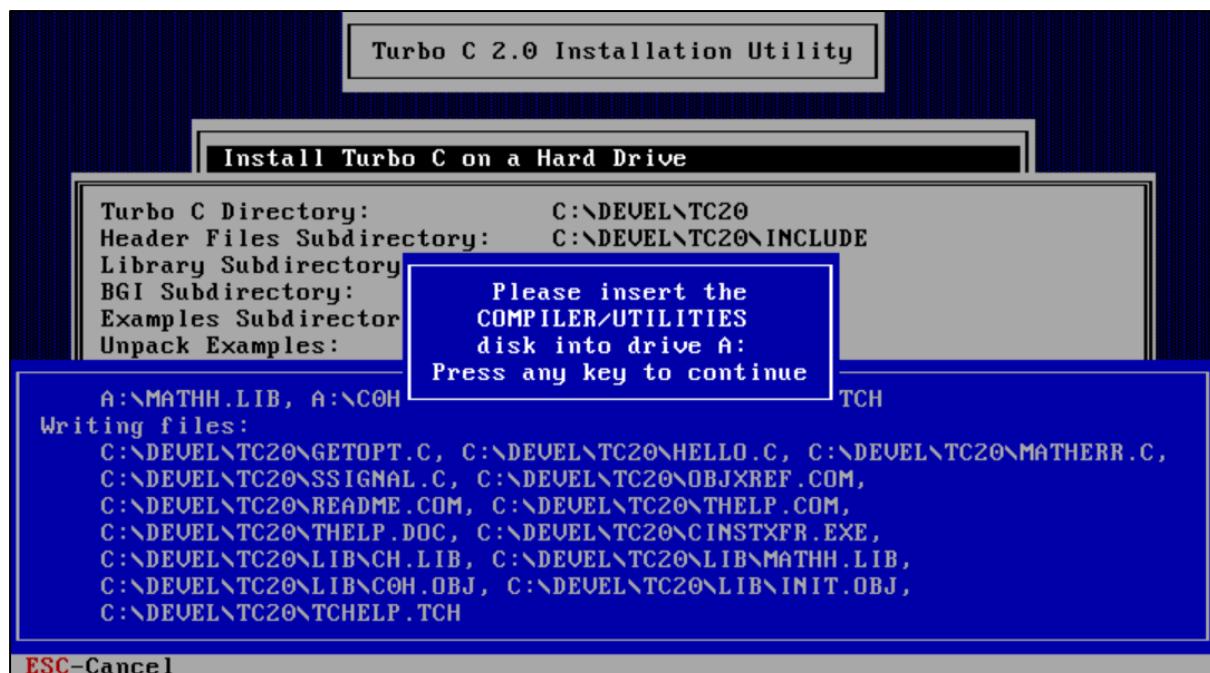
C:\DEVEL\TC20



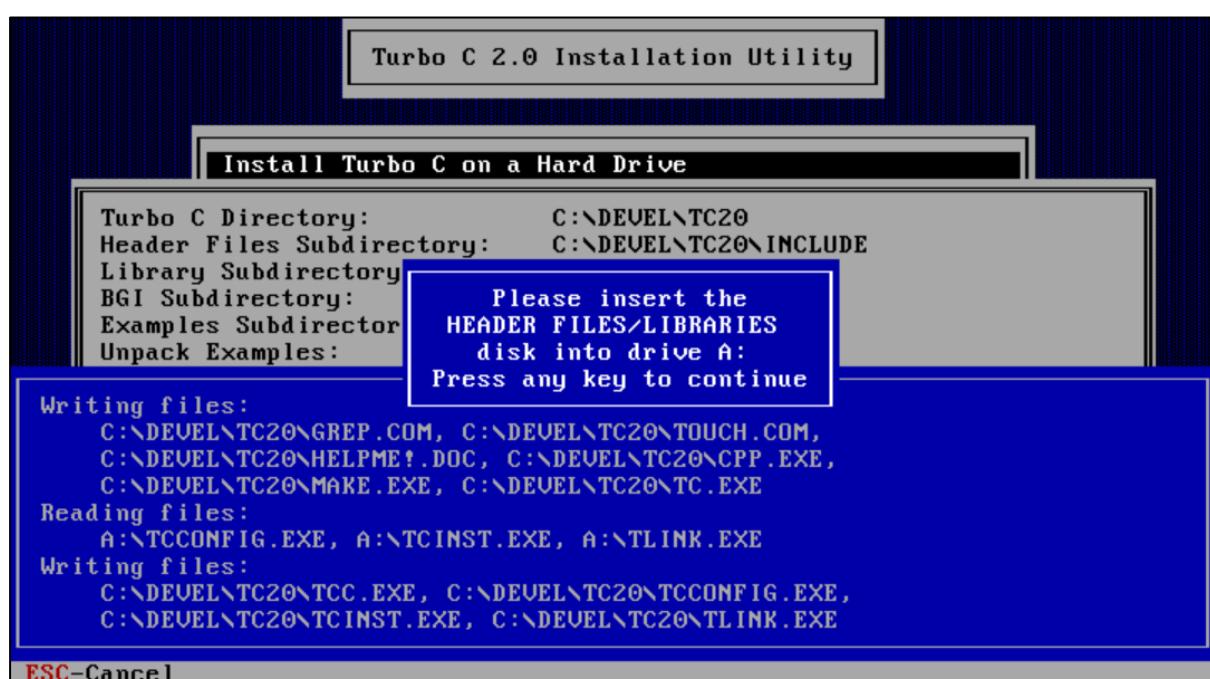
Leave the "Memory Models..." as the default where all are installed. Memory models can be selected at compile time.



Select "Start Installation" and Enter to continue.



Insert disk02.img



Insert disk03.img



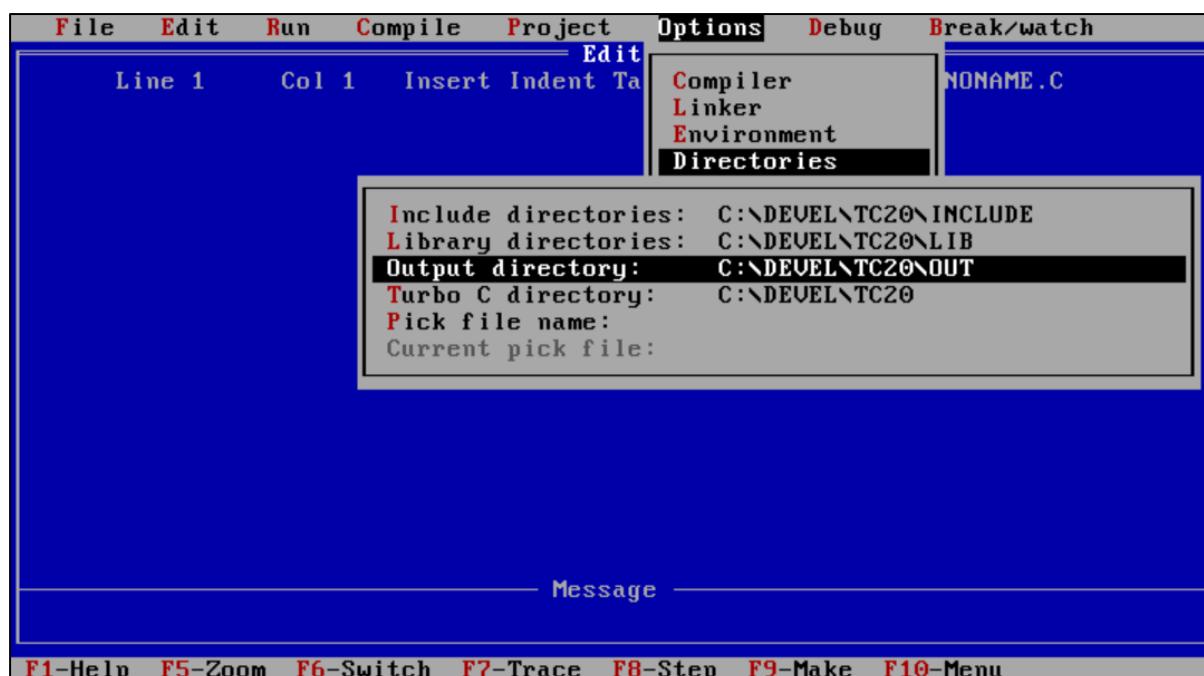
The Turbo C V2.0 install is complete.

FreeDOS sets !FILES=40 by default so don't change it to a lower value. This is only required in Windows DOS Mode and is not required when using DOS in windowed mode.

Paths and environment variable can be set in the batch file to launch TC20 in the same way as is shown for the V1.0 TCxx.BAT

For Windows 95 you can create a shortcut PIF file to the Batch file and place it on your desktop or in the ALaunch menu directory.

Set the Config paths from the TC.EXE IDE. The Turbo C V2.0 set up will create the basic config file for you, but it is still wise to check that they are correct.



Also take note of the TAB size (4 recommended) and other editor options.

Compile HELLO.C found in the root directory.

You will notice that TC20 does not insert a pause when running the application so you may need to place a debug pause as the last line of code: system("PAUSE");

---

## Other C [BCC, CC386]

I may consider including other compilers in future revisions.

---

## FreeBASIC DOS 32-bit V1.07.1a

FreeBASIC is a free, BASIC compiler for Windows (32-bit and 64-bit), 32 bit protected-mode DOS (COFF executables, like DJGPP), and Linux (x86, x86\_64, and ARM). It is open source and licensed under the GPL. It is designed to be syntax compatible with QuickBASIC, while expanding on the language and capabilities. It can create programs for MS-Windows, DOS and Linux, and is being ported to other platforms.

The following guide is based upon the FreeDOS version. You can copy and paste the FreeBASIC directory from FreeDOS to the Windows 95 drive. I would recommend using the same version in both FreeDOS and Windows 95 unless you are very familiar with the FreeBASIC and DJGPP development environments. Some modifications will need to be made to some batch files when moving from FreeDOS to Windows 95.

## A Beginners Guide To DOS Programming

Note that the FreeDOS BONUS CD FD13BNS.ISO contains an older version of FreeBASIC V1.07.1a and the current version at the time of this guide is FreeBASIC V1.10.0 for DOS. If you use the automated FDIMPLES install from the Bonus CD you will install V1.07.1a or if you use FDNPKG from the repository you may install V1.09.0. Check the version numbers from the FreeDOS repository as well as the Sourceforge repository and compare before continuing. If you want to use the latest FreeBASIC version you will need to carry out a manual install.

Please note: The manual install set from Sourceforge V1.10.0 is designed for use in a Windows 9x environment with LFN enabled. I would recommend only using the versions available from FreeDOS.

<https://www.ibiblio.org/pub/micro/pc-stuff/freedos/files/repositories/1.3-devel/>

<https://sourceforge.net/projects/fbc/files/FreeBASIC-1.10.0/Binaries-DOS/>

Update, Current FreeDOS version:

Version: 1.09.0

Entered-date: 2022-01-01

Please take note of the incompatible threads library used between DJGPP and FreeBASIC. The details are found at <https://www.freebasic.net/wiki/DevBuildDos>

“pthread.h”

```
#include <sys/socket.h> /* for sockaddr /
#include <sys/wtime.h> /* for struct timespec */
#include <sys/select.h>
```

socket.h, wtime.h and select.h are not compatible with FreeBASIC and need to be commented out. This applies to compiling the FreeBASIC compiler itself and I am uncertain if it also applies to libraries compiled for FreeBASIC. I have been compiling library files for FreeBASIC with the above 3 includes commented out without failure so far.

Also note that if you use the modified pthread.h file in DJGPP for compiling FreeBASIC you will need to use the unmodified pthread.h for compiling C source code or libraries. I have two separate DJGPP installs to keep my C build and FreeBASIC build separate. One has the modified pthread.h and the other uses the standard pthread.h.

### FreeDOS Bonus CD V1.07.1a

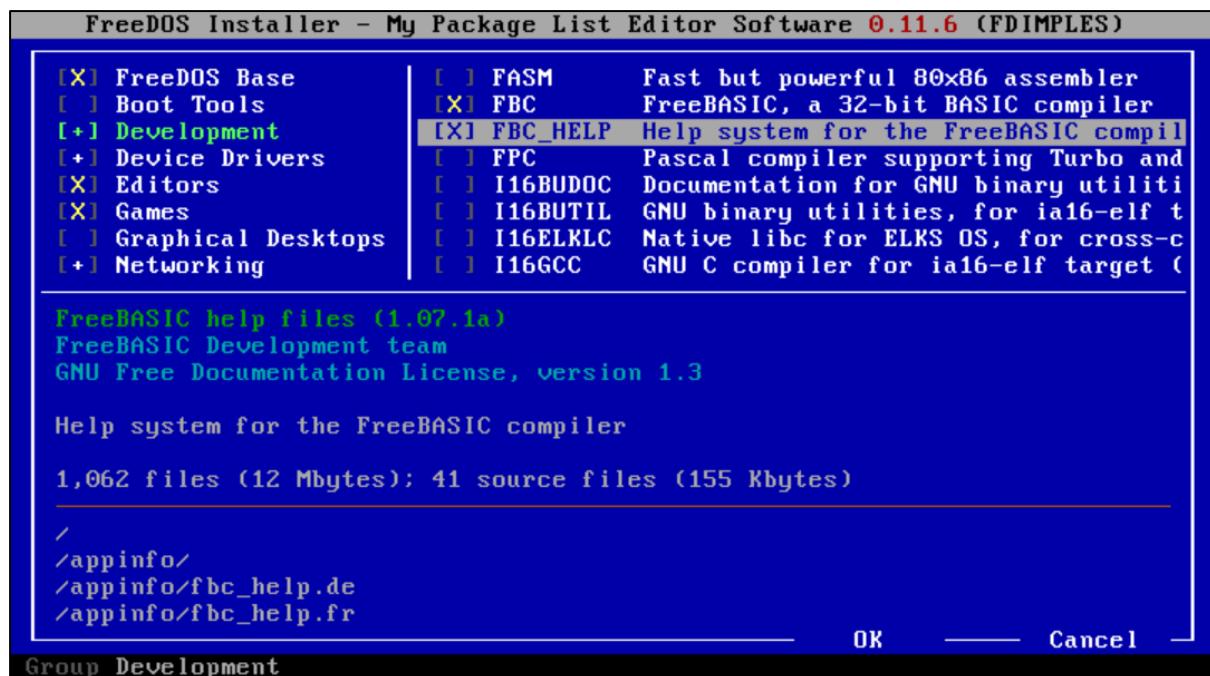
Note the V1.07.1a help file will not unpack correctly. It is best to update to V1.09.0 from the repository as shown in the next section “Use FDNPKG package manager from the repository” to correct this.

**FreeDOS only. I recommend installing to FreeDOS first and then copy the install files to Windows 95 C:\DEVEL\FBC directory.**

Mount the FD13BNS.ISO in VirtualBox.

From the command line run FDIMPLES and navigate to the section [ ] Development.

Select [X] FBC and [X] FBC\_HELP



Select OK and Enter to install the FreeBASIC compiler.

```

devel\fbc\examples\dos\modex.bas -> c:\devel\fbc\examples\dos\
devel\fbc\examples\dos\isrtimer.bas -> c:\devel\fbc\examples\dos\
devel\fbc\doc\lgpl.txt -> c:\devel\fbc\doc\
devel\fbc\doc\gpl.txt -> c:\devel\fbc\doc\
devel\fbc\doc\fbc.1 -> c:\devel\fbc\doc\
devel\fbc\bin\dos\ld.exe -> c:\devel\fbc\bin\dos\
devel\fbc\bin\dos\gprof.exe -> c:\devel\fbc\bin\dos\
devel\fbc\bin\dos\gdb.exe -> c:\devel\fbc\bin\dos\
devel\fbc\bin\dos\dxegen.exe -> c:\devel\fbc\bin\dos\
devel\fbc\bin\dos\as.exe -> c:\devel\fbc\bin\dos\
devel\fbc\bin\dos\ar.exe -> c:\devel\fbc\bin\dos\
appinfo\fbc.tr -> C:\FreeDOS\appinfo\
appinfo\fbc.sv -> C:\FreeDOS\appinfo\
appinfo\fbc.lsm -> C:\FreeDOS\appinfo\
appinfo\fbc.fr -> C:\FreeDOS\appinfo\
appinfo\fbc.de -> C:\FreeDOS\appinfo\
Package fbc installed: 923 files extracted, 0 errors.
install e:\packages\devel\fbc_help.zip
Error: Package contains a file that already exists locally:
c:\devel\fbc\lfnffiles.zip

Help encourage the development of FDIMPLES by becoming a Patron.
https://patreon.com/shidel

C:\>

```

The “fbc\_help.zip” and “lfnffiles.zip” do not unpack or install correctly so the help files will need to be unpacked manually at a later time.

