A BEGINNERS GUIDE TO DOS PROGRAMMING

Book 1

DOS Development Environments

Axle

30 January 2024

Table of Contents

Licence

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 Jacob Palm for providing the Costa GUI desktop for everyone's use.

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

This document is provident in the hope that it will be 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 except 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	14/11/2023	FreeDOS install. Basic applications and drivers.
Draft V 0.3	18/11/2023	Add QBASIC and QuickBASIC. Some corrections.
Draft V 0.4	20/11/2023	Added DJGPP.
Draft V 0.5	30/11/2023	Added FreeBASIC.
Draft V 0.6	30/11/2023	Add common libraries.
Draft V 0.7	30/11/2023	Add utilities. Draft copy complete.
Draft V 0.8	05/12/2023	Proof read and corrections. Add some DOS utilities.
Draft V 0.9	06/12/2023	Minor corrections and updates.
Draft V 0.10	07/12/2023	Added HFS and Dillo instructions.
Draft V 0.11	13/12/2023	Added Visual Basic for DOS and DOjS.

		Some additional information and corrections.
Draft V 0.12	19/12/2023	Corrections. Added full URL list in “Supplemental”.
Draft V 0.13	21/01.2024	Many minor corrections.
Draft V 0.14	24/01/2024	Many minor corrections.
Draft V 0.15	26/01/2024	Correct QB45 images \DEV to \DEVEL
Draft V 0.16	27/01/2024	Proof read and many corrections. This is a “**Preview Draft**”.
Draft V 0.17	30/01/2024	Corrections. Add source code archive. Add Strings25.zip. This is a “**Preview Draft**”.

TODO:

Proof read [Partial]

Preface

This guide is the primary companion to “A BEGINNERS GUIDE TO DOS PROGRAMMING” - Book 2 “Windows 95 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 3rd 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. 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 FreeDOS 1.3 as well as some of the essential applications and some drivers. I will also offer some convenience customisations for using FreeDOS by adding some TUI and GUI navigation as well as some custom start up menus 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 blend together in practice to form a complete setup. This means that some parts of the guide may appear to belong to a different stage but ultimately the order in which applications are installed or when certain customizations are made is somewhat malleable. There are no hard and fast rules to how you choose to set up a DOS OS so this guide just covers some of the options available.

If you are unfamiliar with 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 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. 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. 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.

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 link s to other documents available at the time of writing.

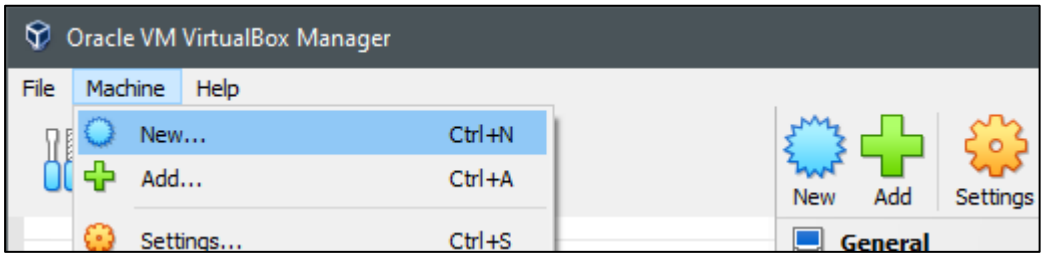
I provide a condensed summary of customisation and installed applications at the end of the guide.

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.

Installing FreeDOS

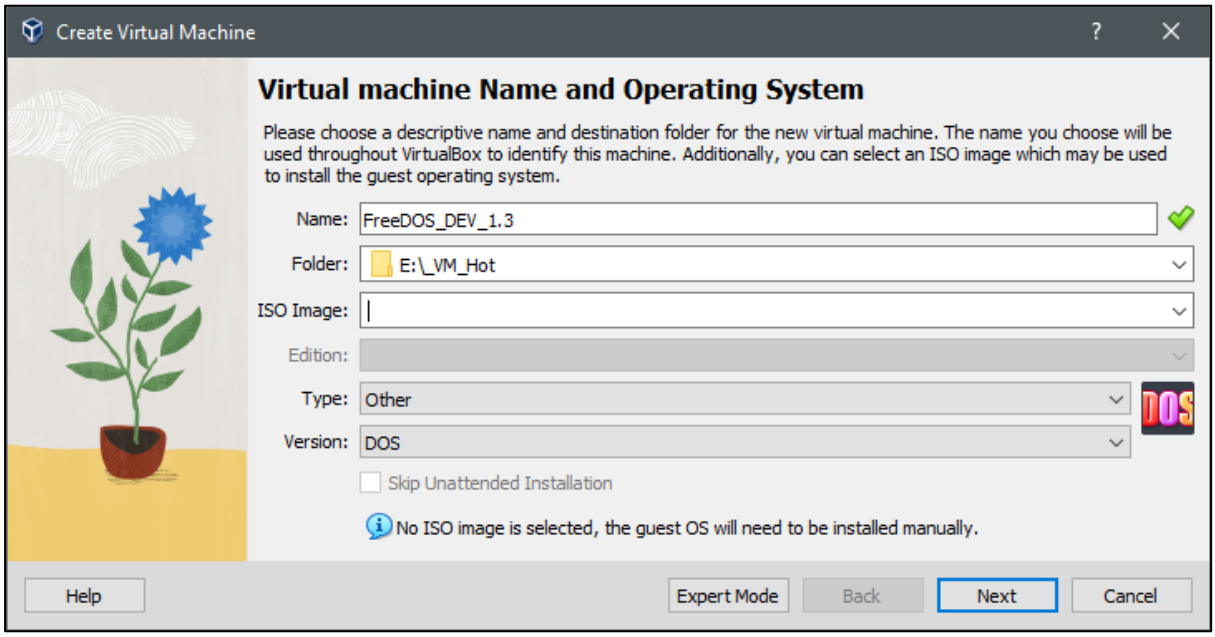
Setting up a VirtualBox guest

In the VitualBox Manager select “Machine -> New...”.

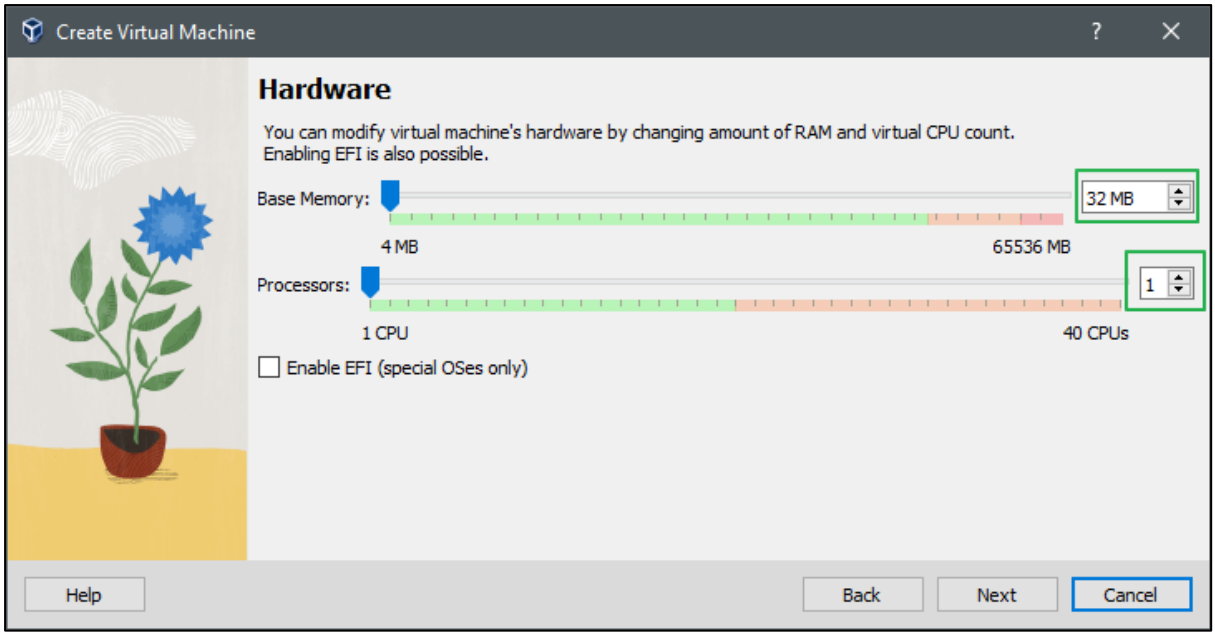


Give the guest machine a recognisable name; select the location where you store your guest operating systems.

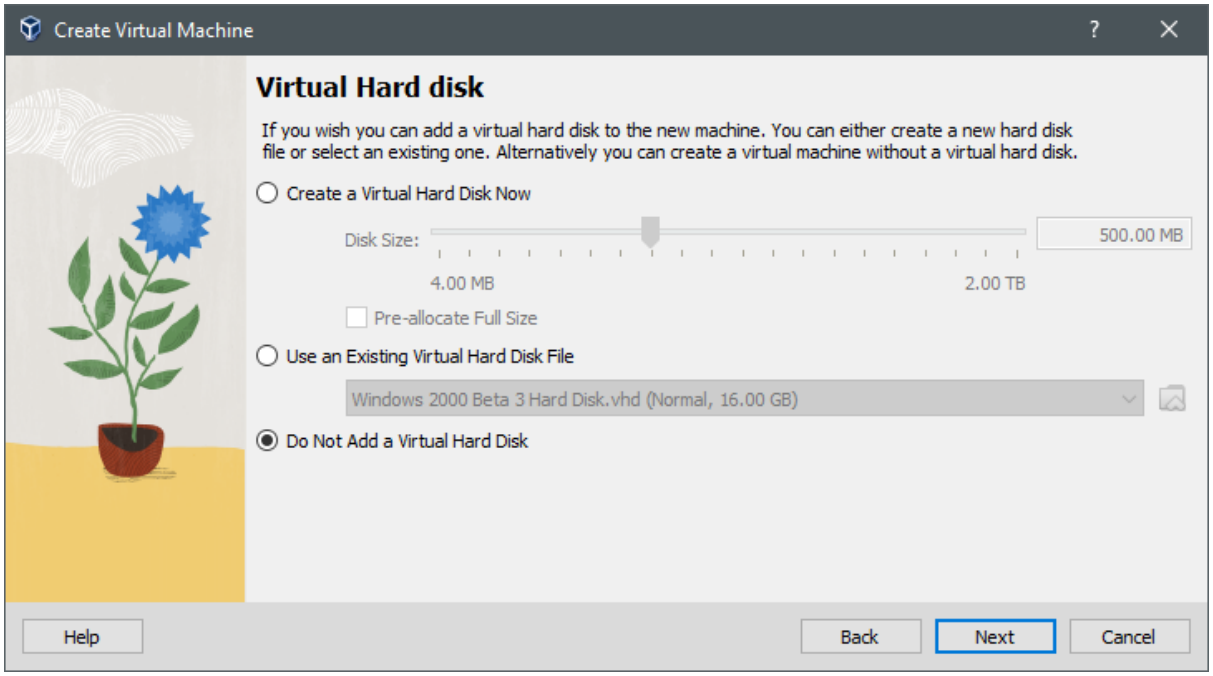
For “Type” select “Other” and then select “DOS” in “Version” and then “Next”.



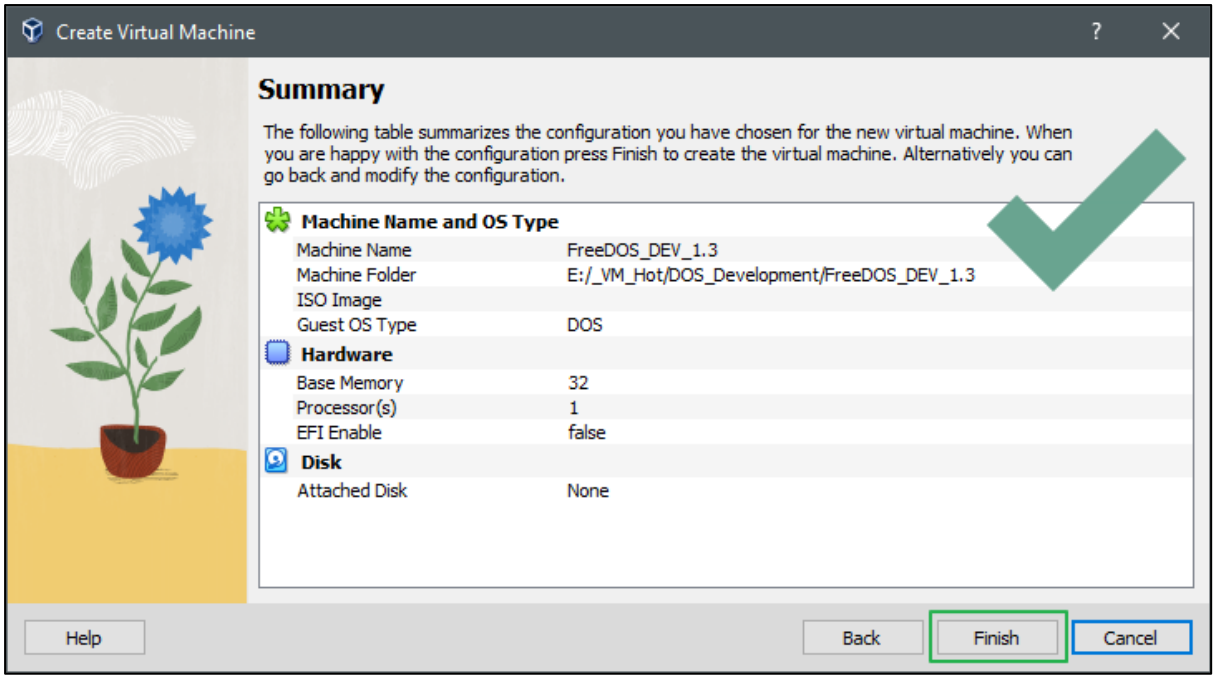
Keep the default hardware settings of 32 MB and 1 CPU and select “Next”. 32MB of RAM is more than sufficient for a DOS environment but you can always alter it latter to emulate low memory machines if needed. Never select more than a single CPU as DOS does not have the native ability to recognise more than One CPU or the native ability to multitask.



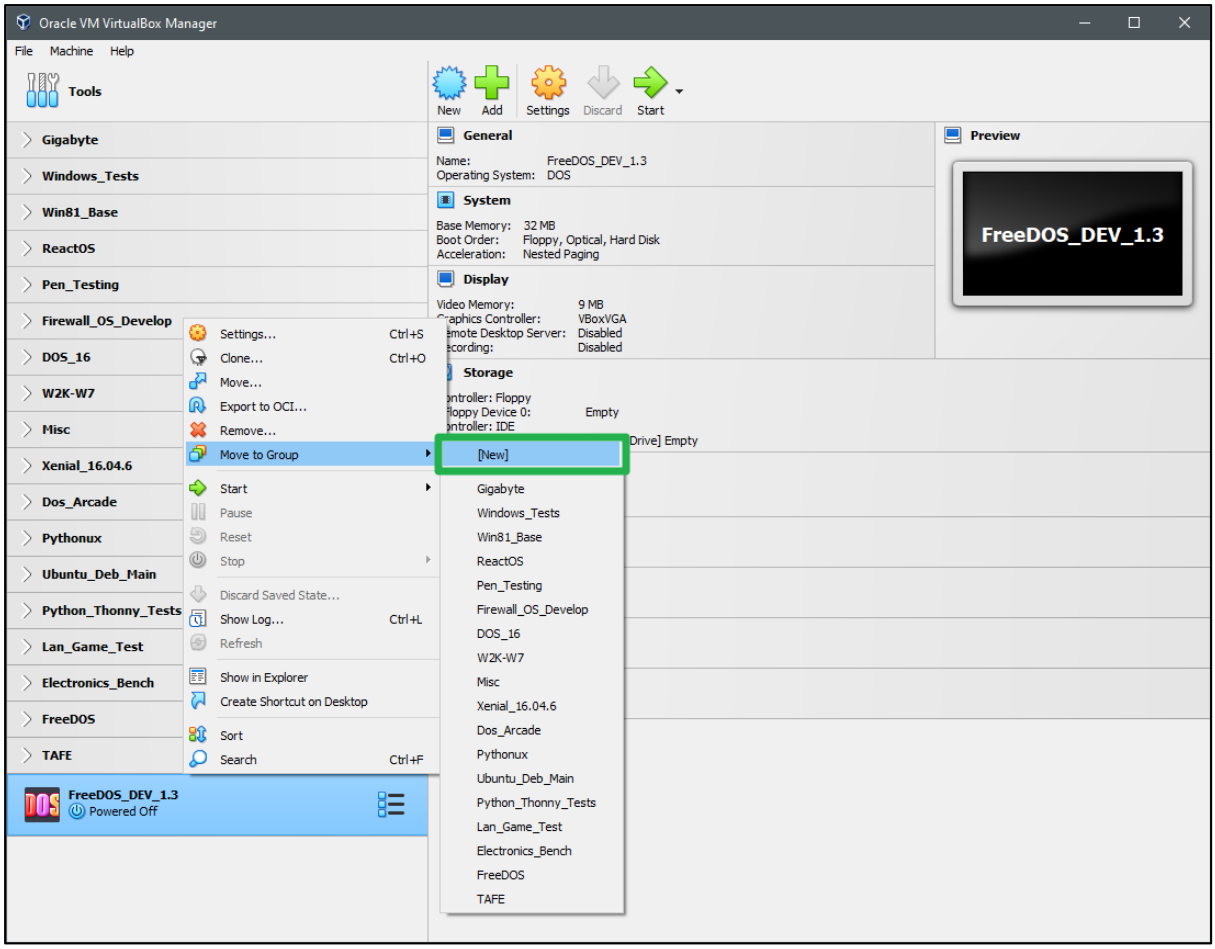
Do not create a virtual hard disk at this stage. VirtualBox V 6.x does allow you to create a Windows .VHD drive at this stage, but for VirtualBox V 7.0 you will need to select expert mode at the beginning to be able to select a .VHD as the drive type. This is OK to leave for now as we will just create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



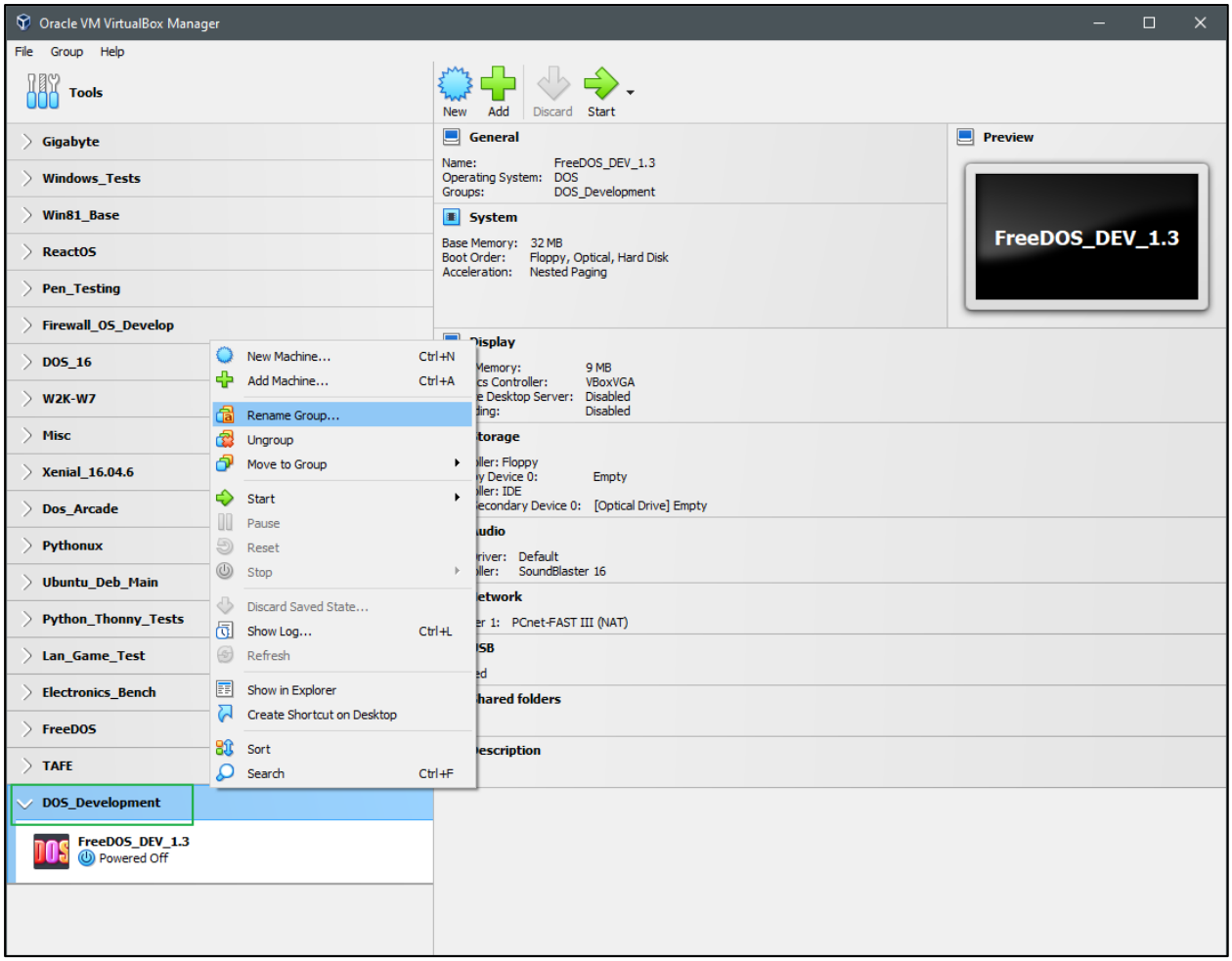
Check that the summary has the correct details and select “Finish” and then “Continue on the warning dialog.



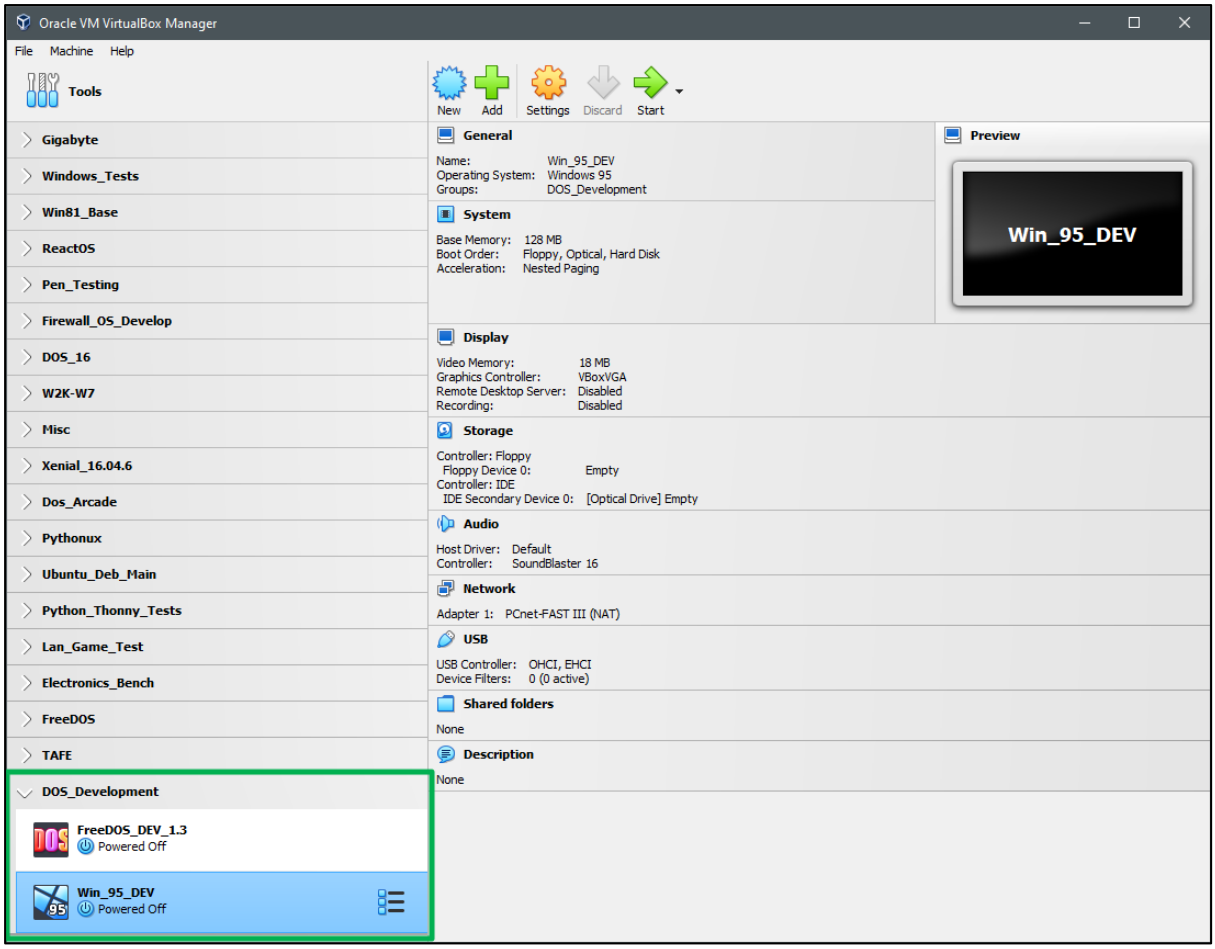
At this point you can place your new DOS Virtual machine into a group for easier organisation if you want. You must have at least one premade virtual machine to create a group. The newly created group will not be named and you will need to give it an appropriate name after it is created. You can also drag and drop a virtual machine into an existing group.



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.

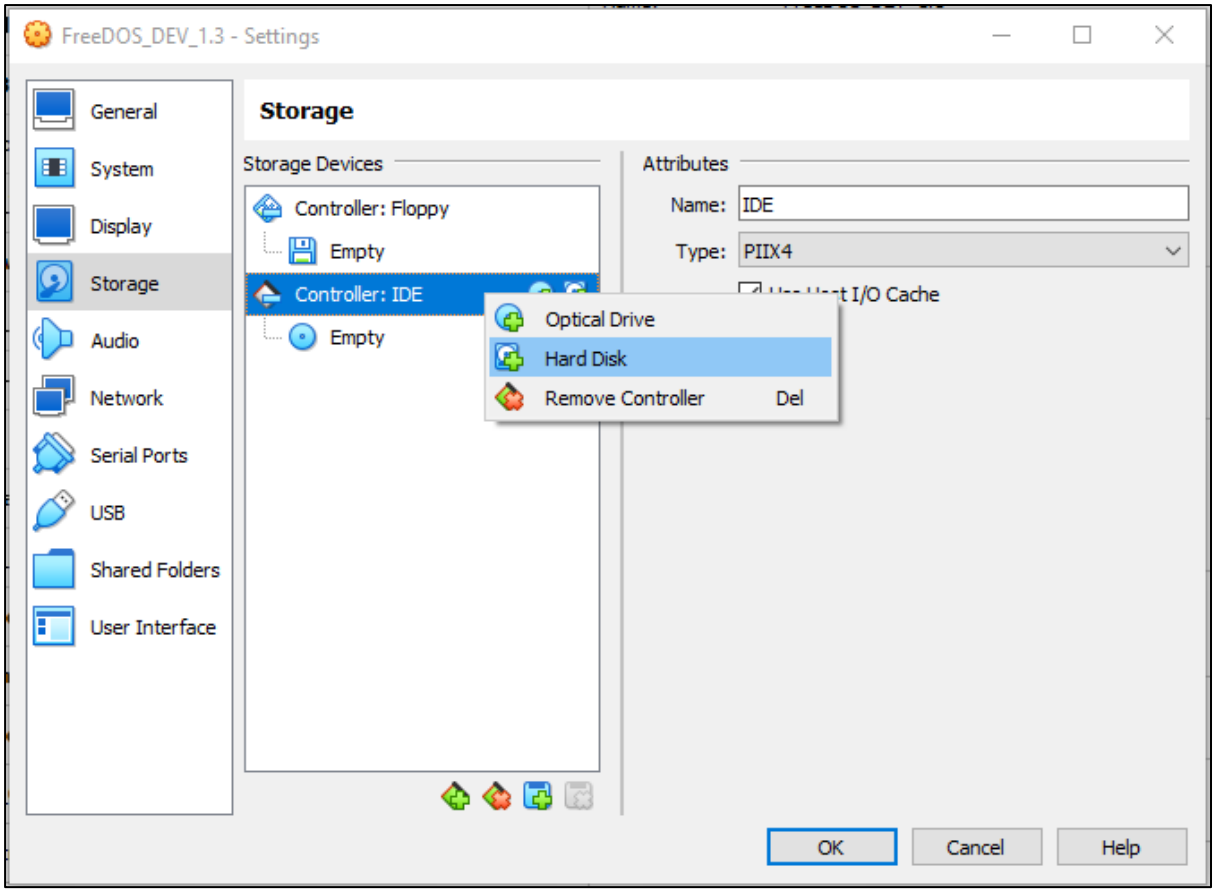


In the accompanying Windows 95 guide I will also be adding the Windows 95 DOS development environment to this group. In the following I have both FreeDOS and Windows 95 in the group “DOS Development”.

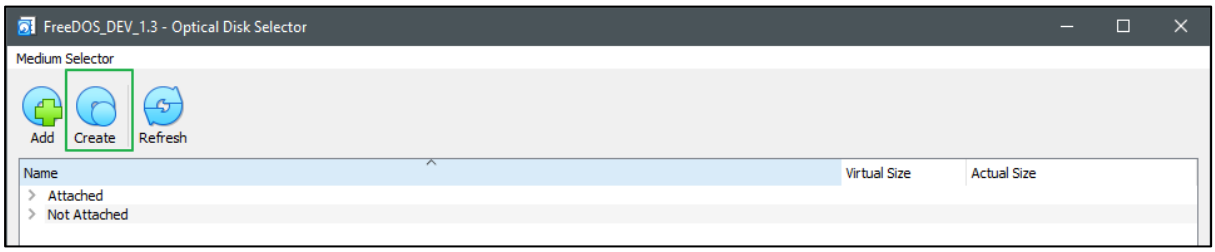


Next go to “Settings” for your virtual machine. We can now create the VHD based virtual drive. I typically use Windows as my host system for virtual machines. Mounting a Virtual Hard Drive (VHD) is a native operation in windows 8.1 onwards making read/write access to the guest OS hard drive easily accessible from the host system. If you are using Linux as your host OS for VitualBox 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”.

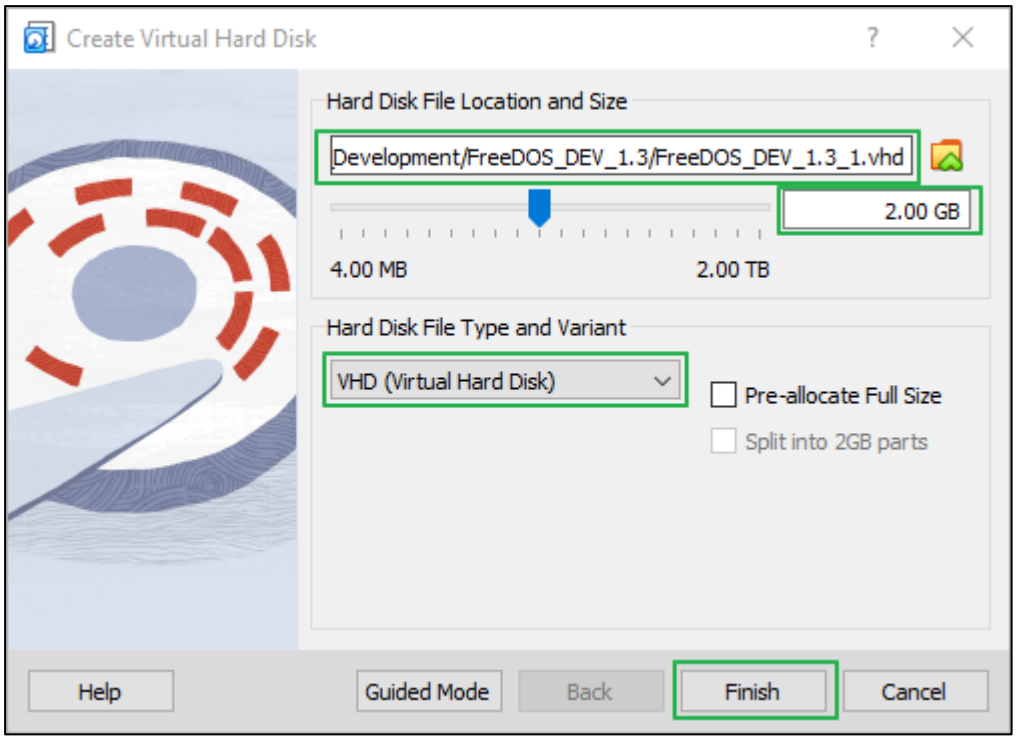


In the next window you can create a hard drive for your machine. DOS does not require a large drive but will expand in size quickly if you are setting up development environments and resource files so I recommend using the maximum drive size allowable of 2GB. Be sure to select the VHD Disk file type. Click “Finish” to continue. Take note that the path for your group and virtual machine are correct. Take note of the virtual hard drive name.

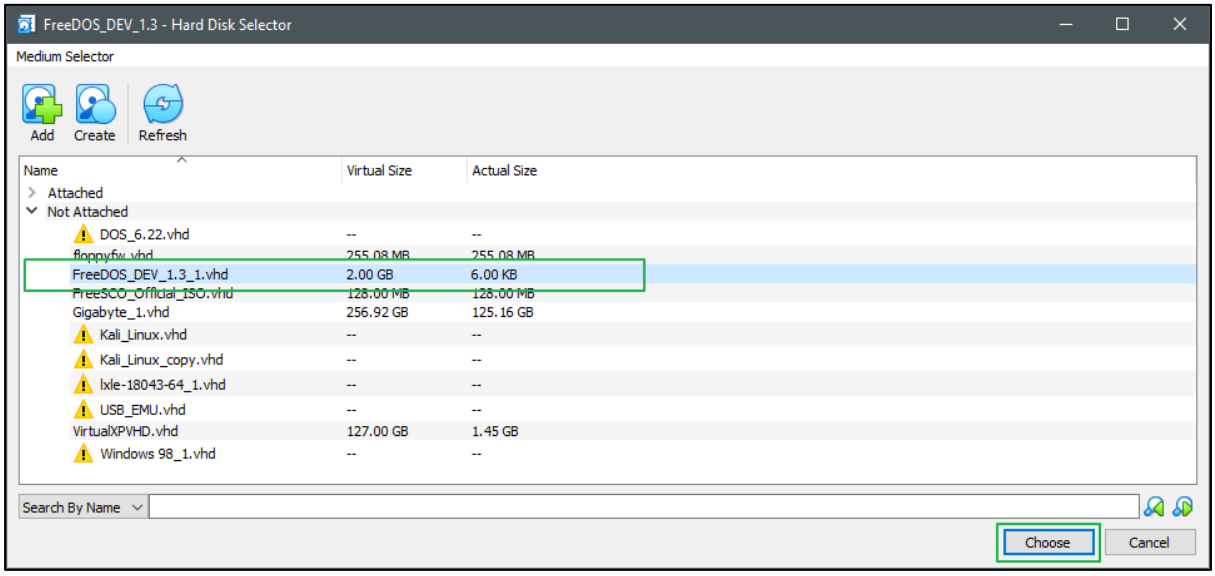
In VitualBox V 6.x select Dynamic drive and for V 7.0 leave [ ] pre-allocate full size unchecked. This allows the drive to expand in size only as needed and saving some drive space on the host OS.

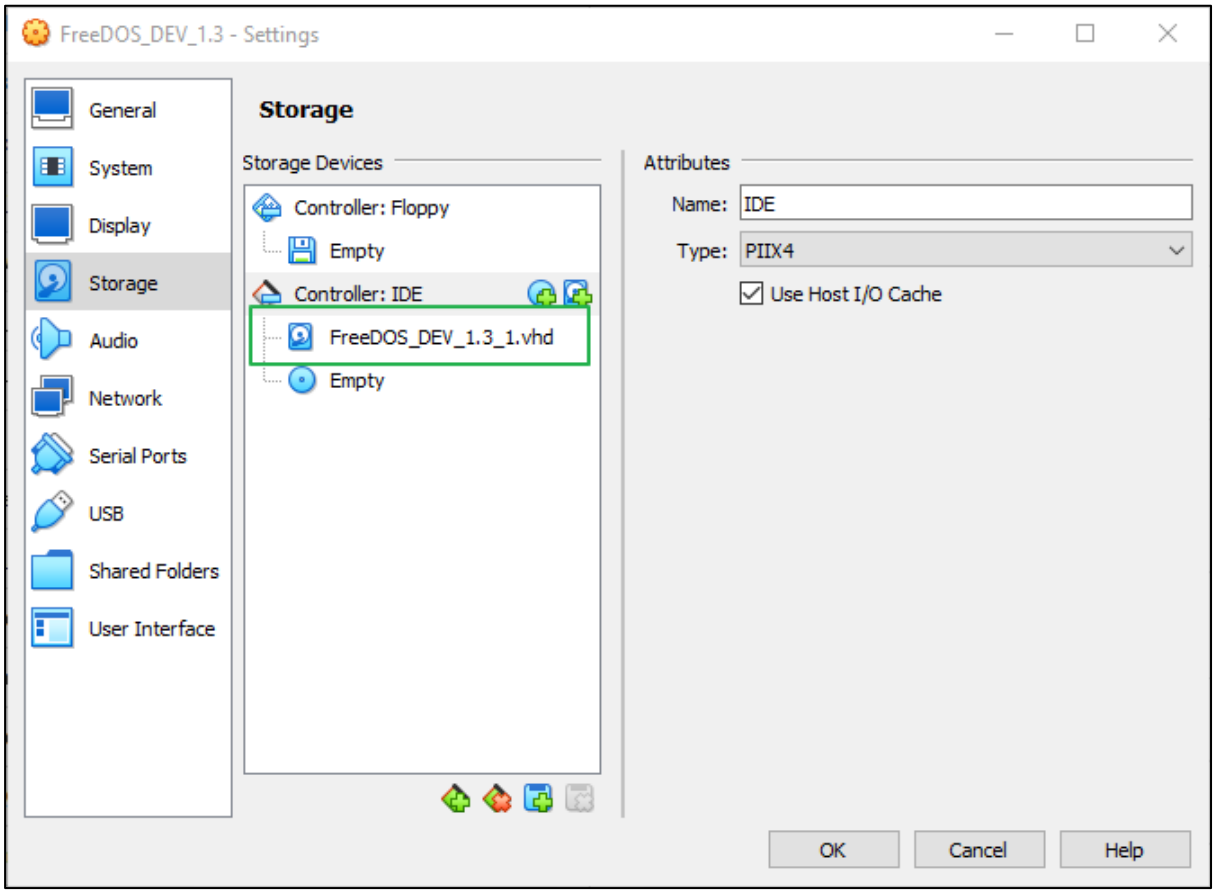
It is possible to use larger hard drives in latter MS-DOS environments as well as FreeDOS V1.3 but I recommend staying with the 2GB drive limit for backward compatibility with some 16-bit applications. Also note that I am using FAT32 whereas some earlier editions of DOS can only use the FAT12/16 file system with a maximum size of 2GB.

Click finish to create the drive.



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 DOS 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 the same 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.

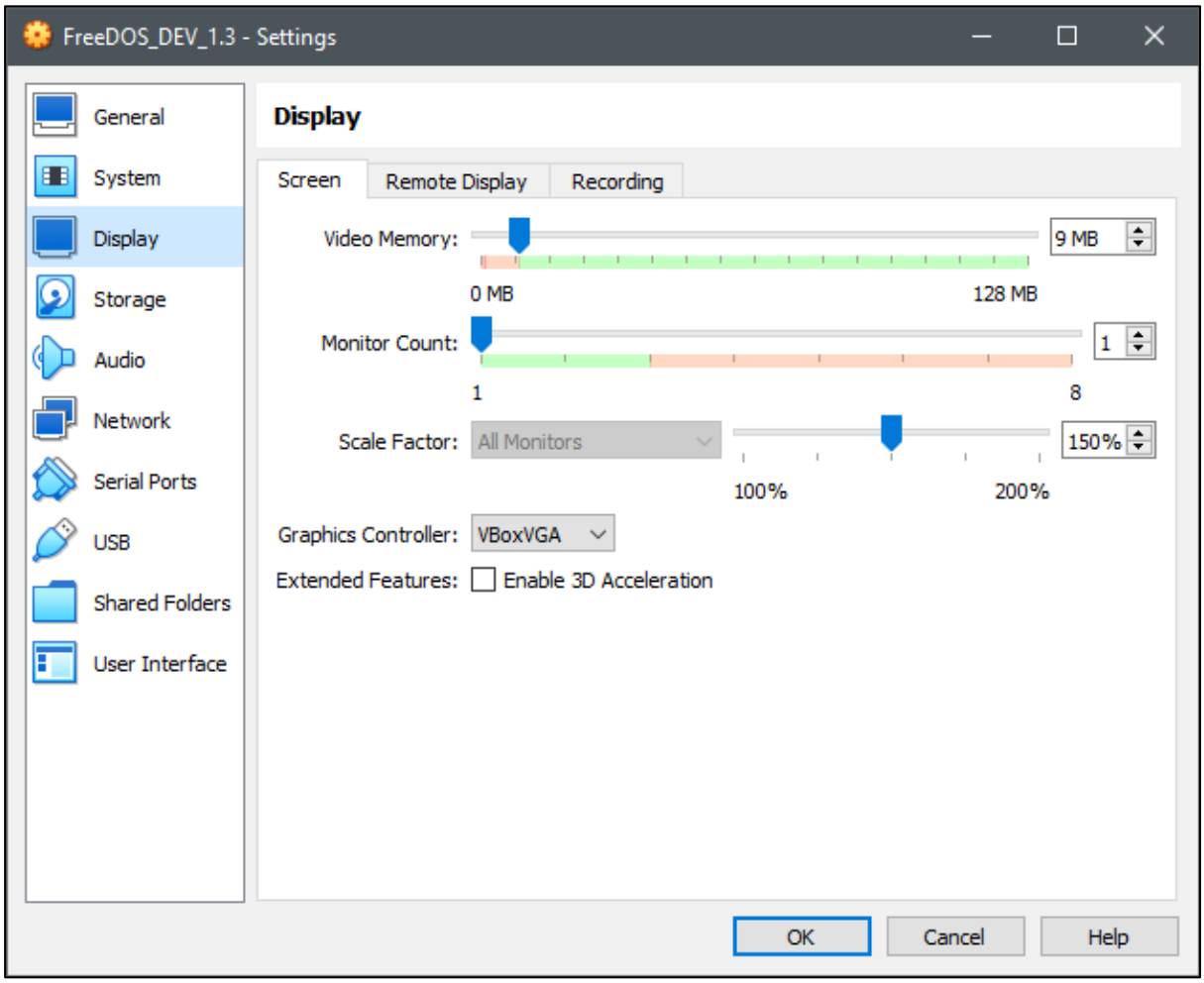
After mounting the new VHD you will see the notice “Not Initialised”. Right click on the notice and select “Initialise Disk”. Be sure to select Master Boot Record (MBR) for the partition type when asked. Next create a New “Simple Volume” and partition the full 2GB of the drive. Select FAT32 as the file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

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.

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.

The default Display Video Memory (9MB) is fine to leave as it is. You can increase it later if your application needs more video memory. The default VBoxVGA is also fine to leave as it is.



At this point you are ready to install the DOS 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.

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 writing to or 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 files 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

A modern DOS install such as FreeDOS also has IP networking capabilities as well as text and basic GUI browsers available. We can use tools such as Http File Server (HFS) on our host machine and then use the DOS web browser in our virtual guest OS to transfer files over http TCP/IP in the browser. I will cover a section on using HFS in a later chapter. Also see Dillo web browser for DOS.

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

See the section “HFS over IP”.

Setting up FreeDOS 1.3

Base install

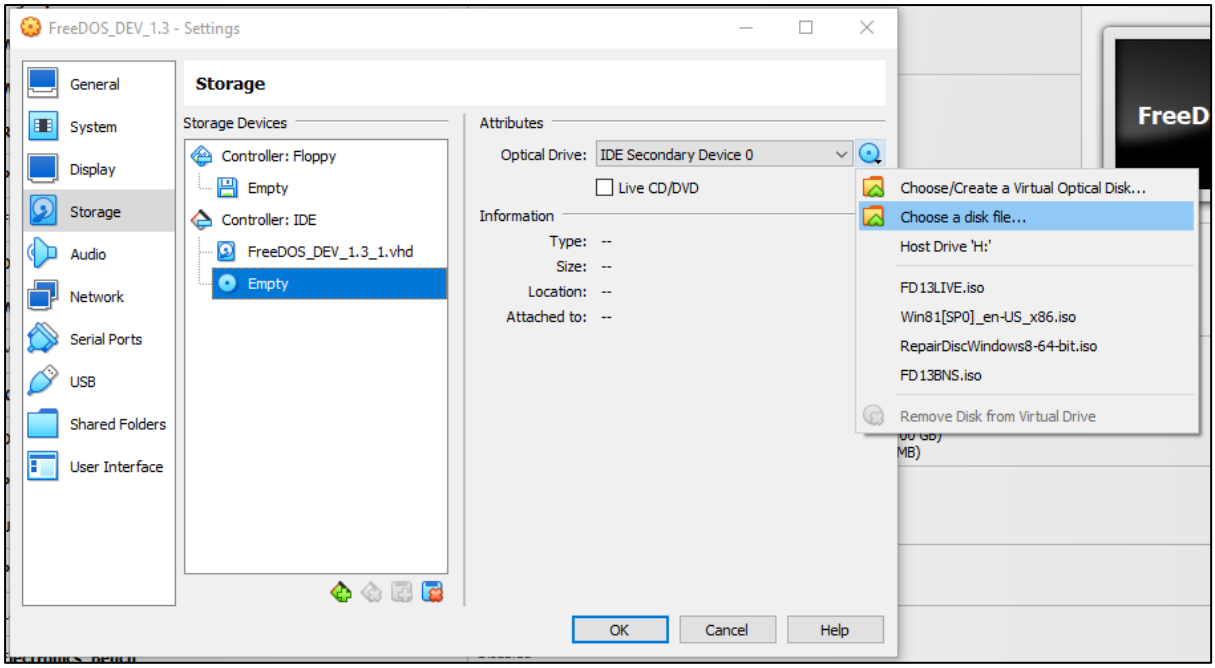
If you have not already done so download both the “FreeDOS 1.3 LiveCD” and the “FreeDOS 1.3 BonusCD” from https://www.freedos.org/download/ and place them in a directory that is easy to find when using the virtual guest. I typically have a dedicated directory of ISO files, or place them in the virtual machine file directory. You will need to extract the ISO file from the downloaded archive.

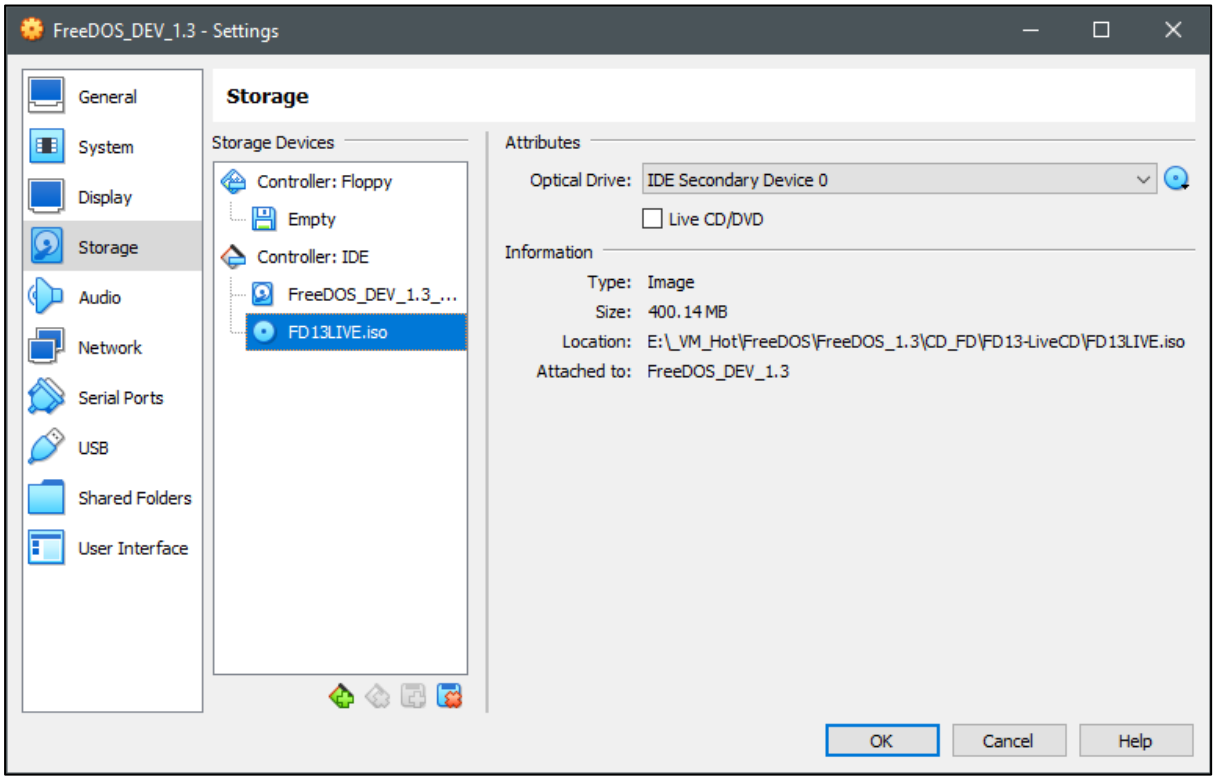
FD13-LiveCD.zip = FD13LIVE.iso (CD drive) and FD13BOOT.img (Floppy drive)

FD13-BonusCD.zip = FD13BNS.iso (CD drive)

Note the bonus CD contains additional applications as well as the development tools for FreeDOS.

From the virtual machine manager open the setting for your FreeDOS guest. Select storage the CD icon under Controller: IDE and select “Choose a disk file...”. Navigate to the location of the FD13LIVE.iso and select OK to mount the CD in the virtual machine drive bay.

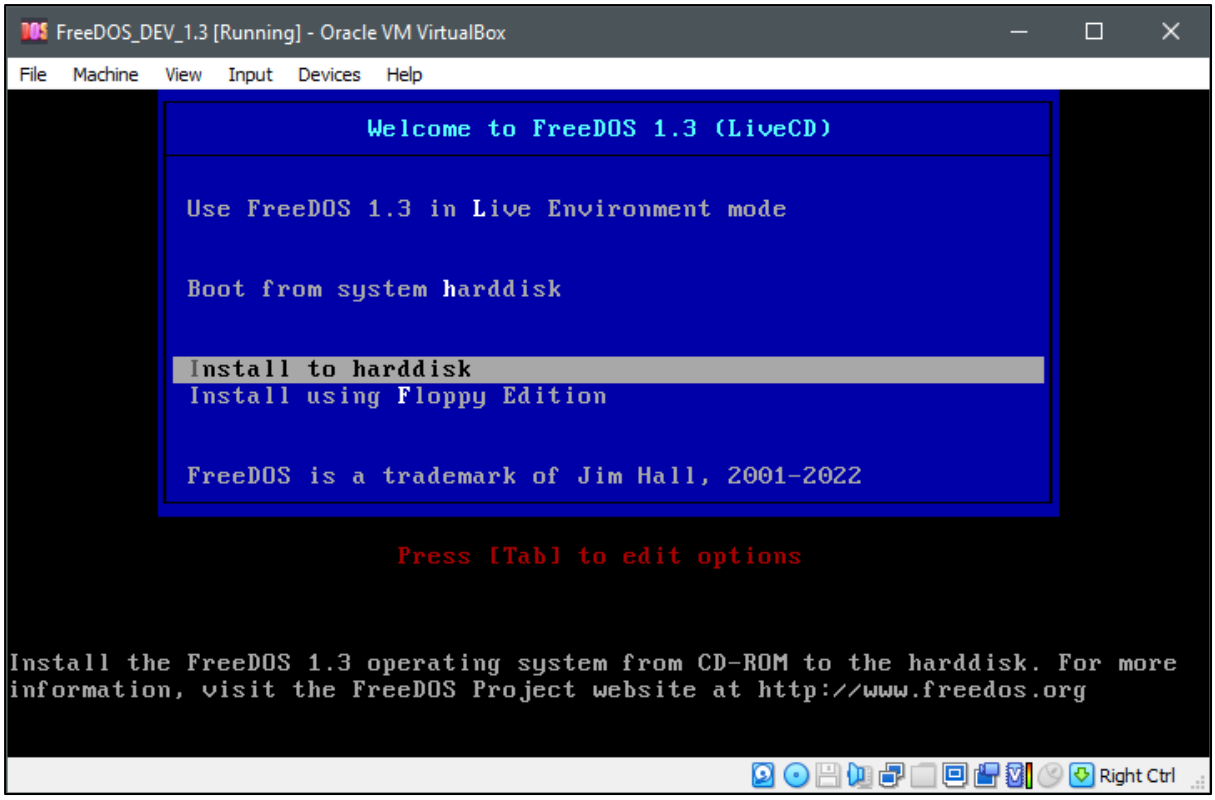




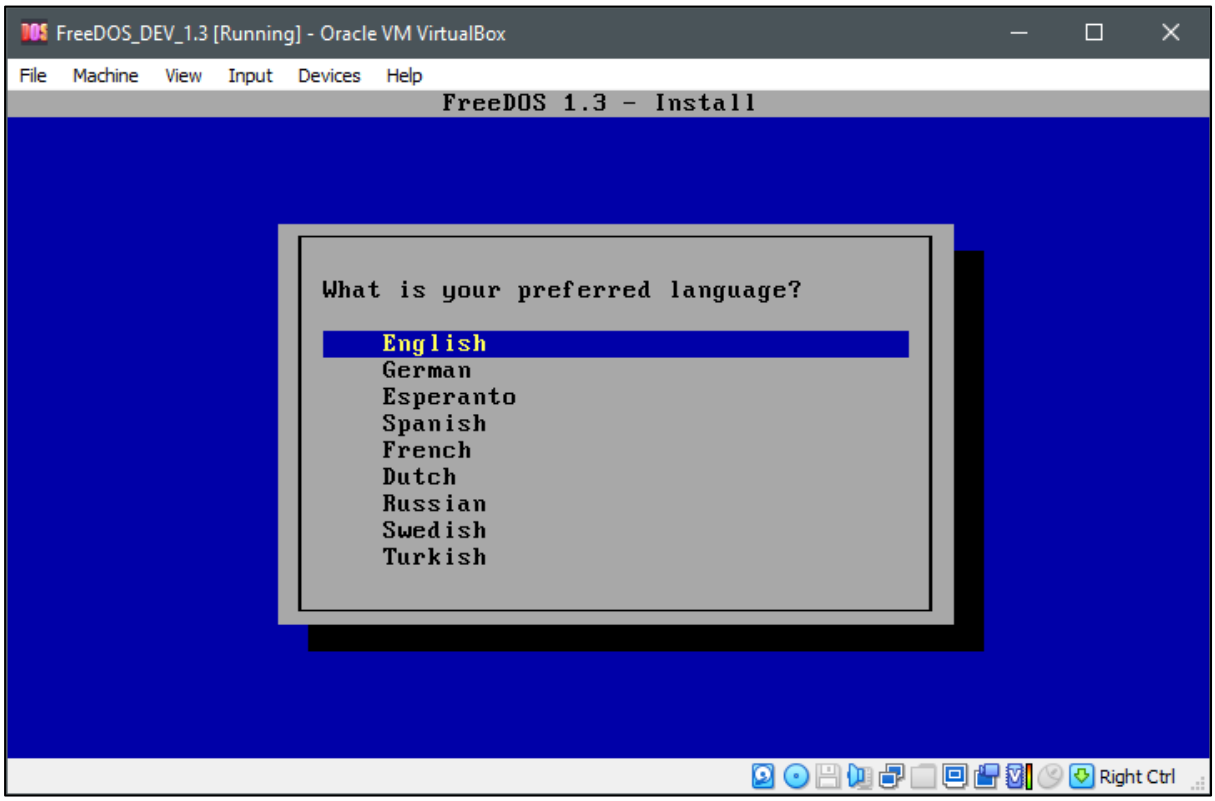
You can now close the Settings window by selecting OK.

We are now ready to begin the install of FreeDOS 1.3. Click on the start icon to turn on the virtual machine. The Live/Install CD (ISO) will be detected and we can begin the install procedure.

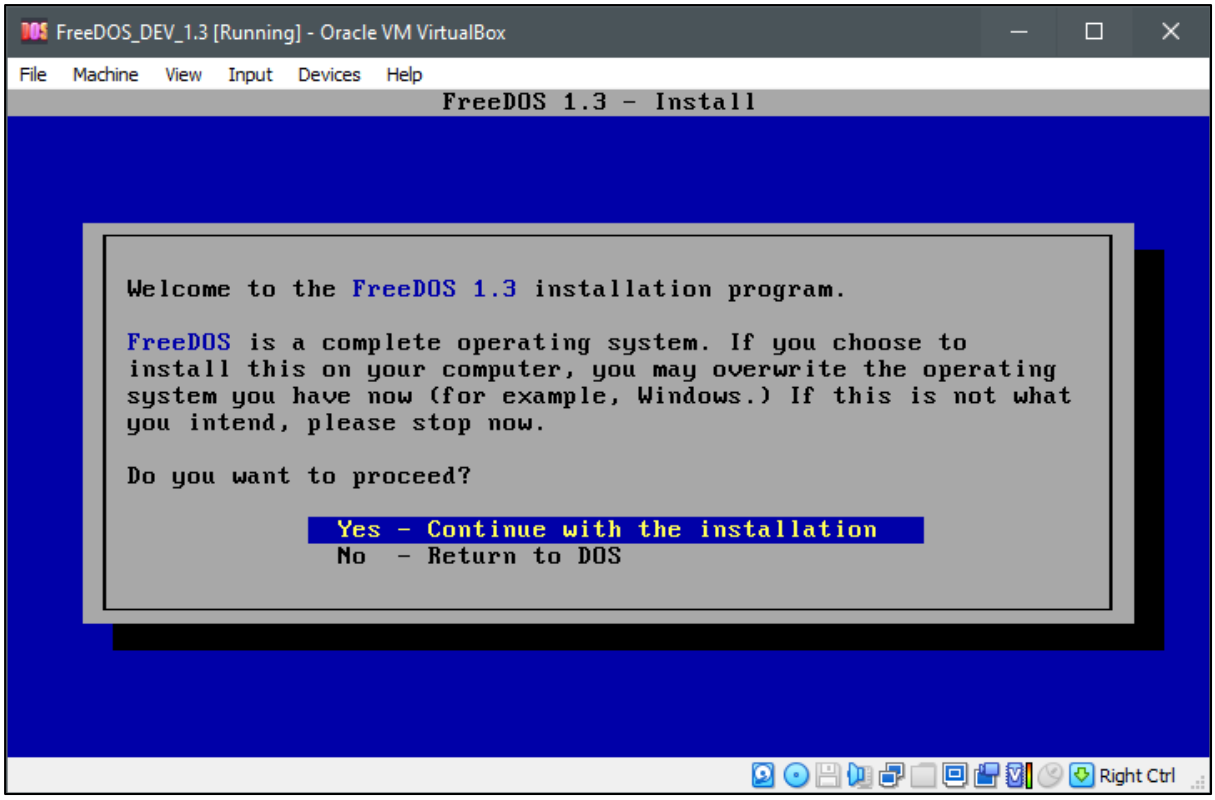
If you want to familiarize yourself with freeDOS before installing you can select the Live Environment Mode, otherwise select the “Install to harddisk” option and press Enter to continue. Note that the timer will select the default Live CD option after 50 seconds if you do not select an option.



Select your preferred language using the navigation keys and press Enter to continue.

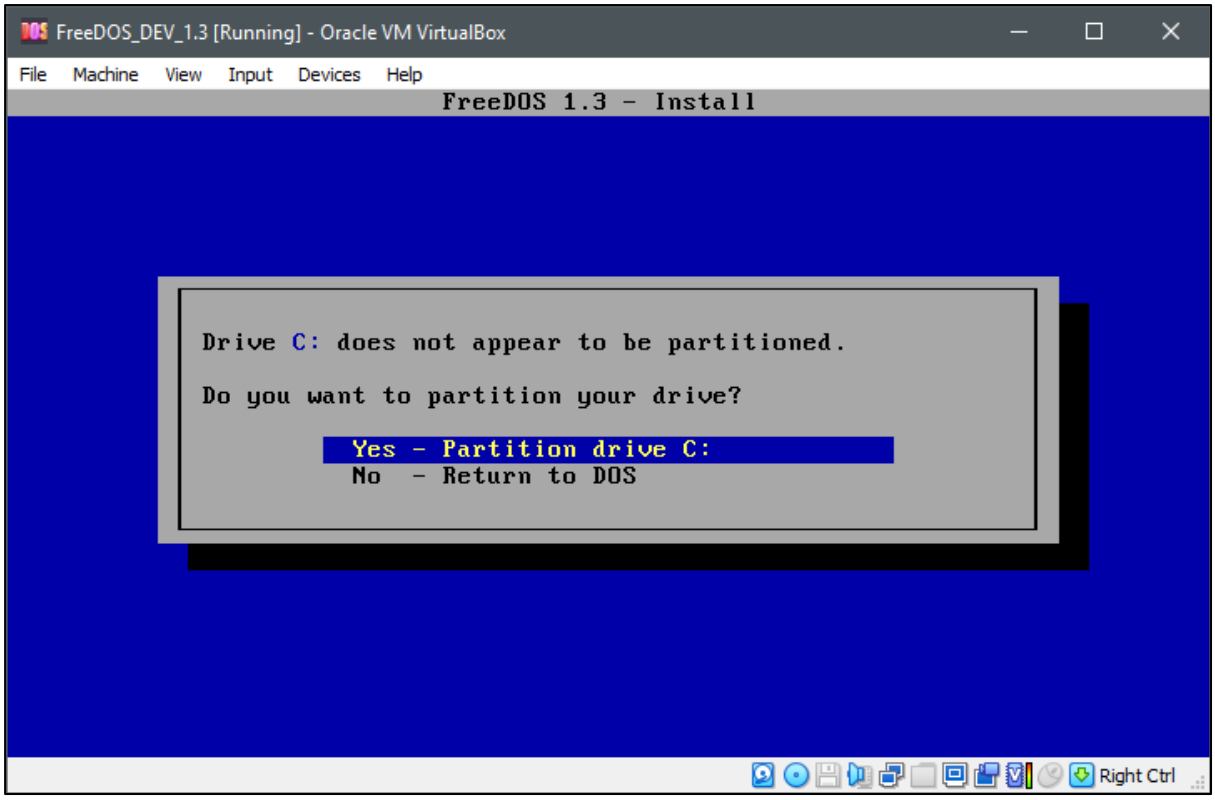


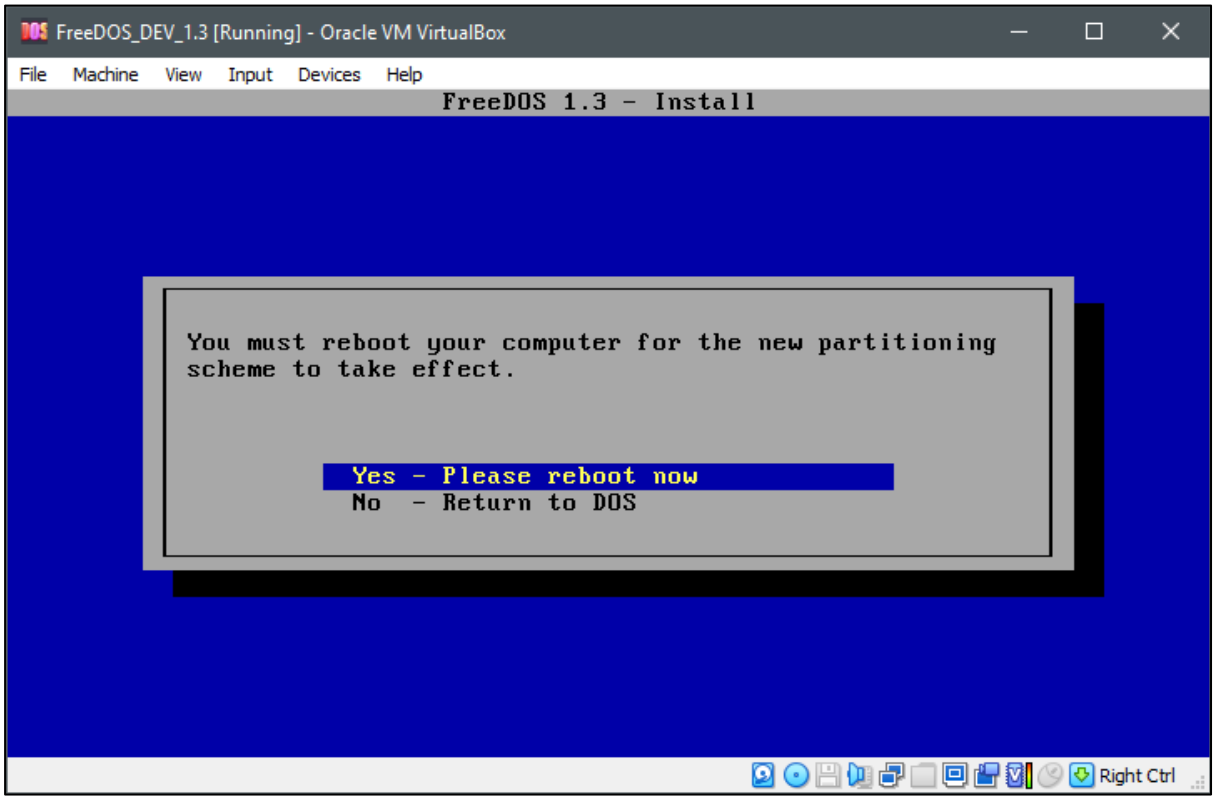
Select YES – Continue with the installation.



Select Yes – Partition drive C:

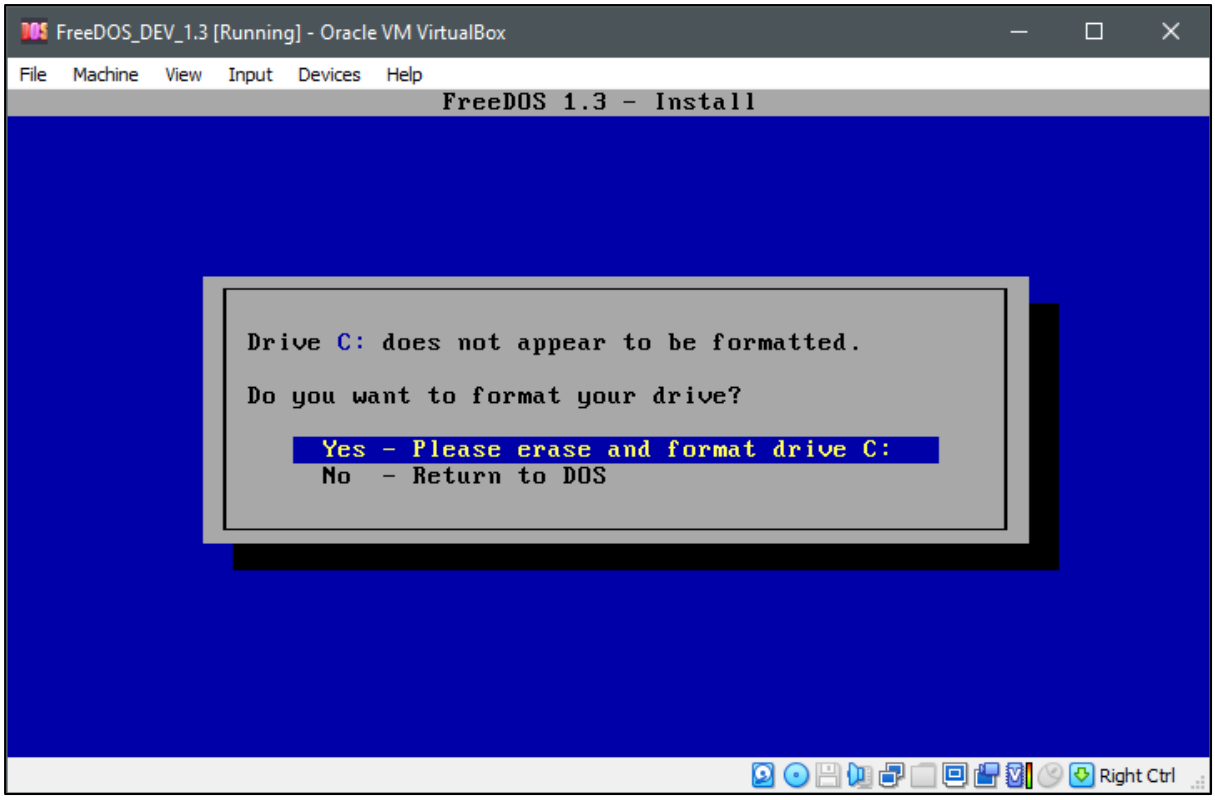
This step will use the built in command line tools to Initialise and partition the disk as we did from the windows Disk Management utility earlier. If the disk has already been partition earlier you will be given the option to erase the disk and re-format it. If this is the first time partitioning the drive a reboot will be required before formatting the drive.

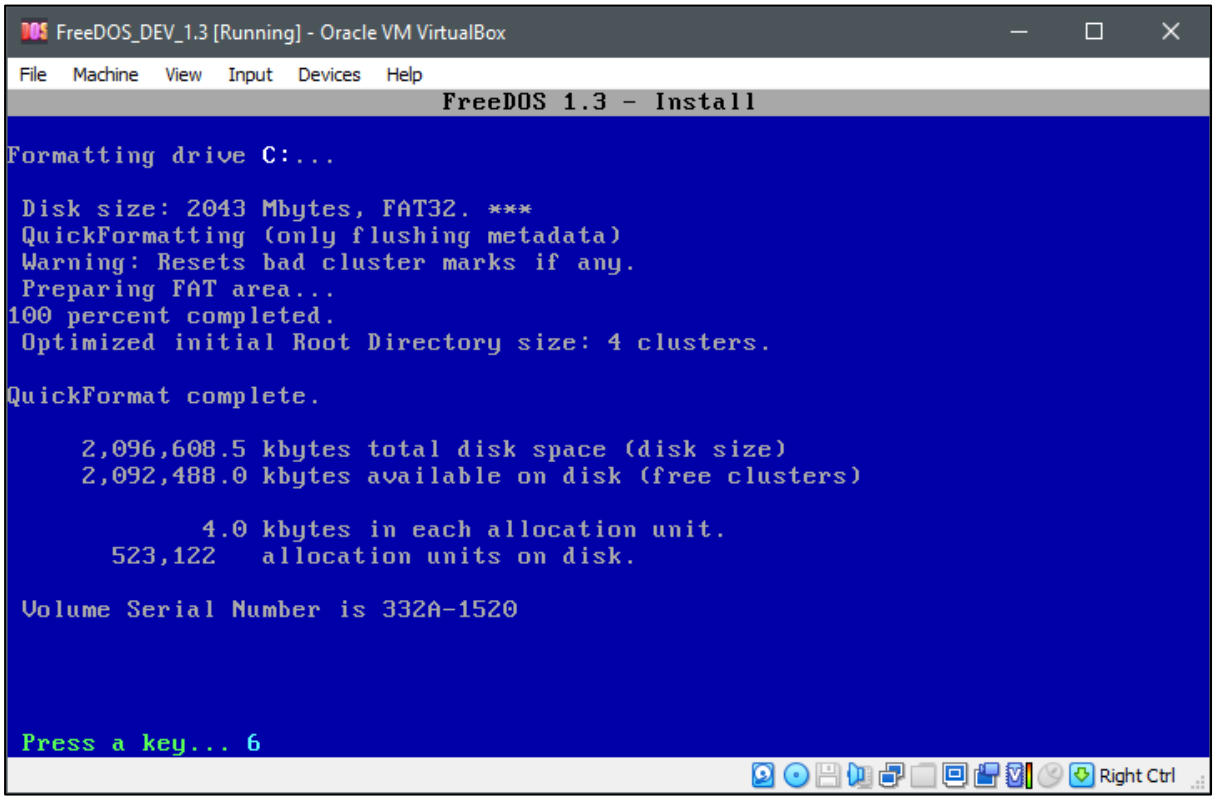




You will need to select “Install to hard disk” as from previously after the reboot..

Or if the drive was already partitioned we will skip to the next screen and continue the format of the drive and then continue the install.

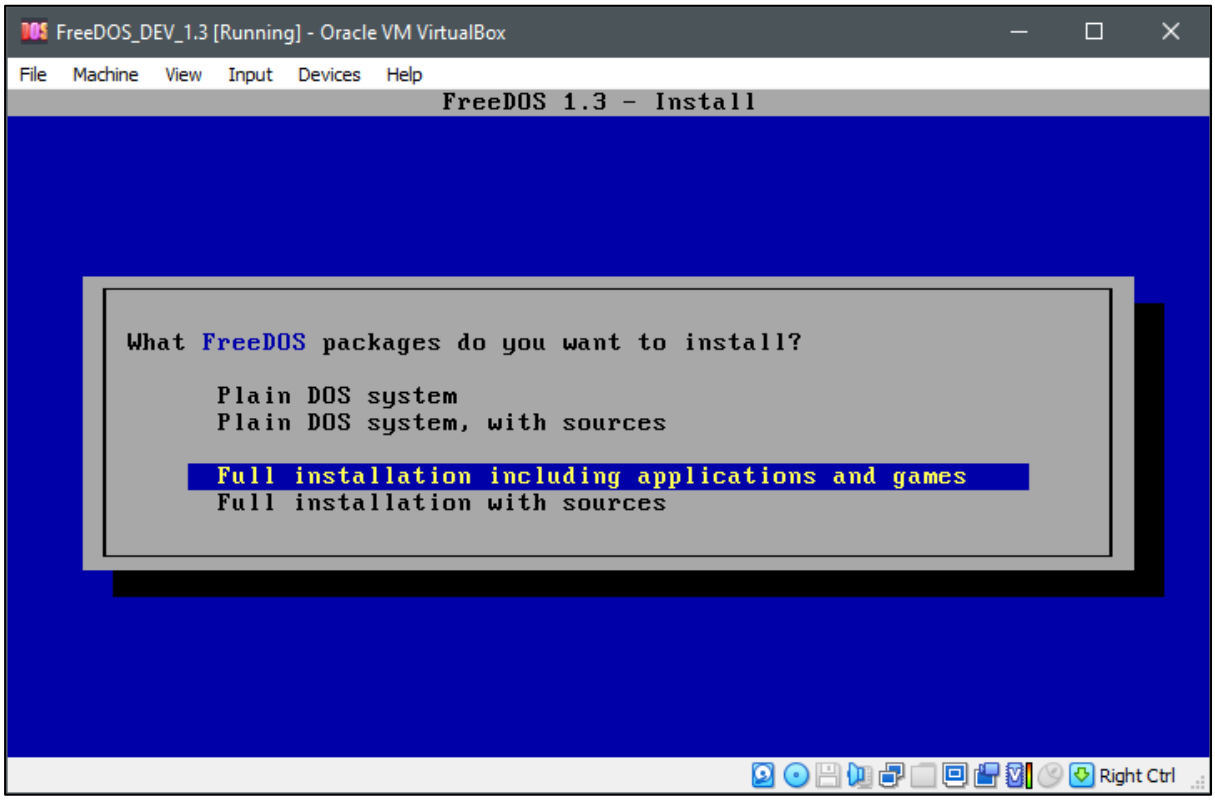




Select your preferred keyboard and select enter.