Navigate to the root directory of FBC “CD \DEVEL\FBC”

Run “FBC[ .EXE ] -version” to test the install.

```
devel\fbc\bin\dos\dxe3gen.exe -> c:\devel\fbc\bin\dos\
devel\fbc\bin\dos\as.exe -> c:\devel\fbc\bin\dos\
devel\fbc\bin\dos\ar.exe -> c:\devel\fbc\bin\dos\
appinfo\fbc.tr -> C:\FreeDOS\appinfo\
appinfo\fbc.sv -> C:\FreeDOS\appinfo\
appinfo\fbc.lsm -> C:\FreeDOS\appinfo\
appinfo\fbc.fr -> C:\FreeDOS\appinfo\
appinfo\fbc.de -> C:\FreeDOS\appinfo\
Package fbc installed: 923 files extracted, 0 errors.
install e:\packages\devel\fbc_help.zip
Error: Package contains a file that already exists locally:
c:\devel\fbc\lfmfiles.zip

Help encourage the development of FDIMPLES by becoming a Patron.
https://patreon.com/shidel
```

```
C:\>fbc -version
Bad command or filename - "fbc".
C:\>cd devel
C:\DEVEL>cd fbc
C:\DEVEL\FBC>fbc -version
FreeBASIC Compiler - Version 1.07.1 (2019-09-27), built for dos (32bit)
Copyright (C) 2004-2019 The FreeBASIC development team.
standalone
C:\DEVEL\FBC>
```

The FreeBASIC compiler and base libraries are now installed. FreeBASIC does not come with an editor or IDE. I will show in the last part of this section how to set up FED as the IDE for FreeBASIC.

### Removal

Use the Bonus CD and run FDIMPLES.

De-select the [ ] FBC and [ ] FBC\_Help in [X] Development, then select OK and enter.

### Use FDNPKG package manager from the repository

This will correct the failed help file install from the BONUS CD.

You can run this as a standalone install, or as an update to the Bonus CD install. Just take note of FDNPKG “install” vs “update”.

### FDNPKG search FBC

```
C:\>FDNPKG search FBC
Package database loaded from local cache.

fbc - FreeBASIC, a 32-bit BASIC compiler
fbc_help - Help system for the FreeBASIC compiler
C:\>
```

### FDNPKG [install|update] FBC

```
C:\>FDNPKG search FBC
Package database loaded from local cache.

fbc - FreeBASIC, a 32-bit BASIC compiler
fbc_help - Help system for the FreeBASIC compiler
C:\>FDNPKG update FBC
Package database loaded from local cache.

Downloading package http://www.ibiblio.org/pub/micro/pc-stuff/freedos/files/repositories/latest-devel/fbc.zip...
Downloading fbc.zip... 1050664 bytes [5%]
```

```
devel\fbc\inc\gs1\gs1_fft.bi -> c:\devel\fbc\inc\gs1\
devel\fbc\inc\gs1\gs1_sum.bi -> c:\devel\fbc\inc\gs1\
devel\fbc\inc\gs1\gs1_math.bi -> c:\devel\fbc\inc\gs1\
devel\fbc\inc\gs1\gs1_dht.bi -> c:\devel\fbc\inc\gs1\
devel\fbc\inc\gs1\gs1_sort.bi -> c:\devel\fbc\inc\gs1\
devel\fbc\inc\gs1\gs1_test.bi -> c:\devel\fbc\inc\gs1\
devel\fbc\inc\gs1\gs1_qrng.bi -> c:\devel\fbc\inc\gs1\
devel\fbc\inc\gs1\gs1_sys.bi -> c:\devel\fbc\inc\gs1\
devel\fbc\inc\png.bi -> c:\devel\fbc\inc\
devel\fbc\inc\libintl.bi -> c:\devel\fbc\inc\
devel\fbc\inc\libexslt.bi -> c:\devel\fbc\inc\libexslt\
devel\fbc\inc\libexslt\libexslt.bi -> c:\devel\fbc\inc\libexslt\
devel\fbc\inc\pcre16.bi -> c:\devel\fbc\inc\
devel\fbc\inc\gif_1.lib.bi -> c:\devel\fbc\inc\
devel\fbc\inc\sqlite3.bi -> c:\devel\fbc\inc\
devel\fbc\doc\lgpl.txt -> c:\devel\fbc\doc\
devel\fbc\doc\fbc.1 -> c:\devel\fbc\doc\
devel\fbc\doc\gpl.txt -> c:\devel\fbc\doc\
appinfo\fbc.sv -> C:\FreeDOS\appinfo\
appinfo\fbc.lsm -> C:\FreeDOS\appinfo\
appinfo\fbc.tr -> C:\FreeDOS\appinfo\
appinfo\fbc.de -> C:\FreeDOS\appinfo\
appinfo\fbc.fr -> C:\FreeDOS\appinfo\
Package fbc installed: 975 files extracted, 0 errors.
C:\>_
```

Navigate to FBC CD \DEVEL\FBC and check the version.

```
FBC[.EXE] -version
```

```
C:\>CD \DEVEL\FBC
C:\DEVEL\FBC>FBC -version
FreeBASIC Compiler - Version 1.09.0 (2021-12-31), built for dos (32bit)
Copyright (C) 2004-2021 The FreeBASIC development team.
standalone
C:\DEVEL\FBC>
```

The package manager FDNPKG has updated the FreeBASIC compiler for version 1.07.1 a to V1.09.0

If the FDIMPLES install failed to install the help files from the Bonus CD then use the install option below to correct

```
FDNPKG [install|update] FBC_HELP
```

After this the help file will be correctly installed and correct the FDIMPLES failure.

```
devel\fbc\html\images\pal2.png -> c:\devel\fbc\html\images\
devel\fbc\html\images\imagecr.png -> c:\devel\fbc\html\images\
devel\fbc\html\images\fbwct.png -> c:\devel\fbc\html\images\
devel\fbc\html\images\putpset.png -> c:\devel\fbc\html\images\
devel\fbc\html\images\put-or.png -> c:\devel\fbc\html\images\
devel\fbc\html\images\rgba.png -> c:\devel\fbc\html\images\
devel\fbc\html\images\put-xor.png -> c:\devel\fbc\html\images\
devel\fbc\html\images\putgf1.png -> c:\devel\fbc\html\images\
devel\fbc\html\images\al1.png -> c:\devel\fbc\html\images\
devel\fbc\html\images\pal16.png -> c:\devel\fbc\html\images\
devel\fbc\html\images\pal256.png -> c:\devel\fbc\html\images\
devel\fbc\html\images\draw1.png -> c:\devel\fbc\html\images\
devel\fbc\html\images\draw2.png -> c:\devel\fbc\html\images\
devel\fbc\html\style.css -> c:\devel\fbc\html\
devel\fbc\fbhelp.txt -> c:\devel\fbc\
devel\fbc\doc\gfdl.txt -> c:\devel\fbc\doc\
devel\fbc\doc\gpl2.txt -> c:\devel\fbc\doc\
devel\fnf files.zip -> c:\devel\
appinfo\fbc_help.lsm -> C:\FreeDOS\appinfo\
appinfo\fbc_help.tr -> C:\FreeDOS\appinfo\
appinfo\fbc_help.fr -> C:\FreeDOS\appinfo\
appinfo\fbc_help.de -> C:\FreeDOS\appinfo\
appinfo\fbc_help.sv > C:\FreeDOS\appinfo\
Package fbc_help installed: 45 files extracted, 0 errors.
C:\DEVEL\FBC>_
```

The FreeBASIC compiler and base libraries are now installed. FreeBASIC does not come with an editor or IDE. I will show in the last part of this section how to set up FED as the IDE for FreeBASIC.

### Removal

Run “FDNPKG remove FBC\_HELP” and “FDNPKG remove FBC”.

This will completely remove the FBC install, including any original files installed via FDIMPLES from the Bonus CD.

### Manual Install – Not recommended

Please note: This install set is designed for use in a Windows 9x environment with LFN enabled. I would recommend only using the versions available from FreeDOS.

Download the version you wish to use in FreeDOS from the FreeBASIC website, or directly from Sourceforge.

<https://sourceforge.net/projects/fbc/files/>

<https://sourceforge.net/projects/fbc/files/FreeBASIC-1.10.0/Binaries-DOS/>

“FreeBASIC-1.10.0-dos.zip” – 2023-05-14

Unpack the “FreeBASIC-1.10.0-dos” directory from the archive and then rename it to FBC.

You should have

.\FBC\\*.\*

.\FBC\FBC.EXE

in the root folder.

Copy the directory \FBC\\*.\* and all sub directories and files to your DOS drive C:\DEVEL\FBC\\*.\*

Restart FreeDOS and navigate to C:\DEVEL\FBC and check the version number.

FBC[.EXE] -version

```
C:\>cd \devel\fbc
C:\DEVEL\FBC>fbc -version
FreeBASIC Compiler - Version 1.10.0 (2023-05-14), built for dos (32bit)
Copyright (C) 2004-2023 The FreeBASIC development team.
standalone
C:\DEVEL\FBC>
```

The manual install has completed.

Note if you use this version in FreeDOS you will encounter errors due to the 8.3 file name convention. You will need to enable Long File Name (LFN) support which I have not recommended in this guide.

### Long File Names

FreeBASIC is constructed in such a way that it is designed to cross compile under the Windows 9x DOS subsystem with long file names. When using FreeBASIC under DOS without LNF enabled you will encounter some issues with the 8.3 file naming convention. In some instances you may need to modify the header and library names in the #include "header.h" and the LIBrary names used by the linker to match the 8.3 names.

### Start-up batch file

If you wish to use FBC.EXE from the command line you can use the following batch file instead of setting the system paths and environment variables in the FCONFIG.SYS or AUTOEXEC.BAT. I don't recommend using CONFIG.SYS or AUTOEXEC in Windows 95. Place any additional environment variables in the Batch file used to launch the application. Place the FBCCMD.BAT in the C:\DOSLINKS directory. You can also create the PIF file as a shortcut on the desktop or in the ALaunch menu directory. Set the "initial environment" to 4096 in the PIF "Properties".

Remember to type "EXIT" followed by Enter to leave the child console screen and return to the parent context.

#### **FBCCMD.BAT (For command line compiling.)**

```
@ECHO OFF
CLS
REM Set the DOS path environments (required for command.com)
SET PATH=%path%;C:\WINDOWS;C:\WINDOWS\COMMAND;C:\DOSLINKS
```

```
REM For DOS-MODE only.  
REM CTMOUSE /R55  
set PATH=%path%;C:\DEVEL\FBC  
REM Optional paths:  
set PATH=%path%;C:\DEVEL\FBC\BIN;C:\DEVEL\FBC\INCLUDE;C:\DEVEL\FBC\LIB\ DOS  
  
REM Change to the FBC directory  
CD \DEVEL\FBC  
REM Launch a new instance of the command console using the temp paths.  
command.com  
CLS
```

A basic example of compiling a source file to an executable from the command line.

```
FBC.EXE -v -w ALL -exx C:\DEVEL\FBC\PROJECT\HELLO.BAS  
C:\DEVEL\FBC\PROJECT\HELLO.EXE
```

### FED IDE setup

Folding text EDitor (FED) works well with FreeBASIC. Like the example shown for the DJGPP setup it requires a couple of batch files to facilitate the compiling and running of the application source code. I recommend using a separate copy of FED in the FreeBASIC directory to keep the configuration and batch files separated from other programming environments.

If you already have a copy of FED set up for DJGPP you can copy and paste the FED directory to the root directory of .\FBC\ and modify the batch files to suite the FBC compiler calls. Alternatively you can use a fresh copy of FED and set up the editor configs and create the 3 batch files.

I am using the BTTR Software version FED v2.24b “fed224b.zip”.

<https://www.btrr-software.de/products/fed/>

Install FED to C:\DEVEL\FBC\FED224B\\*.\*

Create a batch file to launch FED in C:\DOSLINKS\FBFED.BAT. You can also create the PIF file as a shortcut on the desktop or in the ALaunch menu directory. Set the “initial environment” to 4096 in the PIF “Properties”.

The PATH is so the system can find FBC.EXE, STINGS.COM and the compiler batch files. Unlike the FreeDOS version I have also included the Windows and DOS Comspec paths.

### FBFED.BAT (Or any other appropriate name)

```
@ECHO OFF  
REM Set the DOS path environments (required for command.com)  
SET PATH=%path%;C:\WINDOWS;C:\WINDOWS\COMMAND;C:\DOSLINKS  
REM For DOS-MODE only.  
REM CTMOUSE /R55  
REM System launch for FED224 for FBC  
  
set PATH=%path%;C:\DEVEL\FBC
```

```
CD \DEVEL\FBC
CALL C:\DEVEL\FBC\FED224B\FED.EXE

CLS
```

You will need an extra command line application names “STRINGS.COM” for one of the batch file tasks.

Go to the following link and download “strings.zip” or “string25.zip” from the section “STRINGS — Enhanced SET.”

<https://www.btr-software.de/freesoftware/batch1.htm>

DOS and FreeDOS does not have a built in string manipulation library as is found in later version of Windows command console CMD.EXE so we need to use an external command line tool for this. STRINGS.COM can be difficult to find on the internet so I will place a copy of this with the source code for this guide.

<http://www.pl.exim.org/packages/coast/msdos/batutil/> “string25.zip”

<http://www.manmrk.net/tutorials/batch/index.htm> “string25.zip”

<http://www.lanet.lv/simtel.net/msdos/batchutl-pre.html> “string25.zip”

You will need to have STRINGS.COM placed either in the system path

C:\FREEDOS\BIN\STRINGS.COM or in the directory with the batch files for running commands on the DJ GCC compiler.

Copy “STRINGS.COM” to the FBC root directory if it is not in the system path

C:\FREEDOS\BIN\STRINGS.COM

C:\DEVEL\FBC\STRINGS.COM

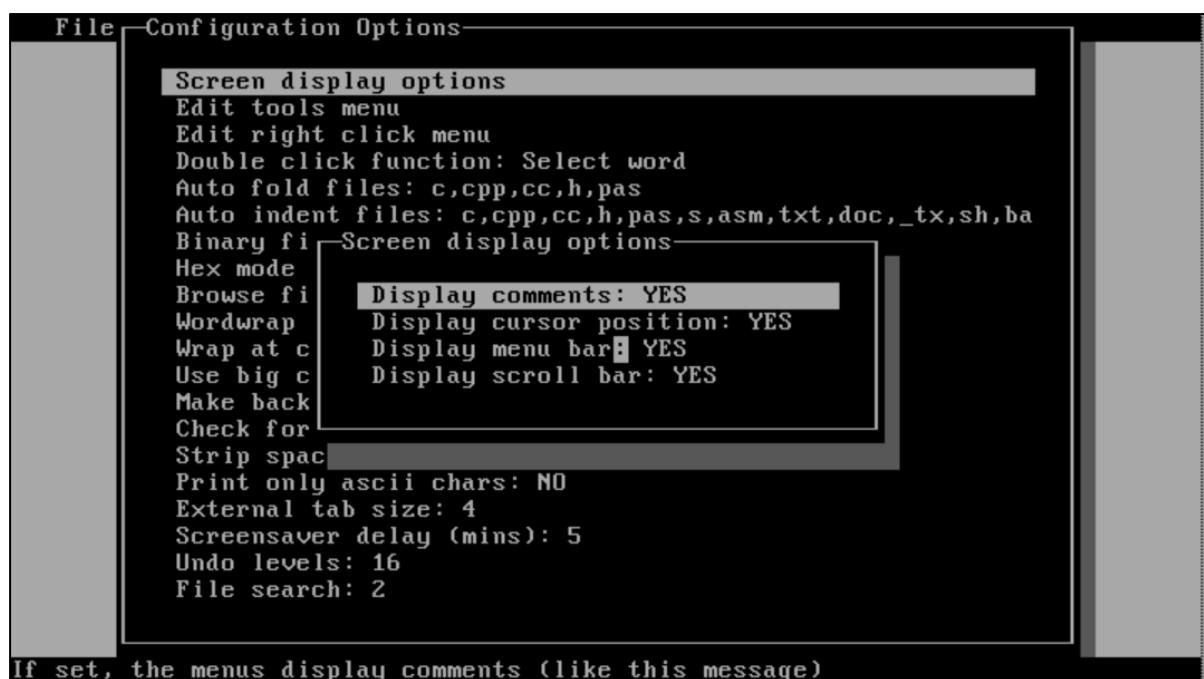
DOS does not have string manipulation tools for use in the command line or batch files, so STRINGS.COM fills this gap.