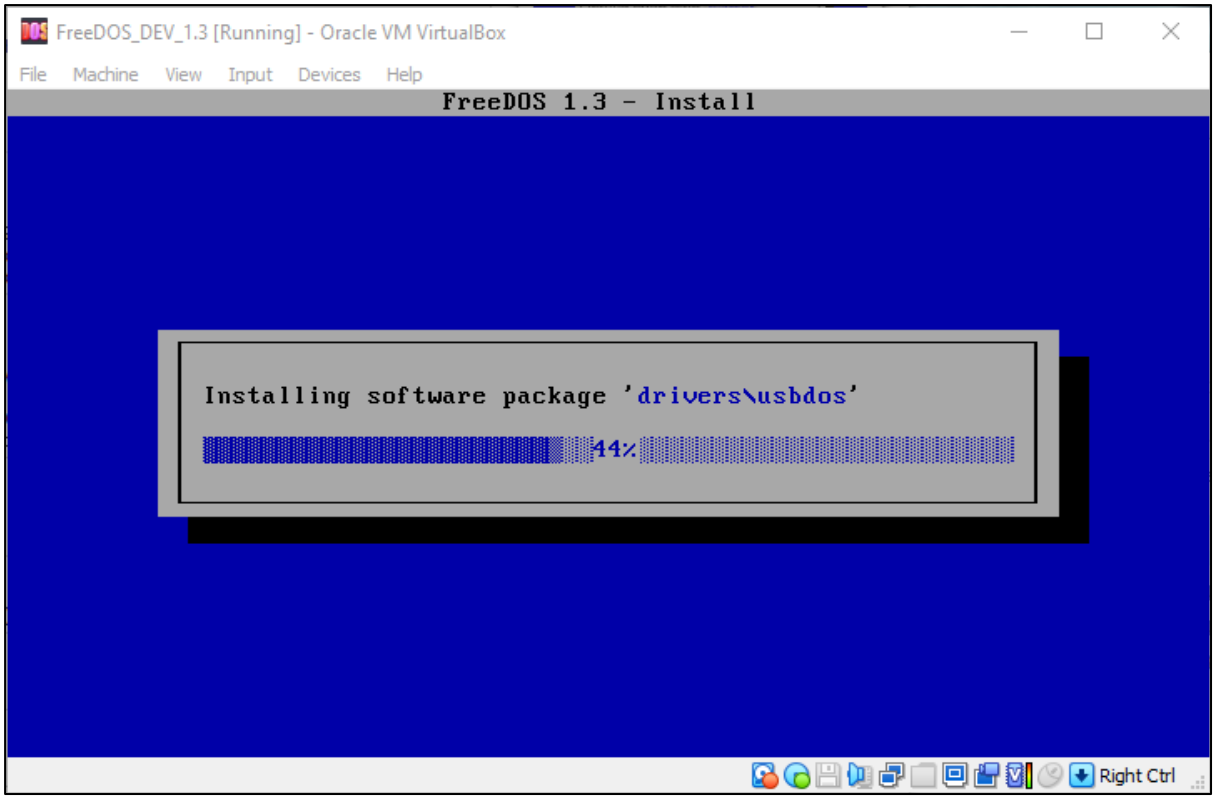
When you get to the screen asking what packages to install I would suggest “Full installation including applications and games”. It requires more disk space but I often find that I end up installing most of the base packages at a later time anyway. If you make a mistake here you can install the

additional packages and sources later but it becomes a little more manual and time consuming. You can also easily remove unwanted packages at a later time.

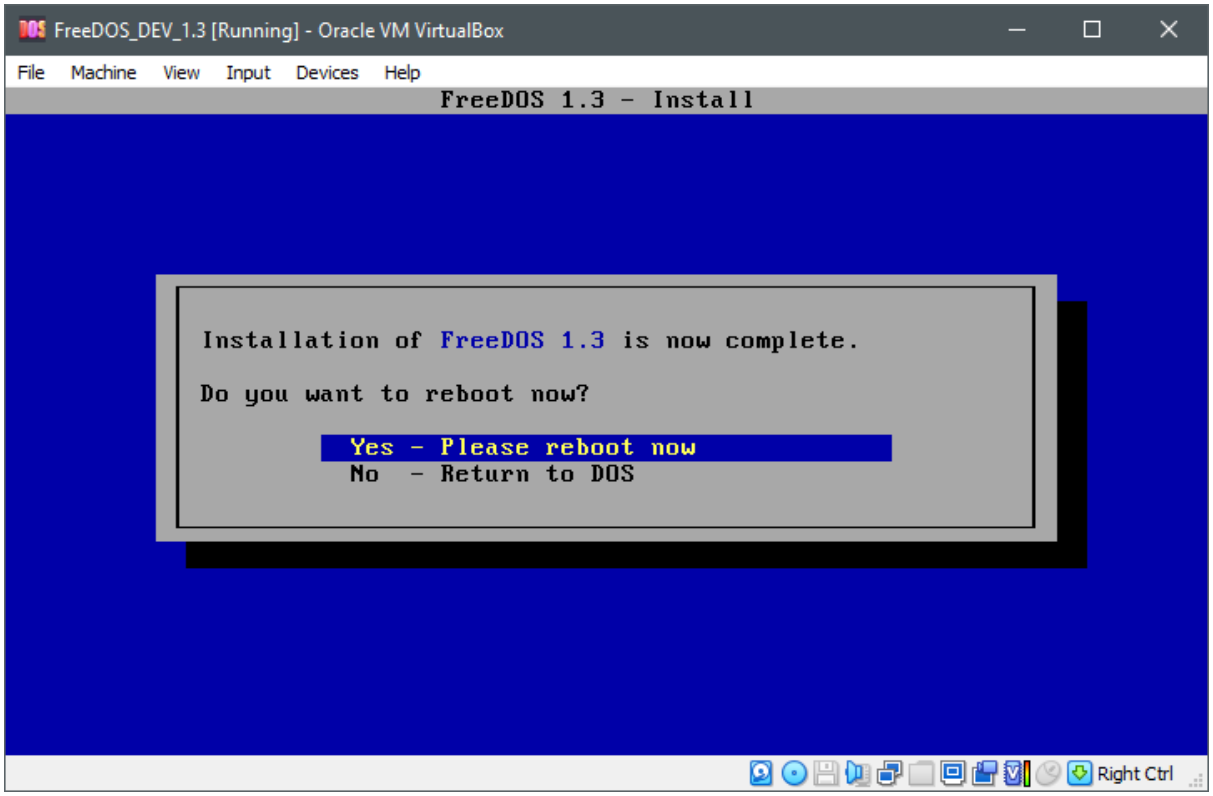


Select Yes – Please install FreeDOS 1.3 and enter to begin the install.

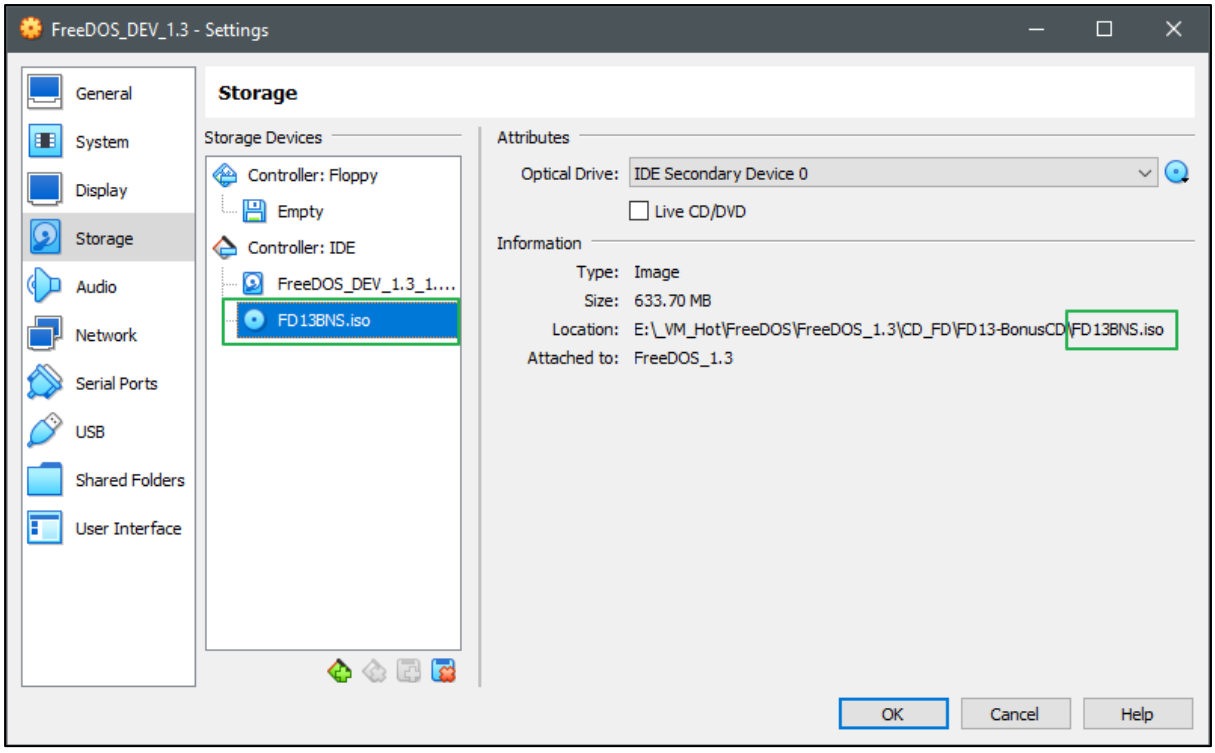
The next step may take a little while depending upon the speed of your host computer.



Next you will be asked to reboot. The FreeDOS file install has essentially been completed at this stage. Note you will need to close and power off the machine just after it reboots.



The Live CD will still be in the drive at reboot and will automatically load. Select “File -> Close” and then “o Power off the machine”. Open settings and select the “Storage” tab and now select the FD13-BonusCD “FD13BNS.iso”. We will need this to install additional development applications when we reboot.



We can now start our FreeDOS Guest OS for its first run. Select the default EMS RAM settings for now.

First run

You will be met with the boot/welcome screen. From here we can add additional applications and software development tools. The next section will focus on a mix of Installing different tools sets, adding some drivers and customisations to assist in DOS navigation.

You may find that the virtual window is a little small and difficult to read. You can “Scale” the window size under the menu options “View -> Virtual Screen -> Scale to nnn%” if it makes it easier. DOS is a text based environment based upon character size, with the number of character across and number of characters deep. This is typically set to 80 characters wide and 25 characters high. When a graphical application is running in DOS the display resolutions of the video adaptor can be used. Vitualbox will typically resize the display to match the size of the display resolution so you will encounter different virtual window sizes for different applications. DOS is meant to run in “Full screen Mode” so using the virtual guest in full screen mode is the only way around this, or use the scaling and accept the occasional change in window size when in emulated windowed mode.

Shutdown

FreeDOS comes with many command line utilities. You can shut down and reboot FreeDOS by typing one of the following to commands to the command line followed by Enter.

FDAPM `poweroff` – Do a complete shutdown of DOS.

FDAPM `warmboot` – Do a "warm" reboot of DOS.

FDISK `/reboot` – Do a "cold" reboot of DOS.

Check the FDAUTO.BAT for a list of "aliases" for the above commands.

App managers

From the command line type in **FDIMPLES** [Enter] to invoke the App Manager. This will automatically read from the CD when it is inserted. You will need to have the bonus CD mounted when using FDIMPLES.

If you are uncertain of how to navigate, type FDIMPLES /? to invoke the help file. MS and MS-DOS like systems typically use the "appname /?" extension for help, but Unix extensions such as "-h, --h, --help" are also common.

Another network package manager **FDNPKG** can also be used for installing and updating from online repositories.

Browse through the FDIMPLES package manager application and view the base installed applications from our FreeDOS installer/setup. We can add and remove most FreeDOS approved packaged from here. The Bonus CD is essentially an offline version of the FreeDOS package repository found at https://www.ibiblio.org/pub/micro/pc-stuff/freedos/files/repositories/1.3/pkg-html/

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

The FDNPKG tool can be used to install or remove packages from the online repository that may not be included on the CD as well as Update to the latest revision of an application from the CD.

FDNPKG is an online update tool for FreeDOS. It can install/remove software on a FreeDOS system using network repositories or local zip packages. Every package is a self-contained archive in zip file format with a specific layout and other requirements.
https://freedos.sourceforge.io/wiki/index.php/FDNPKG

FDNPKG /? for use instructions.

In FDIMPLES you will notice that a large number of "Editors" have been installed with the full base package. Everyone has individual preferences with text editors but I find FED and SETEDIT the 2 most useful for application development as they are about the closest you will get to an IDE in DOS. I use FED almost exclusively in this guide.

Also take note of the "Development" section as I will work through the install of many of these tools in a later section.

Also note that the DOS sound drivers for VirtualBox sound emulation do not exist, but we can use some Sound blaster drivers that will offer sound but are somewhat buggy and inconsistent. So if you

plan on running games or other audio applications you will run into some limits on VirtualBox. Virtual machine environments such as DOSBox and BOCHS are more suitable for legacy gamming.

Restore points

When experimenting with customizations of an OS it is worthwhile creating restore points before hand so if it does not go as expected you can quickly roll back and start again.

If you are using original legacy hardware you will need an additional partition or hard drive for this process. You can simply copy and paste the entire OS drive and files to a backup directory on a separate partition, place the entire system drive into a zip archive on a backup partition or use a drive imaging tool to make a complete bit by bit copy of the system drive. Space can sometimes be limited in this scenario so I suggest keeping only the most important restore points in a safe place. For example if you have created a full install with required apps and all is working well then keep this for the future as a quick disaster recovery.

With a Virtual Machine environment such as VirtualBox this restore point process is relatively quick and easy. In your guest OS directory ensure the guest OS is NOT running. Create an archive of the virtual machine config files and the system virtual hard drive using 7-Zip. Give them a name including the time stamp of when the backup was made i.e. 2023-07-24_13 (YYYY-MM-DD-HH24). Copy this archive to a backup directory in your virtual machine directory. Any time that you need to roll back to a previous point, delete the virtual machine files and replace them from the backup archive with the time stamp required. The same as with the Hardware description you can later delete incremental backups and keep only the most important or recent restore points.

DOS customisations

Long File Names

FreeDOS, MS-DOS 6.22 and prior do not natively support long file names aka names longer than 8 characters or file extensions greater than 3 characters. A convention called 8.3 file naming is supported where long file names are truncated to six characters followed by the Tilde “~” and an index number. For example “mylongfilename.txt” will be reduced to “mylong~1.txt”.

FreeDOS does have the DOSLFN.COM tool available in the default install but I do not recommend using it unless you have a specific reason to do so. The long file name utility was not introduced until MS-DOS 7.0 and 7.1 which corresponds to compatibility with Windows 95 and 98. It is better to stay with the 8 character convention to keep compatibility when transferring files to other DOS machines. This 8 character convention also becomes very important when writing DOS compatible applications. Leaving DOSLFN.COM turned off forces the 8 characters convention and eliminates accidental long names.

32-Bit expanded memory and Extenders

Expanded Memory Modules.

Computers use a variety of different parts of the hardware's available memory. In DOS most applications will run in the limited Conventional memory block of 640kb. UMA allows access to memory from 640kb to 1MB. HMA allows for an additional range of memory from 1MB + 64kb. We also have HMA, EMS and XMS with a variety of possible memory usage configurations. I won't go into detail as it becomes quite complex.

With the advent of the 32-bit 386 hardware they also needed to be able to make use of the 32-bit CPU and memory which is not native to DOS 086 or 286 machines. They used some trickery by creating 32-bit memory extenders to allow the 16-bit DOS pointers to access the 32-bit memory range.

You will encounter a number of different EMM386 drivers and as much as they are created to be compatible, they do not always work well with every application. It is useful to keep a small collection of EMM386 drivers handy just in case.

Some common 386 Memory managers:

- EMM386 – MS-DOS
- JEMM386 – FreeDOS
- QEMM-386 – Quarterdeck
- ++

You can find some more explanation at the following wiki:

https://www.vogonswiki.com/index.php/DOS_memory_management

DOS Extenders

The original DOS 286 was a 100% 16-Bit "Real Mode" OS, As such only ran on 16-Bit CPU hardware. With the coming of the 386 PC hardware the DOS OS, still fundamentally 16-Bit could not make full use of the 32-Bit Hardware. To get around this problem they created the 32-Bit "DOS Extenders" to allow the running of 32-bit protected memory mode applications. Windows 3.1 was originally built on top of the MS-DOS Extended memory module. Although sharing some source code similarities , these DOS extenders should not be confused with the Windows 32 OS kernel in Windows 95. Some Windows 3.1 DOS 32-bit applications can be run on DOS but note they are DOS 32-bit applications and not Windows applications.

Today there are many different compatible DOS 32 extender replacements. Not all are equal and not all extenders work with other applications. Typically an application is coded around a specific DOS extender and that DOS extender will often be required at run time.

You will in time find that you will have a small collection of DOS extenders for different applications. They are typically integrated into the system using the DOS config files, but also compiled into the executable or used as a library in the application directory. I would recommend placing the DOS Extender in the same directory as the application executable. It's a little heavier on disk space usage, but avoids conflicts and complex config files.

Some common 32-bit DOS extenders:

- DOS/4G[W]

- DOS/32
- GO32
- HDPMI32 – Windows MS-DOS

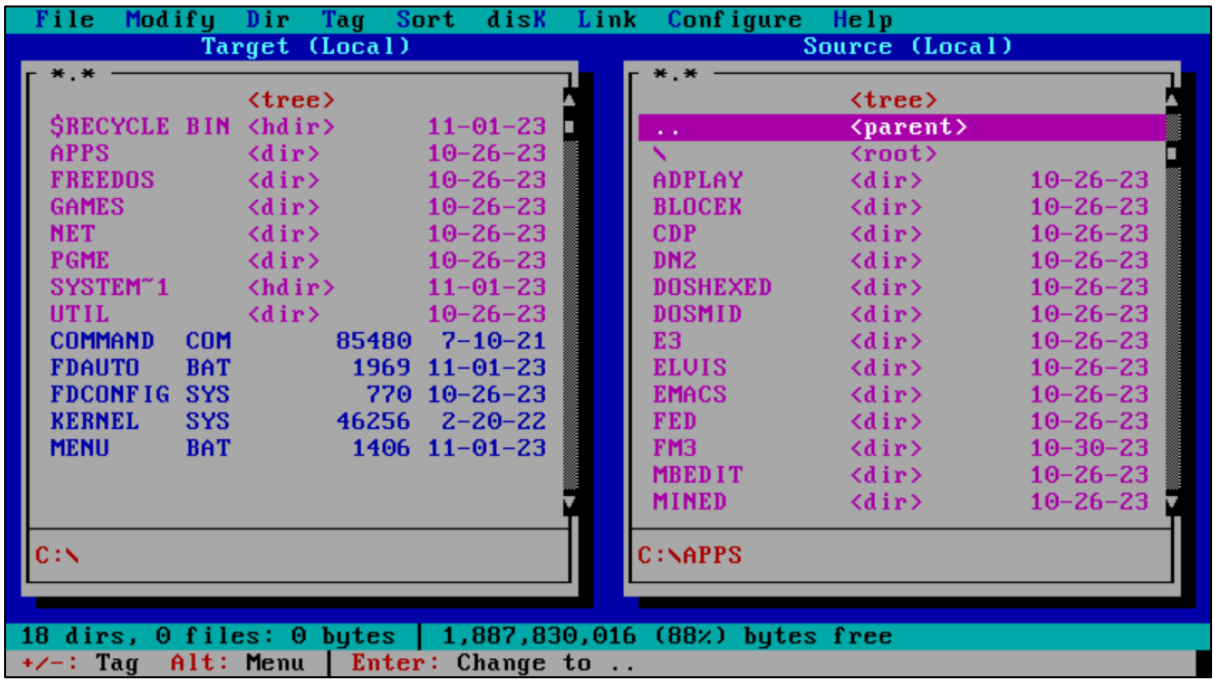
You can find some more explanation at the following wiki:

https://en.wikipedia.org/wiki/DOS_extender

Essential navigation

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. 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.bttr-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

[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:\APPS\FM3 into the destination Directory followed by Enter. Type Y to install FM3.

After the install the current working directory will be C:\APPS\FM3. Type FM3 and 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 FreeDOS has the mouse drivers installed by default, so you can also use the mouse for navigation if that is more convenient. Note that DOS mouse has no scroll wheel so you will need to use the scroll bar at the right.

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

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 the built in FreeDOS “EDIT.EXE” or FED. If you chose the full DOS install then FED Folding Editor will already exist in the APPS directory. If not insert the FreeDOS Bonus CD (ISO) and use FDIMPLES to install 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.

Before becoming familiar with FM3 go to the next chapter “**Application launch BAT**” and follow the instruction to create a batch file to launch the application.

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:\APPS\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:\FREEDOS\LINKS` and open FM3.BAT in a text editor (See: "**Application launch BAT**").

Add "`SET PATH=%path%;%dosdrv%\APPS\pkz204g`" before the call to `FM3.EXE`

FM3.BAT

```
ECHO OFF
REM File Maven 3 application launch.
REM Set a temporary path for PKZIP
SET PATH=%path%;%dosdrv%\APPS\pkz204g
REM Launch File Maven 3
CALL C:\APPS\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

http://www.freedos.org/books/get-started/7-bat-files/

http://wiki.freedos.org/wiki/index.php/Games

Batch files are the glue that brings a DOS OS together in a way that is useful to the user. There is no strict right or wrong as to how you decide to customise your DOS environment as it really comes down to personal preferences. DOS can be become messy very quickly so it is important to attempt some consistency in your methods of setting up your work environment. This may require several practice installs and setups before you have a feel for how you want your own environment to look and feel.

The two most important files that you will work with are CONFIG.SYS (FDCONFIG.SYS) and AUTOEXEC.BAT (FDAUTO.BAT). These two files set up the initial operating system environments and are the most important configuration files in your system. They can also be a point of confusion as they can vary quite a great deal from system to system. There is no one size fits all when it comes to the AUTOEXEC and CONFIG files.

Note that FreeDOS uses FDAUTO.BAT instead of AUTOEXEC.BAT. Some installers will add entries to the AUTOEXEC.BAT file and create an AUTOEXEC.BAT file if it does not exist. You may have to migrate the new entries from autoexec into the FreeDOS FDAUTO.BAT file. Always keep a current back up of each file before modifying in case you make a mistake. You can always boot into the system from an attached floppy boot to replace the FDAUTO or CONFIG from the backup if you do make a mistake.

CONFIG.SYS and AUTOEXEC.BAT set up the systems “Global” environments that are applicable to all applications running on the system. Additional batch files (BAT) set up secondary or temporary environments that are additional to the global environments. It is very easy to duplicate or create conflicting environment setting between our global settings and the temporary settings in a batch file. Always have an understanding of the variables and environment path settings in your CONFIG and AUTOEXEC files when creating batch files. Sometimes it is appropriate to set global environments, and at other times it is better to leave the environments out of the CONFIG and AUTOEXEC files and set them on a case by case need in the start-up batch file for the application.

When we call an application to start from the command line the system first looks for the NAME.BAT followed by the NAME.COM and then NAME.EXE in that order, so if we type and application name FM3 into the command line, the command.com will first look for FM3.BAT and if not found, FM3.COM and last FM3.EXE. If we type the extension FM3.EXE then that file will be selected first. Take great care with regards to duplicate executable names and especially with regards to batch file names that launch an executable application. If a system has two batch files or executables of the same name it will launch the first encountered.

As an additional note please be aware of the difference between DOS COMMAND.COM and the Windows 9x COMMAND.COM and CMD.EXE in Windows NT versions, as they are not the same. COMMAND.COM is a native text only terminal environment found in DOS, The Windows 9x COMMAND.COM can run in both Text Mode and GUI emulated mode and CMD.EXE is graphical Windows command line “emulator application” that has no relationship with DOS.

COMMAND.COM uses batch.BAT file scripting whereas CMD.exe uses Comand.CMD scripts. CMD.EXE can run .BAT files as a legacy script for backward compatibility but is very limited. The batch and command script capabilities vary between different DOS COMMAND.COM implementations. The scripting commands between DOS COMMAND.COM, Windows COMMAND.COM and CMD.EXE have some syntax similarity but are NOT compatible.

Many people and help sites on batch scripting have a tendency to not differentiate between the above which can be very confusing when a script fails to run. And many sites also incorrectly label the Windows command interpreter as DOS. Take great care when looking for help on DOS batch programming and check that it is a DOS batch help document.

Launching our DOS applications with a batch file give us a greater degree of control over how the system and the application will run. If we type an application into the command line we will be met with “Bad command or filename – “name”” if we are not in the application directory or if the application is not in the system search path. Rather than creating system paths to every directory and application in our system we can place a launch.BAT in the system path for each application that we want to call from the command line.

To see all of the current global environment settings type SET into the command console and then Enter.

```
C:\>SET
CONFIG=1
DOSDIR=C:\FreeDOS
COMSPEC=C:\FreeDOS\BIN\COMMAND.COM
DOSDRV=C:
LANG=EN
TZ=UTC
PATH=C:\FreeDOS\BIN;C:\FreeDOS\LINKS
NLSPATH=C:\FreeDOS\NLS
HELPPATH=C:\FreeDOS\HELP
TEMP=C:\FreeDOS\TEMP
TMP=C:\FreeDOS\TEMP
BLASTER=A220 I5 D1 H5 P330
DIRCMD=/P /OGN /Y
COPYCMD=/-Y
OS_NAME=FreeDOS
OS_VERSION=1.3
AUTOFILE=C:\FDAUTO.BAT
CFGFILE=C:\FDCONFIG.SYS
CDROMID=FDCDX001
CDROM=E:
MTCPCFG=C:\FreeDOS\MTCP.CFG
WATTCP.CFG=C:\FreeDOS
EMAIL=anonymous@freedos.org
C:\>_
```

If you only want to check the current system path you can use “`ECHO %path%`”.

```
C:\>ECHO %path%
C:\FreeDOS\BIN;C:\FreeDOS\LINKS
C:\>_
```

As you can see from this we have 2 system paths already set by default:

`C:\FreeDOS\BIN` (The main FreeDOS system executable files.)

`C:\FreeDOS\LINKS` (This directory contains shortcuts and start up batch files for applications.)

We can make use of this default "LINKS" directory to store our batch files for launching applications. Any batch file located in this system path can be called from any current working directory in the command prompt.

We could also create our own special folder named APPSTART for storing our app start up batch files as long as we add the directory location to our system paths in AUTOEXEC.BAT. We already have a "LINKS" folder to use so this is not necessary. It's is also best to keep the number of system search paths to a minimum to avoid system lag due to excessive searches.

Navigate to the directory `C:\FreeDOS\LINKS` (`CD \FreeDOS\LINKS`). If you are using FM3 exit FM3 so that you are in the command prompt. Type EDIT into the command line and then Enter to invoke the default FreeDOS editor. Press Alt + file and select New to create a new ASCII (txt) document.

Add the following lines into the editor:

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

Select Alt + F and Save As...

In the name line *.* use backspace <- to remove the *.* and then type FM3.BAT and press Enter to save.

The title bar of the text document should now contain the name "FM3.BAT".

Select Alt + F and then Exit the editor. Confirm that the FM3.BAT file is in the LINKS directory by typing DIR and pressing Enter.

You now have the basic "Batch File" application launcher set up for FM3 (File Maven 3). You can start File maven at any time by typing FM3 into the command line. This is the most basic batch file for calling an application to start.

When an application has been started via a batch file the called application runs in a DOS command instance created by the batch file, when we exit the application it will return to the batch file and then exit back to the system prompt. We can chain load multiple batch files and applications in this way as "Child Instances" of the batch file. Each time we "exit" we will return to the previous "Parent" batch file. If you call a new COMMAND.COM instance you will need to "Exit" the command prompt to return to the parent batch file or application.

Caution!
Avoid calling parent or sibling batch files as you can find yourself in caught in deep parent/child re-entrance loops and difficulty "unwinding" back to the original parent application.
This behaviour in batch files is no different to that of SUB Routines and Functions in common programming language source code. In effect a batch a file is a "SUB Routine" or "Function" of the COMMAND.COM application. You have to treat batch files as function calls under the system as main(). The OS is for all intents and purposes a C language development environment int main(){...} with calls to SUB Routines, Functions and scripts.

You will see that some other application installers have also created batch files to launch their applications. You can use "MORE appname.bat" to view the batch files, or just select the file and press Enter if using FM3. These are a little more advanced launchers and check for the existence of the application first and offer a message if the application is not found. The only difference to the above FM3.BAT is that FM3.BAT ends quietly without the notification.

The other part you will see is that the batch file passes on any additional command line arguments to the application as variables. %1 %2 etc. If you use an application such as FED that requires additional command line parameters then be sure to include the extra variables to pass on the parameters.

```
@ECHO OFF
REM File Maven 3 application launch.
CALL C:\APPS\FM3\FM3.EXE %1 %2 %3 %4 %5 %6 %7 %8 %9
```

You will also see that the link used to call the application uses the system variable %dosdrv% instead of C:

The following template shows some of the common methods of launching an application via a batch file. You can't use it as it is and you will need to select an individual method that you feel most comfortable with.

```
@ECHO OFF
CLS
REM Generic application launch template
REM
REM SET Option 1
REM SET additional temporary environment variables and paths if needed.
REM for example compilers will need to be able to find the binary and
library
REM directories.
REM Will add the path "C:\APPS\FM3" to the current system environment.
REM The temporary path is deleted and returns to the default system paths
REM upon exiting the batch file.
SET PATH=%path%;C:\APPS\FM3
REM
REM SET Option 2 (Usually not required)
REM The temporary variables are usually cleared when the batch file exits.
REM Make a backup of the current system environment and restore before
REM exiting the batch file.
REM Back up the system path to %pathold%
SET PATHOLD=%path%
REM Add our temporary path to the system paths.
SET PATH=%path%;C:\APPS\FM3
REM Last in the batch file. Restore the previous environment (%pathold%)
SET PATH=%pathold%
REM Clear/empty the variable.
SET PATHOLD=""
REM
REM The application will start with reference to the working directory.
REM Launch Option 1
REM Change working directory.
REM Some applications need this.
```

```
REM Alternatively use the full path to the application.
REM CD DRIVE:\PATH\DIRECTORY
CD \APPS\FM3
CALL FM3.EXE
REM
REM Launch Option 2
REM Call the application using the full path.
CALL C:\APPS\FM3\FM3.EXE
REM
REM Launch Option 3
REM Call the application using the system path variable.
CALL %dosdrv%\APPS\FM3\FM3.EXE
REM
REM Launch Option 4
REM Call the application using the system path variable.
REM Pass the command line options to the called application.
CALL %dosdrv%\APPS\FM3\FM3.EXE %1 %2 %3 %4 %5 %6 %7 %8 %9
CLS
```

You can include the CLS commands where it is convenient to tidy up left over console commands and ECHOs if you like.

From this point on I recommend all applications that are added to your system be checked for an associated batch file and create one if it does not exist. Not all installers will create a batch file to launch the application so it is worthwhile having a look through what applications you want fast command line access to and adding a batch file for it. If you don't like the format of the batch file it is easy to come back and edit it later. Don't add batch files for applications that already exist in the system path such as those in `C:\FREEDOS\BIN*.*`

Some applications will require a more complex batch file to include additional path and start up environments. I will present a number of different batch files for each programming environment in that chapter.

It is not mandatory to use a batch file to launch all applications in DOS, but does offer more convenience as well as allowing us to set correct parameters for the application we are launching. This becomes important in later chapters when we need to set specific environment variables and parameters for launching programming environments and associated editors.

If you skipped here from the FM3 install I would also have a quick look at the batch file used for FED below. The %1 %2 etc. are not required for FM3.exe.

FED (Folding Editor) start-up batch file.

FED is a folding text editor for MS-DOS. It offers text color highlighting and the ability to call other applications such as compilers allowing it to act like and IDE.

The same as for the description in "Application launch BAT" create a batch file and place it into `C:\FREDOS\LINKS`

The following is a generic BAT file.

FED.BAT

```
@ECHO OFF
REM Launcher for FED
CALL %dosdrv%\APPS\FED\FED.EXE %1 %2 %3 %4 %5 %6 %7 %8 %9
```

Or a more complete version of the batch file with an error check.

FED.BAT

```
@ECHO OFF
IF EXIST %dosdrv%\APPS\FED\FED.EXE GOTO FED
GOTO NOFOUND
GOTO END
:FED
REM You can add environment paths or change working directory here.
REM Launch FED
CALL %dosdrv%\APPS\FED\FED.EXE %1 %2 %3 %4 %5 %6 %7 %8 %9
GOTO :END
:NOFOUND
ECHO FED not found!
ECHO Press any key to end...
PAUSE
:END
```

You can call FED from anywhere in the system. The variable %1 %2 etc. allows you to send command line options to FED. For example typing FED C:\FDAUTO.BAT will pass C:\FDAUTO.BAT as a command line argument to the batch file FED.BAT. The batch file will then pass the argument on to FED as variable %1.

If you want to clear the left over text from the console after exiting FED just add the CLS command after :END in the FED.BAT batch file.

Start-up Menu

Another convenience is to boot directly to a start-up menu where we can have the convenience of selecting our default desktop environment and navigation. In the following I will present a generic menu that is called directly from our FDAUTO.BAT file

Create a batch file named MENU.BAT with the following lines in the system drive root alongside of FDAUTO.BAT (AUTOEXEC.BAT). You can use EDIT EXE or any other editor such as FED.EXE for this task. Alternatively you can create the file on you host OS and transfer it into the virtual machine drive.

The COSTA desktop will be installed in a later chapter. It is OK to leave the entry here for now.

The DOSSTART and ENSEMBLE desktops will not be used and are just spares. You can alter the name of the menu display, test for input entry number as well as the label section to reflect your own default start application, or ultimately just delete the unwanted entries. I have left the extra entries so you see how the menu system works.

The code follows the following flow.

Displays a list of menu choices :MENU

Tests the numeric value of the user input. IF %M%==n GOTO LABEL

The :LABEL section acts like a subroutine and contains the specific commands to perform a task such as launching an application. The last line in the label returns back to the :MENU. Remember that after a batch calls another batch file or application (child) it will return to the next line in the parent when the called application closes. So, in the case of FM3, when we exit FM3 we will be returned to our MENU for the next choice. In the case of EXIT to DOS (:EOF) the batch file ends and cannot return to the MENU. To invoke the MENU again navigate to the drive root and type in MENU and Enter.

The reset and shutdown commands are defined in the FDAUTO.BAT file and are an alias for the FDAPM commands.

You can customise this menu in whatever way you want as it is just a template to start from :) You could also call sub menus “call SMENU1.BAT” but I would caution against over complicating things. If you decide to use sub menus please be aware of the menu loop problem. Never call a sibling or parent menu with call. Allow the menu (batch file) to end and natively and fall back to the parent menu that called it. This is the same as the problem that can occur when not correctly unwinding child “functions “or “subroutines” in any other programming language. Child batch files act in the same way as function or subroutines.

Syntax highlighting by Notepad++ on Windows 10 host OS.

MENU.BAT

```
@ECHO OFF
REM Prompt the user for input.
prompt $p$g

:MENU
CLS
ECHO.
ECHO .............................................
ECHO PRESS 1 to 7 to select your task, or 5 to EXIT.
ECHO .............................................
ECHO.
ECHO 1 - Open File Maven 3
ECHO 2 - Open Costa Desktop
ECHO 3 - Open UNUSED
ECHO 4 - Open UNUSED
ECHO 5 - EXIT to DOS
ECHO.
ECHO 6 - Restart PC
ECHO 7 - Shutdown PC
ECHO.

rem @choice /B /C:1234 /N What do you want to do?
rem if errorlevel 4 goto DONOTHING

rem set /P Prompt the user with string, and assigns the user input to the
variable.
SET /P M=Type your choice then press ENTER:
IF [%M%]==[] GOTO ERROR
IF %M%==1 GOTO FM3
IF %M%==2 GOTO COSTA
```

```
IF %M%==3 GOTO DSTART
IF %M%==4 GOTO ENSEMBLE
IF %M%==5 GOTO EOF
IF %M%==6 GOTO RESET
IF %M%==7 GOTO SHUTDOWN


:FM3
rem LH
REM cd %dosdrv%\PROGS\FMAV35A
REM call FM3.exe
REM call %dosdrv%\PROGS\FMAV35A\FM3.EXE
call FM3.BAT
GOTO MENU

:COSTA
cd %dosdrv%\PROGS\COSTA
call COSTA.BAT
GOTO MENU

:DSTART
REM cd %dosdrv%\PROGS\DOSSTART
REM call DOSSTART.BAT
GOTO MENU

:ENSEMBLE
REM cd %dosdrv%\PROGS\ENSEMBLE
REM call ENSEMBLE.BAT
GOTO MENU

:RESET
cd %dosdrv%
reset
GOTO MENU

:SHUTDOWN
cd %dosdrv%
shutdown
GOTO MENU

:ERROR
ECHO Unknown Selection!
ECHO Please try again.
PAUSE
GOTO MENU

#:END
REM End batch and exit to command prompt.
:EOF
cls
```

Once you have your MENU.BAT in place open FDAUTO.BAT (AUTOEXEC.BAT) in a text editor (EDIT or FED) and add the following as the LAST LINE (After :END). The order of operations in batch, config and system files is important.

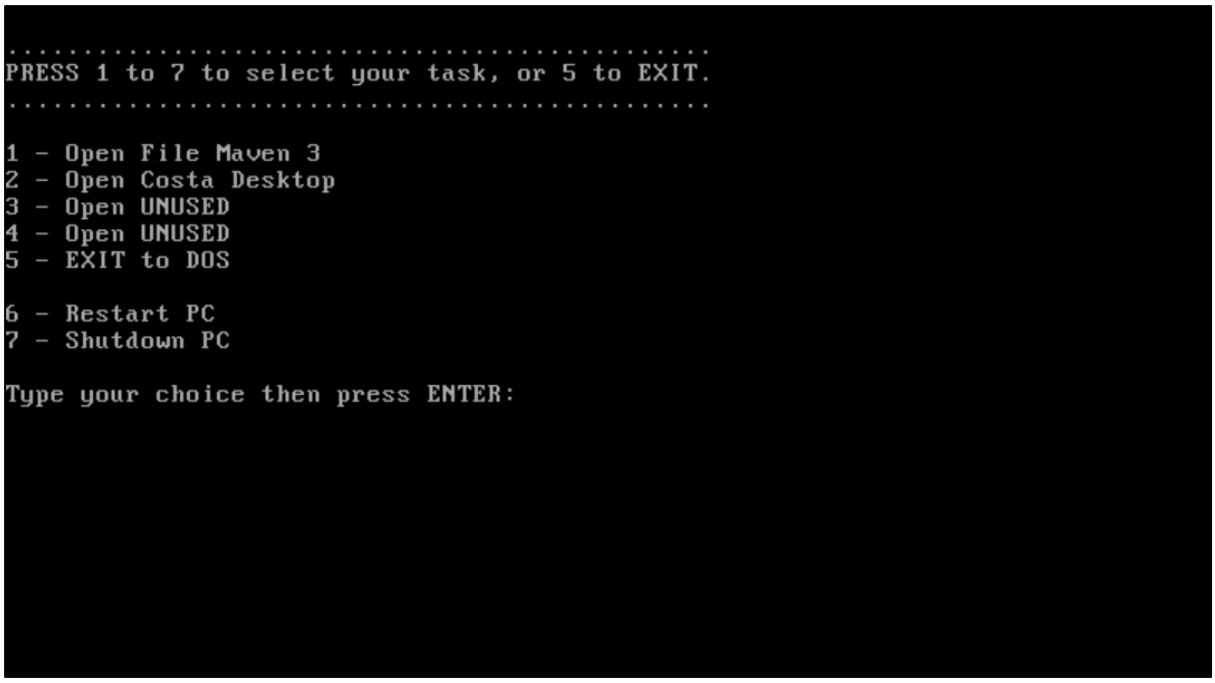
```
...
:END
```

```
PAUSE
CALL %dosdrv%\MENU.BAT
```

The PAUSE is optional. I use it so I can view the boot notifications at start up.

Save the file and reboot the system using “reset” or “shutdown”.

If all has worked well you will now be presented with an options menu to select your choice of application at start up.



Important applications

There are many applications available that will not be contained in the official Bonus CD or in the FreeDOS application repository. To use these application it is typically a matter of copying the application directory name as well as included directories and files into any location on the DOS drive. It is preferable to keep some degree of order to the file system so I would recommend copying applications to the \APPS directory or I sometimes create my own separate directory with a name like “PORTABLE” or PPROGS”. The important thing is to attempt some form of common organisation. Many application installers will install directly to the system root C:\AppName which can all become very messy quite quickly. It is possible to move these applications to you APPS directory but you must take care to also change the paths in any associated batch files as well as entries to AUTOEXEC.BAT to reflect the new location. You can often find the default install locations for these files as well as any environment variable by checking the installer disk for the help docs, installer.bat or install.cfg files for hints.

Once you have placed your application in place you can then create an associated batch file to launch the application. It is not necessary to create a batch file to launch EVERY application, just the ones you may need on a regular basis from the command line.

Manual installation of DOS applications that don't have an installer are simply a matter of copying the application binaries to an appropriate location in your DOS drive such as C:\APPS\Myapp*.*. Create a batch file in C:\FREEDOS\LINKS\Myapp.BAT that points to the main executable. To remove the application, simply delete the app directory and the batch file. This is essentially what the installer, uninstallers do in DOS. Check the README and help files for any special environment variables that may need to be set.

Summary 1

At this point you should have the basics of installing, creating a batch file to launch applications as well as having FM3 set up for easy navigation. You should be able to easily launch the built in FreeDOS editor "EDIT.EXE" and the more advanced folding editor "FED.EXE". Along with the start-up MENU you now have the essential tools to navigate and customise your DOS install.

Costa GUI Desktop

Costa desktop for DOS is a simple yet powerful GUI like desktop launch menu system. It allows you to create shortcuts to your favourite applications and keep them organised as icons on the 5 available desktops. The Costa GUI desktop is very flexible and easy to use as well as offering a great amount of organized convenience for launching your favourite DOS applications.

You can download Costa "costa174.zip" from:

https://costa.jacobpalm.dk/

The latest version at the time of writing this guide is v1.7.4

[Update: Jacob has just added the ability to move icons between the 5 desktops. As soon as the new update is released I will update the version here.]

Unpack the "costa174.zip" archive and copy it to `C:\APPS\Costa174\*.*`

Navigate to `C:\APPS\Costa174\` and view the `COSTA.BAT` file with "`MORE COSTA.BAT`", or enter from FM3. As you will see, the Costa launch batch file is not yet created.

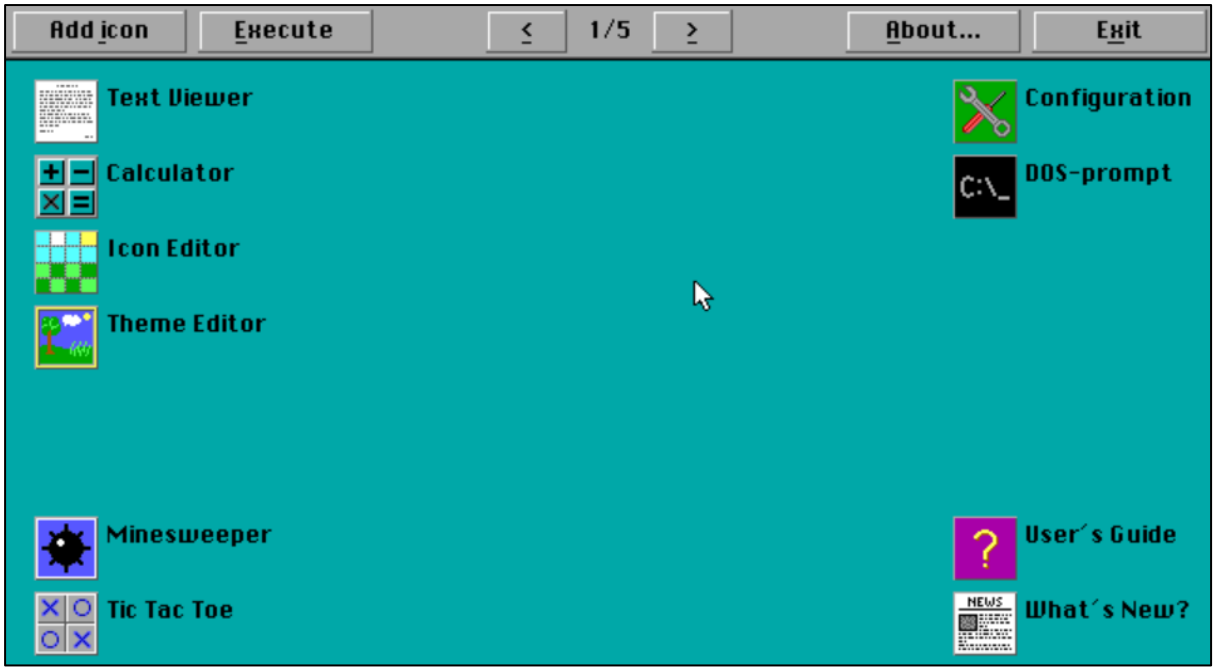
Run the `SETUP.EXE` application either from the command line or from FM3. This will create the `COSTA.BAT` file with the correct path parameters to launch Costa Desktop.

You can again View the contents of `COSTA.BAT` to see the changes. Looking at the Batch files used to launch different applications will assist in your knowledge of creating your own custom batch files to launch other applications as discussed in "Application launch BAT".

Next copy `COSTA.BAT` to the system path `C:\FREEDOS\LINKS`

To do this in FM3 make sure the left pane navigation is highlighting `COSTA.BAT`. Using the arrow keys "->" move focus to the right pane and navigate to the `C:\FREEDOS\LINKS` directory. Change

back to the left pane and with COSTA.BAT highlighted select Alt+F then “Copy”. “Copy to:” should show C:\FREEDOS\LINKS and you can select OK and then press Enter to copy the file. Check the right pane in FM3 and the COSTA.BAT file should now be present. You can type COSTA into the command line to launch the Costa Desktop.



If you created the start-up menu in the section “Start-up Menu” you will notice that I already had an entry for the Costa desktop. Navigate to the System root and View the MENU.BAT with More or in a text editor. Under the label :COSTA you will find “CALL COSTA.BAT” as the 2nd selection from the menu. You can change the order of the menu if you want by changing the text display and making sure to alter the choice tests IF %M%==2 GOTO COSTA to reflect the correct menu number and Labels.

Close all applications and select shutdown or reset (restart). The next time you boot the system you can now select from File Maven or Costa desktop as your work environment. I would use Costa as the default as we can open File Maven at any time from the Costa desktop.

Add shortcuts to Costa

Adding a desktop shortcut or launch icon to Costa is quite easy. The [Add Icon] button allows you to navigate to and select any executable file (.bat, .com or .exe) to launch. You can also add additional parameters (Arguments) to the application being launched. For example adding EDIT.EXE as the program or DOS command and then adding C:\MYDOCS\help.txt will send help.txt as the argument for EDIT opening help.txt in the FreeDOS editor. In the same way you can call MORE < C:\MYDOCS\help.txt to view the document.

At this time you can “Move” icons around and arrange them on individual desktops but you cannot move an icon to another desktop. You have 5 built in desktops available so take a moment to think about how you may want to organize each desktop; for example desktop 2 for games, desktop 3 for

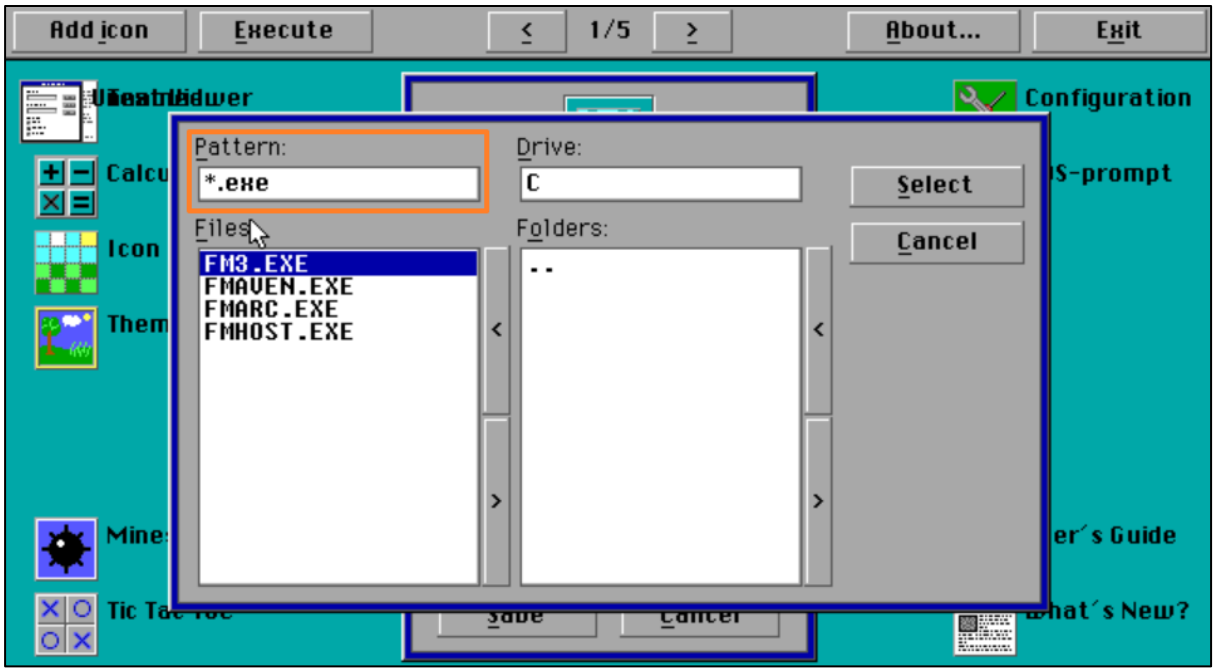
development tools etc. Experiment a little as you can always delete an icon and then recreate it on another desktop.

Let's start by adding the File Maven 3 file manager to our desktop. Note that DOS does not have graphical icons associated with executable files so you will need to use one of the icons included in Costa or use the inbuilt icon editor to create your own icon.

Select [Add Icon] button and use the arrow keys to select an icon for the executable.

Next give the launch icon a name that you can associate with the application; for example FM3 or FileMaven.

Use the [..] button next to Program or DOS command: to navigate to the application. In this case it will be `C:\APPS\FM3\FM3.EXE` unless you have placed it somewhere else. Take note of the scroll buttons <, > or use the keyboard arrow navigation keys and TAB to scroll available directories and files. Remember that ".." moves up one directory. Note that the extension in "Pattern:" will default to *.EXE and filter the results to EXE type files, so if you are searching for a .COM executables change this to *.COM or *.* to list all files.

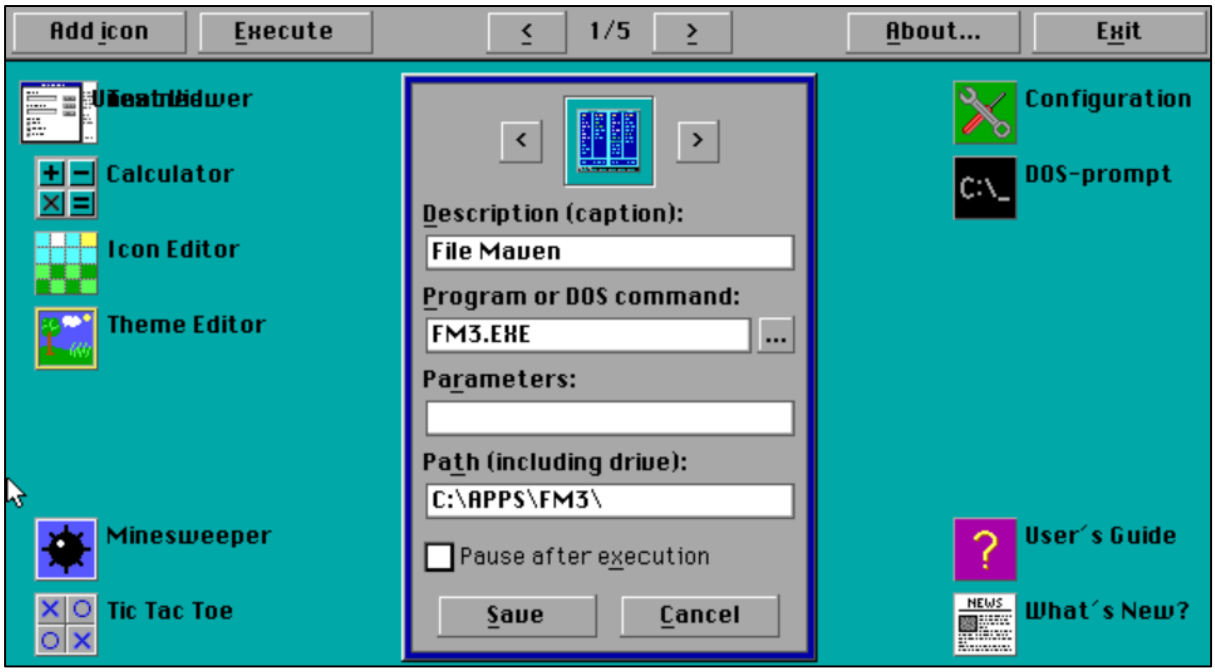


The path to the executable (FM3.EXE) is added automatically.

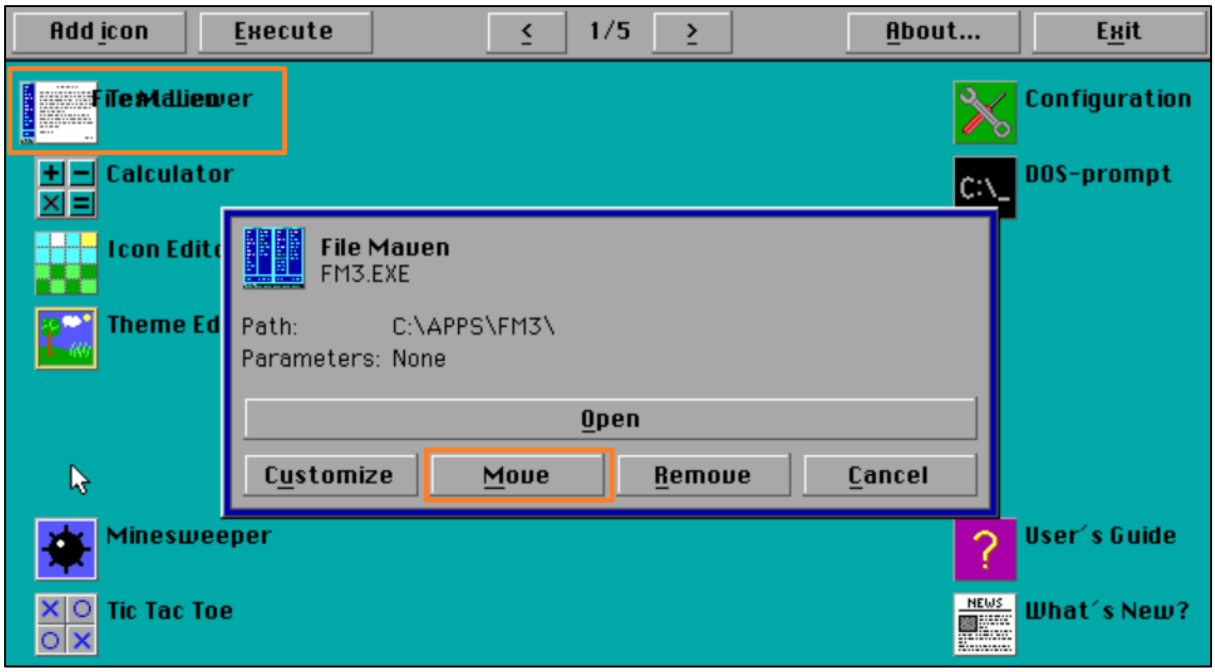
We do not need to start File Maven with any additional commands so we can now save the new launch icon.

Please note that I have linked directly to FM3.EXE rather than calling the associated FM3.BAT in the .\LINKS directory. Calling FM3.EXE directly removes the ability to use the PKZIP unpacker utility as it is set in the FM3 batch file. I would suggest using the FM3.BAT file as this offers the ability to start the application with additional parameters required for the PKZIP utility. When opening other applications such as software development environments (In later chapters) the use of batch files will become important and will need to be used in the Costa desktop launch icons.

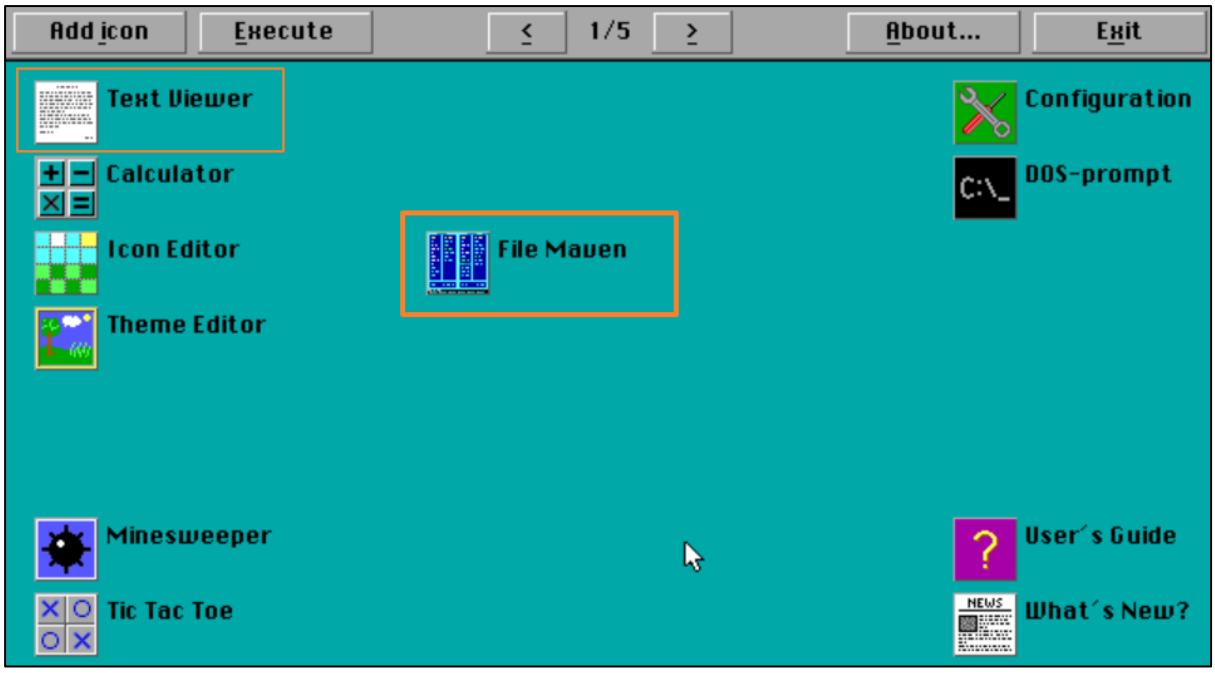
Like with batch files the called application returns to the parent application that called it after the child application closes, so applications launched from Costa will return focus back to the Costa desktop when closed. If you launched Costa from the Start-up Menu then focus will return to the start-up menu (MENU.BAT) when you exit Costa.



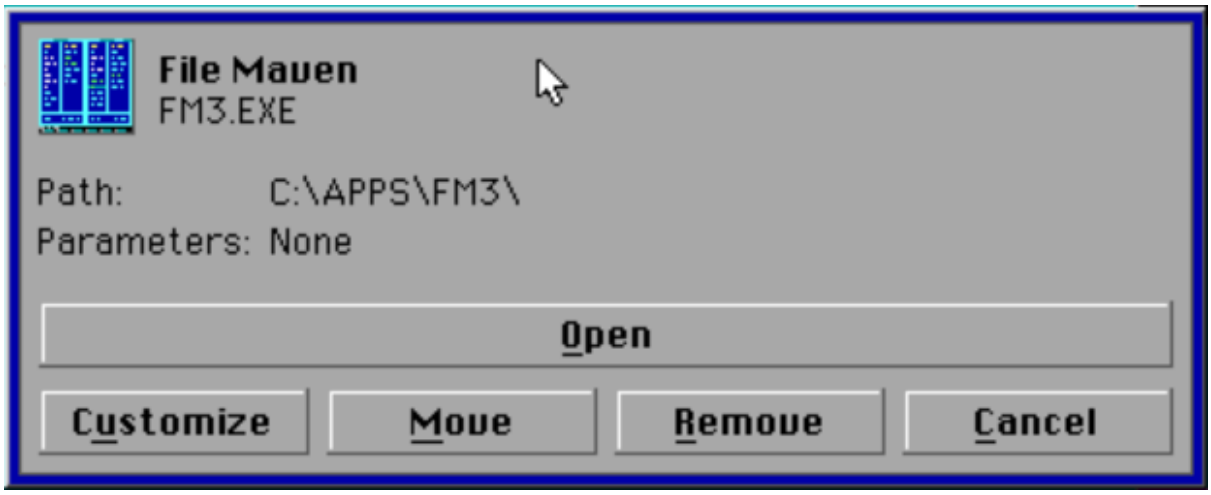
You will notice that new icons are created in the top left corner and will be behind another icon. Click on the visible icon (upper most) and select “Move” from the options.



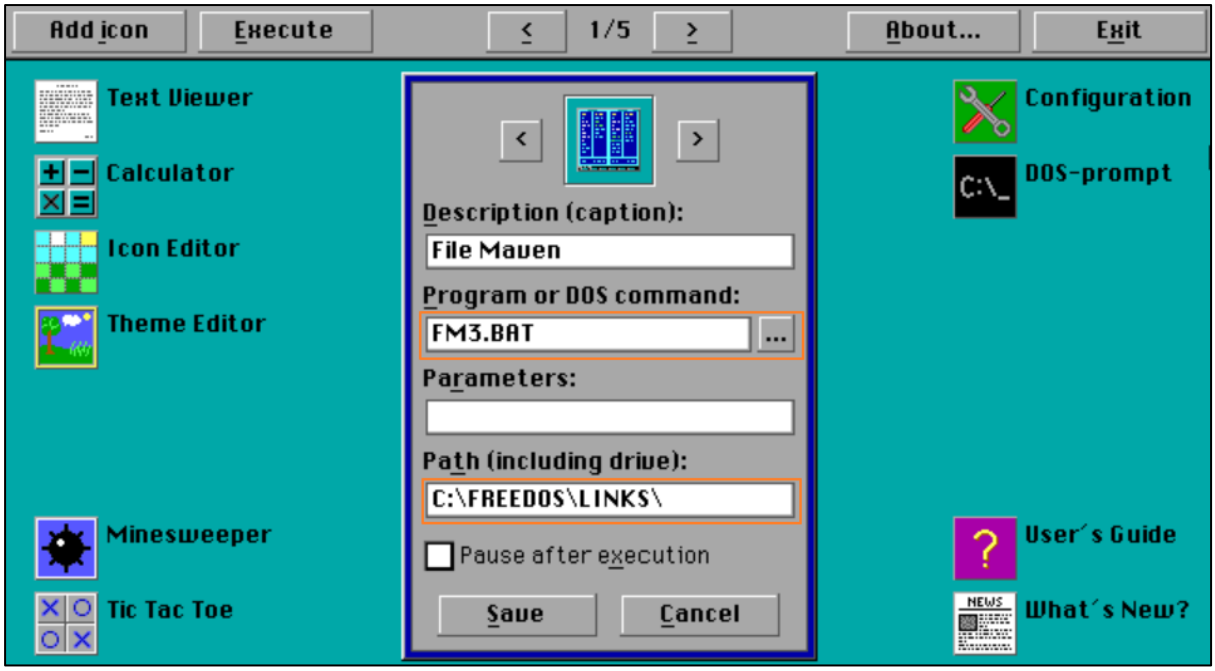
The icon will attach to the mouse pointer and drop in place on left click.



Click on the icon and choose the appropriate button. You can Open the application, use Customize to change the launch icon parameters, Move, Remove the icon.



Note if you followed the above guide and linked directly to the FM3.EXE executable then you will not have access to the PKZIP utility that is set in the FM3.BAT file. I wanted to show both ways so you can see the difference. Either, delete the icon and then create a new icon, or use the customize button to select the `C:\FREEDOS\LINKSFM3.BAT` file. Don't forget to change the filter Pattern *.EXE to *.BAT



As I work through additional application installs I will refer back to this section with regards to creating and organising desktop shortcuts.

Have a think about some of the essential applications that you may want to use at this stage such as editors, games, system test tools. Remember FDIMPLES? Try adding the FreeDOS application manager to the Costa desktop. As this file exists in the system path you can just add FDIMPLES into a new icon “Program or DOS command:” section, or alternatively use the search tool to navigate to the location of FDIMPLES.

Hint: File Maven 3 has a file search utility under the File menu.

Summary2

It is up to you as the user to select what is the most appropriate way of linking to the executable files in your system. Take the time to think and plan as well as keeping notes of what you do. Keep regular backups of important configuration files. If you are unfamiliar with DOS it helps to make a test install and test different ideas. When you have worked out a good plan for your DOS environment create a clean install and follow your notes from the test install to set it up the way you want.

At this stage you should have a boot up launch menu, a Convenient GUI Desktop (Costa), a 2 pane TUI file manager (FM3) and the ability to create batch files and launch applications. Take some time to familiarize yourself with the system using Costa and FM3.

Note that audio is unreliable when DOS is installed in a VirtualBox guest environment. Some games will work, some will not. Continue to the next section to set up the DOS audio drivers, but don’t expect them to work well.

SB 16 drivers VirtualBox

VitualBox only provides very basic SoundBlaster audio support and there are no official drivers for DOS operating systems under virtualBox. That being said the DOS and Windows 3.1 drivers will work in many instances. DOS does not have a universal audio interface like later 32-bit protected user mode operating systems and relies heavily upon the drivers being programmed into the application. As there was no STD audio interface only the common audio card drivers at the time of creating the applications were included at the time. This is a bit of a hit and miss when it comes to some old games. That being said the SB interface was the most popular and many sound card manufacturers and game creators followed the same API. To the best of my knowledge the MIDI sound interface does not work using the drivers in in DOS under VirtualBox.

The following drivers are about the closest match available for DOS on VirtualBox, but other version will work as well. As none of these drivers are designed for VirtualBox emulation it is a bit of a trial and error to find a good match with “SOME” functionality, so don’t expect a perfect audio solution.

If you are using legacy hardware from the period rather than VirtualBox most drivers for the hardware will work as expected. DOS and the hardware at this time typically relied upon the internal system “Beep Speaker” as the default for audio. The beep speaker is not emulated in VirtualBox.

You can obtain a copy of the drivers from the following links or use the driver file name in your search. The philscomputerlab.com downloads are a copy of the official downloads from Creative Labs. If the drivers are no longer available at the following 2 sites then take extreme care with regards to the legitimacy of drivers from other sites.

Search engine “sbbasic.exe” or “Sound blaster 16 DOS”.

https://support.creative.com/downloads/searchdownloads.aspx?nLanguageLocale=1033&filename=SB&nPage=39

“Sound Blaster 16/SB32/AWE32 Basic Disk for DOS/Windows 3.1 Installation” (sbbasic.exe)

This is the essential driver set for DOS.

https://support.creative.com/downloads/searchdownloads.aspx?filename=CTCMBBS

“Creative PnP Configuration Manager (Rev 4)” (ctcmbbs.exe)

This is the essential diagnostics and configuration manager for the SB Drivers above.

The same 2 files can also be downloaded from PhilsComputer Lab...

https://www.philscomputerlab.com/creative-labs-drivers.html

I would also recommend sourcing a 16-bit PCM (uncompressed) WAV audio file and am MP3 format audio file for testing. Some audio sites will provide a range of audio test files for downloading.

https://www.mmsp.ece.mcgill.ca/Documents/AudioFormats/WAVE/Samples.html

https://mauvecloud.net/sounds/index.html

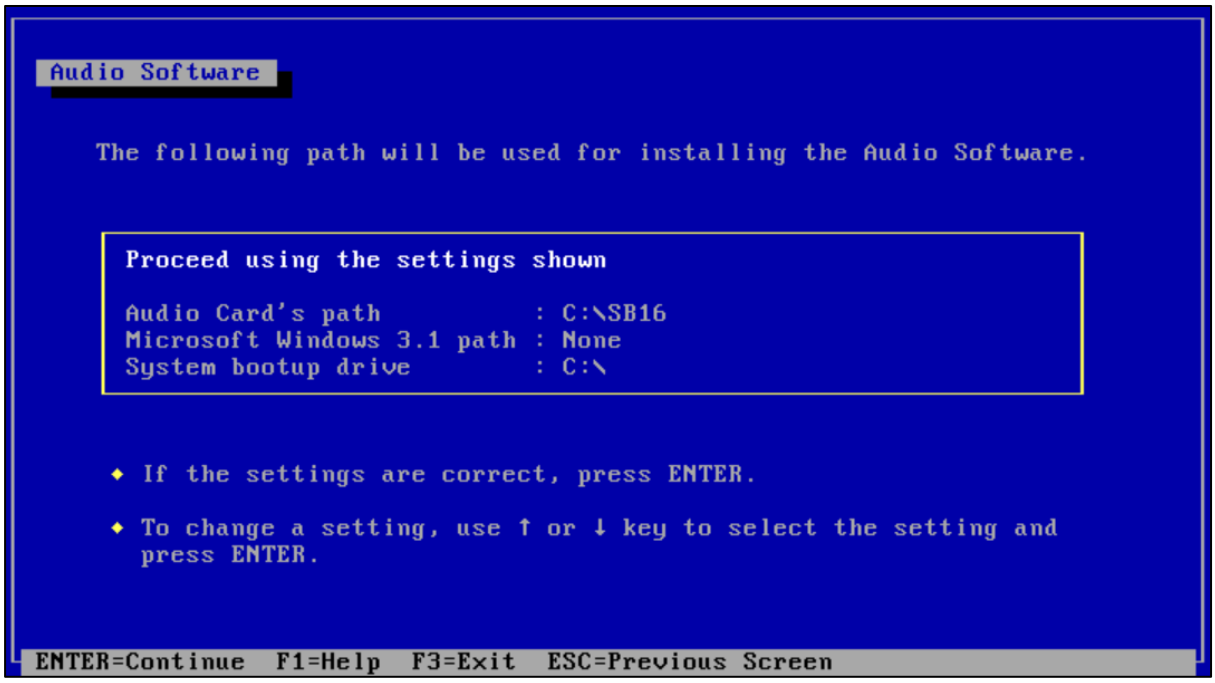
Alternatively if you have a Windows 3.1 or Windows 95 install handy you can extract a wave file from `C:\WINDOWS\MEDIA` (PCM, 22,050Hz, 16 Bit, Stereo) to use as a test file.

DOS media players can and will play other Bit rate wave files but 16-Bit is native and the best place to start for testing.

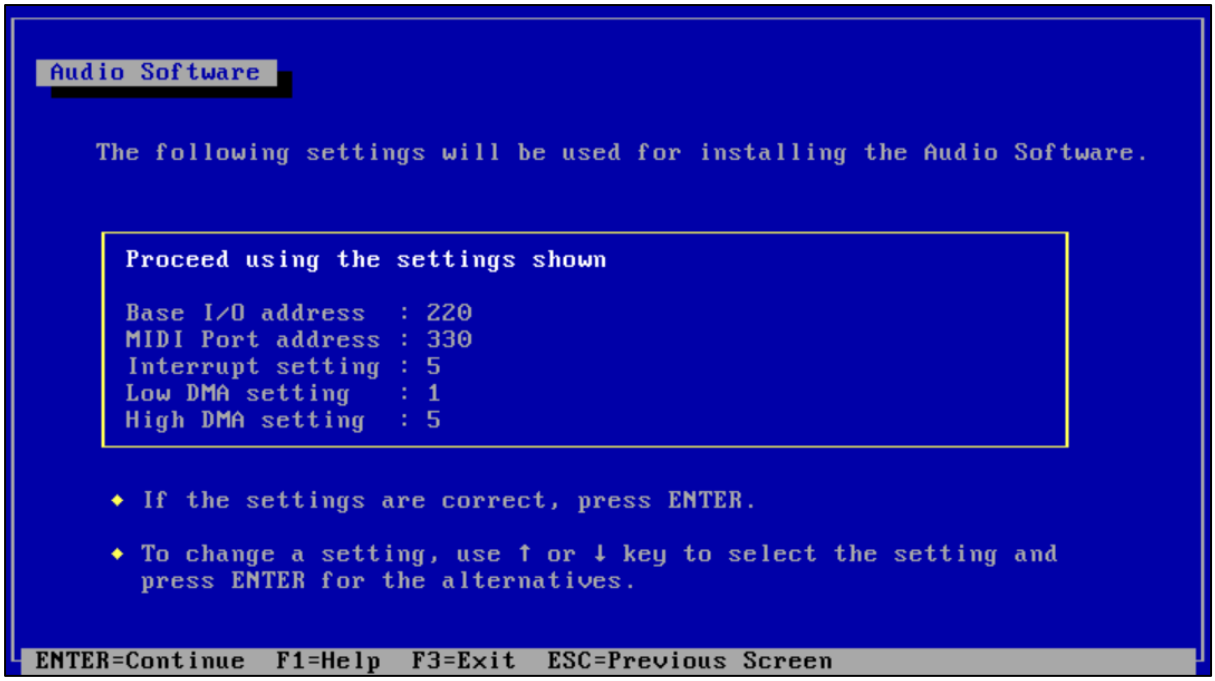
Transfer the “sbbasic.exe” and “ctcmbbs.exe” files to a temporary file in your DOS hard drive as well as your .wav and .mp3 test files. It may be beneficial to keep the executables in separate directories.

Make a backup copy of `FDAUTO.BAT` (AUTOEXEC.BAT) and `FDCONFIG.SYS` (CINFIG.SYS). I usually place them in a directory `C:\BKUP` with a timestamp directory or place a text file next to the backups with a time stamp and information.

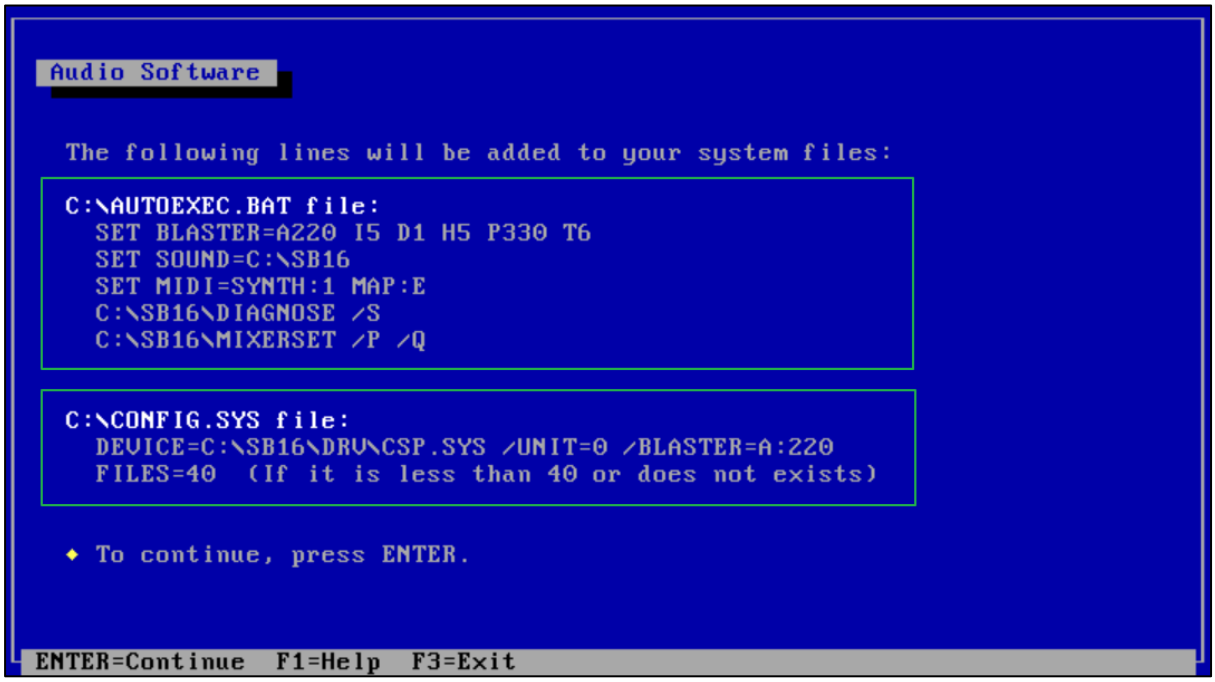
Navigate to the directory where you copied the driver set up files and execute the `sbbasic.exe` file to unpack the drivers install set. Next view the readme.txt file, then execute the `INSTALL.EXE` file. Follow the install guide using the default with the install path `C:\SB16` and the system boot drive as `C:\`.



In the next screen keep the default `AUTOEXEC.BAT` settings:



You will see that the following 2 entries have been made to AUTOEXEC.BAT and CONFIG.SYS



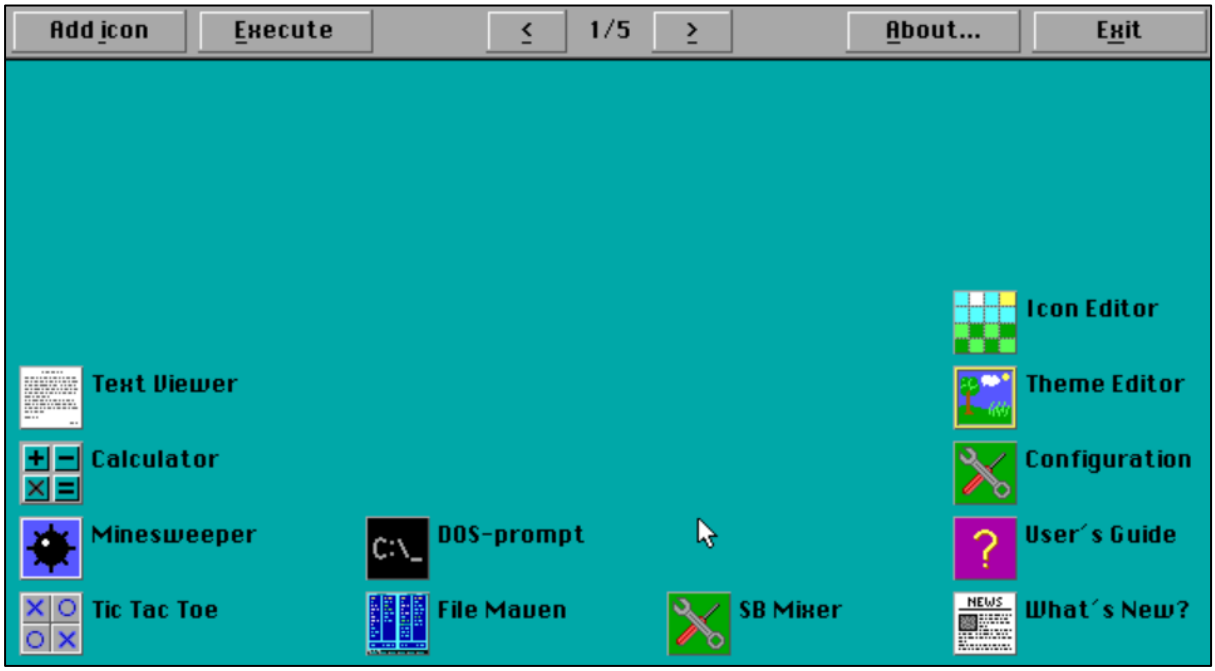
Once the install is complete the 2 original files will have the .B~K extension if you are using MS-DOS. For FreeDOS there will be no .B~K files as FreeDOS uses FDAUTO and FDCONFIG rather than AUTOEXEC and CONFIG. Do NOT reboot and press Enter to exit the installer. Navigate to the C:\ drive root using the command line or FM3.

There are 3 files associated with the Sound Blaster drivers:

- CTMIX.CFG – The SB Mixer settings.

- MIXERSET.EXE – The Mixer settings and volume controls.
- DIAGNOSE.EXE – Checks the settings created in FDCONFIG and FDAUTO.

MIXERSET.EXE is the application you will use to set up volume, balance, speakers etc. so it is worth adding this to your Costa desktop.



Fix FDAUTO.BAT

If you are using MS-DOS or another DOS OS then these steps may not be required.

Freedos 1.3 uses FDCOFIG.BAT and FDAUTO.SYS at boot time rather than the MS-DOS CONFIG.SYS and AUTOEXEC.BAT. The driver installer application expects the legacy MS-DOS CONFIG and AUTOEXEC to be present and creates them if they don't exist. This is very important to note as many application installs will add the entries to the incorrect files. We will need to manually check and correct this every time you run an automated application install in FreeDOS.

Close any open application such as FM3 or Costa. Open a text editor such as EDIT or FED.

Open FDAUTO.BAT and then also open AUTOEXEC.BAT. You can use the Window menu in edit or File -> Next file to change between open text documents.

We need to copy the five lines from AUTOEXEC.BAT to the FDAUTO file and replace the line "`set BLASTER=A220 I5 DI H5 P330`".

Hint Use the Shift key + arrow keys to highlight the text. Use Alt + Edit -> Copy to copy the text. Delete the line in FDAUTO (`set BLASTER=A220 I5 DI H5 P330`) and with the curser at the start of the empty line use Alt+E -> Paste to put the new text in place. Select File and save (Write) to save the changes. Close both files.

Your FDAUTO should now look like following (FIXED) version:

AUTOEXEC.BAT (Created by SB INSTALL)