Open FED224 using the batch file in .\DOSLINKS or from the command line with FBFED.BAT

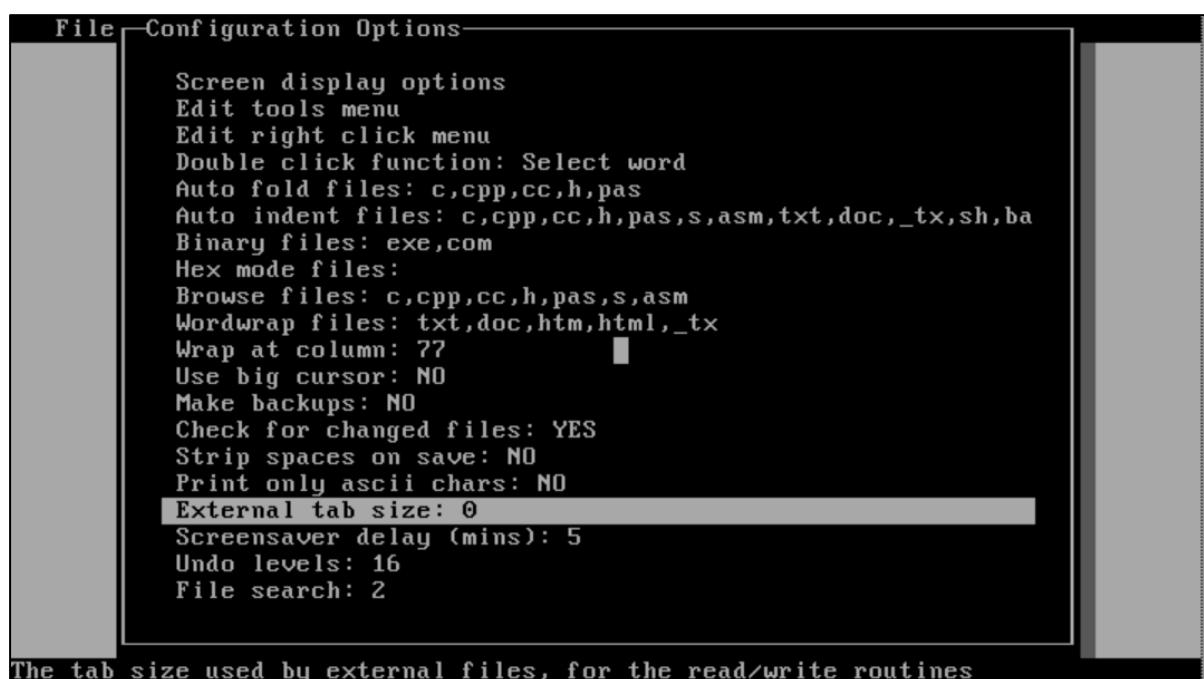
In Windows 95 Launch FED from the desktop PIF shortcut.

You will need to hold down the ALT key to show the menu if you have not changed the default settings.

Select Alt + C to open the Config menu and select Options. You will have a long list of different configuration options for the way the editor behaves. I suggest changing the “Screen display option” to something suitable. I typically change all to YES to add in the TUI navigation.



Next turn off (NO) Strip spaces on save, and set External tab size to 0. The external tab size other than zero will alter the indents when opening other source documents.



You can spend time trying other options if you wish, but these will get you at a point where you can start coding.

Note that the option "Auto fold files: c, cpp etc. will open the file with all blocks folded for these file extensions (Alt + Misc). If you prefer to open the source code with all lines fully expanded, remove the file extensions from the config.

Next in Options -> Config change the “Tab size” to 4(std).



Don't forget to “Save Config” before exiting FED.

To set up FED to compile and run source code we will need to add some custom “Tools” entries as well as creating a couple of batch files. There are many different modifications to the way this can be done including additional batch file for different purposes, but I am just going to show the Compile and Run batch files to keep it simple.

The following 2 batch files are created to work in the current working directory or system path. If you wish to use them from an alternative location you will need to adjust some of the path parameters for both the COMP and RUN batch files as well as the call to the batch files from FED.

The 2 batch files perform similar tasks. They take the source code file name from FED %f (including the fully qualified path), truncate the file extension ‘BAS’ and add ‘EXE’ in its place. HELLO.BAS becomes HELLO.EXE

If an optional file name is input at the FED tools menu then the source code file name is replaced with the optional file name.

Please note: If using these 2 batch files for source that has a different file extension be source to change the token lengths to match \$LEFT. For example .C is 1 and .BAS will require a value of 3. I should get rid of the magic number and use a configurable variable at the top of the page for readability.

In COMP.BAT both the source code path and file name, as well as the modified path and output file name are used at line 73 to invoke the compiler. You can alter the compiler and linker arguments here to create alternative versions of the batch file; for example debug compile, or release compile.

CALL FBC[.EXE] -v -w all -exx %cfile% -x %xfile% (FBC.exe verbose, Warn ALL, source file, compile to object, output to exe name).

In the RUN.BAT this line is used to run the compiled exe using the modified file name sent from FED.

Create or copy the following 2 batch files to the FreeBASIC root; C:\DEVEL\FBC\\*.\*

### COMP.BAT

```
@echo OFF
REM This is a generic command line runner for FED and FreeBASIC_DOS.
REM With small modification the compiler command line call can be used with
REM FED for other compilers or languages.
REM Change "FBC.EXE -v -w all -exx %cfile% -x %xfile%" switches to suite your
REM project requirements.
REM For Script engines you will need to use:
REM Interpreter.exe/com -[switches] %cfile%
REM (Other compilers) Change str=EXE to EXE/COM etc.
REM For Script engines you will need to
REM SEE: Accompanying RUN.BAT for FED
REM Requires STRINGS.COM (Version 2.5) Copyright (c) 1991, 1992 Douglas Boling
REM ftp.sunet.se/pub/simtelnet/msdos/batchutl/string25.zip
set cfile=%1
set efile=%2
REM If %2 (%p) is empty, use source file name.
IF [%2]==[] goto :SOURCENAME
REM Else use custom file name from %2 (%p).
IF NOT [%2]==[] goto :CUSTNAME

goto :ERROR

:SOURCENAME
REM Use %f path\source.ext and change to path\%1.exe
STRINGS plen =LENGTH %cfile%
REM .BAS = 3, .C = 1
STRINGS plen =SUB %plen%, 3
STRINGS xfile =LEFT %cfile%, %plen%

REM Add (concatenate) EXE to the path\Name._____
set str=EXE
set xfile=%xfile%%str%

REM FBC parameters
REM call FBC.EXE -v -w all -exx %cfile% -x %xfile%
call FBC -v -w all -exx %cfile% -x %xfile%
goto :END

:CUSTNAME
REM Build our custom path\filename.EXE/COM
REM ECHO Use %f path\source.ext and change to path\%2
REM Token index number starts at 1
Set index=1
REM Set delimiter "\"
set character=\

:LOOP
REM Loop through each token with "\" delimiter
STRINGS token = PARSE %cfile%, %index%, %character%
```

```

REM Look ahead to see if the next token is empty string
STRINGS lkahdidx = ADD %index%, 1
STRINGS lkahead = PARSE %cfile%, %lkahdidx%, %character%
IF [%lkahead%] == [] goto :FINISH

set xfile=%xfile%%token%%character%
REM Increment the index number
STRINGS index = ADD %index%, 1

REM Safety stop catch endless loop on error.
IF %index% == 10 goto :ERROR

goto :LOOP

:FINISH
REM Add file name.exe/com
REM Build the path\filename.EXE
set xfile=%xfile%efile%

REM Call FBC with custom file name.EXE/COM
REM call FBC.EXE -v -w all -exx %cfile% -x %xfile%

call FBC -v -w all -exx %cfile% -x %xfile%
goto :END

:ERROR
ECHO Unknown Error!
goto :END

:END
REM Release the temp variables
set cfile=
set efile=
set plen=
set str=
set xfile=

REM You could place a PAUSE here followed by a CLS to keep the console
REM window open to view the output of the compiler.
REM FED already adds a pause after invoking the command line.
REM PAUSE
REM CLS

```

#### RUN.BAT

```

@echo OFF
REM This is a generic command line runner for FED and FreeBASIC_DOS.
REM With small modification the compiler command line call can be used with
REM FED for other compilers or languages.
REM (Other compilers) Change str=EXE to EXE/COM etc.
REM SEE: Accompanying COMP.BAT for FED
REM Requires STRINGS.COM (Version 2.5) Copyright (c) 1991, 1992 Douglas Boling
REM ftp.sunet.se/pub/simtelnet/msdos/batchutl/string25.zip
set cfile=%1
set efile=%2
REM If %2 (%p) is empty, use source file name.
IF [%2]==[] goto :SOURCENAME

```

```
REM Else use custom file name from %2 (%p).
IF NOT [%2]==[] goto :CUSTNAME

goto :ERROR

:SOURCENAME
REM Use %f path\source.ext and change to path\%1.exe
STRINGS plen =LENGTH %cfile%
REM .BAS = 3, .C = 1
STRINGS plen =SUB %plen%, 3
STRINGS xfile =LEFT %cfile%, %plen%

REM Add (concatenate) EXE/COM to the path\Name._____
set str=EXE
set xfile=%xfile%%str%

REM Run the executable
call %xfile%
goto :END

:CUSTNAME
REM Build our custom path\filename.EXE/COM
REM ECHO Use %f path\source.ext and change to path\%2
REM Token index number starts at 1
Set index=1
REM Set delimiter "\"
set character=\

:LOOP
REM Loop through each token with "\" delimiter
STRINGS token = PARSE %cfile%, %index%, %character%

REM Look ahead to see if the next token is empty string
STRINGS lkahdidx = ADD %index%, 1
STRINGS lkahead = PARSE %cfile%, %lkahdidx%, %character%
IF [%lkahead%] == [] goto :FINISH

set xfile=%xfile%%token%&character%
REM Increment the index number
STRINGS index = ADD %index%, 1

REM Safety stop catch endless loop on error.
IF %index% == 10 goto :ERROR

goto :LOOP

:FINISH
REM Add file name.exe/com
REM Build the path\filename.EXE
set xfile=%xfile%%efile%

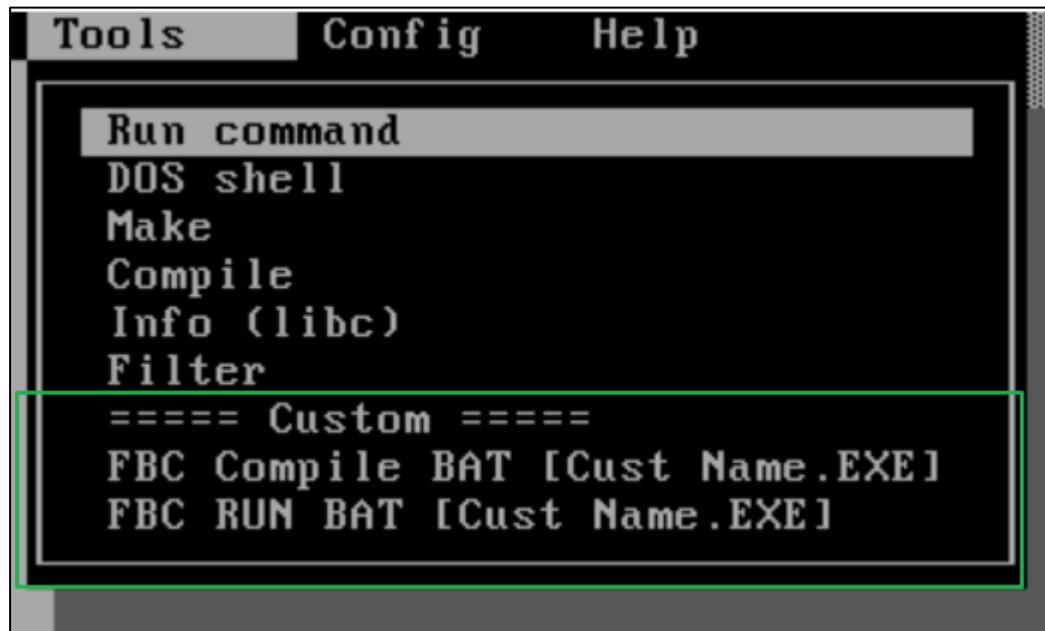
REM Call the custom EXE NAME
call %xfile%
goto :END

:ERROR
ECHO Unknown Error!
goto :END
```

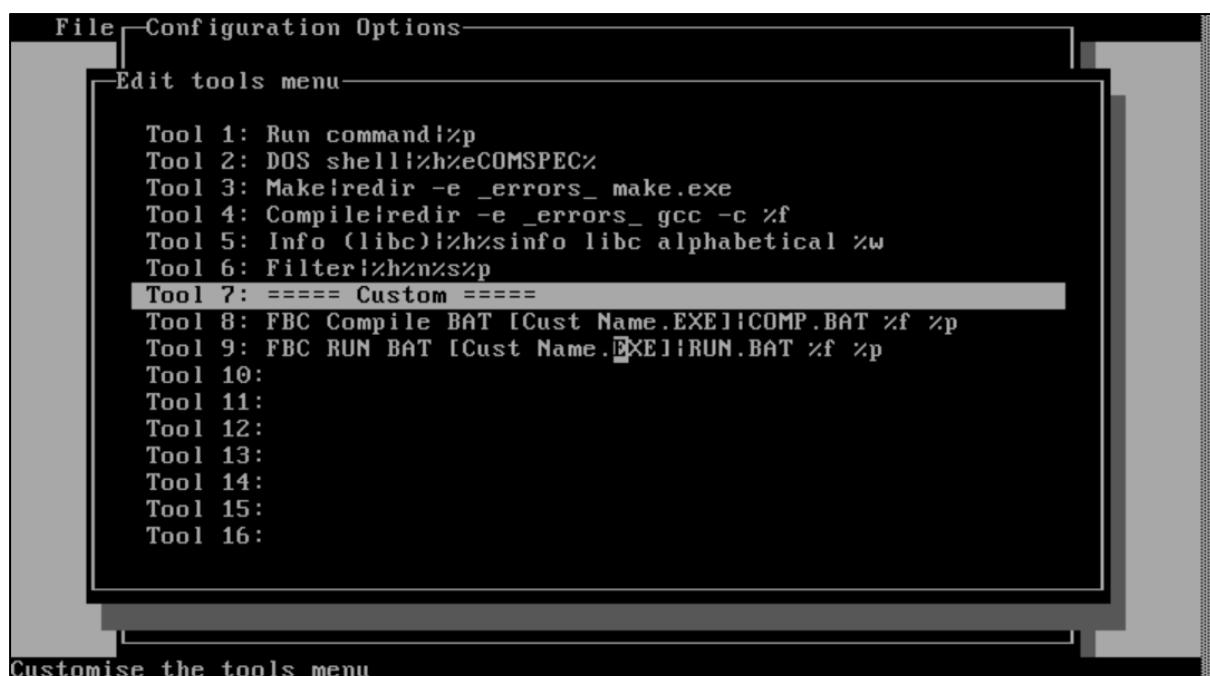
```
:END
REM Release the temp variables
set cfile=
set efile=
set plen=
set str=
set xfile=

REM You could place a PAUSE here followed by a CLS to keep the console
REM window open to view the output of the compiler.
REM FED already adds a pause after invoking the command line.
REM PAUSE
REM CLS
```

Once the 2 batch files are in place, open FED and add the following to the tools menu so we can call the 2 batch files from FED. If you copied the FED224B from the DJGPP directory, just modify the lines.



Open from the menu “Config ->Options -> Edit tools menu” and add the following 2 lines.



Tool6===== Custom =====

Tool7=FBC Compile BAT [Cust Name.EXE]|COMP.BAT %f %p

Tool8=FBC RUN BAT [Cust Name.EXE]|RUN.BAT %f %p

The first part before '!' is the tool description. The section after the '!' is the command line argument **COMP.BAT SourceName UserInput** where f% is the internal FED variable containing the full qualified path and source file name in the editor window and p% is a variable to take a custom output name from the user. If p% is left blank the batch file routines ignore the empty p% variable and use the f%. If p% is given a value then the batch file will substitute the custom name in place of the original source name and the file name.ext component of f% is ignored.

To remove a line just leave it blank.

Make sure you "Save config" before closing FED.



Hint you can copy the FED.CFG to other instances of FED to keep your FED customizations.

The FED website also makes available some alternative pre made syntax highlighting schemes FED.SYN. It is a good idea to keep a backup of your FED configs in case of mistakes or for recovering you settings.

Open FED for your FBC environment and create a “Hello World!” HELLO.BAS source file.

The screenshot shows two windows of the FED application. The top window is titled "File Edit Search Misc Tools Config Help". A dropdown menu is open under "File" with the option "Open (ctrl+enter for binary mode)" highlighted. Below the menu, the path "c:\devel\fbc\proj\hello.bas" is shown. The main text area of the window displays the command "c:\devel\fbc\proj\hello.bas". The bottom window is titled "File Edit Search Misc Tools Config Help" and contains the REMarks section of a FreeBASIC program:

```
REM FreeBASIC DOS - Hello world

Declare Function main_procedure() As Integer
main_procedure()

Function main_procedure() As Integer
    Shell "CLS"
    Print "Hello world!"
    Sleep
    Return 0
End Function
```

At the bottom of the bottom window, the status bar shows the path "- -d c:\devel\fbc\proj\hello.bas - line 1 - col 1 - 0x52 (82)".

From the menu open “Tools -> ...” and select your menu entry to COMPILE the source code followed by you menu entry to RUN the source code. I would also test both menu entries using the custom

name for the output file. You will find that if you compile with a custom name the variable p% will be auto filled when using the Run option.

File Edit Search Misc Tools Config Help

REM FreeBASIC DOS - Hello world

```
Declare Function main_procedure() As Integer
main_procedure()
Function main_procedure() As Integer
    Shell "CLS"
    Print "Hello world!"
    Sleep
    Return 0
End Function
```

Run command  
DOS shell  
Make  
Compile  
Info (libc)  
Filter  
===== Custom =====  
FBC Compile BAT [Cust Name.EXE]  
FBC RUN BAT [Cust Name.EXE]

Run external tool

File Edit Search Misc Tools Config Help

REM FreeBASIC DOS - Hello world

```
Declare Function main_procedure() As Integer
main_procedure()
Function main_procedure() As Integer
    Shell "CLS"
    Print "Hello world!"
    Sleep
End Function
```

S FBC Compile BAT [Cust Name.EXE]—  
R

Run external tool

```
Run external tool
COMP.BAT c:\devel\fbc\proj\hello.bas
FreeBASIC Compiler - Version 1.09.0 (2021-12-31), built for dos (32bit)
Copyright (C) 2004-2021 The FreeBASIC development team.
standalone
target:      dos, 486, 32bit
backend:     gas
compiling:   c:\devel\fbc\proj\hello.bas -o c:\devel\fbc\proj\hello.asm (main module)
assembling: C:\devel\fbc\bin\dos\nasm.exe --32 --strip-local-absolute "c:\devel\fbc\proj\hello.asm" -o "c:\devel\fbc\proj\hello.o"
ld options in 'c:\devel\fbc\proj\ldopt.tmp':           -o "c:\devel\fbc\proj\hello.EXE" -T "C:\devel\fbc\lib\dos\i386go32.x" -s -L "C:\devel\fbc\lib\dos" -L "."
" C:\devel\fbc\lib\dos\crt0.o" "C:\devel\fbc\lib\dos\fbrt0.o" "c:\devel\fbc\proj\hello.o" "-(" -lfb -lgcc -lc -lm "-)"
linking:     C:\devel\fbc\bin\ld.exe @c:\devel\fbc\proj\ldopt.tmp

<press a key>_
```

Now select run the output executable.



The screenshot shows a window titled "FreeBASIC DOS - Hello world". The menu bar includes File, Edit, Search, Misc, Tools, Config, and Help. The main area contains the following FreeBASIC code:

```
REM FreeBASIC DOS - Hello world

Declare Function main_procedure() As Integer

main_procedure()

Function main_procedure() As Integer
    Shell "CLS"
    Print "Hello world!"
    S-FBC RUN BAT [Cust Name.EXE]
    R
End F
```

Below the code, there is a black rectangular box containing the command "S-FBC RUN BAT [Cust Name.EXE]". The status bar at the bottom left says "Run external tool".

The output window below shows the program's execution:

```
Hello world!
-
```

Using the alternative output name (Make sure to add the extension “WORLD.EXE”)...

The screenshot shows a window titled "REM FreeBASIC DOS - Hello world". The menu bar includes File, Edit, Search, Misc, Tools, Config, and Help. The main area contains the following FreeBASIC code:

```
Declare Function main_procedure() As Integer
main_procedure()

Function main_procedure() As Integer
    Shell "CLS"
    Print "Hello world!"
    S-GCC Compile BAT [Cust Name.EXE]-
R
End F      WORLD.EXE_
```

Below the code, a message box displays the command: "GCC Compile BAT [Cust Name.EXE]". The window has a "Run external tool" button at the bottom.

The screenshot shows a terminal window with the following text:

```
Run external tool
COMP.BAT c:\devel\fbc\proj\hello.bas WORLD.EXE
FreeBASIC Compiler - Version 1.09.0 (2021-12-31), built for dos (32bit)
Copyright (C) 2004-2021 The FreeBASIC development team.

standalone
target:      dos, 486, 32bit
backend:     gas
compiling:   c:\devel\fbc\proj\hello.bas -o c:\devel\fbc\proj\hello.asm (main module)
assembling:  C:\devel\fbc\bin\dos\nasm.exe --32 --strip-local-absolute "c:\devel\fbc\proj\hello.asm" -o "c:\devel\fbc\proj\hello.o"
ld options in 'c:\devel\fbc\proj\ldopt.tmp':           -o "c:\devel\fbc\proj\WORLD.EXE" -T "C:\devel\fbc\lib\dos\i386go32.x" -s -L "C:\devel\fbc\lib\dos" -L "."
" "C:\devel\fbc\lib\dos\crt0.o" "C:\devel\fbc\lib\dos\fbrt0.o" "c:\devel\fbc\proj\hello.o" "-(" -lfb -lgcc -lc -lm "-)"
linking:      C:\devel\fbc\bin\dos\ld.exe @c:\devel\fbc\proj\ldopt.tmp

<press a key>
```

When we “run” the compiled exe the custom name already exists...

The screenshot shows a window titled "REM FreeBASIC DOS - Hello world". The menu bar includes File, Edit, Search, Misc, Tools, Config, and Help. The code area contains:

```
File      Edit      Search      Misc      Tools      Config      Help
REM FreeBASIC DOS - Hello world

Declare Function main_procedure() As Integer
main_procedure()

Function main_procedure() As Integer
    Shell "CLS"
    Print "Hello world!"
    S-FBC RUN BAT [Cust Name.EXE]-
R
End F      WORLD.EXE
```

The output window below shows the result of running the program:

```
Hello world!
```

If you had any problems with compiling check though the GCC output for hints and recheck the 2 batch files for possible syntax errors using debug prints.

```
ECHO Debug Pause
ECHO %1
ECHO %cfile%
ECHO %efile%
REM etc.
PAUSE
```

There are many valid ways to set up your development environment. The above is just one method to get you started creating with FreeBASIC using FED as your IDE. Take care on large projects as you may need to invoke more memory in your FDCONFIG.SYS or FDAUTO.BAT file. Take note of !BUFFERS=, !FILES=, !STACKS=, !FCBS= etc. NOTE: The ‘!’ means that it is “forced or mandatory” to be loaded. In Windows 95 These values are set in your launching batch file and PIF file.

When including libraries remember that you may still have to add additional paths and linker options in the batch when using specific libraries. Follow the instructions from the library as well as the manual for GCC.

If you are using ALaunch then copy your shortcut PIFs for FED and FBC to an appropriate directory for the menu.

### FreeBASIC additional

#### **Some additional batch files for profiling and debugging.**

Create the EXE with profile information.

```
CALL FBC.EXE -g -profile -v -w all -exx %cfile% -x %xfile%
```

Run the EXE through the profiler and output a report.

```
CALL .\BIN\ DOS\GPROF.EXE -b -p %xfile% > fprofile.txt
```

Run the executable in the debugger.

```
CALL .\BIN\ DOS\GDB.EXE %xfile%
```

You can use the generic COMP.BAT file with the modified call to the executable above to create different profiles for profiling and debugging. Give the batch file a related name such as FBCDBG.BAT and then create an additional entry into FED Tools menu for each profile.

### Using libraries.

FreeBASIC comes with the header files “header.bi” for many common libraries but may not have the LIB binaries due to the size of including every library in the default install. In most cases you can go to the DJGPP repository and download a copy of the precompiled library.

For example if you go to the DJGPP repository and look under Tool Kits we can find the Public domain curses library “pdcur39a.zip”. ‘a’ stands for library “Archive” file or just LIB file.

<https://www.delorie.com/pub/djgpp/current/v2tk/>

In the directory pdcur39a\lib you will find the two library files that match the header.bi files found in FreeBASIC.

libpanel.a

libpdcurses.a

Copy these 2 files to your DOS FreeBASIC LIB directory C:\DEVEL\FBC\LIB

Take care with the 8.3 naming conventions and check the names between the header file .bi and the library archive name. You can remove the LIB prefix from libpdcurse.a to make pdcurse.a, (Not recommended) or change the include line in the .\FBC\INC\PDCURSES.BI from

```
#inlib "pdcurse" to
```

```
#inlib "pdc~1"
```

"pdc~1" will find the 8.3 named library file .\FBC\LIB\DOS\LIBPDC~1.A (libpdcurse.a)

**Be aware of this 8.3 naming issue with any library you bring into the DOS environment. If you use the FBC V1.10.0 from the FreeBASIC repository you may encounter long file names as well.**

You can also compile the LIB archive files yourself from source using the DOS DJGPP compiler. Most libraries will have the instructions and premade make files to compile the DOS binaries.

---

### DOjS (+ Jsh)

DOjS is a JavaScript-able canvas with WAV and MIDI sound support. DOjS runs in Dosbox and on real hardware or a virtual machine with MS-DOS, FreeDOS or any DOS based Windows like Windows 95/98/ME. If you run it on real hardware you need at least a 80386 with 4MB. I recommend a Pentium class machine (>= 100MHz) with at least 32MB RAM.

Note: Midi hardware is not available in VirtualBox.

<https://github.com/SuperIlu/DOjS>

<https://www.ibiblio.org/pub/micro/pc-stuff/freedos/files/repositories/1.3/pkg-html/dojs.html>

Note: DOjS will only run on a x386 or higher and is a 32-bit application.

DOjS is a work in progress and evolving so be sure to check the GitHub page for bugs and updates.

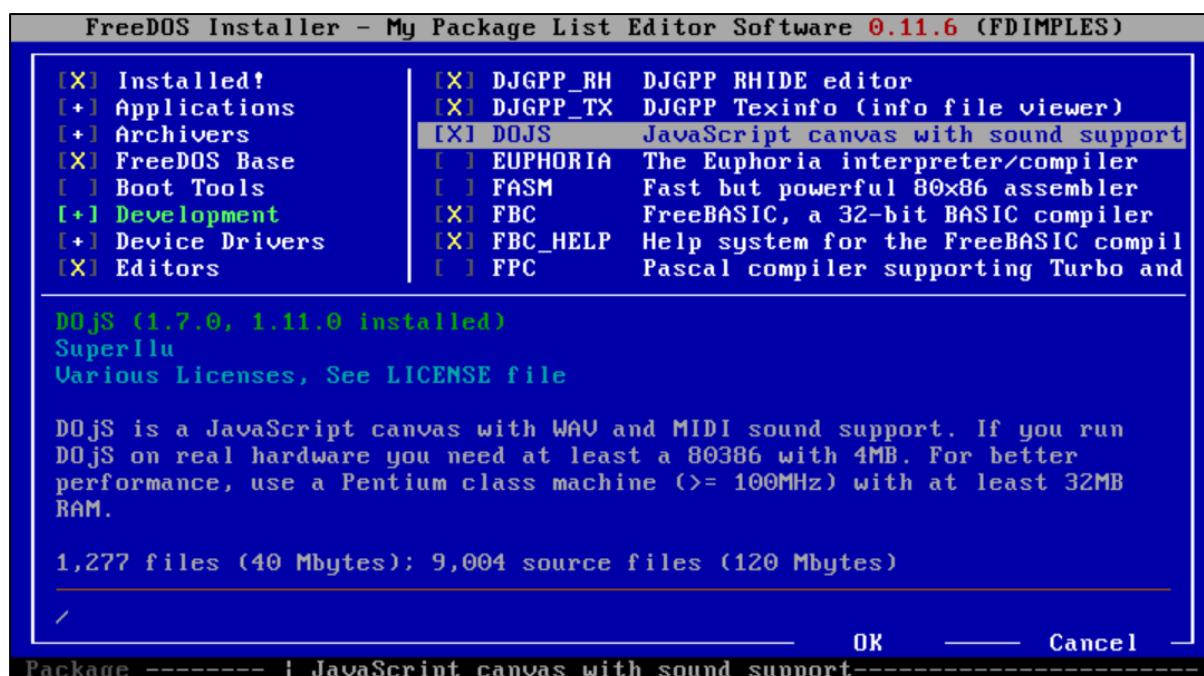
### Install

DOjS can be installed from the FreeDOS bonus CD via FDIMPLES, or via FDNPKG.

For Windows 95 you can copy the C:\DEVEL\DOJS directory from FreeDOS to the Windows 95 drive. Alternatively you can reconstruct the directory structure from the GitHub files.

Insert the Bonus CD and run FDIMPLES.

Select DOjS from the Development category.



or

If updating a currently installed version of DOJS:

FDNPKG update DOJS

```
est/drivers/...
Loading http://www.ibiblio.org/pub/micro/pc-stuff/freedos/files/repositories/latest/edit/...
Loading http://www.ibiblio.org/pub/micro/pc-stuff/freedos/files/repositories/latest/emulator/...
Loading http://www.ibiblio.org/pub/micro/pc-stuff/freedos/files/repositories/latest/games/...
Loading http://www.ibiblio.org/pub/micro/pc-stuff/freedos/files/repositories/latest/gui/...
Loading http://www.ibiblio.org/pub/micro/pc-stuff/freedos/files/repositories/latest/net/...
Loading http://www.ibiblio.org/pub/micro/pc-stuff/freedos/files/repositories/latest/sound/...
Loading http://www.ibiblio.org/pub/micro/pc-stuff/freedos/files/repositories/latest/unix/...
Loading http://www.ibiblio.org/pub/micro/pc-stuff/freedos/files/repositories/latest/util/...

dojs - A DOS JavaScript Canvas with sound
C:\>fdnpkg update dojs
Package database loaded from local cache.

Downloading package http://www.ibiblio.org/pub/micro/pc-stuff/freedos/files/repositories/latest/devel/dojs.zip...
Downloading dojs.zip... 16326644 bytes [31%]
```

or for a first clean install:

FDNPKG install DOJS

```
C:\>FDNPKG install DOJS
Package database loaded from local cache.
Downloading package http://www.ibiblio.org/pub/micro/pc-stuff/freedos/files/repositories/latest/devel/dojs.zip...
Downloading dojs.zip... 4270284 bytes [8%]
```

Or alternatively you can download DOjS from the GitHub home page and do a manual install.

Once it is installed you can use the built in editor to confirm DOjS is working by the following command line. Note that you will need to supply a JS script file as an argument to DOJS.EXE.

Navigate to C:\DEVEL\DOJS

```
DOJS.EXE C:\DEVEL\DOJS\EXAMPLES\DEMO.JS
```

This will open the source in the DOjS built in editor.

```
DOJS V1.11    C:\DEVEL\DOJS\EXAMPLES\DEMO.JS          0001/0001

function Setup() {
    ClearScreen(EGA.BLACK);
    SetFramerate(30);

    Xmax = SizeX();
    Ymax = SizeY();
    XmaxHalf = Xmax / 2;
    stepX = Xmax / 128;
    stepY = Ymax / 128;

    x = 0;
    y = Ymax;

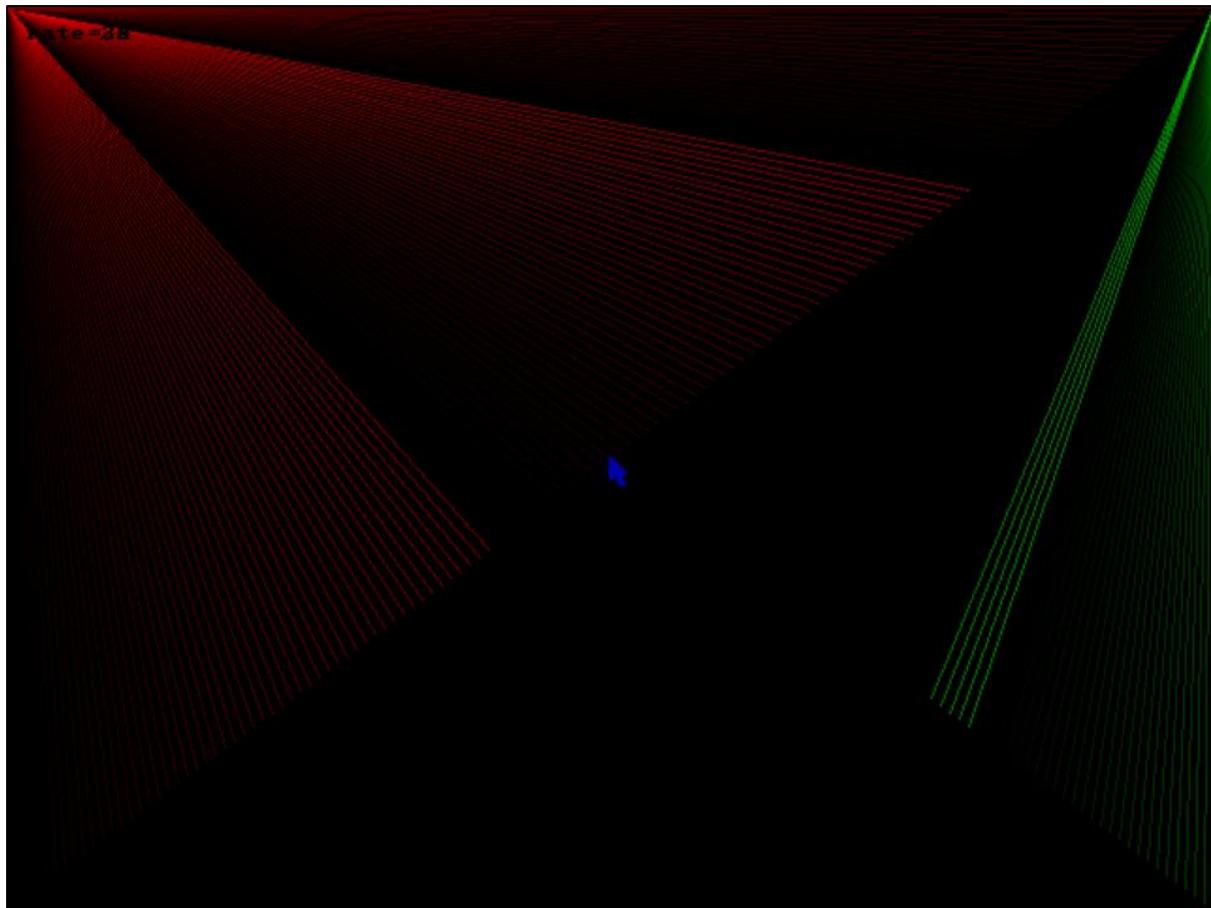
    state = 1;
    fontCount = 0;

    fnt = new Font(JSBOOTPATH + "fonts/luct38.fnt");
    fntW = fnt.maxwidth;
    fntH = fnt.height;

    Println("stepX=" + stepX + ", stepY=" + stepY);
}

1Help  2      3Save  4Run  5      6      7Find  8      9Log  10Exit
```

Next press F4 to run the application.