```
SET SOUND=C:\SB16
SET BLASTER=A220 I5 D1 H5 P330 T6
SET MIDI=SYNTH:1 MAP:E
C:\SB16\DIAGNOSE /S
C:\SB16\MIXERSET /P /Q
```

FDAUTO.BAT (OLD)

```
@ECHO OFF
REM Standard AutoExec Batch File

set DOSDRV=C:
set DOSDIR=C:\FreeDOS
set LANG=EN
set TZ=UTC
set PATH=%dosdir%\BIN
if exist %dosdir%\LINKS\NUL set PATH=%path%;%dosdir%\LINKS
set NLSPATH=%dosdir%\NLS
set HELPPATH=%dosdir%\HELP
set TEMP=%dosdir%\TEMP
set TMP=%TEMP%
SET BLASTER=A220 I5 D1 H5 P330 T6
set DIRCMD=/P /OGN /Y
set COPYCMD=/-Y
set OS_NAME=FreeDOS
set OS_VERSION=1.3
set autofile=C:\FDAUTO.BAT
set cfgfile=C:\FDCONFIG.SYS
alias cfg=edit %cfgfile%
alias auto=edit %autofile%
alias reboot=fdapm warmboot
alias reset=fdisk /reboot
alias halt=fdapm poweroff
alias shutdown=fdapm poweroff

if "%config%"=="4" goto END

if not exist %dosdir%\bin\vinfo.com goto Only8086
vinfo /m
if errorlevel 3 goto Support386
if errorlevel 2 goto Support286

:Only8086
MEM /C /N
goto FINAL

:Support286
FDAPM APMDOS
CTMOUSE
MEM /C /N
goto FINAL

:Support386

rem codepage settings
REM NLSFUNC %dosdir%\BIN\COUNTRY.SYS
```

```
REM DISPLAY CON=(EGA),858,2)
REM MODE CON CP PREP=((858) %dosdir%\CPI\EGA.CPX)
REM KEYB US,858,%dosdir%\bin\keyboard.sys
REM CHCP 858
REM MKEYB UK
rem keyboard settings


if "%config%"=="3" goto Support386Low

LH FDAPM APMDOS
rem LH SHARE

rem if EXIST %DOSDIR%\BIN\DOSLFN.COM goto UseLFN
goto NoLFN
:UseLFN
LH %DOSDIR%\BIN\DOSLFN.COM
set DIRCMD=%DIRCMD% /LFN
:NoLFN

CTMOUSE
goto InitCDROM

:Support386Low
FDAPM APMDOS
CTMOUSE

:InitCDROM
if not exist %dosdir%\bin\cdrom.bat FINAL
echo.
call %dosdir%\bin\cdrom.bat

:FINAL
MEM /C /N
echo.
if not exist %dosdir%\bin\fdnet.bat goto NoNetwork
call %dosdir%\bin\fdnet.bat start
if errorlevel 1 goto NoNetwork
REM Custom networking stuff once packet driver has loaded

:NoNetwork

if exist %dosdir%\bin\fdassist.bat call %dosdir%\bin\fdassist.bat
if exist %dosdir%\bin\cdrom.bat call %dosdir%\bin\cdrom.bat display
if exist %dosdir%\bin\welcome.bat call %dosdir%\bin\welcome.bat

:END
PAUSE
CALL %dosdrv%\MENU.BAT
```

FDAUTO.BAT (FIXED)

```
@ECHO OFF
REM Standard AutoExec Batch File

set DOSDRV=C:
set DOSDIR=C:\FreeDOS
set LANG=EN
set TZ=UTC
```

```
set PATH=%dosdir%\BIN
if exist %dosdir%\LINKS\NUL set PATH=%path%;%dosdir%\LINKS
set NLSPATH=%dosdir%\NLS
set HELPPATH=%dosdir%\HELP
set TEMP=%dosdir%\TEMP
set TMP=%TEMP%
SET SOUND=C:\SB16
SET BLASTER=A220 I5 D1 H5 P330 T6
SET MIDI=SYNTH:1 MAP:E
C:\SB16\DIAGNOSE /S
C:\SB16\MIXERSET /P /Q
set DIRCMD=/P /OGN /Y
set COPYCMD=/-Y
set OS_NAME=FreeDOS
set OS_VERSION=1.3
set autofile=C:\FDAUTO.BAT
set cfgfile=C:\FDCONFIG.SYS
alias cfg=edit %cfgfile%
alias auto=edit %autofile%
alias reboot=fdapm warmboot
alias reset=fdisk /reboot
alias halt=fdapm poweroff
alias shutdown=fdapm poweroff

if "%config%"=="4" goto END

if not exist %dosdir%\bin\vinfo.com goto Only8086
vinfo /m
if errorlevel 3 goto Support386
if errorlevel 2 goto Support286

:Only8086
MEM /C /N
goto FINAL

:Support286
FDAPM APMDOS
CTMOUSE
MEM /C /N
goto FINAL

:Support386

rem codepage settings
REM NLSFUNC %dosdir%\BIN\COUNTRY.SYS
REM DISPLAY CON=(EGA),858,2)
REM MODE CON CP PREP=((858) %dosdir%\CPI\EGA.CPX)
REM KEYB US,858,%dosdir%\bin\keyboard.sys
REM CHCP 858
REM MKEYB UK
rem keyboard settings


if "%config%"=="3" goto Support386Low

LH FDAPM APMDOS
rem LH SHARE

rem if EXIST %DOSDIR%\BIN\DOSLFN.COM goto UseLFN
```

```
goto NoLFN
:UseLFN
LH %DOSDIR%\BIN\DOSLFN.COM
set DIRCMD=%DIRCMD% /LFN
:NoLFN

CTMOUSE
goto InitCDROM

:Support386Low
FDAPM APMDOS
CTMOUSE

:InitCDROM
if not exist %dosdir%\bin\cdrom.bat FINAL
echo.
call %dosdir%\bin\cdrom.bat

:FINAL
MEM /C /N
echo.
if not exist %dosdir%\bin\fdnet.bat goto NoNetwork
call %dosdir%\bin\fdnet.bat start
if errorlevel 1 goto NoNetwork
REM Custom networking stuff once packet driver has loaded

:NoNetwork

if exist %dosdir%\bin\fdassist.bat call %dosdir%\bin\fdassist.bat
if exist %dosdir%\bin\cdrom.bat call %dosdir%\bin\cdrom.bat display
if exist %dosdir%\bin\welcome.bat call %dosdir%\bin\welcome.bat

:END
PAUSE
CALL %dosdrv%\MENU.BAT
```

Next open FDCONFIG.SYS and CONFIG.SYS in your editor. We are going to transfer the settings to FDCONFIG.

FDCONFIG.SYS is arranged a little differently to a STD MS-DOS style CONFIG.SYS due to the variables and menu system. You will need to look at the existence of some of the variable settings before just copying and pasting. New entries that don't exist would be best placed after the menu system, but to keep with the MENU 4 item (No drivers) I will use the Menu system.

You will see that "!FILES=40" is already set at the beginning of FDAUTO so we do not need to change this.

In the menu you will see a list of different options after the menu text. The prefix numbers before say to load the specific lines on any of those menu number selections. So, a line beginning with 12? will load if menu 1 or menu 2 was selected, 123? will load if any of the first 3 menu items were selected. We want the audio drives to load on any selection except 4 so the line will begin with 123?

```
123?DEVICE=C:\SB16\DRV\CSP.SYS /UNIT=0 /BLASTER=A:220
```

You can copy and paste the line from CONFIG to FDCONFIG and place it in a new line as follows.

FDCONFIG.SYS

```
SET DOSDIR=C:\FreeDOS

!COUNTRY=001,858,C:\FreeDOS\BIN\COUNTRY.SYS
!LASTDRIVE=Z
!BUFFERS=20
!FILES=40
!MENUCOLOR=7,0

MENUDEFAULT=1,5
MENU 1 - Load FreeDOS with JEMM386 (no EMS, max RAM free)
MENU 2 - Load FreeDOS with JEMM386 (Expanded Memory)
MENU 3 - Load FreeDOS low with some drivers (Safe Mode)
MENU 4 - Load FreeDOS without drivers (Emergency Mode)

12?DOS=HIGH
12?DOS=UMB
12?DOSDATA=UMB
123?DEVICE=C:\FreeDOS\BIN\HIMEMX.EXE
1?DEVICE=C:\FreeDOS\BIN\JEMM386.EXE NOEMS X=TEST I=TEST I=B000-B7FF NOVME
NOINVLPG
2?DEVICE=C:\FreeDOS\BIN\JEMM386.EXE X=TEST I=TEST I=B000-B7FF NOVME
NOINVLPG
34?SHELL=C:\FreeDOS\BIN\COMMAND.COM C:\FreeDOS\BIN /E:1024
/P=C:\FDAUTO.BAT
12?SHELLHIGH=C:\FreeDOS\BIN\COMMAND.COM C:\FreeDOS\BIN /E:1024
/P=C:\FDAUTO.BAT
```

FDCONFIG.SYS (FIXED)

```
SET DOSDIR=C:\FreeDOS

!COUNTRY=001,858,C:\FreeDOS\BIN\COUNTRY.SYS
!LASTDRIVE=Z
!BUFFERS=20
!FILES=40
!MENUCOLOR=7,0

MENUDEFAULT=1,5
MENU 1 - Load FreeDOS with JEMM386 (no EMS, max RAM free)
MENU 2 - Load FreeDOS with JEMM386 (Expanded Memory)
MENU 3 - Load FreeDOS low with some drivers (Safe Mode)
MENU 4 - Load FreeDOS without drivers (Emergency Mode)

12?DOS=HIGH
12?DOS=UMB
12?DOSDATA=UMB
123?DEVICE=C:\FreeDOS\BIN\HIMEMX.EXE
123?DEVICE=C:\SB16\DRV\CSP.SYS /UNIT=0 /BLASTER=A:220
1?DEVICE=C:\FreeDOS\BIN\JEMM386.EXE NOEMS X=TEST I=TEST I=B000-B7FF NOVME
NOINVLPG
```

```
2?DEVICE=C:\FreeDOS\BIN\JEMM386.EXE X=TEST I=TEST I=B000-B7FF NOVME
NOINVLPG
34?SHELL=C:\FreeDOS\BIN\COMMAND.COM C:\FreeDOS\BIN /E:1024
/P=C:\FDAUTO.BAT
12?SHELLHIGH=C:\FreeDOS\BIN\COMMAND.COM C:\FreeDOS\BIN /E:1024
/P=C:\FDAUTO.BAT
```

Save the FDCONFIG.SYS file and close the editor. You can copy the 2 files AUTOEXEC.BAT and CONFIG.SYS to a backup directory for later reference or delete the files if using FreeDOS.

Reboot the operating system with “reset”. When the FreeDOS OS loads it will also load the audio device drivers configured in FDCONFIG and FDAUTO.

You can find more information about setting up FDCONFIG.SYS and FDAUTO.BAT at the following FreeDOS wiki.

http://wiki.freedos.org/wiki/index.php/Games

Audio playback

There are a number of different Audio file, CD, and MIDI players available for DOS. Try to remember that digital Audio playback was still in its infancy and relied heavily upon MIDI audio files and RAW PCM (WAV) audio files on CD ROM at the time. There is no one application that fits all possible playback scenarios. Also keep in mind that VirtualBox SB card emulation does not support MIDI playback so many games will not be able to play the common MIDI and Sound Font files of the period.

I have tested a number of audio players with mixed success. MPlayer and Quick View (QV) both appear to work OK to some degree. MPXPLAY, CDP, DOSAMP, ADPLAY, OPENCP work but the sound is terrible.

You may need to try a few different apps and experiment a little if you want to play music.

SB MIXER

SB Mixer (MIXERSET.EXE) allows you to control the Audio levels and balance and is found in the `C:\SB16` Drivers install.

MPlayer

Can be found on and installed from the FreeDOS Bonus CD. Use FDIMPLES to install from the inserted CD(ISO). MPlayer is a media player capable of playing both audio and video file formats. This is about the most usable media player for DOS in a VirtualBox client.

The executable file will be found in `C:\FreeDOS\BIN\MPLAYER.EXE`

QuickView [Pro]

QuickView and QuickView Pro are proprietary Media players that can be found from the following site:

http://www.multimediaware.com/qv/index.html

QuickView is one of a number of applications capable of playing wave files from the command line or batch file. This can be convenient if you want to add a little bit of sound bling to your DOS install or an application you write. You can use it entering

"`[Drive:][Path]\QV\QV.EXE [Drive:][Path]\AUDIO.WAV`"

or call it from a batch file

"`CALL [Path]\QV\QV.EXE [Drive:][Path]\AUDIO.WAV`".

There are a number of other free wave player applications for DOS if you search around the internet. Each that I have tested has worked with varying degrees of success so you may have to try a few before finding something suitable for your needs.

Audio Players found on the FreeDOS Bonus CD:

- ADPLAY (Requires OPL2 midi synth)
- CDP (can play file, but poor sound)
- MPLAYER (good MMedia player)
- OPENCP (Requires DOS4GW.EXE)

Some other audio players names that I had mixed success with.

MPXPLAY (poor, skips in VirtualBox)

https://mpxplay.sourceforge.net/

DOSAMP

https://www.rarewares.org/rrw/dosamp.php

https://www.rarewares.org/rrw/programs.php

SBPLAY

SBMIX - SBPro mixer control

https://www.bttr-software.de/products/sbmix/

There are many more hiding in the DOS archives.

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 APMDOS
CTMOUSE /R33

:InitCDROM
if not exist %dosdir%\bin\cdrom.bat FINAL
echo.
...
```

Image View Edit

2 of the following image editors are available via the FreeDOS package manager FDPKG. Others will require a manual install.

Image formats varied greatly during the DOS period and many are not compatible with modern image formats. It is helpful to have a few applications for editing some of the common image types of the time. The following are just a few useful applications out of the multitude of “niche” image editors and viewers available for DOS. You can install/test any of these applications at any time

when you may need them and are not essential DOS installs, but are useful when it comes to graphical programming.

VGAPaint 386

VGAPaint 386 (or VP386) was an attempt by Avery Lee (now famous for VirtualDub) to recreate in DOS the Electronic Arts Deluxe Paint IV experience known to Commodore Amiga users. As such VP386 is a bitmap graphics editor.

https://www.bttr-software.de/products/vp386/

You will need 4 files from the download page. Version 1.4 (and V1.3)

- vp386.zip – Main application. (Required)
- ext_dos4gw.zip – 386 32-bit memory extender. (Required)
- vp386drv.zip – Additional VGA drivers.
- vp386ext.zip – Additional extensions.

Install Instructions

If the DOS4GW executable already exists in `C:\FreeDOS\BIN` then copying the "ext_dos4gw" files is not required. If you are not sure then copy the "ext_dos4gw" files into the \vp386 directory.

Unpack all 4 zip packages above. Copy the contents of ext_dos4gw, vp386drv and vp386ext into the root of the vp386 directory (there should be no sub directories).

The "COPYING" file is the same notice in all 3 directories so you can overwrite or rename as long as 1 COPYING exists.

Copy the vp386 directory to a suitable location in your DOS install; for example
`C:\APPS\VP386\*.*`

Create a batch file to launch the VP.EXE application as shown in "**Application launch BAT**" with the addition of adding the environment path to dos4gw.exe

```
SET PATH=%path%;%dosdrv%\APPS\VP386
```

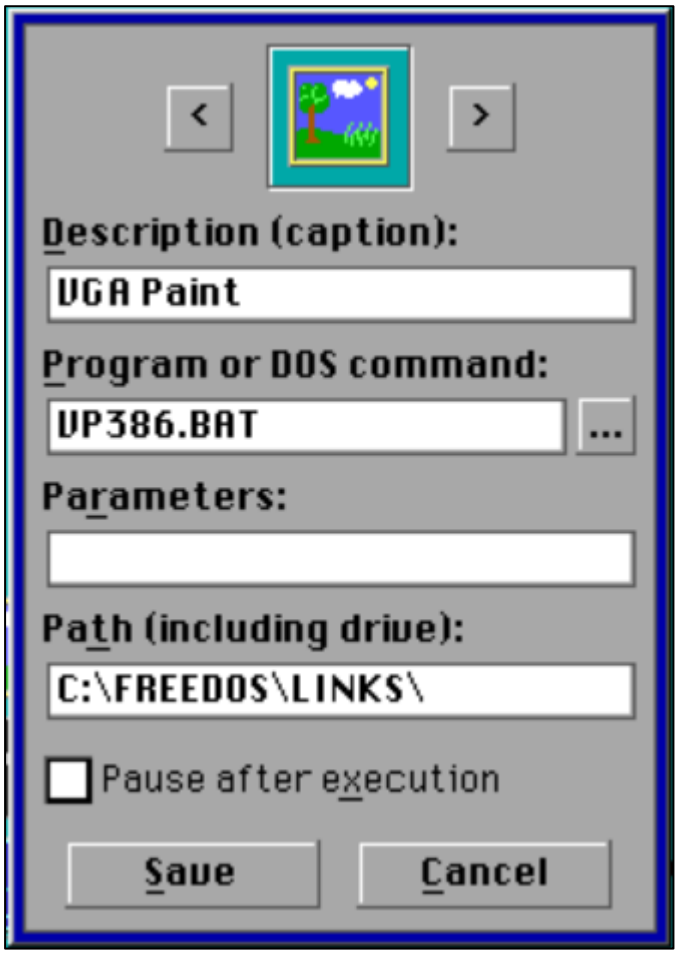
This allows the VP.EXE to find the DOS4GW system file. It is also sufficient to CD to the VP386 directory before calling it but the path method is preferred. If DOS4GW is located in the C:\FreeDOS\BIN directory then this in not needed.

VP.BAT or VP386.BAT

```
@ECHO OFF
IF EXIST %dosdrv%\APPS\VP386\VP.EXE GOTO VP
GOTO NOFOUND
GOTO END
:VP
REM You can add environment paths or change working directory here.
SET PATH=%path%;%dosdrv%\APPS\VP386
REM Launch VGA PAINT
CALL %dosdrv%\APPS\VP386\VP.EXE %1 %2 %3 %4 %5 %6 %7 %8 %9
GOTO :END
:NOFOUND
ECHO VGA PAINT not found!
ECHO Press any key to end...
PAUSE
:END
```

Move your VP.BAT or VP386.BAT file to C:\FREEDOS\LINKS and test the application launch from the command line.

Next add a link/icon in Costa to the Batch file.



PC_Paint

PC Paint, from Mouse Systems Corporation, is a Macintosh MacPaint-like paint program for the PC. It was often bundled with Mouse Systems mice. Despite the similar sounding name and appearance, it is NOT related to Microsoft/ZSoft PC Paintbrush.

https://winworldpc.com/product/pc-paint/3x

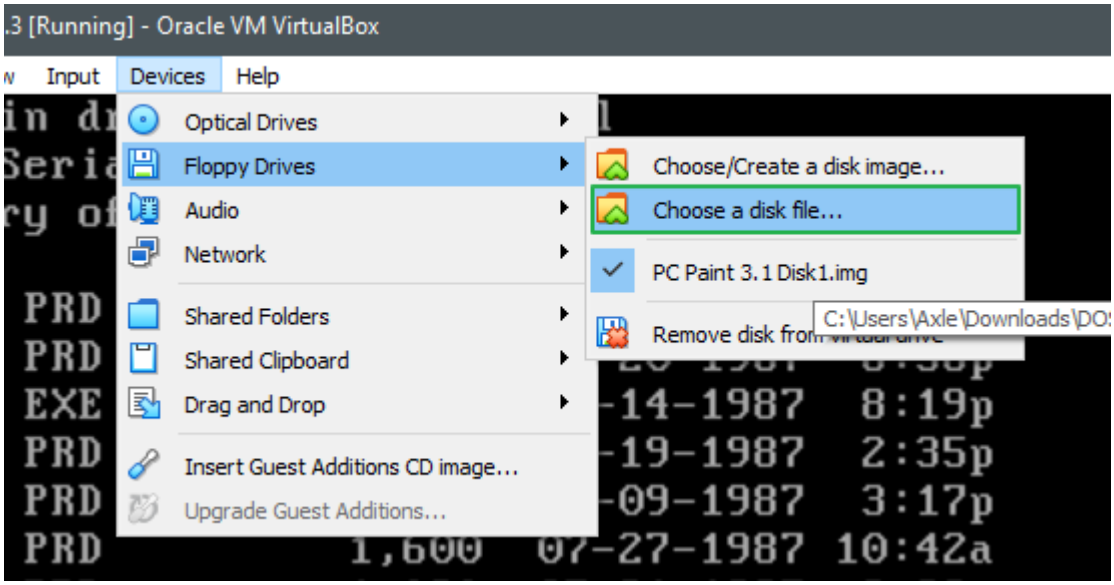
“PC Paint 3.1 (3.5).7z"

Install Instructions

Unpack the PC “Paint 3.1 (3.5).7z” archive. You will now have 2 disk images.

- PC Paint 3.1 Disk1.img
- PC Paint 3.1 Disk2.img

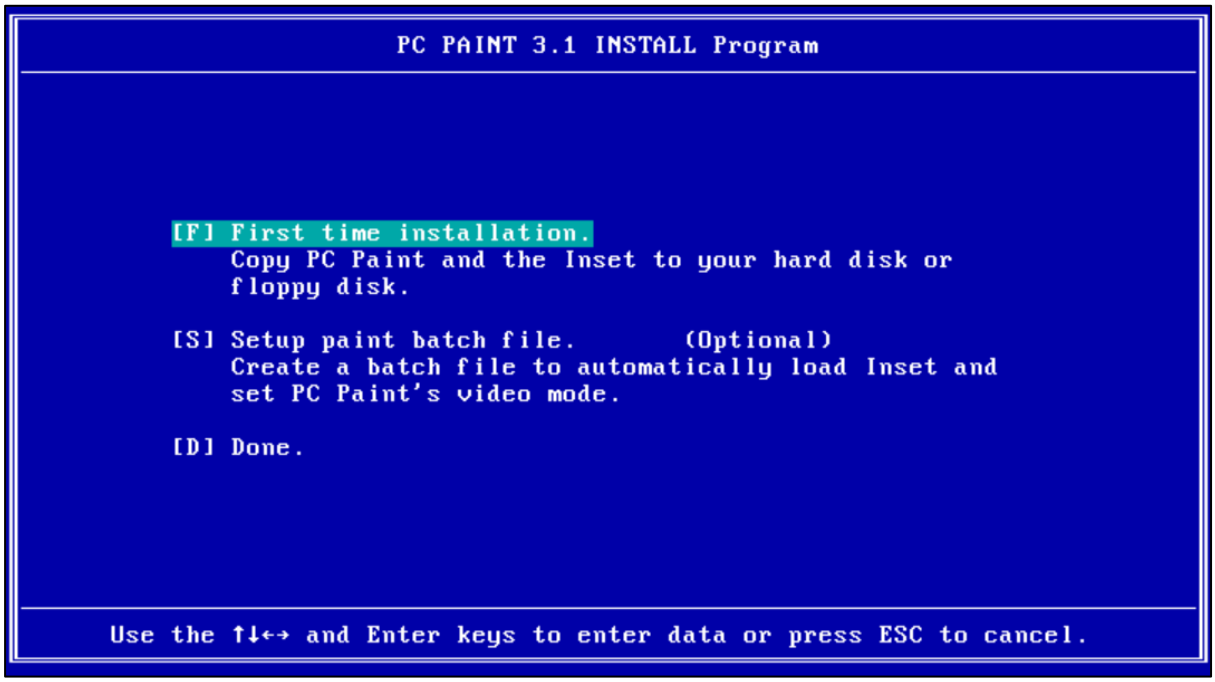
Mount the first floppy disk PC Paint 3.1 Disk1.img from VirtualBox.



Next navigate to the A: or B: drive in the FreeDOS guest and execute the "INSTALL.EXE" file from the floppy disk and follow the install steps. It is best to do this directly from the command line as FM3 sometimes has difficulty in locating multiple floppy images in an install.

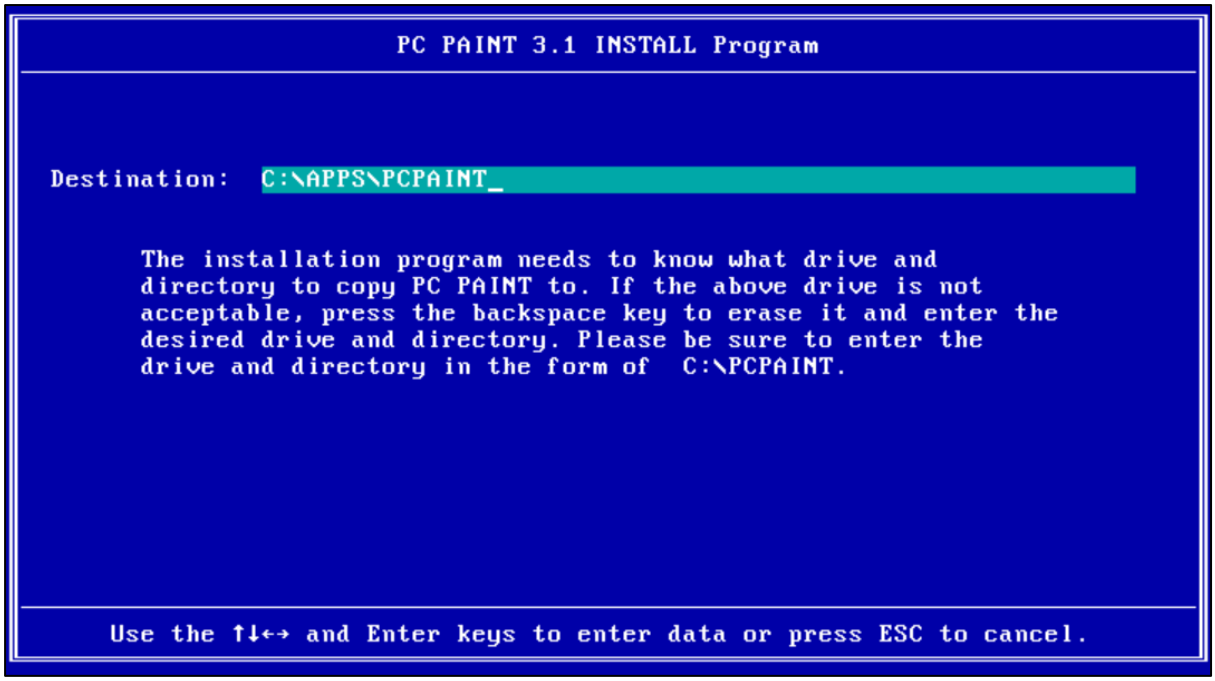
Note: The help document is PCP31.DOC

Select [F] First time installation.



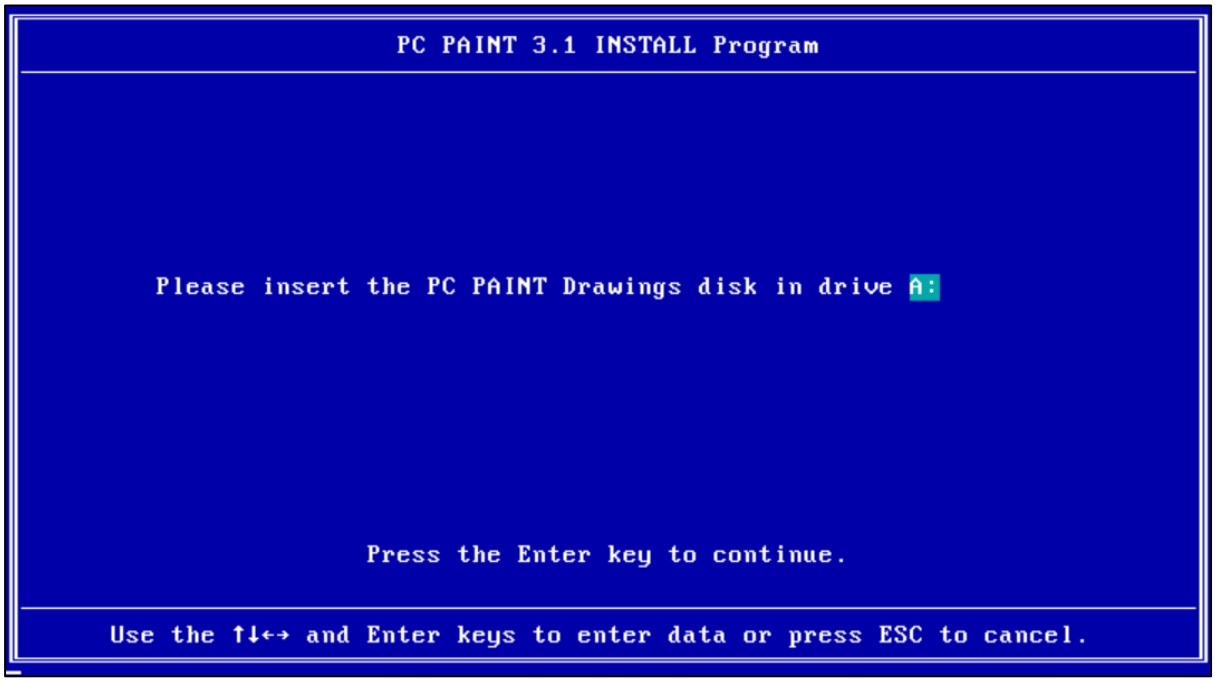
In the next screen I would suggest changing the directory to your APPS directory if using FreeDOS.

C:\APPS\PCPAINT

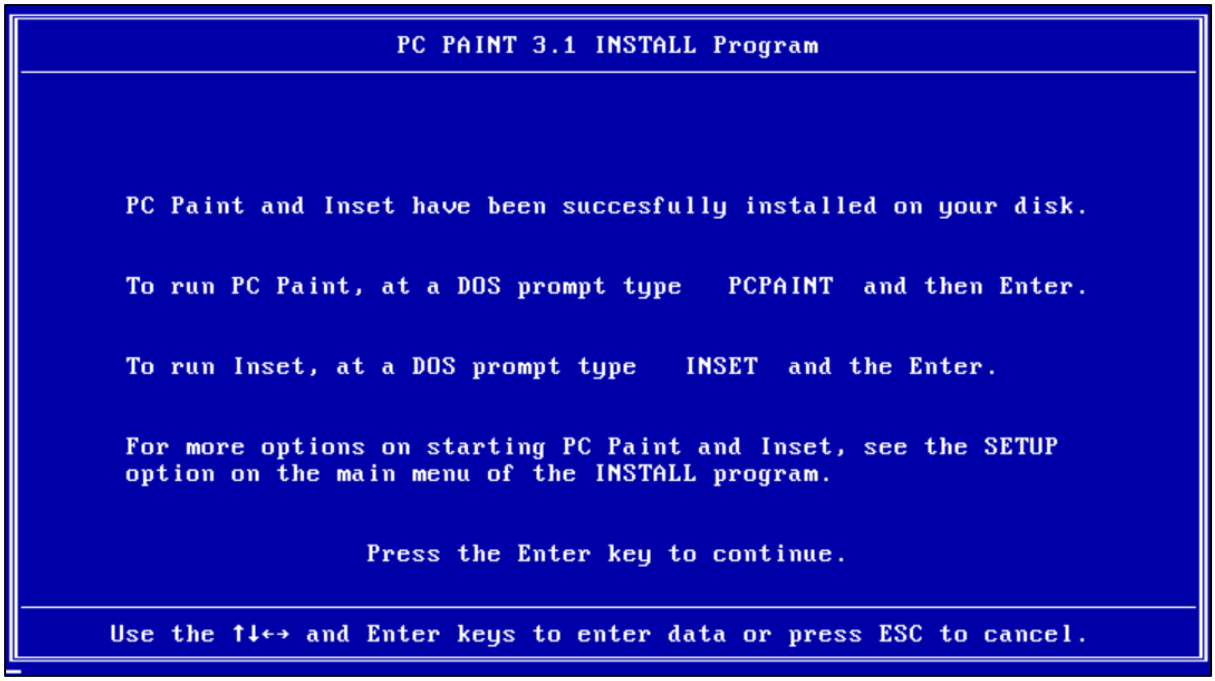


When asked use the VirtualBox menu to choose the second floppy disk file PC Paint 3.1 Disk2.img.

Press enter to continue when the second disk is mounted.

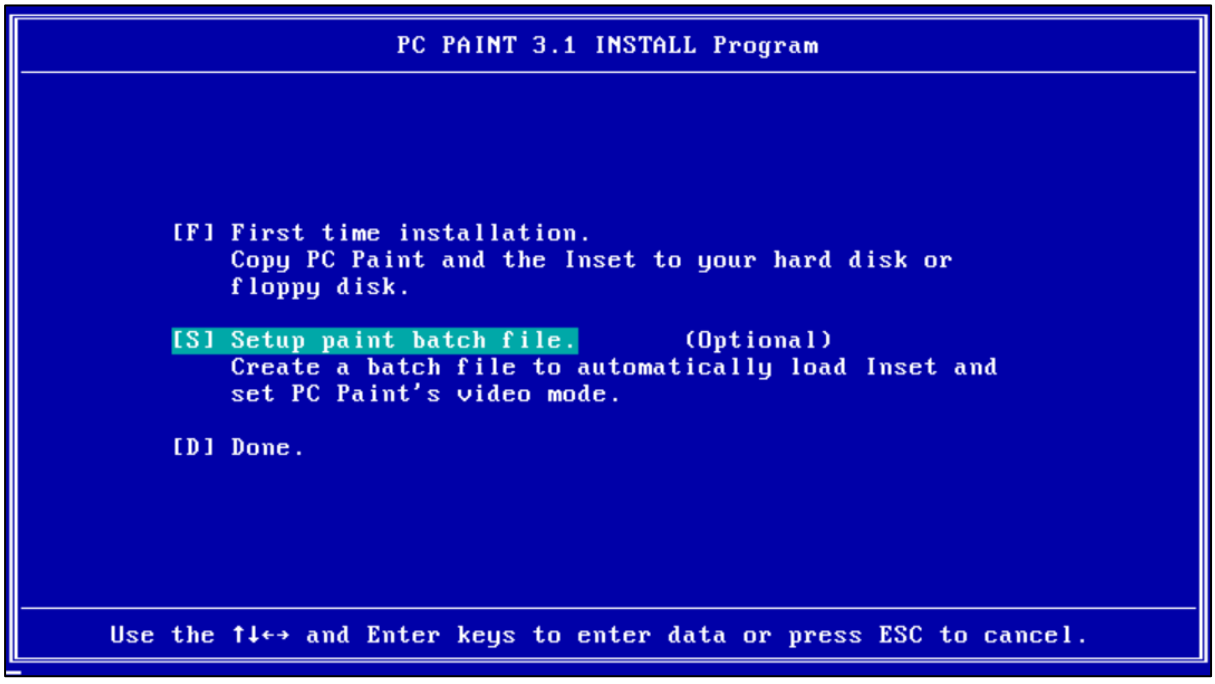


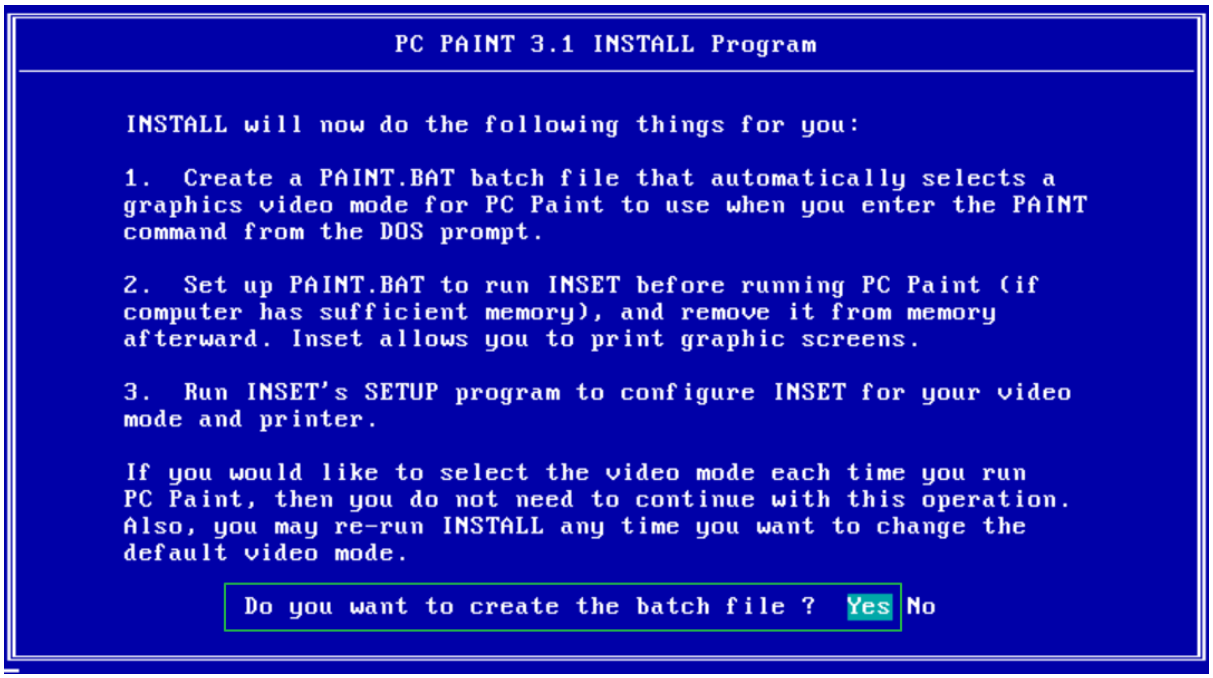
The install has completes successfully.

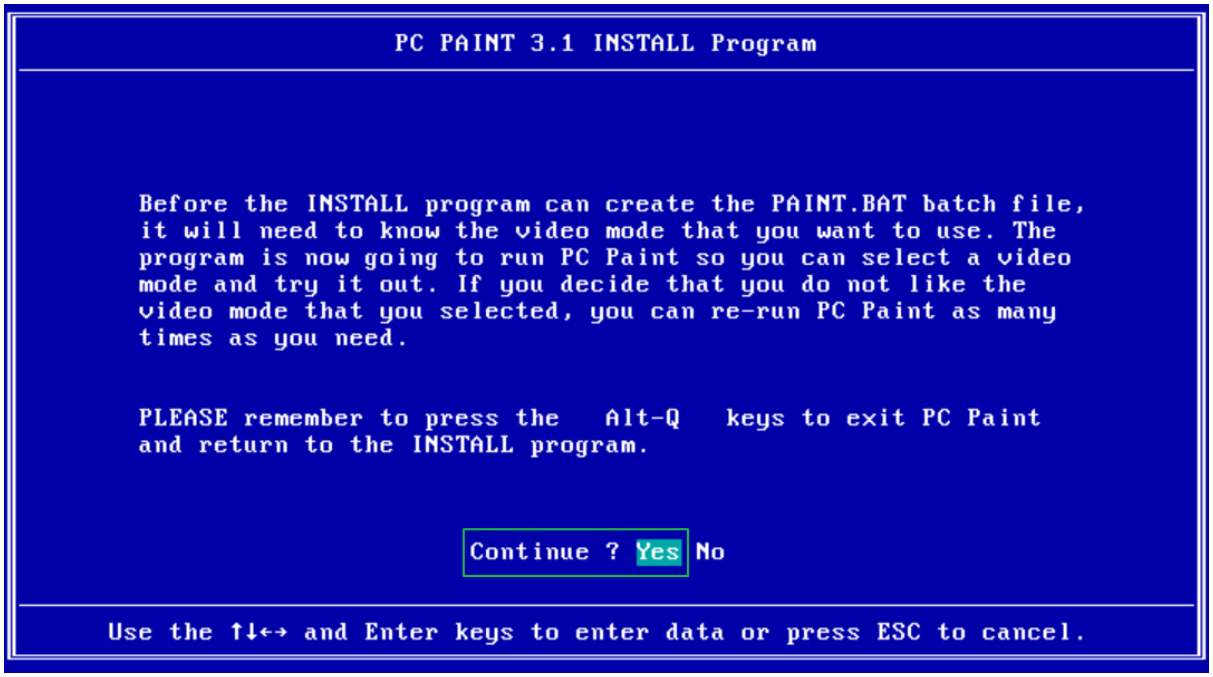


The following is optional and allows you to set up the screen mode, a printer driver and create a batch file. I have shown it here only for an illustration. Do not run the setup unless you have a specific reason to do so. I would recommend reading on to the "Create PC Paint Batch File" and skipping the following setup tasks.

Next you will be offered the option to set up PC PAINT and create a batch file to launch the application. This step is optional and can be done manually later if you wish. Note: you can run PCPaint directly without setting the video mode or creating a batch and do the setup and batch file at any later time.





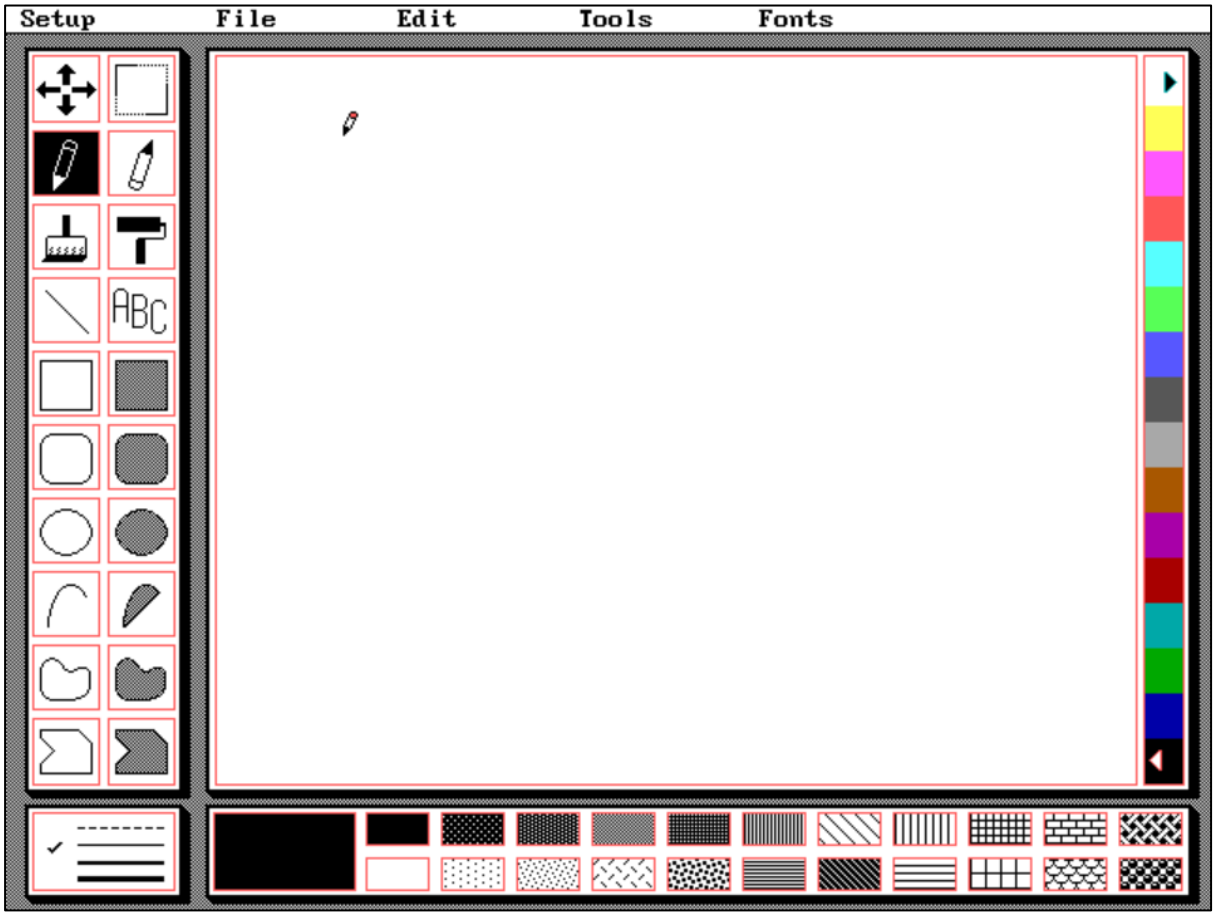


This VGA mode is entirely optional and depends upon your image use requirements. If you are unsure for the start select either G EGA 640x350 16-color or M EGA/VGA 640x48016 colors. Unlike a direct hardware monitor the virtualBox display window will automatically re-adjust to the video size unless you are in VirtualBox “Full screen” mode. Start with a smaller display size if you are unsure.

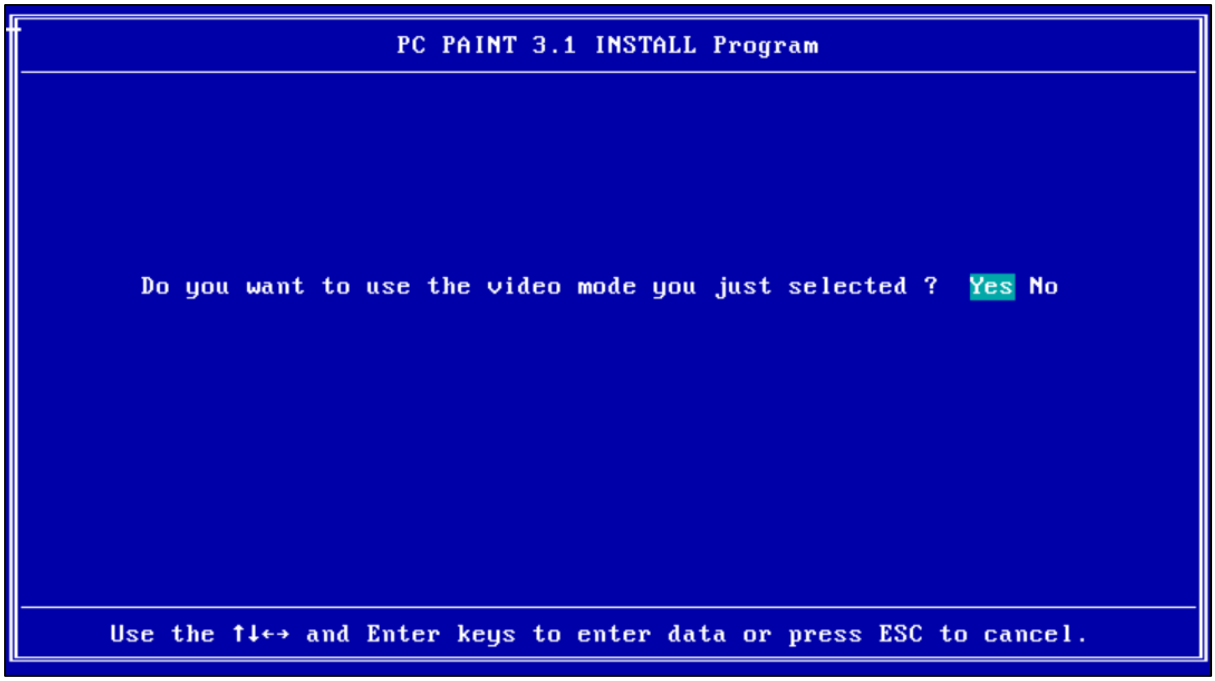
16 colors is the most common in DOS, but some advanced applications can use 256 colors or higher.

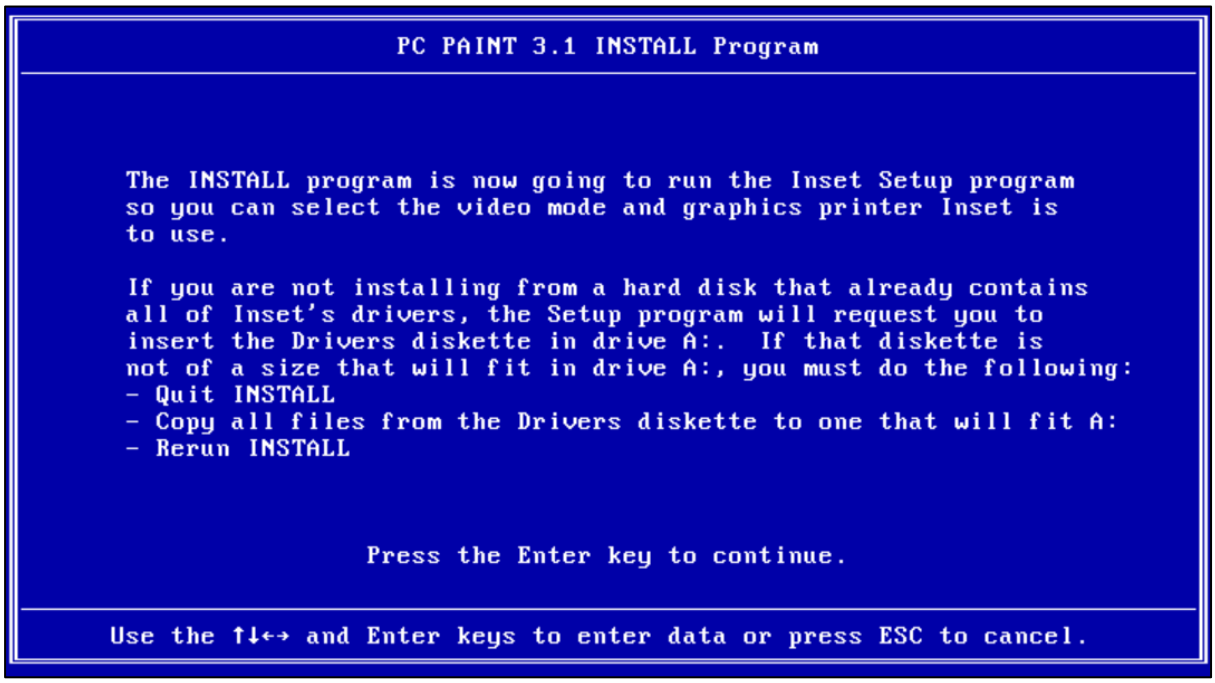
```
                    P C P A I N T  3.1
            Written by John Bridges and Doug Wolfgram
       Choose a Video Mode by Pressing a Letter or Press ESC to Quit

            A -> CGA                320x200    4 colors
          * B -> Tandy 1000         320x200   16 colors
            C -> CGA                640x200    2 colors
            D -> EGA                640x200   16 colors
            E -> EGA                640x350    2 colors
            F -> EGA                640x350    4 colors
            G -> EGA                640x350   16 colors
          * H -> Hercules Mono      720x348    2 colors
          * I -> Plantronics        320x200   16 colors
            J -> EGA                320x200   16 colors
          * K -> AT&T/Toshiba       640x400    2 colors (mode 40h, 72h)
            L -> MCGA/VGA           320x200  256 colors
            M -> EGA/VGA            640x480   16 colors
          * N -> Hercules InColor   720x348   16 colors
            O -> MCGA/EGA/VGA       640x480    2 colors
            P -> EGA/VGA            800x600    2 colors (mode 29h)
            Q -> EGA/VGA            800x600   16 colors (mode 29h)

    Modes marked with '*' are DEFINITELY not available on this system.
WARNING: Certain high-resolution modes may not be available on this system!
            Copyright(C) 1989 - Mouse Systems Corporation
```

After entering M.



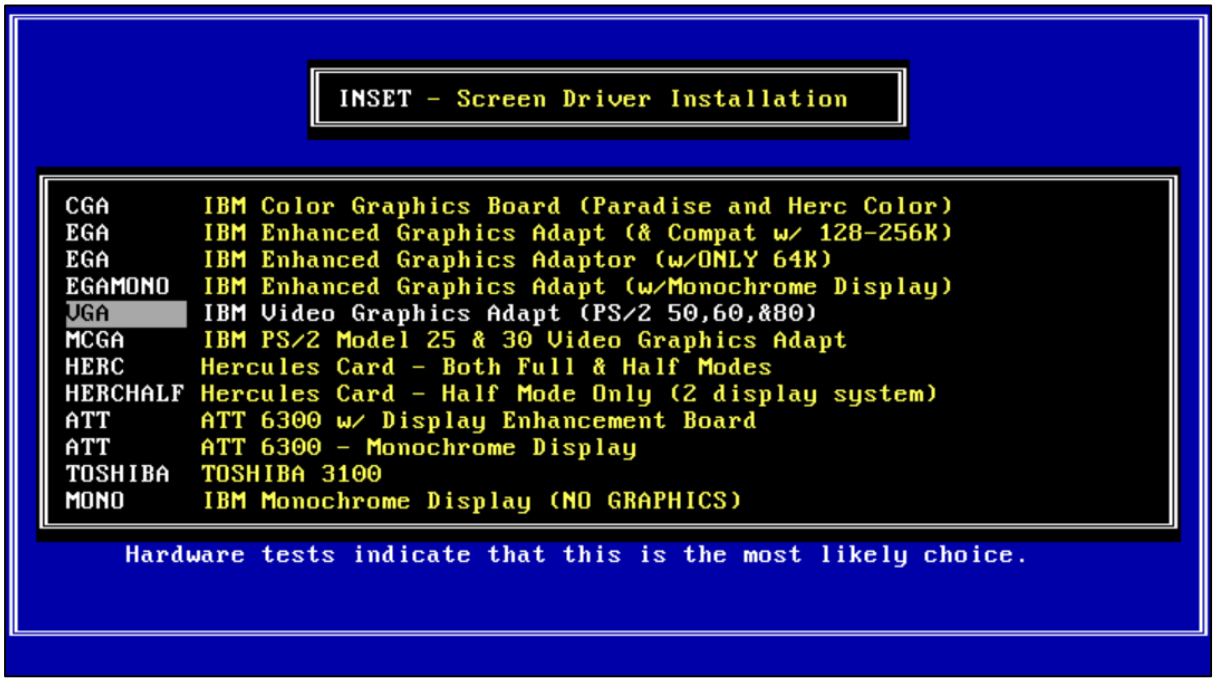
After testing the video mode select yes to continue creating the batch file.



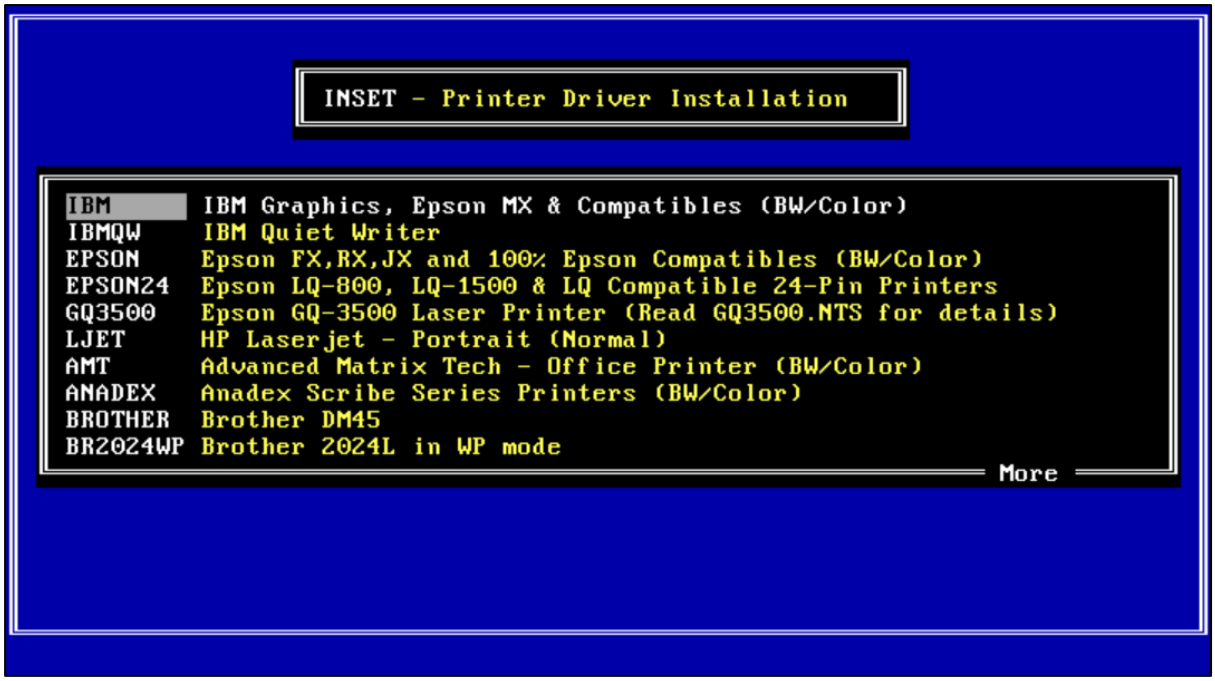


In the next steps the setup program will ask to set up your default hardware printer. Unless you have a specific hardware printer for the period just enter the default values as we won't need to use the printer functions.

Select VGA unless you have a specific driver you need to use.

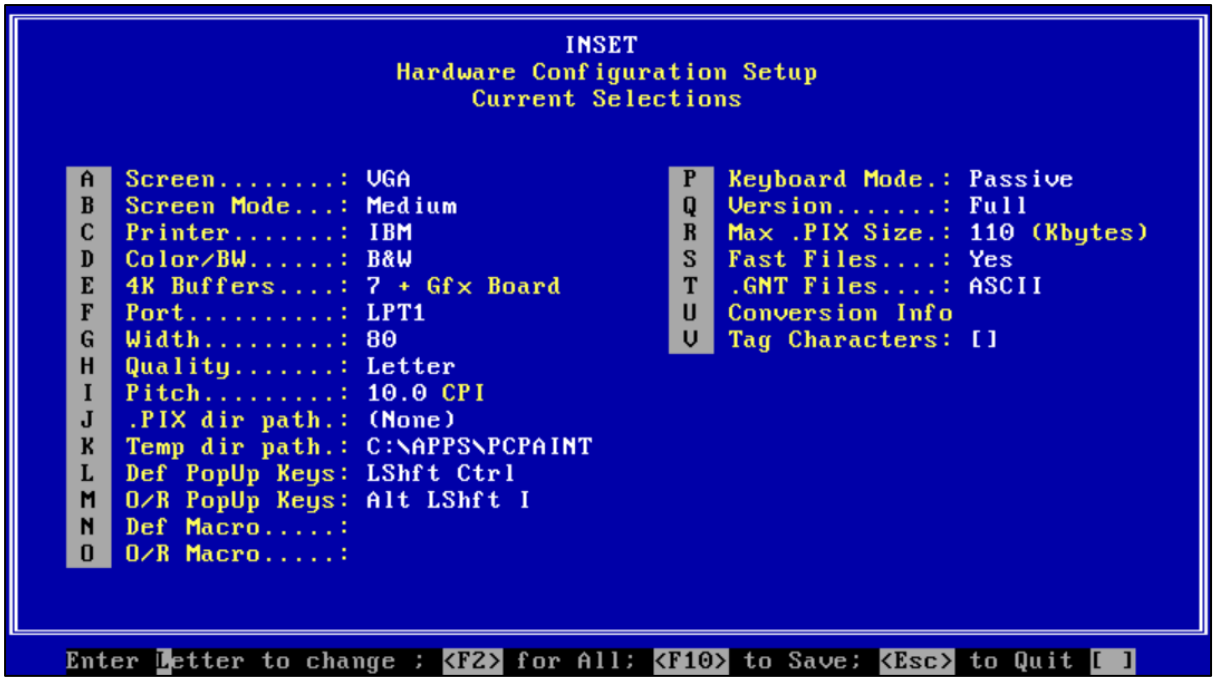


Select the default printer device if using VirtualBox.



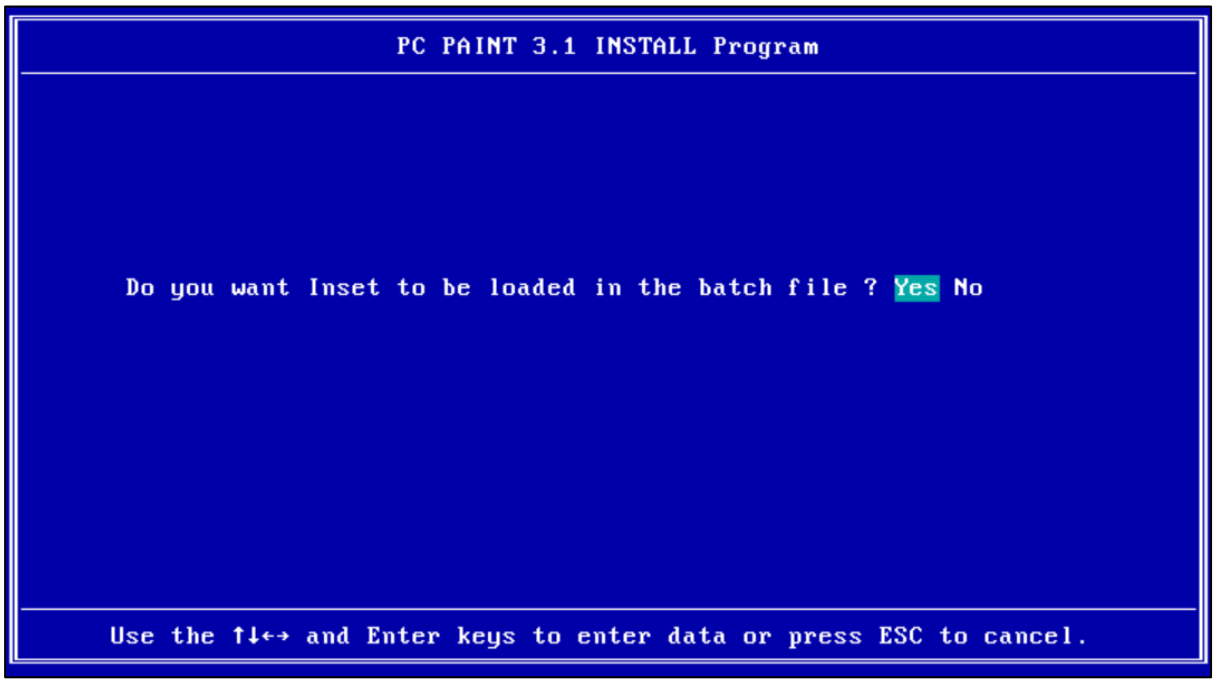
Then select B&W.

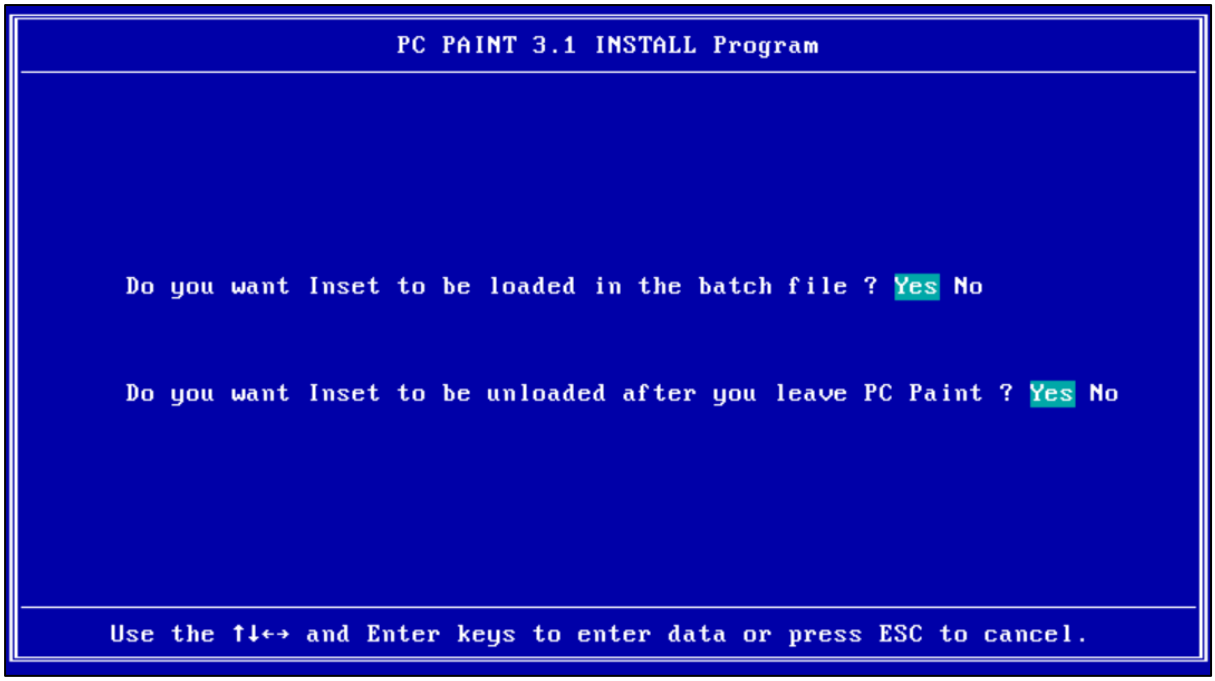
Press F10 to save.



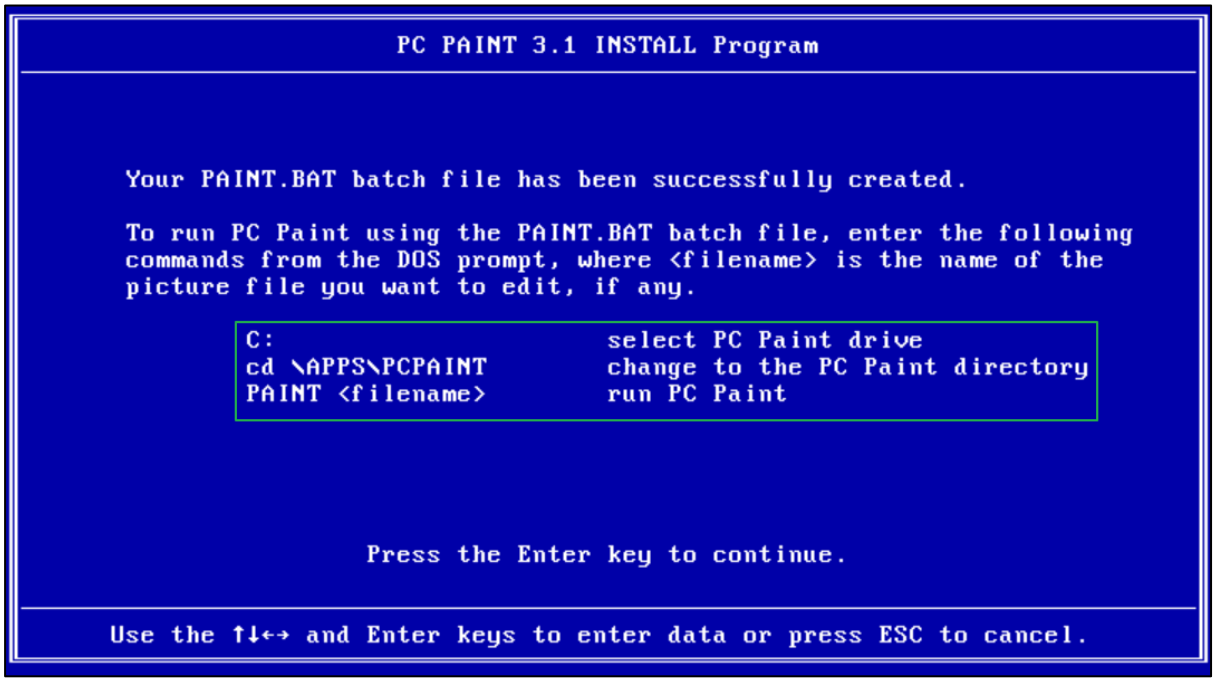
Insert the disk 1 install disk.

Select yes for both options.

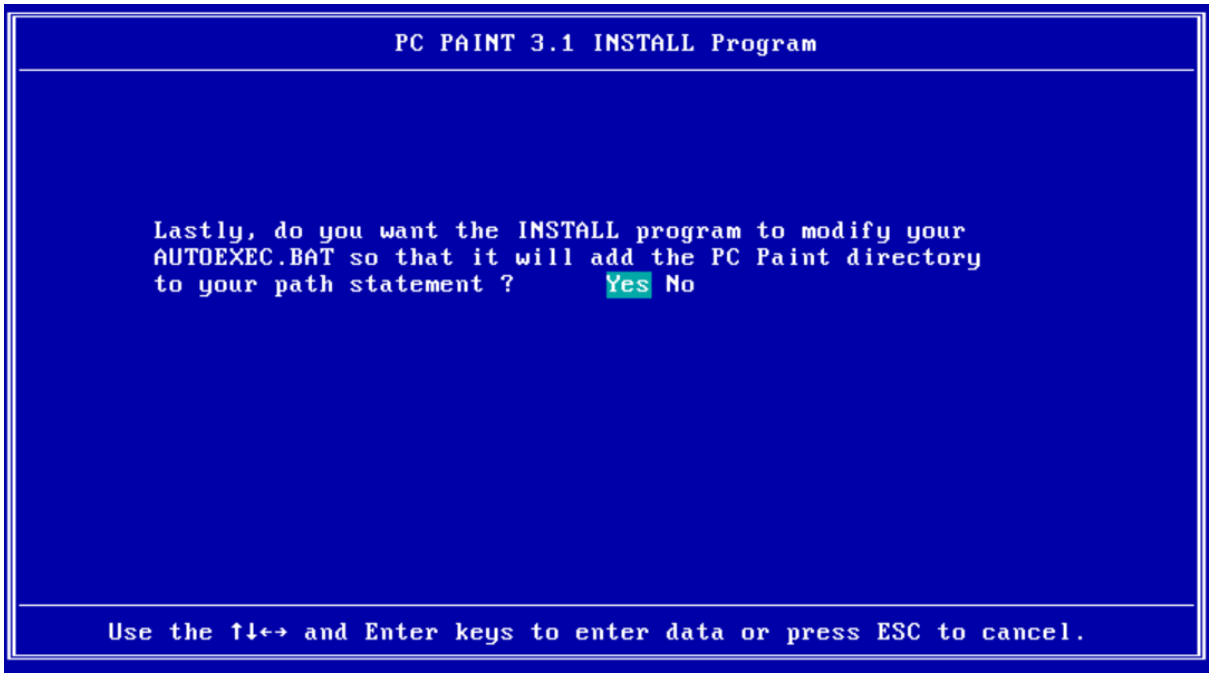


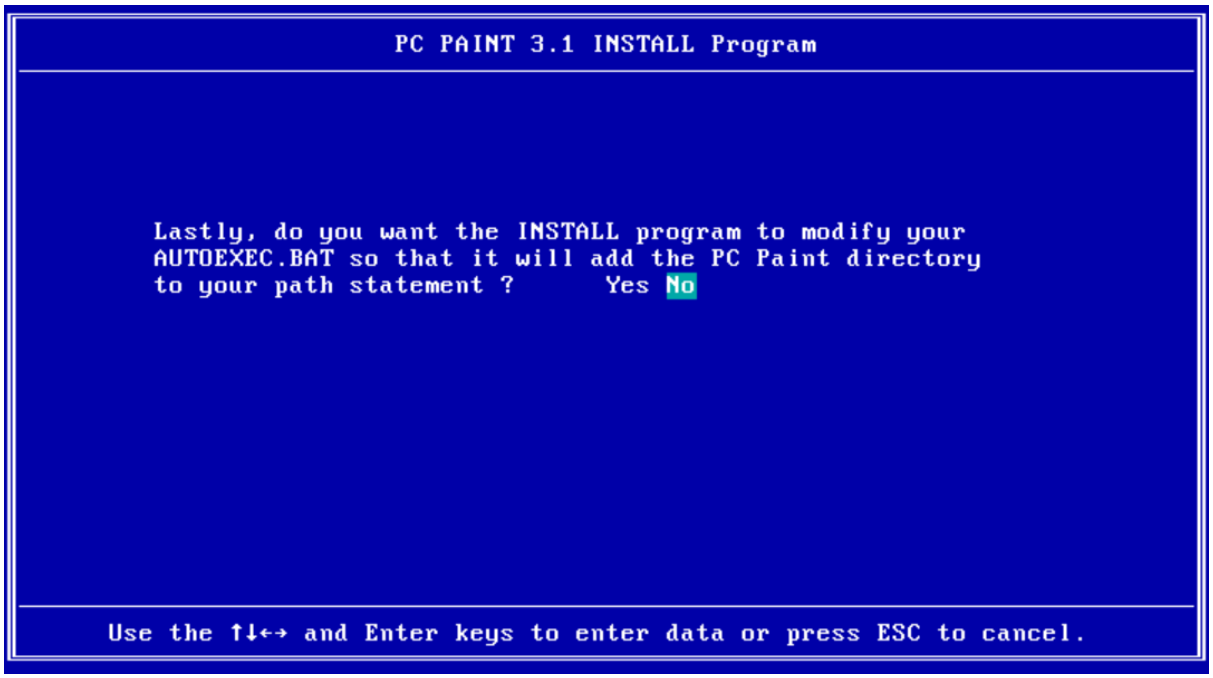


Next you will be notified of the batch file location.