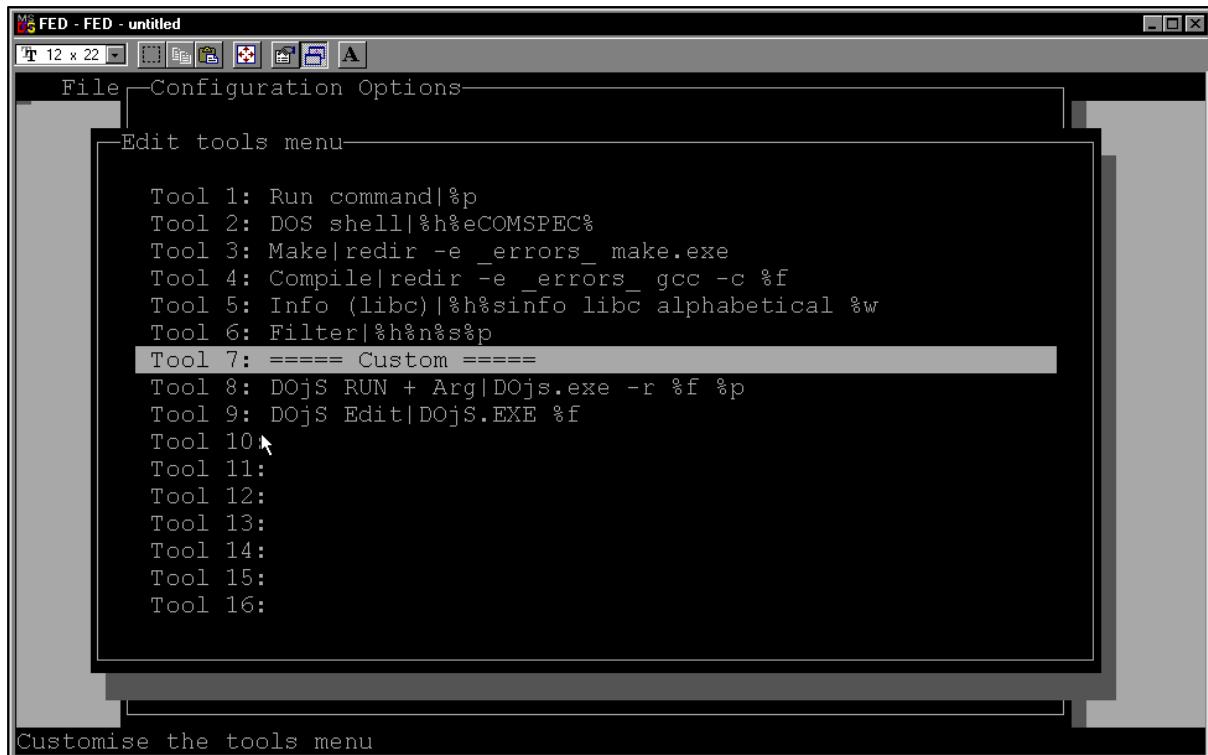
Take careful note of the placement of your source code as it needs to be in or below the root directory. You will need to keep the directory structure of your source related to the directory of DOJS.

If you find the built in editor a little cumbersome you can also use FED as your IDE. Edit the Tools menu of FED to run “DOJS.EXE –r %f %p”. %p is for if you need to send additional arguments to your script at startup and will only be taken into account if it is not left blank.

DOJS RUN + Arg|DOJS.EXE – r %f %p

DOJS Edit|DOJS.EXE %f

Screen capture from Windows 95B.

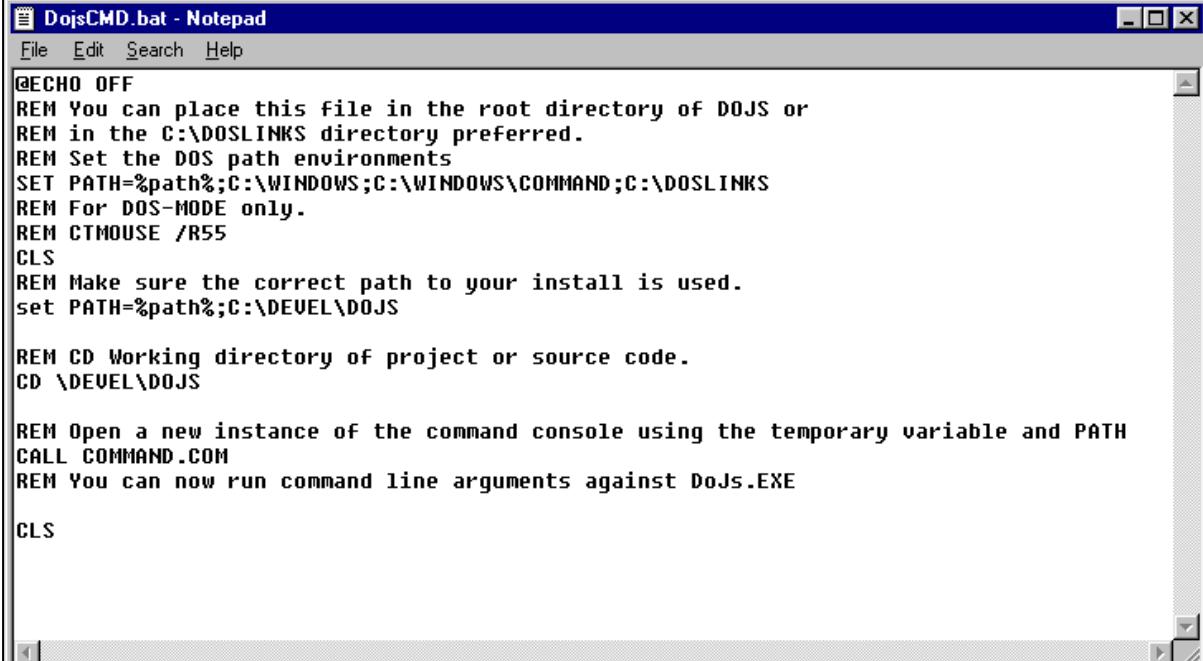


FED will not be able to reproduce the syntax highlighting of the built in editor.

You can now experiment with JavaScript in DOS.

Use the previous guides to create the DOJS.BAT file for the .\DOSLINKS directory. Because you will need to supply the source name you must A. call the batch file followed by the source name, or include your source in the batch file or Windows PIF. Don't forget to add the %1 %2 %3 after DOJS.EXE in your batch files.

Hint: you can drop a source file on the Windows PIF or batch file if the batch file is set up correctly and the source is in the correct location. Don't forget to add the Windows (MS-DOS) paths to your batch file.



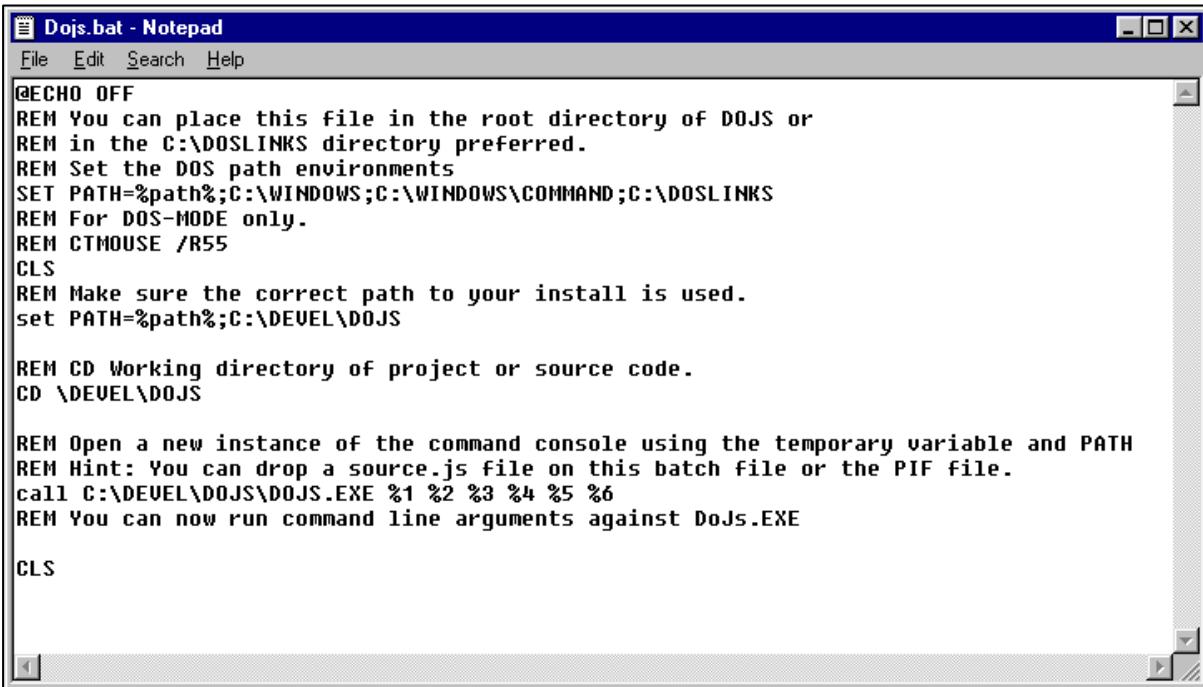
The screenshot shows a Microsoft Notepad window titled "DojsCMD.bat - Notepad". The window has a standard Windows 95-style title bar with "File", "Edit", "Search", and "Help" menu options. The main text area contains a batch file script:

```
@ECHO OFF
REM You can place this file in the root directory of DOJS or
REM in the C:\DOSLINKS directory preferred.
REM Set the DOS path environments
SET PATH=%path%;C:\WINDOWS;C:\WINDOWS\COMMAND;C:\DOSLINKS
REM For DOS-MODE only.
REM CTMOUSE /R55
CLS
REM Make sure the correct path to your install is used.
set PATH=%path%;C:\DEVEL\DOJS

REM CD Working directory of project or source code.
CD \DEVEL\DOJS

REM Open a new instance of the command console using the temporary variable and PATH
CALL COMMAND.COM
REM You can now run command line arguments against DoJs.EXE

CLS
```



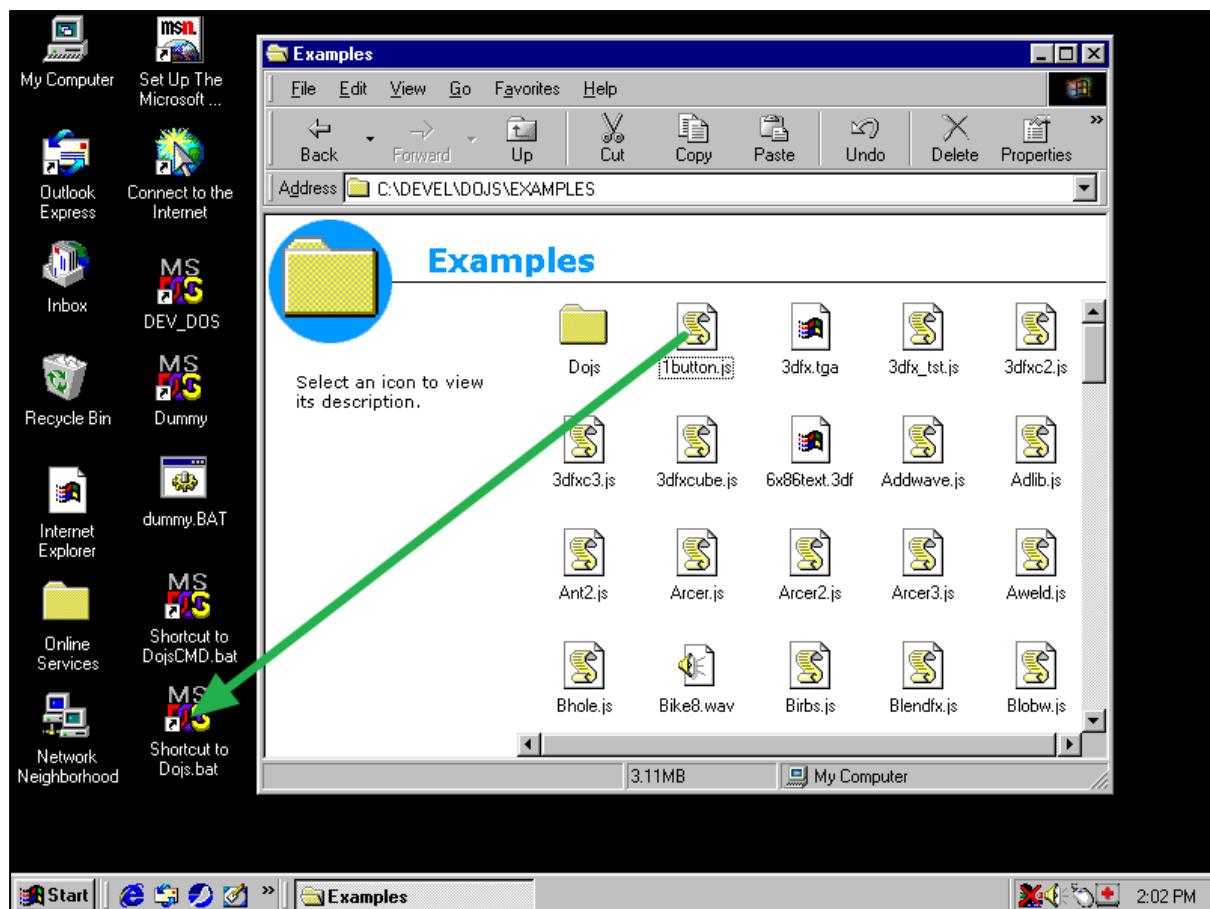
The screenshot shows a Microsoft Notepad window titled "Dojs.bat - Notepad". The window has a standard Windows 95-style title bar with "File", "Edit", "Search", and "Help" menu options. The main text area contains a batch file script, identical to the one in the previous window:

```
@ECHO OFF
REM You can place this file in the root directory of DOJS or
REM in the C:\DOSLINKS directory preferred.
REM Set the DOS path environments
SET PATH=%path%;C:\WINDOWS;C:\WINDOWS\COMMAND;C:\DOSLINKS
REM For DOS-MODE only.
REM CTMOUSE /R55
CLS
REM Make sure the correct path to your install is used.
set PATH=%path%;C:\DEVEL\DOJS

REM CD Working directory of project or source code.
CD \DEVEL\DOJS

REM Open a new instance of the command console using the temporary variable and PATH
REM Hint: You can drop a source.js file on this batch file or the PIF file.
call C:\DEVEL\DOJS\DOJS.EXE %1 %2 %3 %4 %5 %6
REM You can now run command line arguments against DoJs.EXE

CLS
```



## Win 32 Development environments

### GCC, MinGW Dev-C++(Bloodshed V401, Orwell V 511(x86)-GCC4.9.2)

This is the open source C/C++ compiler and IDE of the time (v4.01). The legacy V4.01 works well but has a few bugs in the IDE. You may need at least Windows 95C for the IDEs to run.

#### Bloodshed Legacy V4.01

<https://www.bloodshed.net/Dev-Cpp-4.0>

#### Bloodshed V 4.x to V5

<https://www.bloodshed.net/>

Orwell v5.11 will run on windows 95 but you will need to take care with the GCC and MinGW library versions.

#### Orwell V5 to ...

<https://sourceforge.net/projects/orwelldevcpp/files/Setup%20Releases/>

## Windows Scripting Host (WSH) V 5.6

MS CScript/wscript

This exists in most versions of windows and has similarities to its derivative VBA used in MS-Office.

Windows scripting host is an interpreted scripting system that supports JScript (\*.js/\*.jse) and VBScript (\*.vbs/\*.vbe). Often used with Windows HTML Applications (HTA) to create scripted GUI applications.

Note that MS no longer provides downloads for V5.6 or earlier. You will have to hunt around for the installer/update.