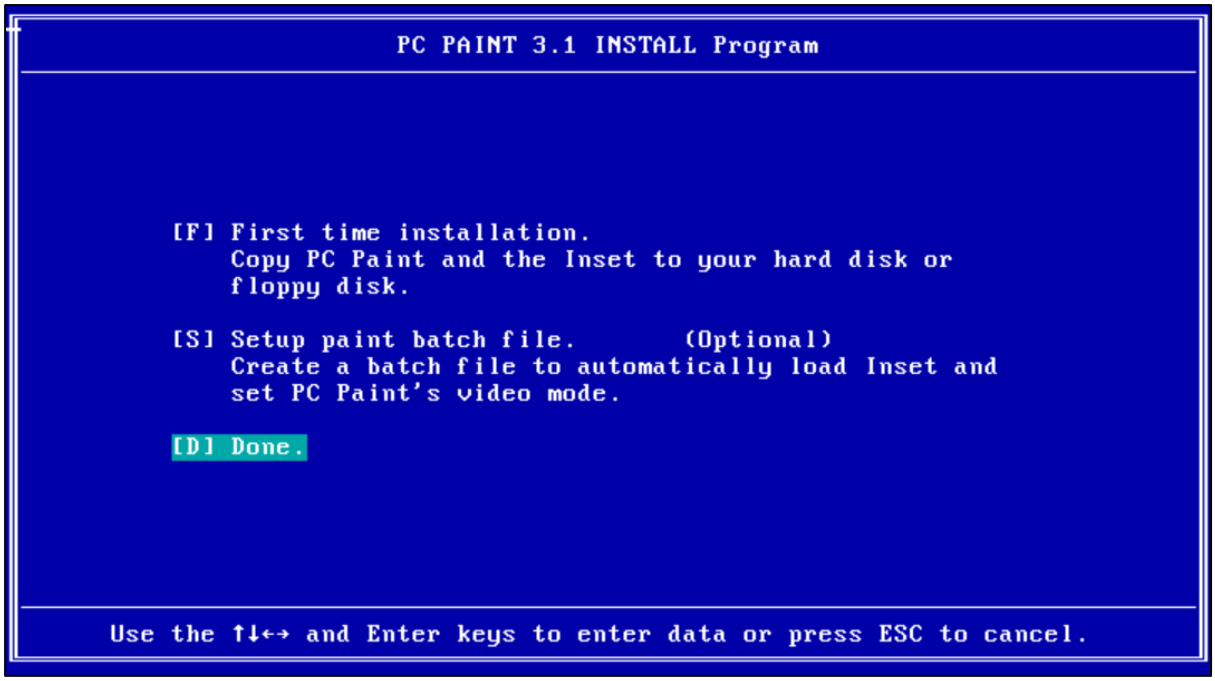


In the next screen select Yes/No. Selecting yes will add a path entry to your AUTOEXEC.BAT file. I would recommend NO unless you have reason to add the path to AUTOEXEC.BAT

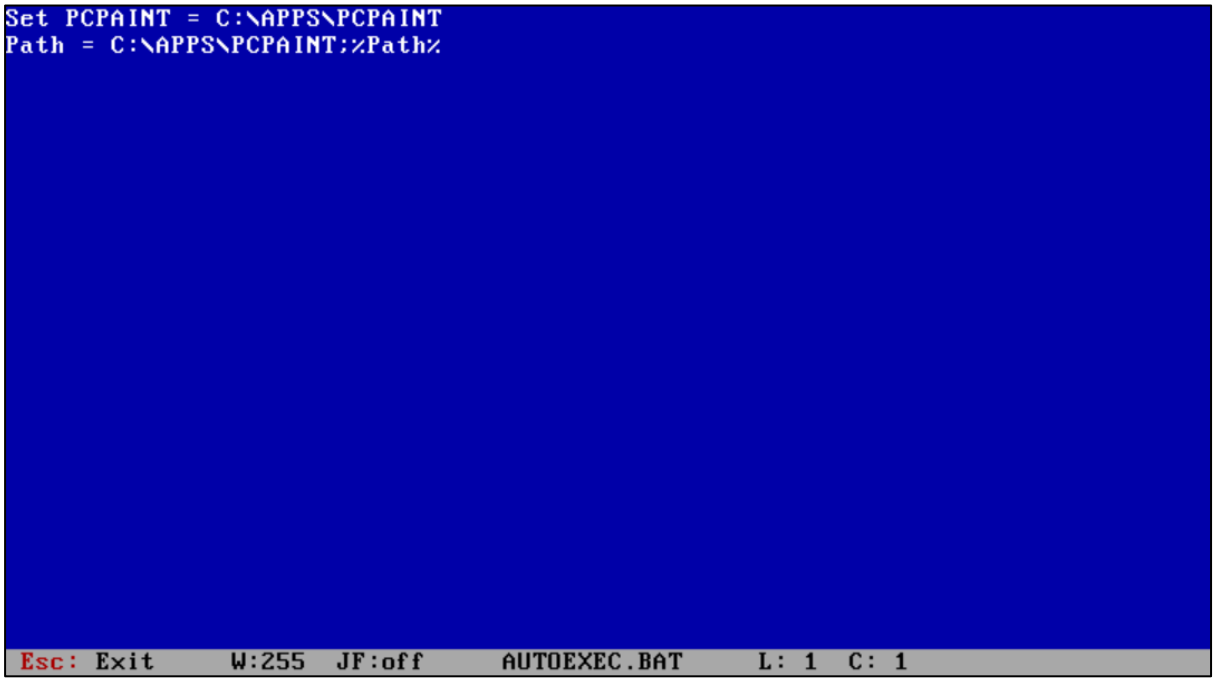




Select [D] Done to exit the setup.



At the end you will have this entry in the AUTEXEC.BAT file if you selected YES.



and this PAINT.BAT file in the PC PAINT directory.

```
echo off
cls
Inset
Set PCPaint = C:\APPS\PCPAINT
PCPaint /V:M %1
Af
Cls
```

```
Esc: Exit      W:255  JF:off        PAINT.BAT      L: 1  C: 1
```

Create PC Paint Batch File

As you may see from above, the setup goes through a lot of tasks to create the batch file. You can skip the above by creating the single batch file shown below using only the required lines from the above 2 batch files. When done copy the batch file to C:\FREEDOS\LINKS.

Inset is to initiate the printer drivers and is not required. "Af" is to unload the printer drivers and is not required.

All we need is to set the path in the system environment and change the working directory to .\PCPAINT.

/V:M sets the video mode as shown in the selection options above. You can also find the additional screen modes in PCP31.DOC

Create the following batch file PAINT.BAT

Note: Be aware that other image editor applications may also use PAINT.BAT. If this is the case rename the batch file to PCPAINT.BAT or some other name that does not conflict.

PAINT.BAT PCPAINT.BAT

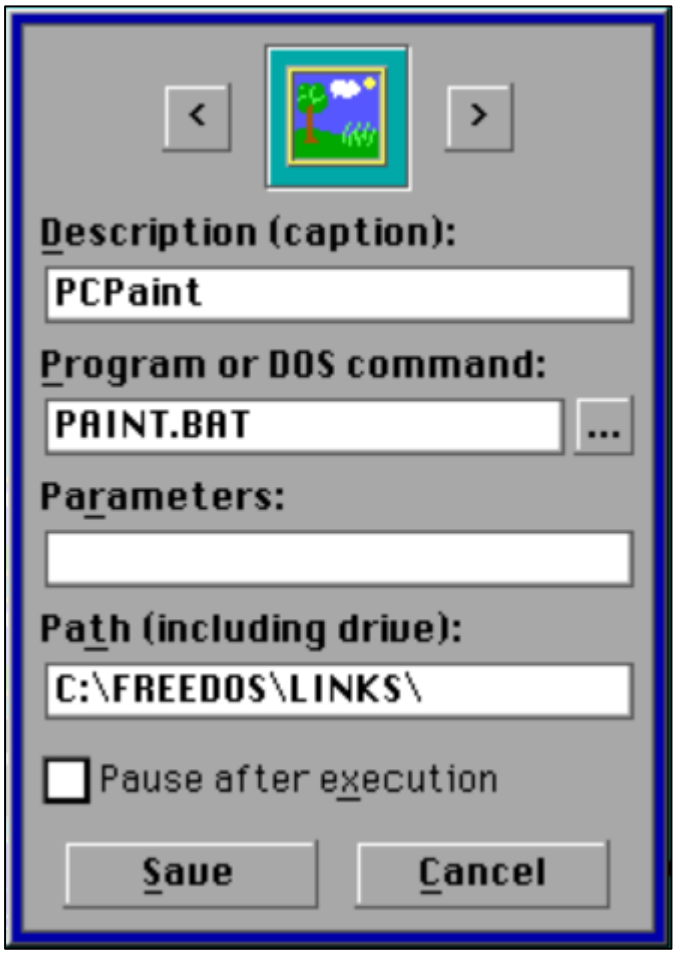
```
@echo off
CLS
REM SET PCPAINT=C:\APPS\PCPAINT\
SET PATH=%path%; C:\APPS\PCPAINT
CD \APPS\PCPAINT
PCPAINT.EXE /V:M %1 %2 %3 %4 %5 %6 %7 %8 %9
CLS
```

You will notice that it is very similar to the example batch files created in "**Application launch BAT**".

It is important to change the working directory to the PCPAINT directory otherwise the application has trouble finding some of its files.

You can also navigate to the PCPAINT directory and call the executable file with the arguments found in PCP13.DOC "`PCPAINT.EXE /v:M`" where /v:M is the video mode.

Copy your batch file to `C:\FREEDOS\LINKS` and create a Costa icon that points to the batch file.



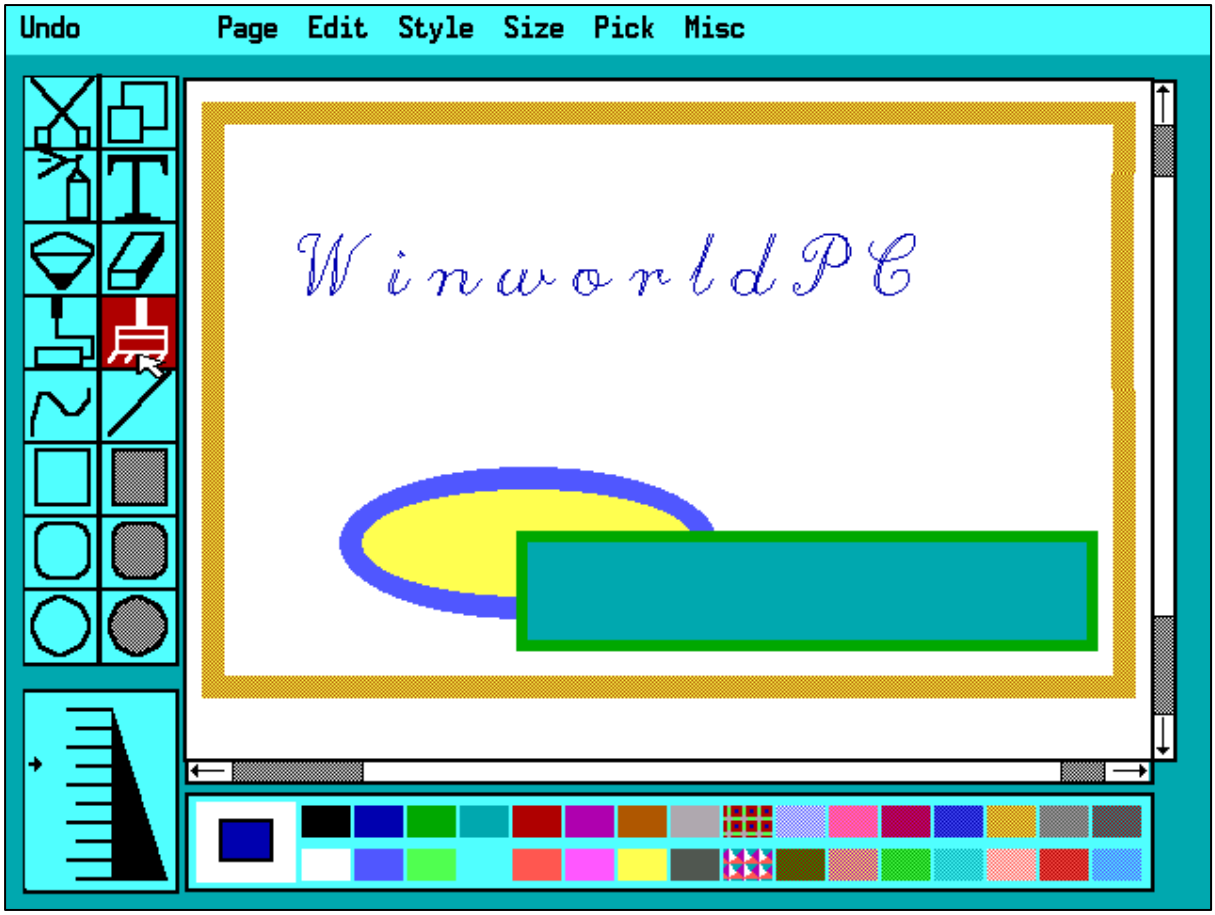
Note that all of the installed files are under the single .\PCPAINT directory. You can move the location of the PCPAINT directory to another location as long as you update the batch file to reflect the new path.

The sample images from the second disk are copied into root of .\PCPAINT. If you want them in a separate location follow the manual archive unpacking section in PCP31.DOC. Note you will need to unpack the files of the floppy drive images or do a basic install to gain access to the help file.

To uninstall PCPAINT simply delete the .\PCPAINT directory and the batch file.

PC Paintbrush

https://winworldpc.com/product/pc-paintbrush/3x-dos



“PC Paintbrush 3.77 (5.25-360k).7z”

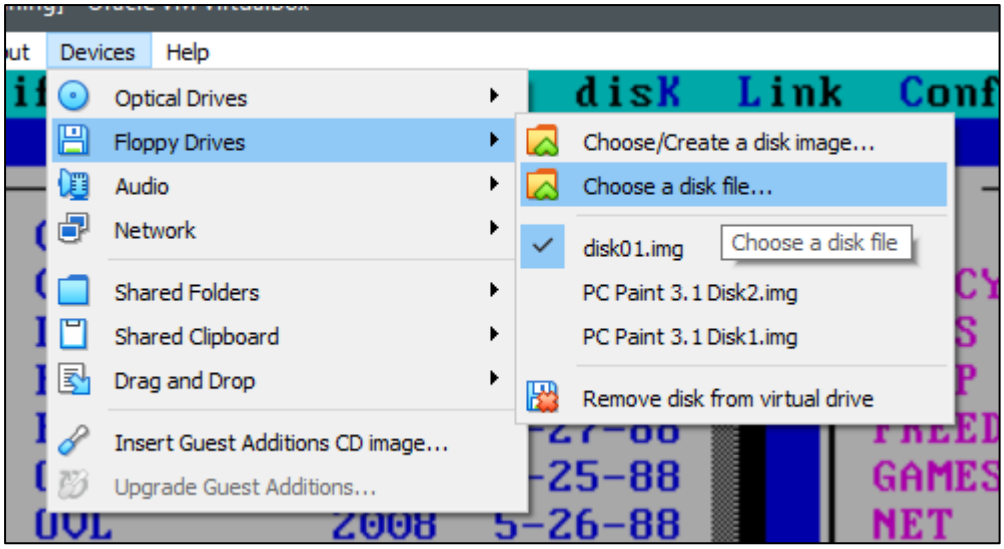
“PC Paintbrush 3.7x Manuals.7z”

Install Instructions

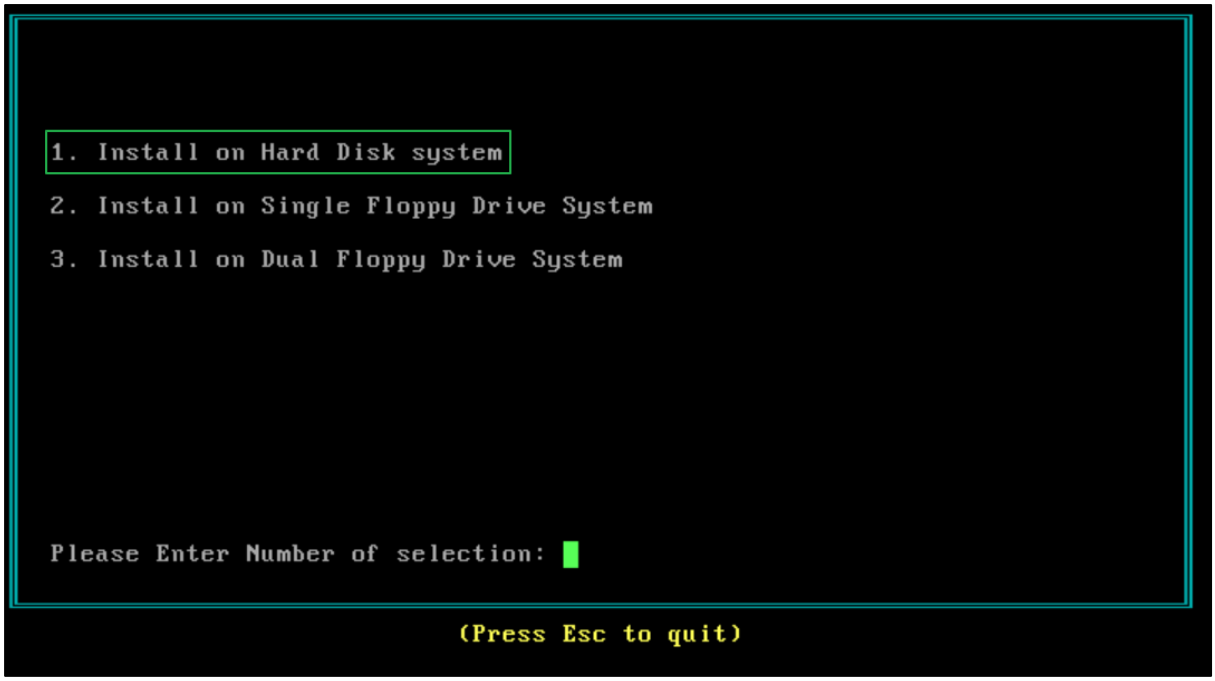
Unpack the PC Paintbrush 3.77 (5.25-360k).7z archive. It will contain 3 floppy disk images.

- disk01.img
- disk02.img
- disk03.img

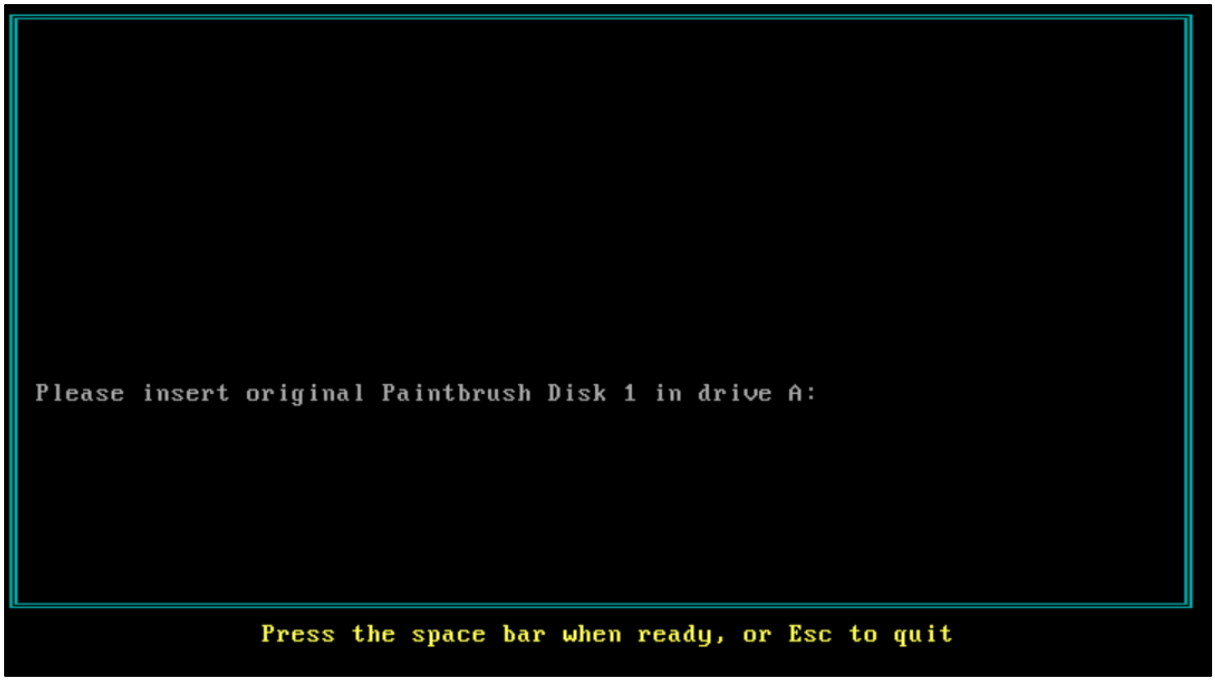
Mount the first floppy disk disk01.img from VirtualBox.



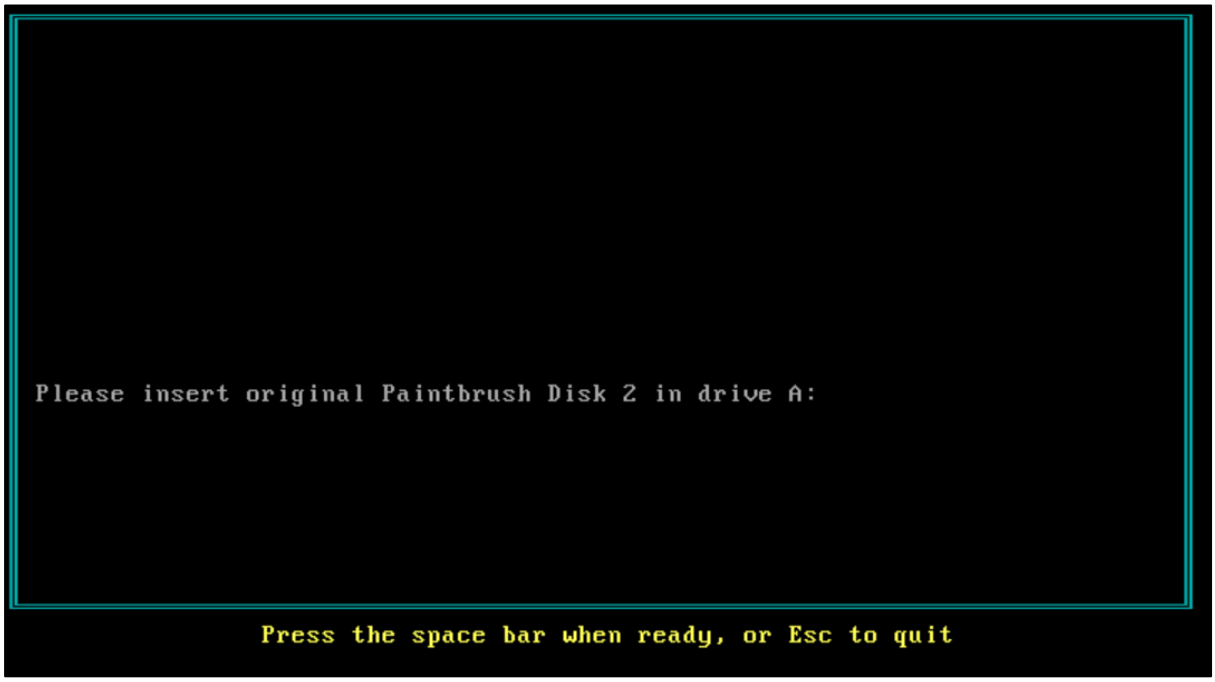
Navigate to the floppy drive directory A: and run the SETUPPB.EXE file and follow the install instructions.



Change the install location to C:\APPS\PBRUSH and then press Enter.



Mount disk02.img in VirtualBox.



PC Paintbrush is installed. You will next need to run the setup application hardware.

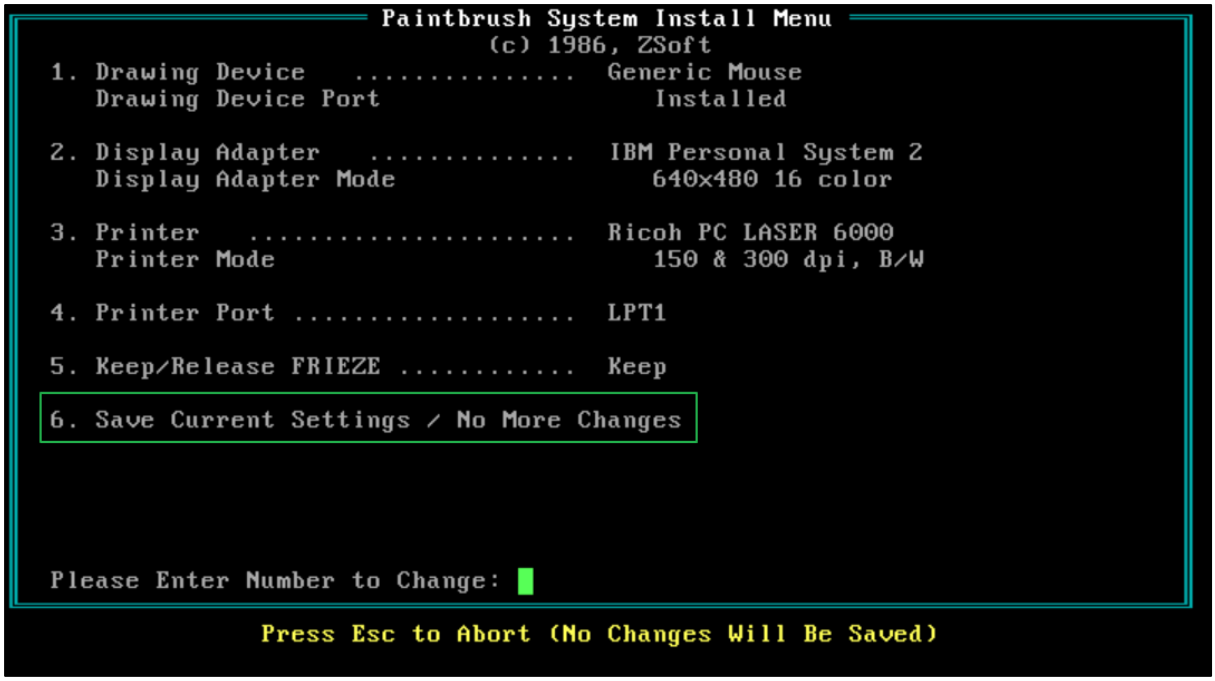
```
  The Paintbrush files have been successfully copied.

  To complete the installation, you must run PCINSTAL.
  PCINSTAL will help you specify the correct hardware (graphics
  display adapter, printer, drawing device) that will be used.

  If you are installing for a hard disk system, PCINSTAL is on
  the new directory you just copied the files into.
  If you are installing for a floppy disk system, PCINSTAL is
  on the original Paintbrush Disk 1.

Press any key ...
```

Navigate to the `C:\APPS\PBRUSH` directory and run `PCINSTAL.EXE`.

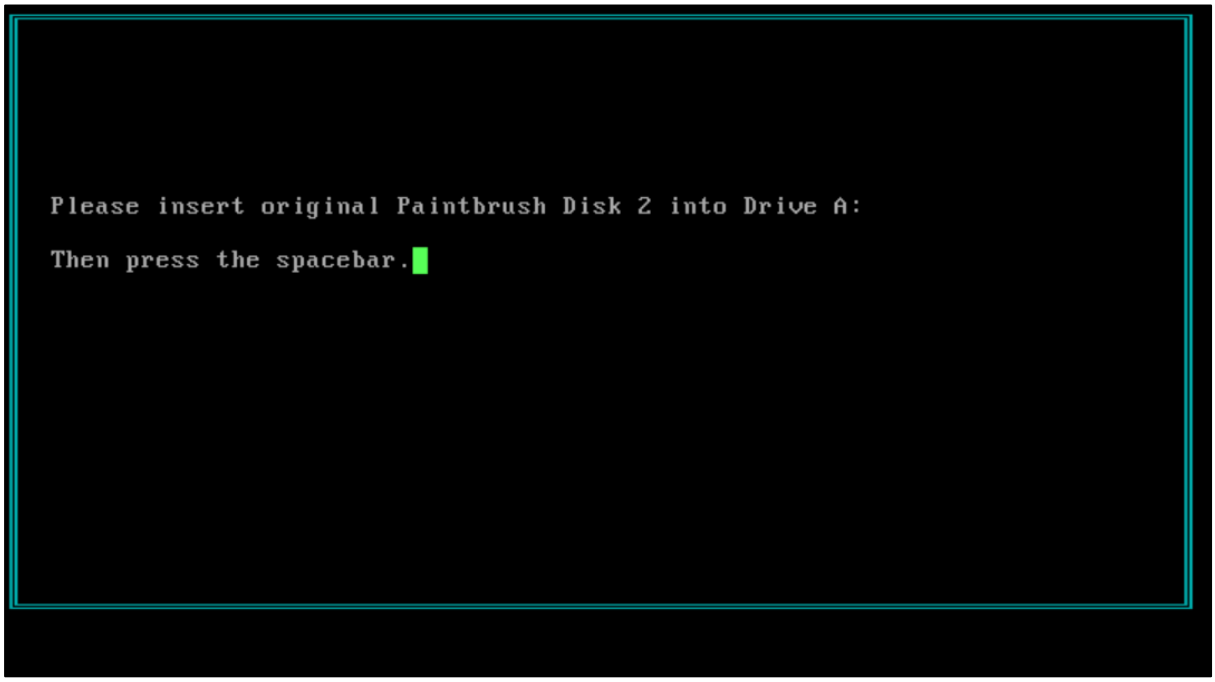
```
                    P C   P A I N T B R U S H

                Copyright 1986, ZSoft Corporation

                    Installation Program
                            for
                    ZSoft Paintbrush systems

                        Installer Version: 3.3


   Press the spacebar to continue.
```

In the next screen the generic default options are fine for VirualBox guest.

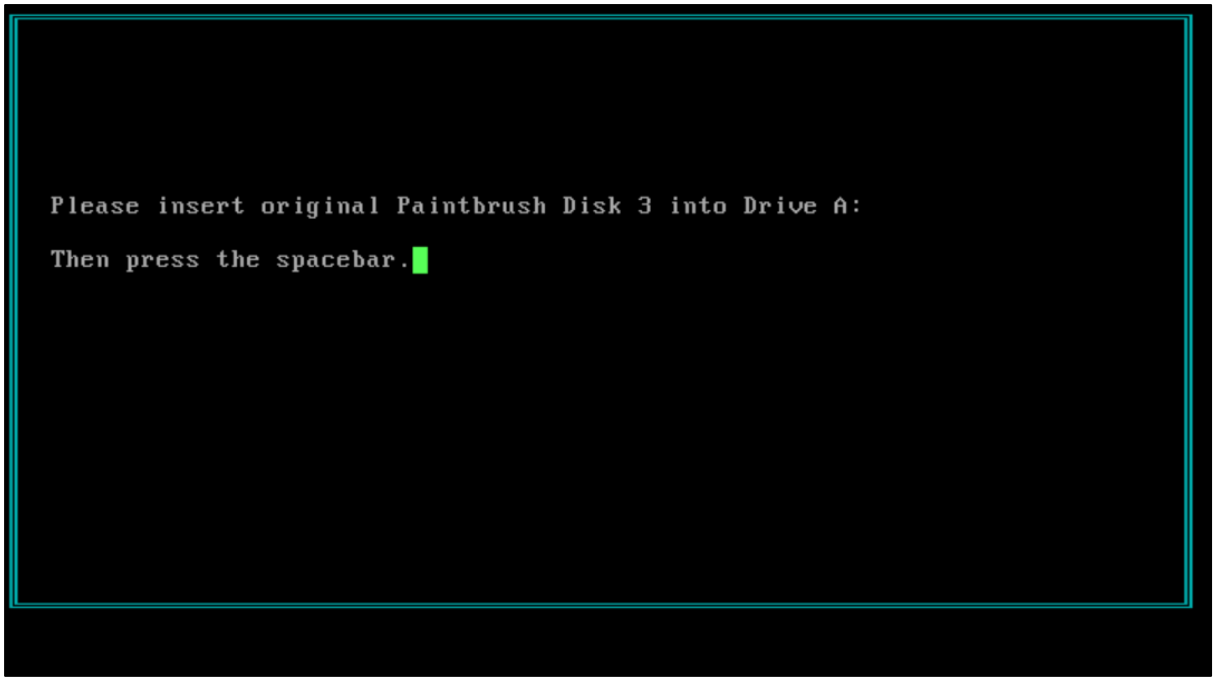
Select 6 to save and continue.



Remount disk02.img in VirtualBox if it was removed and then continue.

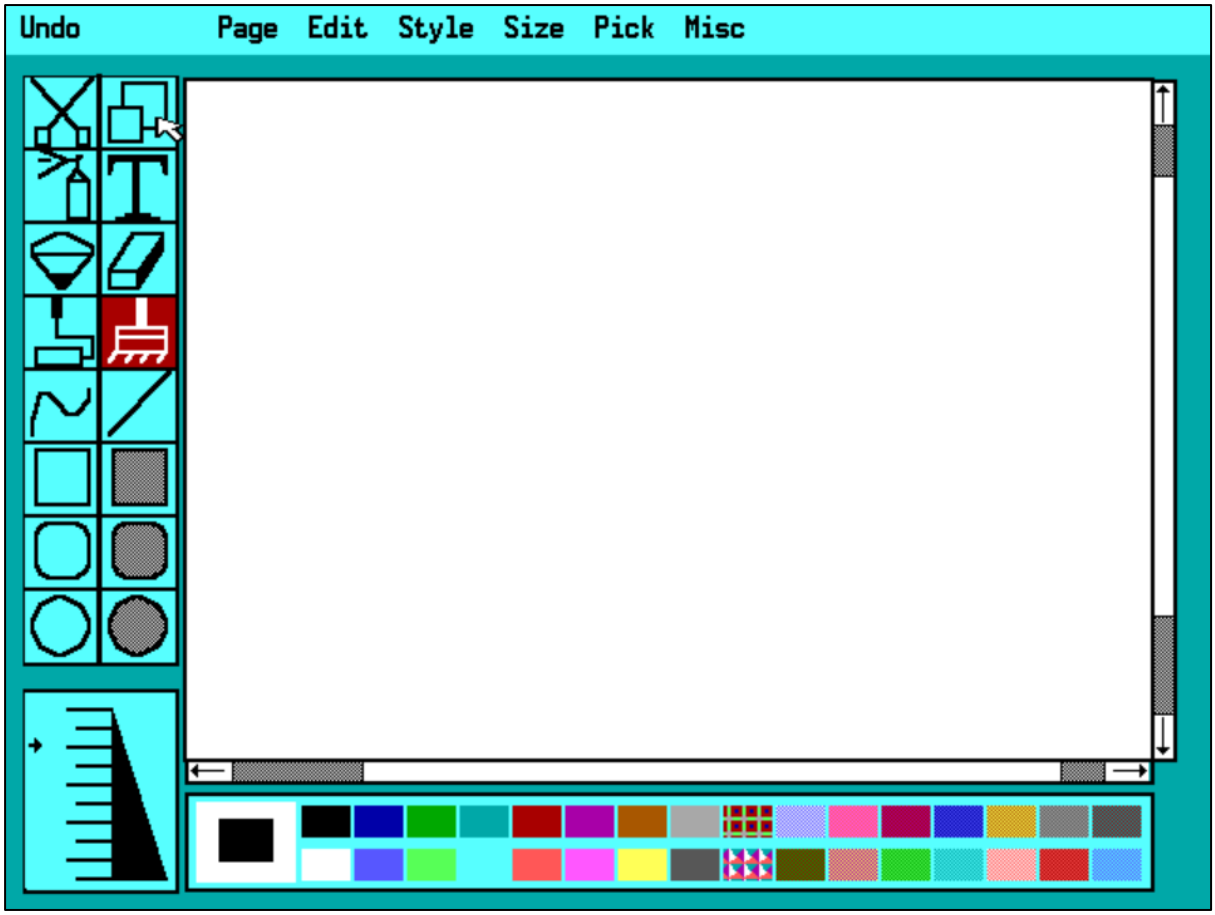


Next mount disk03.img in VirtualBox and then continue.



The setup and install is now complete.

Run the `PAINT.BAT` found in the `C\APPS\PBRUSH` directory and test the application.



Caution! Before proceeding check for other batch files named PAINT.BAT and change the name of the batch file if it is in conflict.

Move PAINT.BAT to `C:\FREEDOS\LINKS\PAINT.BAT`

Open `C:\FREEDOS\LINKS\PAINT.BAT` in an editor and add the following lines.

PAINT.BAT PBRUSH.BAT

```
@ECHO OFF
CD \APPS\PBRUSH
FRIEZE RICOHLP P1 0000 1CNQ 640 480 4
PBrush M 1CNQ 640 480 4
CLS
```

Test that the PAINT.BAT file works correctly. You can now create an icon/link to the batch file in Costa.

The help documents are contained in a separate downloaded archive “PC Paintbrush 3.7x Manuals.7z” as PDF files. PDF readers are available for FreeDOS but I have not tested any at this time. You can find DOSPDF in the repository using the package manager (FDNPKG).

https://www.ibiblio.org/pub/micro/pc-stuff/freedos/files/util/file/dospdf/

PC Piantbrush V4.x will follow a similar install to V3.7

https://winworldpc.com/product/pc-paintbrush/iv-dos

“ZSoft PC Paintbrush IV Plus (5.25-360kb) (3.5-720kb).7z”

“ZSoft PC Paintbrush IV Plus Manual.7z”

Install Instructions

I am not going to go through the installation procedure for the commercial version but expect it will be similar to the V3 installer.

GrafX2 (FreeDOS)

A graphics editor ported to FreeDOS.