Windows 95 users must have OSR2 installed or be running Microsoft Internet Explorer 4.0 or later for these files to work properly. Windows 95 users without Microsoft Internet Explorer 4.0 or OSR2 will need to install DCOM before using these files. To install the DCOM components visit the DCOM Download page. More information on OSR2 can be found here.

Download DCOM95 for Windows 95, version 1.3

“DCOM95.exe”

V5.6 is said to be the last version to be able to run on Windows 95. Please note that the Windows 9x (716 KB) version is different from the Windows NT (665 KB) version even though they have the same file and installer name.

scr56en.exe (716 KB)

[https://web.archive.org/web/20060715000000\\*/https://download.microsoft.com/download/winscript56/Install/5.6/W9XNT4Me/EN-US/scr56en.exe](https://web.archive.org/web/20060715000000*/https://download.microsoft.com/download/winscript56/Install/5.6/W9XNT4Me/EN-US/scr56en.exe)

Documentation.

scrdoc56en.exe

<https://download.microsoft.com/download/winscript56/Install/5.6/W982KMeXP/EN-US/scrdoc56en.exe>

Other versions and tools.

[http://www.installsite.org/pages/en/tt\\_winup.htm](http://www.installsite.org/pages/en/tt_winup.htm)

<https://am.net/lib/tools/Microsoft/scripting/>

Autolt (Old Versions 3.2.12.x )

Windows GUI scripting engine.

<https://www.autoitscript.com/autoit3/files/archive/autoit/>

Microsoft Visual C++ 6.0 Standard Edition

Microsoft Visual CPP 6.0 Standard.7z

<https://winworldpc.com/product/visual-c/6x>

Microsoft Visual C++ 6.0 Standard Edition.zip

<https://archive.org/details/microsoft-visual-c-6.0-standard-edition>

Not to be confused with Microsoft Visual Studio 6.0.

Microsoft Visual Studio 6.0 is an IDE (integrated development environment) for Microsoft Windows that allows developers to develop a variety of applications. It is a bundle of separate programming products, including Microsoft Visual C++, Microsoft J++, Microsoft Visual Basic, and Foxpro

<https://winworldpc.com/product/microsoft-visual-stu/60>

Microsoft Visual Basic 6.0 (Enterprise)

<https://winworldpc.com/product/microsoft-visual-bas/60>

---

## Debuggers & Utilities

The following is a short list of some useful programming utilities. Some can be found as part of the FreeDOS installs and others in various places across the internet. This is by no means a comprehensive list and the DOS utilities available are almost endless.

At the end of the FreeDOS (DOS) application list there are additional applications for Windows 95.

FreeDOS Debug - Debug is an extended MS Debug clone.

DEBUG.COM, which can debug DOS 16-bit real-mode applications only and DEBUGX.COM, which additionally can debug protected-mode DPMI applications, both 16-bit and 32-bit.

DEBUG.COM is part of the FreeDOS base install.

<http://web.archive.org/web/20230610211605/https://www.japheth.de/debxxf.html>

Insight Debugger V1.24 (DOS)

FreeDOS base install

Insight is a very small debugger for analyzing real-mode DOS programs. It features an i80486 disassembler, an i8086 assembler, 'Trace into' and 'Step over' functions, simple breakpoint handling, extended code or data navigation, simple color-highlighting, and a nice menu-driven interface comparable to Borland's Turbo Debugger.

<http://www.bttr-software.de/products/insight/>

<https://www.ibiblio.org/pub/micro/pc-stuff/freedos/files/repositories/1.3/pkg-html/insight.html>

### **Debug (FreeDOS \BIN)**

Command line.

32 bit debugger.

### **Debugx (FreeDOS \BIN)**

Command line.

32 bit debugger eXtended.

### **386SWAT - A Protected Mode Debugger (DOS, Win32)**

Protected mode debugger.

Requires some setup tasks to work.

<http://www.sudleyplace.com/swat/>

See also:

displib command line

dispobj command line

### **UPX**

Universal executable packer.

<https://www.ibiblio.org/pub/micro/pc-stuff/freedos/files/repositories/1.3/pkg-html/upx.html>

### **FED 224b**

Folding text EDitor IDE

<http://www.bttr-software.de/products/fed/>

### **STRINGS.COM**

String manipulation tools for batch files.

<https://www.bttr-software.de/freesoft/batch1.htm>

### DOS HexED

Command line hex editor.

<http://www.dcee.net/Files/Utils/>

### UHEX

A simple and fast multiplatform hex editor.

<https://uhex.sourceforge.net/>

### Biew – Beye (Hex)

Biew is a binary/hexadecimal editor. Command line.

FreeDOS Bonus CD.

### HexCompare

Visual File Comparison Utility. Command line.

<https://sourceforge.net/projects/hexcompare/files/hexcompare%20v1.0.4/>

### VBEDIAG

vbediag is a diagnostic tool for the VESA Video BIOS Extensions.

<https://drv.nu/vbediag/>

### Video Test

386 video adapter id and test utility.

<https://www.elhvb.com/webhq/download/index.htm>

### MONITORS

This excellent 486 tool tests both monitor & adapter.

<https://www.elhvb.com/webhq/download/index.htm>

### NSSI

NSSI is an up-to-date, freeware, hardware information tool.

<https://www.bttr-software.de/freesoft/system.htm>

### Info Plus

Hardware information tool.

[https://archive.org/details/msdos\\_shareware\\_fb\\_INFO](https://archive.org/details/msdos_shareware_fb_INFO)

### HWInfo 16-bit 32-bit

Hardware information tool.

<https://www.hwinfo.com/download/>

### Hexit

TUI Text editor.

<https://mklasson.com/hexit.php>

<https://www.bttr-software.de/freesoft/filutil2.htm>

<https://www.sac.sk/search.php>

### HIEW

Hex editor.

Requires usage info.

<https://www.bttr-software.de/freesoft/filutil2.htm>

<https://www.sac.sk/files.php?d=17&l>

### Executing Inspector DBG

Real mode debugger.

<http://web.archive.org/web/20050205203014/http://phantom.urbis.net.il/inspector/>

### Mem (FreeDOS \BIN)

Display used and free memory.

### Memstat (FreeDOS \BIN)

Display memory.

**Mode (FreeDOS \BIN)**

Show current console mode.

**XMSTest (FreeDOS \BIN)**

Test XMS memory.

**XMSStat (FreeDOS \BIN)**

XMS memory stats.

**ListVESA**

ListVESA is a utility to report which VESA video modes are supported by the system's hardware. You can use command line options to tailor information listed to one specific mode, modes supporting a specific color bit depth, modes supporting linear frame buffer, or simply general information on the video adapter itself or a table summarizing available modes with no detailed screen data.

<http://www.ibiblio.org/pub/micro/pc-stuff/freedos/files/repositories/1.3/pkg-html/listvesa.html>

**FoxCalc**

Calculator with a TUI and mouse support.

<http://www.ibiblio.org/pub/micro/pc-stuff/freedos/files/repositories/1.3/pkg.html/foxcalc.html>

**RCal**

A big-numbers calculator with floating point that mimics the "paper rolling" calculators. 8086+

<http://www.ibiblio.org/pub/micro/pc-stuff/freedos/files/repositories/1.3/pkg-html/rcal.html>

**p7Zip**

Command line 7-Zip archive tools.

FreeDOS Bunus CD

**Windows 95 utilities:**

**System Monitors:**

### Process Viewer (PrcView)

View all running processes including memory and I/O information.

I recommend installing this application on any Windows 95 system.

See chapter “Essential Applications”.

### Windows System Monitor

Monitor Windows system resources.

Windows accessories. Add/Remove programs.

### Hex Editors:

“Yuri Software” Yuri Diomin” **HEdit 2.0.04** (V3.2 may work) Shareware

This can be a little difficult to find. Check any download for malware.

“hedit2.zip” (v2.0.04) 178 KB

<https://heninv.tripod.com/codedit.html>

<https://www.cdmediaworld.com/hardware/cdrom/files.shtml>

[http://cd.textfiles.com/pslv4nv08/WIN/UT\\_DSKFI/](http://cd.textfiles.com/pslv4nv08/WIN/UT_DSKFI/)

HEdit v3.0

<https://web.archive.org/web/20061115075815/http://www.yurisw.com/HEdit.htm>

HEdit v2.1 (30 day trial)

<https://web.archive.org/web/19980701161517/http://www.yurisw.com/hedit/index.htm>

HEdit.exe (V2.0, 2.1, 3.2)

<https://yuri-software-hedit.software.informer.com/>

<https://texteditors.org/cgi-bin/wiki.pl?HexEditorFamily>

HEDIT.EXE (Windows 3.1)

<https://www.pccorner.com/list/WINDOWS/HEDIT.ZIP/INFO/>

### Biew-610

BEYE (Binary EYE)

<https://sourceforge.net/projects/beye/files/biew/6.1.0/>

### Debuggers:

### Drwatson.exe

C:\WINDOWS\SYSTEM\Drwatson.exe

Microsoft Windows Dr. Watson is a diagnostic tool for the Microsoft Windows operating system.

### GoBug

GoBug is a symbolic debugger which is able to run another program (the "debuggee") in closely controlled conditions.

<https://www.godevtool.com/GobugFrame.htm>

### Inspect\_1b

black & white phantoms - executing inspector v1.1b

"The complete White & Black Phantoms Executing Inspector package"

insp1b.zip

<http://ftp.lanet.lv/ftp/mirror/x2ftp/msdos/programming/lang/>

### Ollydbg 100/200

<https://www.ollydbg.de/>

### PE-Info:

Depends v2.2.60000

Dependency Walker

depends22\_x86.zip

<https://www.dependencywalker.com/>

### eXeScope 630/650

eXeScope can analyze, display various information, and rewrite resources of executable files, that is, EXE, DLL, OCX, etc. without source files.

EXESC630.ZIP

<https://archive.org/details/exescope63>

<https://web.archive.org/web/20020211132347/http://hp.vector.co.jp:80/authors/VA003525/emysoft.htm>

eXeSc650.zip

<https://web.archive.org/web/20071121193238/http://hp.vector.co.jp/authors/VA003525/emysoft.htm>

**PE Tools v1.5**

PE\_Tools.zip

[http://uinc.ru/files/neox/PE\\_Tools.shtml](http://uinc.ru/files/neox/PE_Tools.shtml)

<https://github.com/petoolse/petools>

<https://petoolse.github.io/petools/>

**Resource edit:**

ResHacker V3.2.10.64

<https://archive.org/details/res-hackv3210>

**Image tools:**

JRuler v1.4 (2001)

JR Screen Ruler (Free)

jruler.zip

<https://web.archive.org/web/20011219030039/http://www.spadixbd.com/freetools/>

GIMP V1.2 (2000/12/26)

The GIMP was in early development but the following version will run on Windows 95.

NOTE: It is really difficult to find the precompiled Windows binaries. This is essentially the GTK build found elsewhere.

gimp-setup-20001226.zip

updates-20001226-20001229.zip

lzw-plugins-20001226.zip

<https://web.archive.org/web/20010602181213/http://user.sgc.fi/~tml/gimp/win32/downloads.html>

<ftp://ftp.gimp.org/pub/gimp/win32/gimp-setup-20001226.zip>

gimp-setup-20001023.zip

<https://web.archive.org/web/20010714101546/http://www.gimp.org/~tml/gimp/win32/downloads-20001023.html>

More:

<https://web.archive.org/web/20020201165506/http://www2.arnes.si/~sopisimo/gimp/>

### **MS-Paint**

C:\Program Files\Accessories\MSPAINT.EXE

### **IrfanView**

V 4.44

See section “Essential Applications”.

Office tools:

### **MS Calculator**

C:\WINDOWS\Calc.exe

### **Calc16 V3.3 (Flow Simulation International Ltd. 1995)**

Calc95 is a pocket calculator simulator program for the Microsoft Windows operating system.

calc16s.zip

<http://winspace.retropc.se/winspace-mar96-4/win95/desktop/index.html>

### **Calc95 V3.3 Beta Two (Flow Simulation International Ltd. 1995)**

Calc95 is a pocket calculator simulator program for the Microsoft Windows operating system.

calc95s.zip

<http://winspace.retropc.se/winspace-mar96-4/win95/desktop/index.html>

### **metapad V3.0 (Alexander Davidson 2001)**

Better version of the Windows Notepad. I have not tried V3.6 with Windows 95.

<https://aadavids.tripod.com/>

### **NotePad2 V1.0.12 (2004 or earlier)**

NotePad2 is a small, fast and free text editor with syntax highlighting for HTML and other common languages.

Note: There are a number of forks of this application by other authors. I recommend staying with the Flow's Freeware versions. The old versions (ANSI) can be a little difficult to find.

V1.0.12 (Best link)

<https://web.archive.org/web/20040929130954/http://www.flos-freeware.ch/notepad2.html>

V1.0.12

<http://koos.50webs.com/>

V1.1.0.8

<http://www.oldversion.com/windows/notepad2/>

### **NotePad++ V3.1 - V3.7**

See the main section “Essential Applications” NotePad++.

Archive:

**7-Zip**

See chapter “Essential Applications” 7-Zip.

---

## **Libraries**

The following libraries are for DOS applications. The Libraries will be used in Windows 95 MS-DOS in the same way as they would be used in FreeDOS. This is a copy of the libraries section from Book 1.

Take care with the 8.3 files naming when using Windows 95.

There are a wide variety of libraries available for DOS programming. Although very few are easy to use or compatible with modern programming libraries there are some that offer similar APIs and functionality.

The following is a list of libraries that can be used in both DOS and modern OS programming and some that share similar APIs.

Make sure that you compile the libraries with the same GCC compiler version in which you intend to use the library. Many of the libraries can be used in both DJGPP and FBC DOS as long as you have the correct headers.

PDCurses, Allegro and GRX are the 3 most widely supported libraries available for DOS programming. PDCurses and Allegro both have modern libraries available for 64-bit Windows and Linux systems. GRX does have implementations for Win32 and Linux x86-64 as well as SDL\_BGI as an alternative. If you are programming on modern Windows or Linux you will find more modern libraries that are better suited.

Sound programming in DOS can be particularly precarious as there were no set universal audio drivers. Most applications had to target as many different sound devices as were available at the time. There are some Sound libraries available for DOS but they are somewhat limited compared to modern sound/audio libraries. Also remember that the VirtualBox sound card emulation does not include an FM synthesiser for playing MIDI files.

Another consideration to keep in mind for both Video and Audio is that 32-bit DOS does not natively give DMA (Direct Memory Access) in protected mode which can slow down video and audio playback making it glitchy. There are advanced programming techniques and APIs to deal with this.

### DJGPP FBC defaults: Allegro, GRX, PDCurses.

[http://www.delorie.com/djgpp/v2faq/faq22\\_4.html](http://www.delorie.com/djgpp/v2faq/faq22_4.html)

### FreeBASIC Library list (DOS, Win9x, Win x96-64)

<https://www.freebasic.net/wiki/ExtLibTOC>

Also see the older FreeBASIC library list at the end of this section.

### Some Audio libraries

- Allegro has Audio playback APIs.  
<http://www.s2.org/hmqaudio/>
- Housemarque Audio System  
<http://www.shdon.com/dos/sound#mixing>
- Sound Blaster  
<http://www.shdon.com/dos/sound#mixing>
- Apogee Sound System (Open Watcom)  
[https://github.com/jimdose/Apogee\\_Sound\\_System](https://github.com/jimdose/Apogee_Sound_System)
- Judas Sound Library  
<https://github.com/volkertb/JUDAS>
- Miles Sound System  
<http://www.ke5fx.com/>  
<http://www.thegleam.com/ke5fx/misc/AIL2.ZIP>
- IMYplay  
<https://imyplay.sourceforge.io/>
- Libao  
<https://xiph.org/ao/>

This is a port of Cross-platform Audio Library 1.2.0 to MSDOS/DJGPP

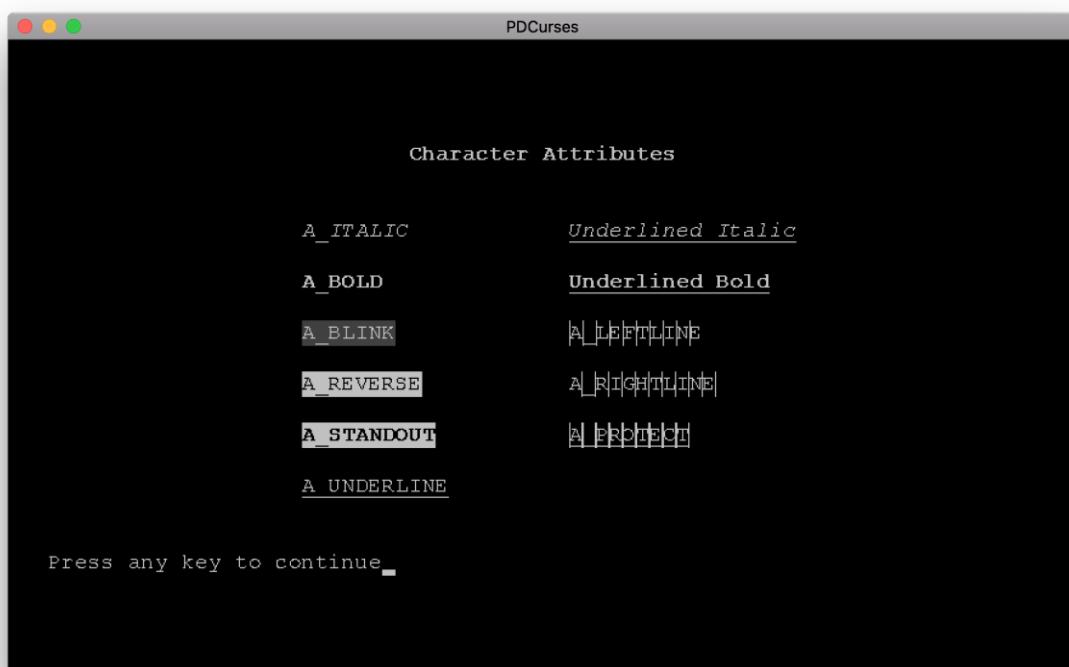
<https://www.delorie.com/pub/djgpp/current/v2tk/>

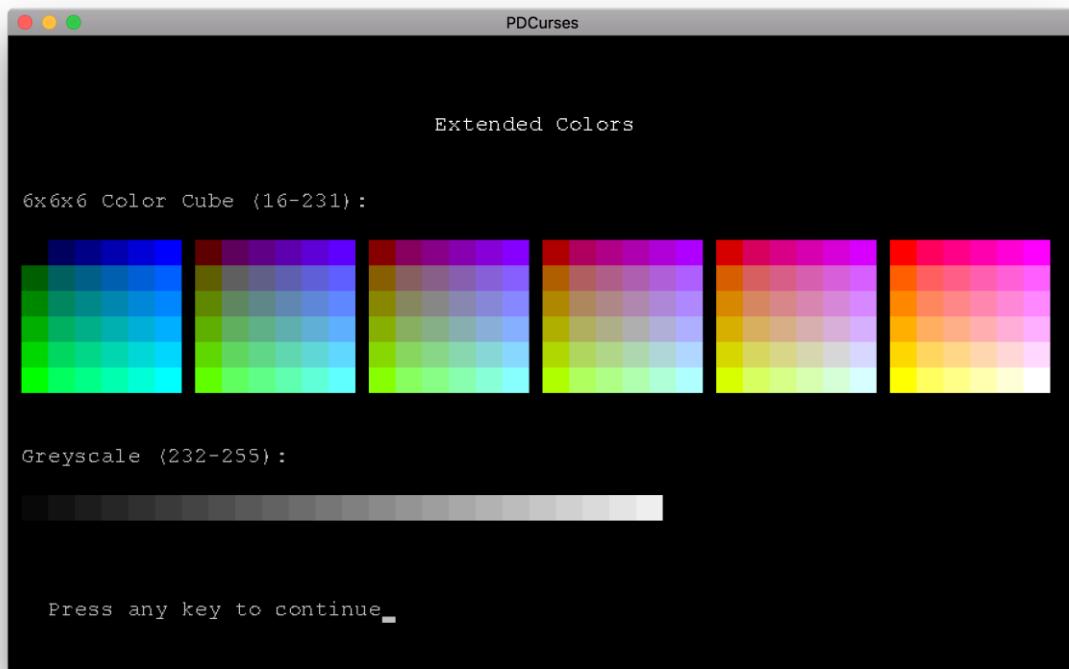
"lao120b.zip"

- Many other fragments.

### PDCurses (Curses)

PDCurses (Public Domain Curses) is a cross platform TUI library for console programming. It is available in DOS, Windows 32-bit and 64-bit as well as Unix and Linux. Precompiled library .LIB .a binaries are often available for download for the different compilers and platforms. The source code also provides make files for compiling PDCurses in any of the supported platforms using the different version of GCC.





### DJGPP and FreeBASIC DOS

<https://www.delorie.com/pub/djgpp/current/v2tk/>

### FBC 0.90.1 (Old version)

This repository has the PDCurses V3.4 which is usable in most DJGPP and FBC versions. The FBC header uses the V3.4 but will also work with PDC v 3.9

<https://sourceforge.net/projects/fbc/files/Older%20versions/0.90.1/Binaries%20-%20Windows/Libraries/>

Source code V3.9 – All supported platforms.

<https://pdcurse.org/>

<https://github.com/wmcbrine/PDCurses>

Make sure you use the same compiler version to compile the library as you intend to use the library with. For example if you want to use PDCurses with FreeBASIC V1.09.0 you will need to download and compile the library in the winlibs-gcc-9.3.0, or DJGPP For DOS etc.

### GRX graphics library

## A Beginners Guide To DOS Programming

GRX is a 2D graphics library originally written by Csaba Biegł for DJ Delorie's DOS port of the GCC compiler.

GRX loosely supports the BGI API.

<http://grx.gnu.de/>

You will need to compile the GRX library yourself.



GRX (FreeBASIC) 2012 [#\_\_FB\_DOS\_\_\*.BI]

2.4.9 (x86)

**GRC20.BI**

**GRXKEYS.BI**

**MGUI (C++, GRX)**

MGUI Library and Designer for GRX.

**MGRX (GRX) 2023**

Alternative Light fork of GRX

MGRX is a 2D graphics library derived from the [GRX library](#). GRX was originally written by Csaba Biegl for DJ Delorie's DOS port of the GCC compiler.

MGRX supports four platforms: DOS (DJGPPv2), Linux console, Linux X11 and Win32 (TDM-GCC). On DOS it supports VGA and VESA compliant cards. On Linux console it comes with framebuffer and KMS/DRM drivers. On X11 and Win32 it runs in a window.

Note mGRX Has no BGI API.

<https://www.fgrim.com/mgrx/>

### Allegro

Allegro is a cross-platform library intended for use in computer games and other types of multimedia programming. Only Allegro 4.x.x versions are suitable for DOS programming.



<https://sourceforge.net/projects/alleg/>

<https://liballeg.org/>

<https://www.allegro.cc/about>

<https://sourceforge.net/projects/alleg/>

<https://phoxis.org/2009/02/13/allegro-422/>

Allegro is available from the DJGPP repository.

<http://www.delorie.com/pub/djgpp/current/v2tk/allegro/>

[#\*\_\_FB\_DOS\_\_.BI]

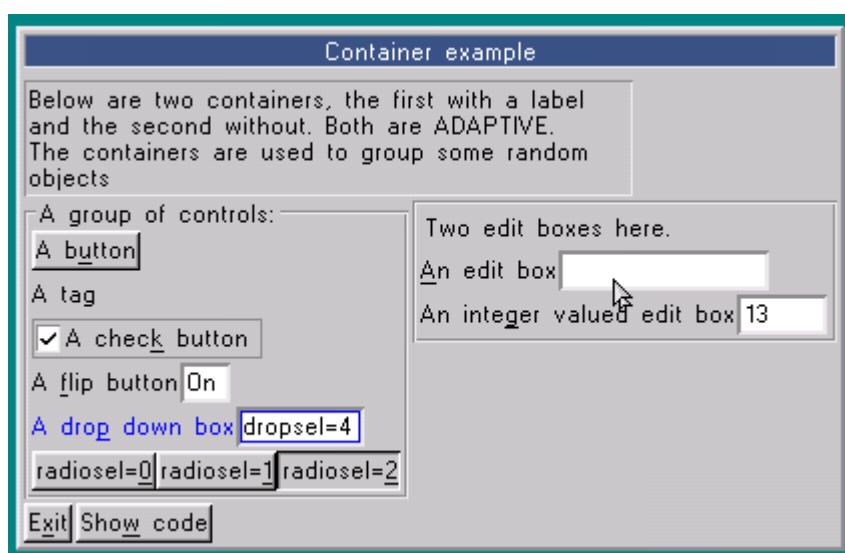
Allegro.bi V 4.4.2

Done

### CGUI (Allegro)

[#\*\_\_FB\_DOS\_\_.BI]

CGUI is a library that provides you with a set of C-functions to create GUI applications, i.e. windows and window controls like buttons, check-boxes, menus, input boxes etc. Available for DOS – FBC and DJGPP.



### GfxLib\_FB (libgfx2)

[#\*.BI] FBGFX.BI

The library named GfxLib is the built-in graphics library included in FreeBASIC. As well as re-creating every QuickBASIC graphics command, GfxLib has built-in commands to handle input from the keyboard and mouse.

<https://www.freebasic.net/wiki/GfxLib>

<https://github.com/freebasic/fbc/tree/master/src/gfxlib2>

<https://sourceforge.net/p/fbc/code/ci/master/tree/src/gfxlib2/>

GPL 2.0

### Lib Static DOS:

libfbgfx.a

libfbgfxmt.a

### Include DOS:

```
fbgfx.bi  
" fbgfx  
" fbgfxmt thread-safe  
" fbgfxpic position independent code  
" fbgfxmtpic thread-safe & PIC
```

## LibGD

[#\*\_\_FB\_DOS\_\_.BI]

### V2.1.1

gd is a graphics library. It allows your code to quickly draw images complete with lines, arcs, text, multiple colors, cut and paste from other images, and flood fills, and write out the result as a PNG or JPEG file. This is particularly useful in World Wide Web applications, where PNG and JPEG are two of the formats accepted for inline images by most browsers.

gd is not a paint program. If you are looking for a paint program, you are looking in the wrong place. If you are not a programmer, you are looking in the wrong place, unless you are installing a required library in order to run an application.

gd does not provide for every possible desirable graphics operation. It is not necessary or desirable for gd to become a kitchen-sink graphics package, but version 2.0 does include most frequently requested features, including both truecolor and palette images, resampling (smooth resizing of truecolor images) and so forth.

<https://libgd.github.io/manuals/2.1.1/files/preamble-txt.html>

Nuklear (Could be ported to DOS) (JS Capable) See Microwindows

<https://github.com/vurtun/nuklear>

<https://github.com/Immediate-Mode-UI/Nuklear>

FLTK (DOS, C++) see Nano-x, Microwindows

<https://www.seriss.com/people/erco/fltk-videos/tutorial-fltk-hello.html>

Microwindows (DOS, Win32, Linux) Nano-X

Microwindows or Nano-X is a small graphical windowing system that implements both Win32 and Nano-X (X11-like) APIs for clipped graphics drawing in windows on Linux, Mac OS X, EMSCRIPTEN, Android and other platforms. It is Open Source and licenced under the Mozilla Public License. For creating GUIs, the Nuklear immediate mode GUI, Win32 builtin controls, and TinyWidget's controls based on Nano-X are included. FLTK can be used with the X11 compatibility library NX11.

<https://github.com/ghaerr/microwindows>

### JMGUI (GfxLib, MyGL, YAGL)

jmgui is a Graphical User Interface (GUI) Library intended for use with game programs. The application program interface (API) has been designed around the idea that the GUI could be inserted in to an existing game application with few modifications to the application.

<https://www.execulink.com/~coder/freebasic/jogui.html>

coderJeff

Windows v0.3b: - jm\_gui-v0.3-win32.zip 100 K

Dos v0.3b: - jm\_gui-v0.3-dos.zip 100 K

Source v0.3b: - jm\_gui-v0.3-source.zip 91 K

### FreeBASIC Libraries BINs (Old versions for reference)

<https://sourceforge.net/projects/fbc/files/Older%20Versions/0.90.1/Binaries%20-%20DOS/Libraries/>

#### Home / Older versions / 0.90.1 / Binaries - DOS / Libraries

- FB-dos-mxml-2.7.zip 2014-09-14 29.6 kB
- FB-dos-pcre-8.32.zip 2014-09-14 282.1 kB
- FB-dos-pdcurses-3.4.zip 2014-09-14 48.6 kB
- FB-dos-quicklz-1.5.0.zip 2014-09-14 2.4 kB
- FB-dos-tiff-4.0.3.zip 2014-09-14 191.4 kB
- FB-dos-tinyptc-0.1.zip 2014-09-14 4.2 kB
- FB-dos-tre-0.8.0.zip 2014-09-14 30.4 kB
- FB-dos-xz-5.0.4.zip 2014-09-14 85.5 kB
- FB-dos-zlib-1.2.7.zip 2014-09-14 49.2 kB
- FB-dos-libpng-1.5.14.zip 2014-09-14 93.2 kB
- FB-dos-libxml2-2.9.0.zip 2014-09-14 652.2 kB
- FB-dos-libxslt-1.1.28.zip 2014-09-14 147.7 kB
- FB-dos-libzip-0.11.zip 2014-09-14 42.3 kB
- FB-dos-lua-5.2.2.zip 2014-09-14 114.3 kB
- FB-dos-lzo-2.06.zip 2014-09-14 59.2 kB
- FB-dos-lcms2-2.4.zip 2014-09-14 165.9 kB
- FB-dos-lcms-1.19.zip 2014-09-14 113.4 kB
- FB-dos-libmng-1.0.10.zip 2014-09-14 163.6 kB
- FB-dos-jasper-1.900.1.zip 2014-09-14 162.1 kB
- FB-dos-jpeglib-9.zip 2014-09-14 133.7 kB

|                              |            |          |
|------------------------------|------------|----------|
| • FB-dos-grx-2.4.9.zip       | 2014-09-14 | 210.2 kB |
| • FB-dos-gsl-1.15.zip        | 2014-09-14 | 1.1 MB   |
| • FB-dos-gmp-5.1.1.zip       | 2014-09-14 | 272.4 kB |
| • FB-dos-giflib-5.0.4.zip    | 2014-09-14 | 15.9 kB  |
| • FB-dos-expat-2.1.0.zip     | 2014-09-14 | 73.0 kB  |
| • FB-dos-gd-73cab5d8af96.zip | 2014-09-14 | 123.1 kB |
| • FB-dos-gdsl-1.6.zip        | 2014-09-14 | 43.1 kB  |
| • FB-dos-cunit-2.1-2.zip     | 2014-09-14 | 33.6 kB  |
| • FB-dos-devil-1.7.8.zip     | 2014-09-14 | 273.9 kB |
| • FB-dos-big_int-1.0.7.zip   | 2014-09-14 | 24.1 kB  |
| • FB-dos-bzip2-1.0.6.zip     | 2014-09-14 | 29.3 kB  |
| • FB-dos-aspell-0.60.6.1.zip | 2014-09-14 | 509.3 kB |

Totals: 32 Items      5.2 MB 0

### TCL/Tk

An implementation of the TCL Windowing widget kit for DOS. It is experimental and does not support most Windowing features found in Windows or Linux.

<https://wiki.tcl-lang.org/page/Tcl+for+DOS>

### DUGL

DUGL the DOS Ultimate Game Library, is a DOS only 32bits Games, Graphics and multimedia library.

It aims to give a modern and the fastest graphic software rendering library on DOS, and maybe ever :)

Note: DUGL is a bit blurred between C and C++.

<http://dugl.50webs.com/>



### Library Summary

Many of the common libraries for DJGPP and FreeDOS are available via the DJGPP repository. They will often be an earlier but compatible version for the 2 development environments. If you wish to use the latest version of the libraries you will need to compile them using DJGPP yourself. Check the library compiling instructions for how to compile the library. Typically the common libraries source will have the "MAKE" files and instructions available to build the library for the DOS platform. Call the library make file using a batch file that contains the path to the compiler as well as any environment variables (mandatory), the path to the source (optional), and in some cases you may need to "CD to" or make the source make file the "Working Directory". Read the library build instructions carefully.

I would suggest compiling PDCurses as your first library as the instruction are well described and it is a relatively easy library to build.

You can build some libraries in FreeDOS but most are compiled from a Win32 command line environment such as Windows 9x, and then transferred to the FreeDOS install. Be aware that most of the libraries are designed for the DOS 32-bit environment.

DOS 16-bit real mode libraries require a different compiler environment such as IA16. 16 bit libraries are more complicated and are often built using assembly language.

## DOS Development Summary

Be aware of the different methods for transferring files to your DOS and Windows 95 system drive.

Set up your Virtual machine environment for Windows 95 MS-DOS.

Keep a log of install tasks and customisations for use in later DOS and Windows 95 installs.

Remember DOS has no official directory structures and it is up to the user to organise the system in a meaningful way. Take a look at some common DOS installs. FreeDOS begins with a sound and well organised directory structure and offers a sound template. If you are using FreeDOS keep and make use of the default directory structure as all FD installs will make use of this. For Windows 95 follow the directory structure of FreeDOS except for the Windows binary file directories C:\WIN95 and C:\WINDOWS\COMMAND.

Install Windows 95 with additional components.

Make intermittent backups of the Windows 95 virtual hard drive for roll backs in case of an error.

Become familiar with the Windows emulated vs DOS-Mode system environment settings.

Make backup copies of other custom batch files as well as creating a batch “Template” directory. You can use the batch file templates for other common tasks.

Install additional drivers such as sound, video and mouse if required for DOS Mode. Adjust the mouse speed for CTMouse with the /Rnn switch in DOS Mode.

Install Filemaven 3 for ease of navigation in DOS Mode.

Install some other basic convenience tools such as Image editors, text editors and system information utilities.

Install the required development environments. Some development environments will rely upon DJGPP for compiling library files. Note that most development environments can be copied to Windows 95 directly from the FreeDOS install.

Set up the custom path and environment variables for each software development environment using batch files. Include your preferred editor/IDE as part of the batch file. Keep the Windows path environment in mind.

Keep your software development environments and the editor/IDE separate from each other and use alternative names for directories and batch files.

Set up and customise your Editor/IDE. If using FED, create the custom compile and run batch files as well as the FED tools menu entries.

If using ALaunch to organise your work environment, add the shortcuts (PIFs) to your menu directory.

Become familiar with the DOS environments under Windows 95 as well as your development environment. Check for errors and make personal adjustments as required.

Research and get know some of the additional utility tools such as debuggers, system information tools, hex editors and other conveniences to aid software development.

---

## Supplemental

### Additional Information

The following is a collection of current sites with DOS and Windows 95 associated downloads and information. The list is currently unsorted draft stage and under construction.

This is provided to give an idea of where to find DOS related resources and some of the search keywords as finding DOS resources in search engines can be difficult. Many of the following sites will also reference other DOS related sites so take the time to have a look around.

There are also many good web sites and reference books available that cover both the DOS system and DOS programming in more detail.

Some of the links point to FTP and HTTP[s] file archives that can be quite large. I recommend having an idea of the file information and file name before searching an archive site. Some archive sites contain a file INDEX\_00.HTML which will contain the file descriptions for each file name whereas other will not, and some post the description next to the file name. In all cases it is best to have an idea of the file you are looking for as well as descriptions and version information beforehand.

### DOS and DOS development links and guides (Unsorted)

#### URL List from the main document:

[http://www.installsite.org/pages/en/tt\\_winup.htm](http://www.installsite.org/pages/en/tt_winup.htm)

<https://www.7-zip.org/>

<https://www.trustfm.net/software/utilities/Folder2Iso.php>

<https://sourceforge.net/projects/imdisk-toolkit/>

<https://www.rejetto.com/hfs/>

<https://winworldpc.com/product/microsoft-windows-boot-disk/98-se>

<https://www.ibm.com/support/pages/scrollpoint-ii-mouse-windows-9598nt-40-qj6z34us-ibm-intellistation-z-pro-type-6866>

<https://www.mdgx.com/speed.htm>

<https://www.mdgx.com/web.htm#OSR2>

## A Beginners Guide To DOS Programming

<https://web.archive.org/web/20170828234629/https://www.mdgx.com/web.htm#OSR2>

<https://www.my-internet-explorer.com/ie5/>

<https://archive.org/details/internet-explorer-5-5.5>

<https://winworldpc.com/product/windows-95/patches>

<http://www.oldversion.com/windows/directx-8-0a>

[https://web.archive.org/web/20070226050026/https://download.microsoft.com/download/vc60pr0/update/1/w9xnt4/en-us/vc6redistsetup\\_enu.exe](https://web.archive.org/web/20070226050026/https://download.microsoft.com/download/vc60pr0/update/1/w9xnt4/en-us/vc6redistsetup_enu.exe)

<https://sourceforge.net/projects/vb6extendedruntime/>

<http://ftpmirror.your.org/pub/misc/ftp.microsoft.com/Softlib/MSLFILES/>

<https://sourceforge.net/projects/sevenzip/files/7-Zip/9.20/>

<http://www.oldversion.com/windows/notepad-3-7>

<https://winworldpc.com/product/adobe-acrobat/4x>

<https://www.alentum.com/alaunch/>

<http://cleanup.stevengould.org/>

[https://www.majorgeeks.com/files/details/microsoft\\_regclean.html](https://www.majorgeeks.com/files/details/microsoft_regclean.html)

<https://www.portablefreeware.com/index.php?id=406>

[https://math.vanderbilt.edu/schectex/wincd\\_files/local\\_archive/powertoys/](https://math.vanderbilt.edu/schectex/wincd_files/local_archive/powertoys/)

<https://www.philscomputerlab.com/vga-test-patterns.html>

<http://www.oldversion.com/windows/irfanview/>

<https://www.seamonkey-project.org/releases/seamonkey1.0.9/>

<https://www.seamonkey-project.org/releases/1.0.9>

<http://www.oldversion.com/windows/tiny-personal-firewall-2-0-15a>

<https://vetusware.com/download/Tiny%20Personal%20Firewall%202.0.9/?id=5209>

<https://docplayer.net/21485376-User-s-guide-tiny-personal-firewall-v-2.html>

<https://www.rejetto.com/hfs/?f=dl>

<https://www.vogons.org/viewtopic.php?t=41865>

<https://cutemouse.sourceforge.net/>

<http://help.fdos.org/en/hhstndrd/ctmouse.htm>

<https://www.briggsoft.com/fmdos.htm>

<https://www.sac.sk/files.php?d=7&l>

<https://www.sac.sk/files.php?d=7&l>

## A Beginners Guide To DOS Programming

<https://www.sac.sk/files.php?d=7&l>

<https://www.bttr-software.de/freesoft/arc1.htm>

<https://www.computerhope.com/software/pkutil.htm>

[http://wiki.freedos.org/wiki/index.php/Dos\\_commands](http://wiki.freedos.org/wiki/index.php/Dos_commands)

<https://www.ibiblio.org/pub/micro/pc-stuff/freedos/files/repositories/1.3/pkg-html/index.html>

<https://en.wikipedia.org/wiki/QBasic>

<https://en.wikipedia.org/wiki/QuickBASIC>

<https://www.qbasic.net/en/top-ten-downloads/>

<https://hwiegman.home.xs4all.nl/qb-man/index.html>

<https://archive.org/details/003495-MicrosoftQuickbasic45>

<https://winworldpc.com/product/quickbasic/45>

<https://www.qbasic.net/en/qbasic-downloads/compiler/qbasic-compiler.htm>

[https://archive.org/details/Microsoft\\_QuickBASIC\\_4.5\\_2nd\\_Edition\\_Manual](https://archive.org/details/Microsoft_QuickBASIC_4.5_2nd_Edition_Manual)

<https://www.robvanderwoude.com/path.php>

<https://www.qbasic.net/en/qbasic-downloads/compiler/qbasic-compiler.htm>

<https://archive.org/details/qb71zip>

<https://winworldpc.com/product/microsoft-basic/pds-71>

<https://winworldpc.com/product/microsoft-basic/pds-71>

<https://www.qbasic.net/en/qbasic-downloads/compiler/qbasic-compiler.htm>

<https://winworldpc.com/product/microsoft-visual-bas/10-for-dos>

<https://archive.org/details/ms-vbdos10>

<https://www.ibiblio.org/pub/micro/pc-stuff/freedos/files/repositories/1.3/pkg-html/group-devel.html>

<https://www.ibiblio.org/pub/micro/pc-stuff/freedos/files/repositories/1.3/pkg-html/group-devel.html>

<https://www.delorie.com/digpp/zip-picker.html>

<http://www.delorie.com/pub/digpp/current/>

<https://www.freebasic.net/wiki/DevBuildDos>

[http://www.delorie.com/digpp/v2faq/faq4\\_4.html](http://www.delorie.com/digpp/v2faq/faq4_4.html)

<https://www.emse.fr/~boissier/enseignement/sdao/Rhide.htm>

<http://www.bttr-software.de/products/fed/>

<https://www.bttr-software.de/freesoft/batch1.htm>

## A Beginners Guide To DOS Programming

<http://www.pl.exim.org/packages/coast/msdos/batutil/>

<http://www.manmrk.net/tutorials/batch/index.htm>

<http://www.lanet.lv/simtel.net/msdos/batchutl-pre.html>

<https://www.delorie.com/pub/djgpp/current/v2tk/>

<http://cc.embarcadero.com/Default.aspx>

<http://cc.embarcadero.com/item/25636>

<http://cc.embarcadero.com/Item/26014>

<https://winworldpc.com/product/borland-turbo-c/1x>

<https://winworldpc.com/product/borland-turbo-c/1x>

<https://eecs.wsu.edu/~cs150/prog/tcinstall.htm>

<https://eecs.wsu.edu/~cs150/prog/tcinstall.htm>

<https://www.ibiblio.org/pub/micro/pc-stuff/freedos/files/repositories/1.3-devel/>

<https://sourceforge.net/projects/fbc/files/FreeBASIC-1.10.0/Binaries-DOS/>

<https://www.freebasic.net/wiki/DevBuildDos>

<https://sourceforge.net/projects/fbc/files/>

<https://sourceforge.net/projects/fbc/files/FreeBASIC-1.10.0/Binaries-DOS/>

<https://www.bttr-software.de/products/fed/>

<https://www.bttr-software.de/freesoft/batch1.htm>

<http://www.pl.exim.org/packages/coast/msdos/batutil/>

<http://www.manmrk.net/tutorials/batch/index.htm>

<http://www.lanet.lv/simtel.net/msdos/batchutl-pre.html>

<https://www.delorie.com/pub/djgpp/current/v2tk/>

<https://github.com/SuperIlu/DOjS>

<https://www.ibiblio.org/pub/micro/pc-stuff/freedos/files/repositories/1.3/pkg-html/dojs.html>

<https://www.bloodshed.net/Dev-Cpp-4>

<https://www.bloodshed.net/>

<https://sourceforge.net/projects/orwelldevcpp/files/Setup%20Releases/>

[https://web.archive.org/web/20060715000000\\*/https://download.microsoft.com/download/winscript56/Install/5.6/W9XNT4Me/EN-US/scr56en.exe](https://web.archive.org/web/20060715000000*/https://download.microsoft.com/download/winscript56/Install/5.6/W9XNT4Me/EN-US/scr56en.exe)

<https://download.microsoft.com/download/winscript56/Install/5.6/W982KMeXP/EN-US/scrdoc56en.exe>

[http://www.installsite.org/pages/en/tt\\_winup.htm](http://www.installsite.org/pages/en/tt_winup.htm)

## A Beginners Guide To DOS Programming

<https://am.net/lib/tools/Microsoft/scripting/>

<https://www.autoitscript.com/autoit3/files/archive/autoit/>

<https://winworldpc.com/product/visual-c/6x>

<https://archive.org/details/microsoft-visual-c-6.0-standard-edition>

<https://winworldpc.com/product/microsoft-visual-stu/60>

<https://winworldpc.com/product/microsoft-visual-bas/60>

<http://web.archive.org/web/20230610211605/https://www.japheth.de/debxxf.html>

<http://www.bttr-software.de/products/insight/>

<https://www.ibiblio.org/pub/micro/pc-stuff/freedos/files/repositories/1.3/pkg-html/insight.html>

<http://www.sudleyplace.com/swat/>

<https://www.ibiblio.org/pub/micro/pc-stuff/freedos/files/repositories/1.3/pkg-html/upx.html>

<http://www.bttr-software.de/products/fed/>

<https://www.bttr-software.de/freesoftware/batch1.htm>

<http://www.dcee.net/Files/Utils/>

<https://uhex.sourceforge.net/>

<https://sourceforge.net/projects/hexcompare/files/hexcompare%20v1.0.4/>

<https://drv.nu/vbediag/>

<https://www.elhb.com/webhq/download/index.htm>

<https://www.elhb.com/webhq/download/index.htm>

[https://www.bttr-software.de/freosoftware/system.htm](https://www.bttr-software.de/freesoftware/system.htm)

[https://archive.org/details/msdos\\_shareware\\_fb\\_INFO](https://archive.org/details/msdos_shareware_fb_INFO)

<https://www.hwinfo.com/download/>

<https://mklasson.com/hexit.php>

<https://www.bttr-software.de/freosoftware/filutil2.htm>

<https://www.sac.sk/search.php>

<https://www.bttr-software.de/freosoftware/filutil2.htm>

<https://www.sac.sk/files.php?d=17&l>

<http://web.archive.org/web/20050205203014/http://phantom.urbis.net.il/inspector/>

<http://www.ibiblio.org/pub/micro/pc-stuff/freedos/files/repositories/1.3/pkg-html/listvesa.html>

<http://www.ibiblio.org/pub/micro/pc-stuff/freedos/files/repositories/1.3/pkg-html/foxcalc.html>

<http://www.ibiblio.org/pub/micro/pc-stuff/freedos/files/repositories/1.3/pkg-html/rcal.html>

<https://heninv.tripod.com/codedit.html>

<https://www.cdmediaworld.com/hardware/cdrom/files.shtml>

[http://cd.textfiles.com/pslv4nv08/WIN/UT\\_DSKFI/](http://cd.textfiles.com/pslv4nv08/WIN/UT_DSKFI/)

<https://sourceforge.net/projects/beye/files/biew/6.1.0/>

<https://www.godevtool.com/GobugFrame.htm>

<http://ftp.lanet.lv/ftp/mirror/x2ftp/msdos/programming/lang/>

<https://www.ollydbg.de/>

<https://www.dependencywalker.com/>

<https://archive.org/details/exescope63>

<https://web.archive.org/web/20020211132347/http://hp.vector.co.jp:80/authors/VA003525/emysoft.htm>

<https://web.archive.org/web/20071121193238/http://hp.vector.co.jp/authors/VA003525/emysoft.htm>

[http://uinc.ru/files/neox/PE\\_Tools.shtml](http://uinc.ru/files/neox/PE_Tools.shtml)

<https://github.com/petoolse/petools>

<https://petoolse.github.io/petools/>

<https://archive.org/details/res-hackv3210>

<https://web.archive.org/web/20011219030039/http://www.spadixbd.com/freetools/>

<https://web.archive.org/web/20010602181213/http://user.sgic.fi/~tml/gimp/win32/downloads.html>

<ftp://ftp.gimp.org/pub/gimp/win32/gimp-setup-20001226.zip>

<https://web.archive.org/web/20010714101546/http://www.gimp.org/~tml/gimp/win32/downloads-20001023.html>

<https://web.archive.org/web/20020201165506/http://www2.arnes.si/~sopjismo/gimp/>

<http://winsite.retropc.se/winsite-mar96-4/win95/desktop/index.html>

<http://winsite.retropc.se/winsite-mar96-4/win95/desktop/index.html>

<https://aadavids.tripod.com/>

<https://web.archive.org/web/20040929130954/http://www.flos-freeware.ch/notepad2.html>

<http://koos.50webs.com/>

<http://www.oldversion.com/windows/notepad2/>

[http://www.delorie.com/digpp/v2faq/faq22\\_4.html](http://www.delorie.com/digpp/v2faq/faq22_4.html)

<https://www.freebasic.net/wiki/ExtLibTOC>

## A Beginners Guide To DOS Programming

<http://www.s2.org/hmqaudio/>

<http://www.shdon.com/dos/sound#mixing>

[https://github.com/jimdose/Apogee\\_Sound\\_System](https://github.com/jimdose/Apogee_Sound_System)

<https://github.com/volkertb/JUDAS>

<http://www.ke5fx.com/>

<http://www.thegleam.com/ke5fx/misc/AIL2.ZIP>

<https://imyplay.sourceforge.io/>

<https://xiph.org/ao/>

<https://www.delorie.com/pub/digpp/current/v2tk/>

<https://www.delorie.com/pub/digpp/current/v2tk/>

<https://sourceforge.net/projects/fbc/files/Older%20versions/0.90.1/Binaries%20-%20Windows/Libraries/>

<https://pdcourses.org/>

<https://github.com/wmcbrine/PDCurses>

<http://grx.gnu.de/>

[GRX library](#)

<https://www.fgrim.com/mgrx/>

<https://sourceforge.net/projects/alleg/>

<https://liballeg.org/>

<https://www.allegro.cc/about>

<https://sourceforge.net/projects/alleg/>

<https://phoxis.org/2009/02/13/allegro-422/>

<http://www.delorie.com/pub/digpp/current/v2tk/allegro/>

<https://www.freebasic.net/wiki/GfxLib>

<https://github.com/freebasic/fbc/tree/master/src/gfxlib2>

<https://sourceforge.net/p/fbc/code/ci/master/tree/src/gfxlib2/>

<https://libgd.github.io/manuals/2.1.1/files/preamble-txt.html>

<https://github.com/vurtun/nuklear>

<https://github.com/Immediate-Mode-UI/Nuklear>

<https://www.seriss.com/people/erco/fltk-videos/tutorial-fltk-hello.html>

<https://github.com/ghaerr/microwindows>

## A Beginners Guide To DOS Programming

<https://www.execulink.com/~coder/freebasic/jogui.html>

<https://sourceforge.net/projects/fbc/files/Older%20versions/0.90.1/Binaries%20-%20DOS/Libraries/>

<https://wiki.tcl-lang.org/page/Tcl+for+DOS>

<http://dugl.50webs.com/>

<http://web.archive.org/web/20070222072653/www.microsoft.com/windows95/downloads/>

<https://msfn.org/board/topic/149063-requesting-windows-95-updates-tools-etc/>

<https://www.andrews.edu/~steele/steele.cgi?winsock32>

<http://cd.textfiles.com/winfiles3/required/required.html>

[http://www.installsite.org/pages/en/tt\\_winup.htm](http://www.installsite.org/pages/en/tt_winup.htm)

---