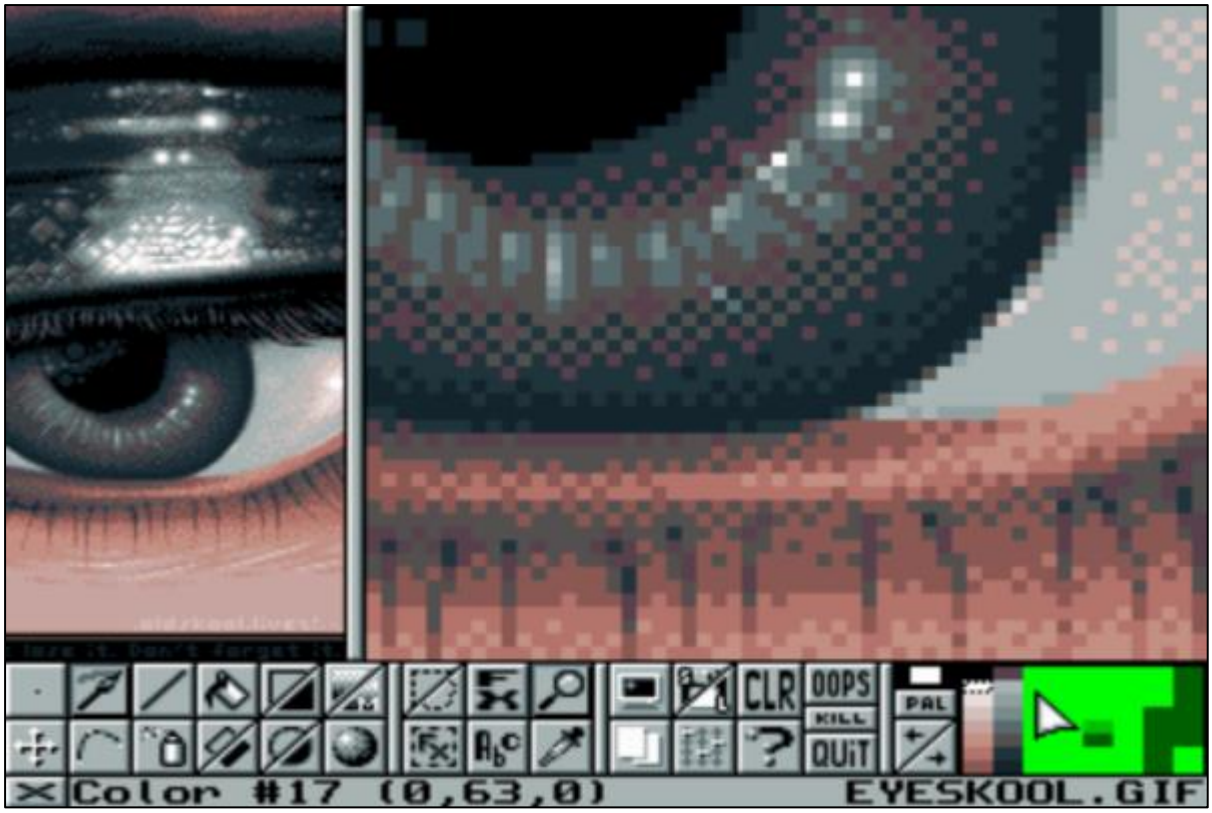
“grafx2.zip”

http://grafx2.chez.com/

https://clasqm.github.io/freedos-repo/Graphics.html

https://sourceforge.net/p/freedos/news/2022/04/grafx2-port-to-freedos-alpha-release/

https://www.ibiblio.org/pub/micro/pc-stuff/freedos/files/repositories/unstable/apps/



Install Instructions

GrafX2 can be manually downloaded from https://www.ibiblio.org/pub/micro/pc-stuff/freedos/files/repositories/unstable/apps/grafx2.zip

or installed using FDNPKG.

As this is the same and only grafx2 version available for FreeDOS I will use the Network Package Manager. Note that grafx2 is not found on the FreeDOS bonus CD.

Checking updates will update the repository cache.

```
FDNPKG checkupdates | more
```

You can also use the redirect to output the file version list to a text file if it is easier to review.

```
FDNPKG checkupdates >> C:\pkgver.txt
```

You can use this to see what applications have newer versions than what is found on the Bonus CD.

The following will also update the repository cache, and then check for the existence of grafx2 in the repository.

```
FDNPKG search GRAFX2
```

To check for the file location in the repository use:

```
FDNPKG vsearch GRAFX2
```

To install to the default FreeDOS location:

```
FDNPKG install GRAFX2
```

```
C:\> fdnpkg vsearch grafx2
Package database loaded from local cache.

grafx2 - A graphics editor
 -> ver 2.2 at http://www.ibiblio.org/pub/micro/pc-stuff/freedos/files/repositor
ies/latest/apps/ [8F14345A]

C:\>fdnpkg install grafx2
Package database loaded from local cache.
Downloading package http://www.ibiblio.org/pub/micro/pc-stuff/freedos/files/repo
sitories/latest/apps/grafx2.zip...
Downloading grafx2.zip... 1733082 bytes [68%]_
```

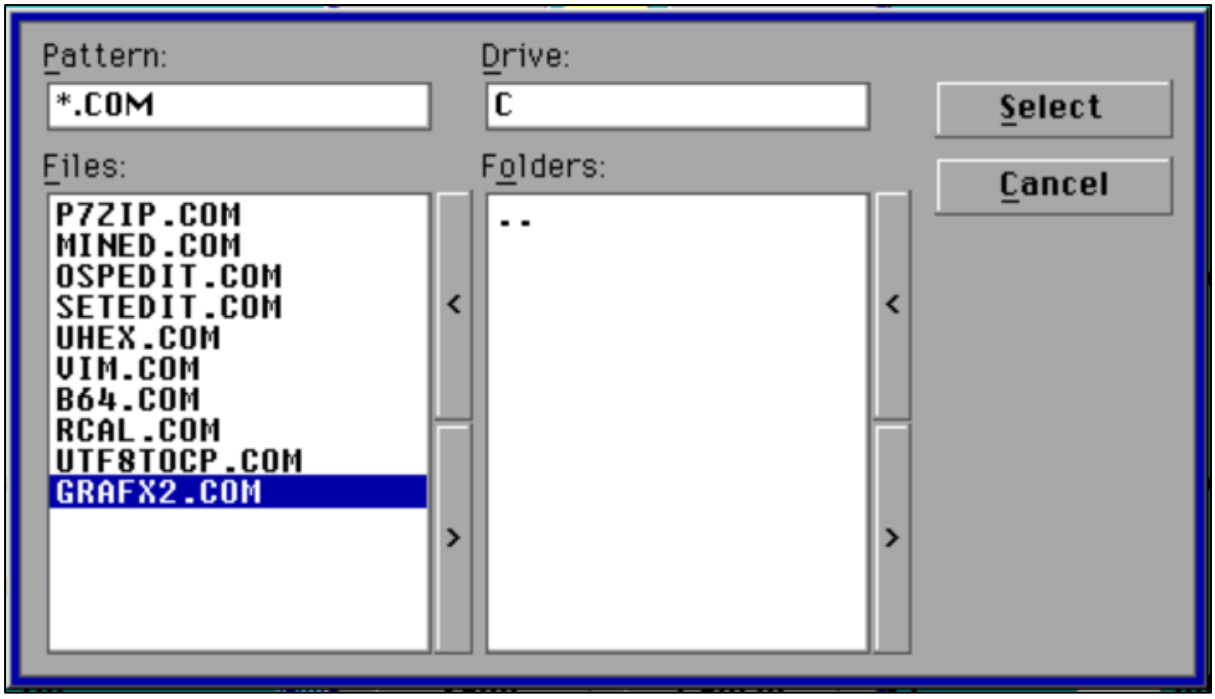
```
 apps\grafx2\share\grafx2\skins\font_mel.png -> c:\apps\grafx2\share\grafx2\skin
s\
 apps\grafx2\share\grafx2\skins\font_fun.png -> c:\apps\grafx2\share\grafx2\skin
s\
 apps\grafx2\doc\hotkeys.txt -> c:\apps\grafx2\doc\
 apps\grafx2\doc\readme.txt -> c:\apps\grafx2\doc\
 apps\grafx2\doc\compilin.txt -> c:\apps\grafx2\doc\
 apps\grafx2\doc\gpl-2_0.txt -> c:\apps\grafx2\doc\
 appinfo\grafx2.lsm -> C:\FreeDOS\appinfo\
Package grafx2 installed: 56 files extracted, 0 errors.
C:\>_
```

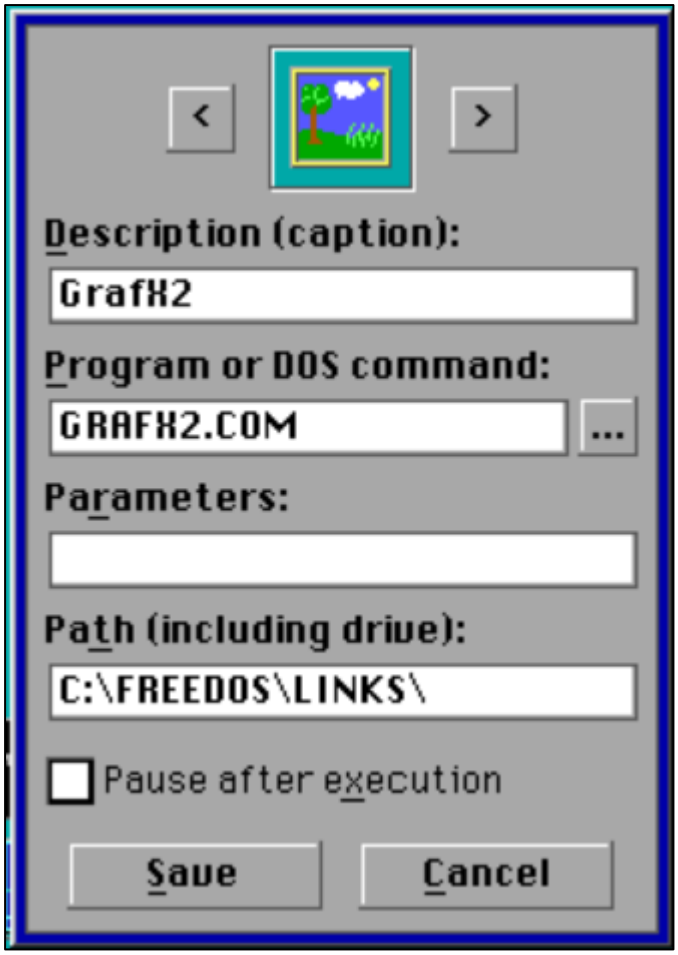
The GrafX2 package will now be found in the C:\APPS\ directory

```
File  Modify  Dir  Tag  Sort  disK  Link  Configure  Help
          Source (Local)                          Target (Local)
 *.*                                     *.*
 CDP          <dir>      10-26-23                 <tree>
 COSTA174     <dir>      11-01-23    $RECYCLE BIN <hdir>     11-01-23
 DN2          <dir>      10-26-23    APPS         <dir>      10-26-23
 DOSHEXED     <dir>      10-26-23    BKUP         <dir>      11-04-23
 DOSMID       <dir>      10-26-23    FREEDOS      <dir>      10-26-23
 E3           <dir>      10-26-23    GAMES        <dir>      10-26-23
 ELVIS        <dir>      10-26-23    NET          <dir>      10-26-23
 EMACS        <dir>      10-26-23    PGME         <dir>      10-26-23
 FED          <dir>      10-26-23    SB16         <dir>      11-04-23
 FM3          <dir>      10-30-23    SYSTEM~1     <hdir>     11-01-23
 GRAFX2       <dir>      11-05-23    TEST         <dir>      11-04-23
 MBEDIT       <dir>      10-26-23    UTIL         <dir>      10-26-23
 MINED        <dir>      10-26-23    COMMAND  COM      85480  7-10-21
 OPENCP       <dir>      10-26-23    FDAUTO   BAT       2060 11-04-23
 OSPEDIT      <dir>      10-26-23    FDCONFIG SYS        825 11-05-23
 PKZ204G      <dir>      10-28-23    KERNEL   SYS      46256  2-20-22
C:\APPS                                 C:\
21 dirs, 0 files: 0 bytes | 1,872,723,968 (87%) bytes free
+/-: Tag  Alt: Menu | Enter: Change to GRAFX2
```

Grafx2 will install a link command in `C:\FreeDOS\Links\GRAFX2.COM` so you can call it any time from the command line. GRAFX2.COM is somewhat similar to the Batch file launchers explained in "**Application launch BAT**".

When adding a Grafx2 icon to the Costa desktop, link to the above file rather than directly to GRAFX2.EXE in the C:\APPS\GRAFX2 directory.





To remove the package use:

```
FDNPKG remove GRAFX2
```

If you choose to do a manual install, copy the contents of grafx2.zip into `C:\APPS\GRAFX2\*.*` and create a batch file GRAFX2.BAT as shown in the section "**Application launch BAT**".

PictView

PictView is a DOS-based multi-format image viewer and converter, one of the fastest in the world. PictView supports more than 40 file formats in more than 140 versions. PictView is freeware, and you do not have to register to use it, however donations for the future development are very welcome.

http://www.pictview.com/

https://clasqm.github.io/freedos-repo/Graphics.html

“pictview.zip”

Additional applications...
The following 2 applications from the author can offer assistance when attempting to work out video adaptor mode and other problems.
“vesamod.zip”
A tool we originally used to track problems with our PictView. It detects various types of video adapters, scans VESA BIOS and saves video ROM into a file. The video mode information can be viewed by VESAInfo. We will be glad if you send us the generated file, it will help us in keeping PictView updated to reflect new trends in video adapters.
“vesainfo.zip”
Lists details about VESA BIOS modes stored in a file created by VESAMod.

Install Instructions

Unpack the downloaded archive “pictview.zip”.

Copy the pictview directory and contents to your DOS guest drive `C:\APPS\PICTVIEW`

Create a PICTVIEW.BAT file.

PICTVIEW.BAT

```
@ECHO OFF
CLS
REM PICTVIEW application launch.
CALL C:\APPS\PICTVIEW\PICTVIEW.EXE %1 %2 %3 %4 %5 %6 %7 %8 %9
REM
REM Alternative; change working directory to \PICTVIEW
REM CD \APPS\PICTVIEW
REM PICTVIEW. EXE %1 %2 %3 %4 %5 %6 %7 %8 %9
```

Move the `PICTVIEV.BAT` to `C:\FREEDOS\LINKS`

Create a link/icon in Costa.

To uninstall PictView just delete the \PICTVIEW directory and associated batch file.

Other useful tools

Below is a list of other tools that may be useful in a basic DOS install. In most cases just unpack the archive and place the application into somewhere convenient in your DOS install such as `C:\APPS\*.*` and create a batch file to launch the application.

DOSPDF

You can find DOSPDF in the repository using the package manager (FDNPKG).

https://www.ibiblio.org/pub/micro/pc-stuff/freedos/files/util/file/dospdf/

VBEDIAG

https://drv.nu/vbediag/

vbediag is a diagnostic tool for the VESA Video BIOS Extensions. vbediag is a DOS (DPMI) program compiled with DJGPP. It displays information about your graphics adapter and the modes it claims to support. This tool is designed to help diagnose problems with FreeBASIC's graphics library on DOS. All output is printed to standard output, so run the program with stdout redirected: vbediag > output.txt

You can add an output to the batch file with `CALL VBEDIAG.EXE >> [Drive][Directory]BVEOUT.TXT`

Alternatively use `VBEDIAG.EXE | MORE` to view the console output one screen at a time.

Video Adaptor Test – Aaron Johnson

386 video adapter ID and test utility.

https://www.elhvb.com/webhq/download/index.htm

https://www.os2site.com/sw/dos/util/video/index.html

“videot.zip”

This one can be a bit tricky to find, so if the above sites fail search for “videot.zip” [+ DOS] and other links/archives may come up.

MONITORS

Tests video cards for characteristics and graphic modes. This excellent 486 tool tests both monitor & adapter.

https://www.elhvb.com/webhq/download/index.htm

https://www.os2site.com/sw/dos/util/video/index.html

“monitors.zip”

SB MIXER

Included with the DOS SB16 Drivers.

“MIXERSET.EXE”

NSSI

Navrátil Software System Information. Version 0.60.45 released 2010-09-09

Professional freeware system information tool, a flagship of Navrátil Software's products.

https://www.navsoft.cz/products.htm

I have only tested version 0.59.16

“nssi059.exe” self-extracting archive.

Info Plus

INFOPLUS Version 1.45, by Andrew Rossmann, February 11, 1991

THIS IS A PROGRAM THAT CAN TELL ALL THE CONFIGURATION OF YOUR COMPUTER.

https://www.bttr-software.de/freesoft/system.htm

https://www.uselesssoftware.com/download/infoplus-zip

“INFOPLUS.ZIP”

HWInfo16

Exhausting information about hardware components displayed in hierarchy unfolding into deep details.

Useful for obtaining a detailed hardware inventory report or checking of various hardware-related parameters.

https://www.hwinfo.com/download/

Look for the DOS Portable version.

“HWINF622.ZIP”

Display Doctor DOS

This is an older DOS version of Display doctor for Windows 9x.

Only use this if you have a specific need for advanced VGA drivers for DOS.

I have used the Windows 95 version in VirtualBox/VirtualPC for a long time with great success but have not yet tested the DOS version.

http://web.archive.org/web/20000302035130/http://www.scitechsoft.com/down_dos.html

“sdd653-d.exe”

LISTVESA FreeDOS

ListVESA is a utility to report which VESA video modes are supported by the system's hardware.

Found on the Bonus CD FDIMPLES or via the online Repo FDPKG.

P7ZIP FreeDOS

P7ZIP is a quick port of 7za.exe (command line version of 7zip). 7-Zip is a file archiver with the highest compression ratios.

Found on the Bonus CD FDIMPLES or via the online Repo FDPKG.

Dillo

Note:
I will cover more on Dillo in the section “HFS over IP” for host client file transfers.

A web browser known for its speed and small footprint. Dillo is a graphical browser which I have found quite usable (with patience) in FreeDOS guest on VirtualBox. It is a tool I sometimes use in conjunction with Rejetto HFS for transferring files between the host and guest DOS OS using VirtualBox.

https://www.ibiblio.org/pub/micro/pc-stuff/freedos/files/repositories/1.2/pkg-html/dillo.html

Dillo can be installed from the FreeDOS bonus CD using FDIMPLES or downloaded/upgraded using the network package manger FDNPKG.

You can check which packages are out of date by FDNPKG `checkupdates | more`

To update all currently installed FreeDOS packages FDNPKG `update`

HFS over IP

For file transfers over HTTP TCP/IP you will need the Dillo graphical web browser for DOS and the HTTP Files Server (HFS) for Windows. I don't have a current file server solution for Linux.

Dillo is found on the FreeDOS bonus CD and can be installed with FDIMPLES.

HFS can be downloaded from the applications home site:

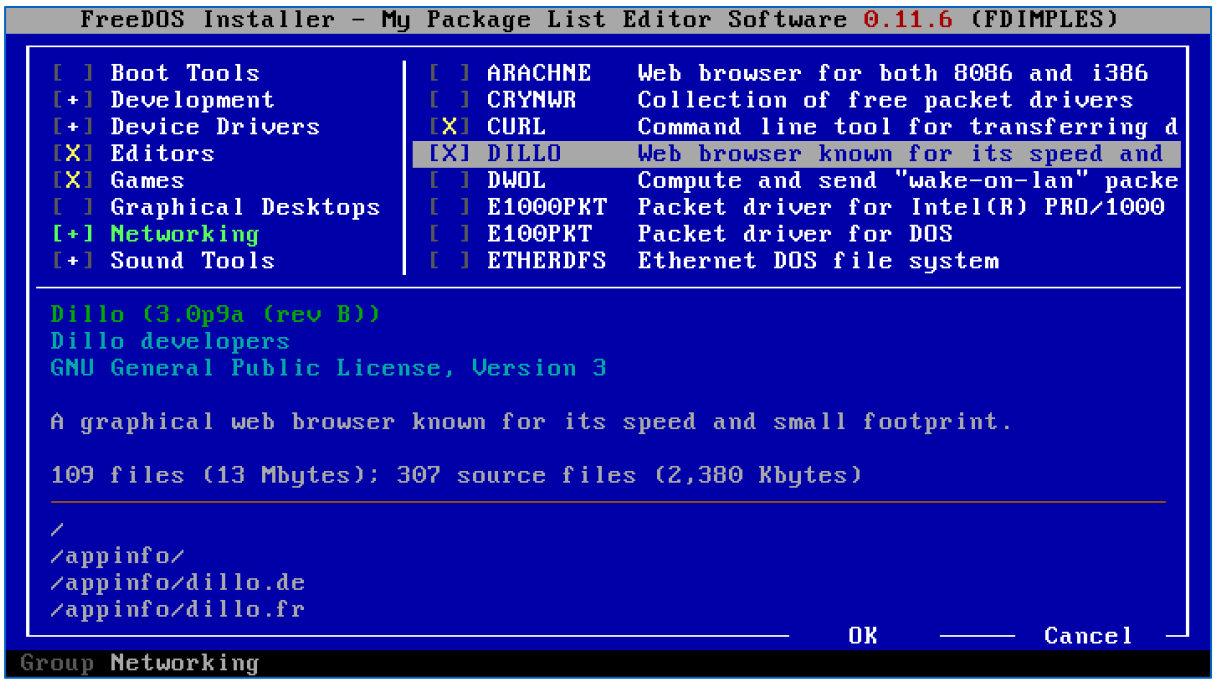
https://www.rejetto.com/hfs/?f=dl

"hfs.exe" Version 2.3m

First install Dillo in your current DOS OS.

Mount the FD13BNS.ISO and run FDIMPLES.

Navigate to the "Networking" section and then select [X] DILLO and OK to install.



You can also install or update Dillo from the FreeDOS repository using FDNPKG

```
FDNPKG search Dillo

FDNPKG install dillo
```

Or if already installed with FDIMPLES from the FD Bonus CD

```
FDNPKG update dillo
```

Navigate to the C:\NET\DILLO directory and make a backup copy of DILLO.BAT

Open C:\NET\DILLO\DILLO.BAT with EDIT or FED.

Alter the environment variables path SELF to reflect the DILLO and DILLO.BAT directory.

```
set SELF=C:\DILLO\DILLO.BAT

set SELF=C:\NET\DILLO\DILLO.BAT
```

DILLO.BAT (DEFAULT)

```
@echo off
REM Customize these lines for your system:
REM ------------------------------------
REM set DILLO=%DOSDIR%\DILLO

:Search
vfdutil /f %0 | set /p SELF=
if "%SELF%" == "" goto Search
if exist %SELF% goto Found
if exist %SELF%.BAT goto Found
set SELF=C:\DILLO\DILLO.BAT
:Found
vfdutil /p %SELF% | set /p DILLO=
if "%DILLO%" == "" goto Found
set SELF=

set NANOSCR=800 600 565

REM ----------------------------------------
REM You shouldn't need to edit below this line.
REM ----------------------------------------

set WATTCP.CFG=%DILLO%\ETC
set ETC=%DILLO%\ETC
set STDOUT=%DILLO%\STDOUT.TXT
set HOME=%DILLO%

%DILLO%\bin\redir.exe -o %STDOUT% -eo -t -x %DILLO%\bin\dillo.exe %1 %2 %3 %4 %5
%6 %7 %8 %9
%DILLO%\bin\vcls.exe
```

DILLO.BAT (Fixed)

```
@echo off
REM Customize these lines for your system:
REM ------------------------------------
REM set DILLO=%DOSDIR%\DILLO

:Search
```

```
vfdutil /f %0 | set /p SELF=
if "%SELF%" == "" goto Search
if exist %SELF% goto Found
if exist %SELF%.BAT goto Found
set SELF=C:\NET\DILLO\DILLO.BAT
:Found
vfdutil /p %SELF% | set /p DILLO=
if "%DILLO%" == "" goto Found
set SELF=

set NANOSCR=800 600 565

REM ------------------------------------------
REM You shouldn't need to edit below this line.
REM ------------------------------------------

set WATTCP.CFG=%DILLO%\ETC
set ETC=%DILLO%\ETC
set STDOUT=%DILLO%\STDOUT.TXT
set HOME=%DILLO%

%DILLO%\bin\redir.exe -o %STDOUT% -eo -t -x %DILLO%\bin\dillo.exe %1 %2 %3 %4 %5
%6 %7 %8 %9
%DILLO%\bin\vcls.exe
```

Save and close.

Note: The first DILLO.BAT needs to remain in the original \DILLO directory to work correctly.

Create a new batch file DILLOW.BAT in C:\FREEDOS\LINKS\DILLOW.BAT

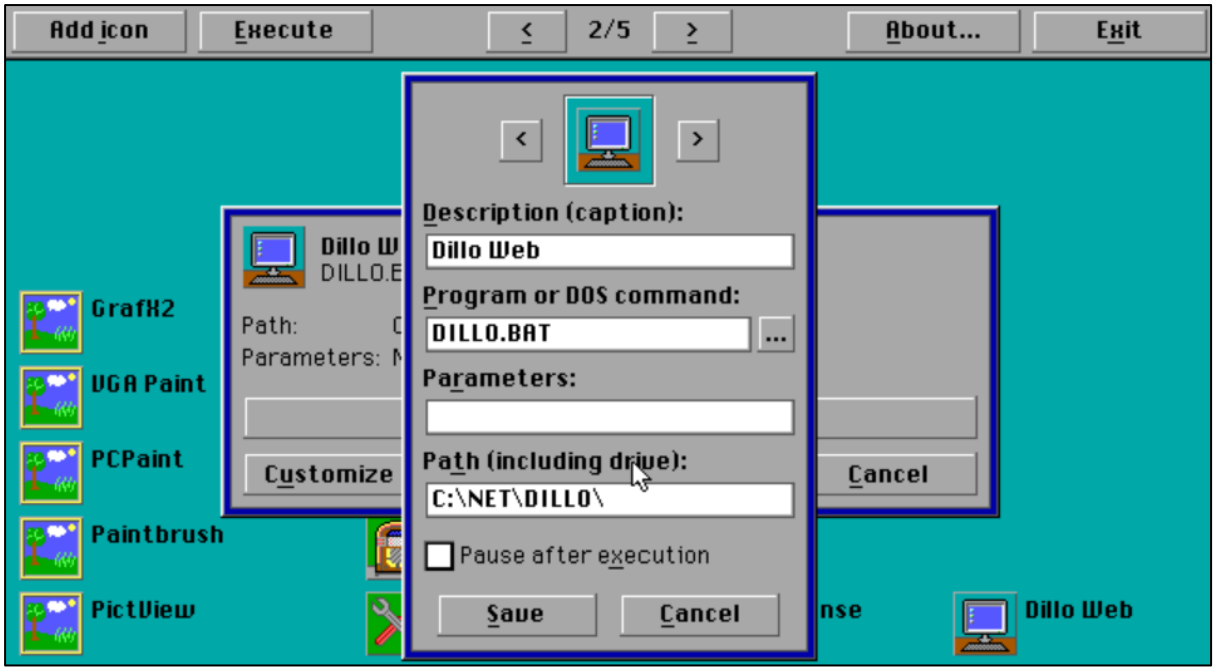
Add the following lines and save the batch file. This is our quick launch file for the command line.

DILLOW.BAT

```
@ECHO OFF
CD \NET\DILLO
CALL DILLO.BAT
```

You can now launch Dillo from any directory with DILLOW or from the Dillo directory with DILLO.BAT.

If you wish to add Dillo to your Costa desktop then I would suggest linking directly to the C:\NET\DILLO\DILLO.BAT file.



Give it a test and then close the browser when finished.

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.

If you are already running a web server on port 80 you can change the port number in HFS settings.

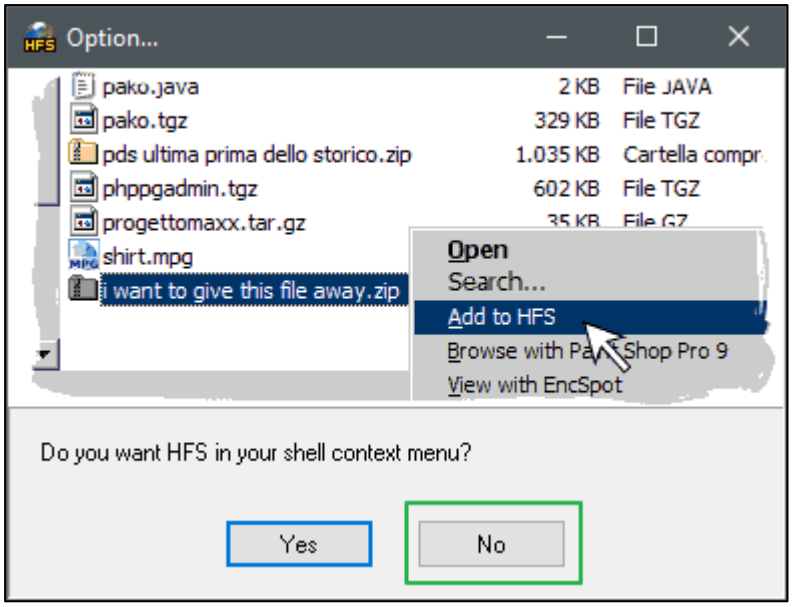
I typically have a dedicated directory for my portable applications aka applications that don't require an installer. Usually something like `C:\Portable\Appname` or `C:\Potable_Apps\Appname`

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.

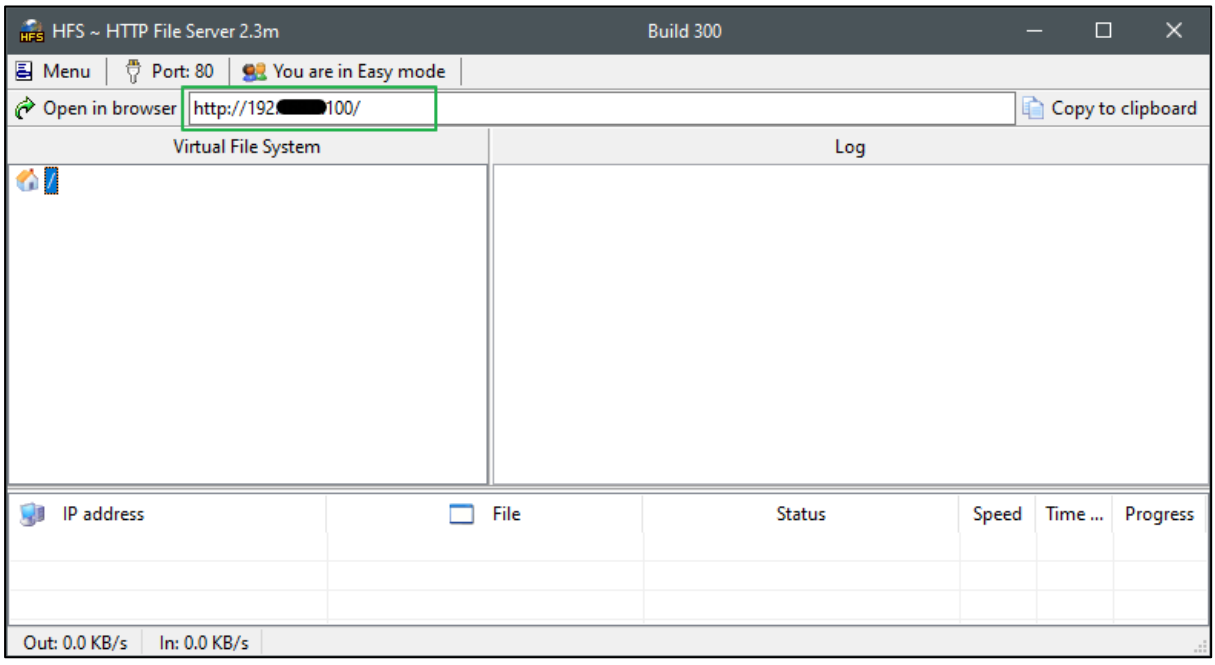
Start HFS.exe

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.

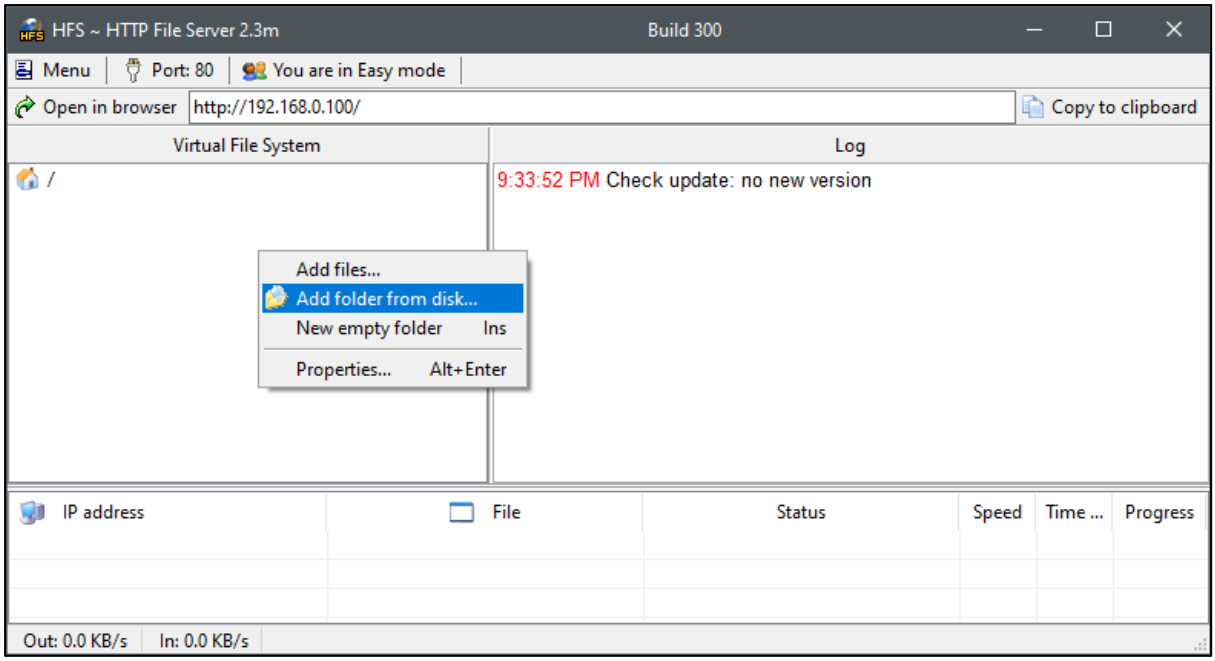


Next you will be met with the server control and configuration window.

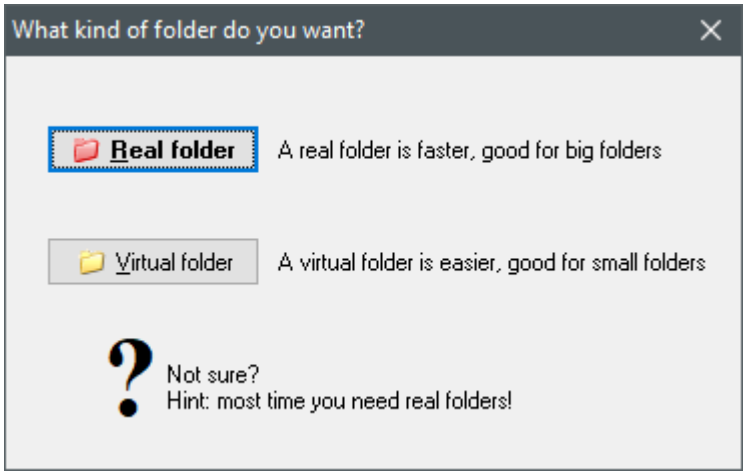
Take note of the server IP address on the network.

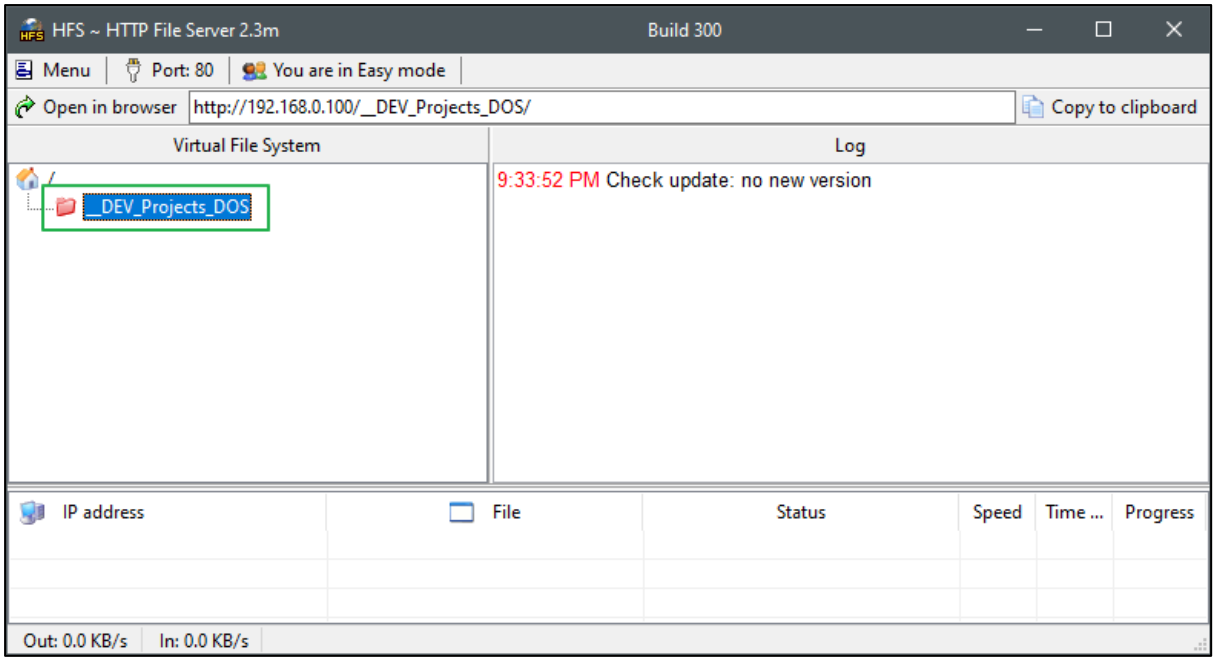


Next right click on the panel next to the home icon and select a home directory for the files you wish to make available for HTTP sharing. This is just the root directory and you can add as many sub directories and files as you need at any time later.

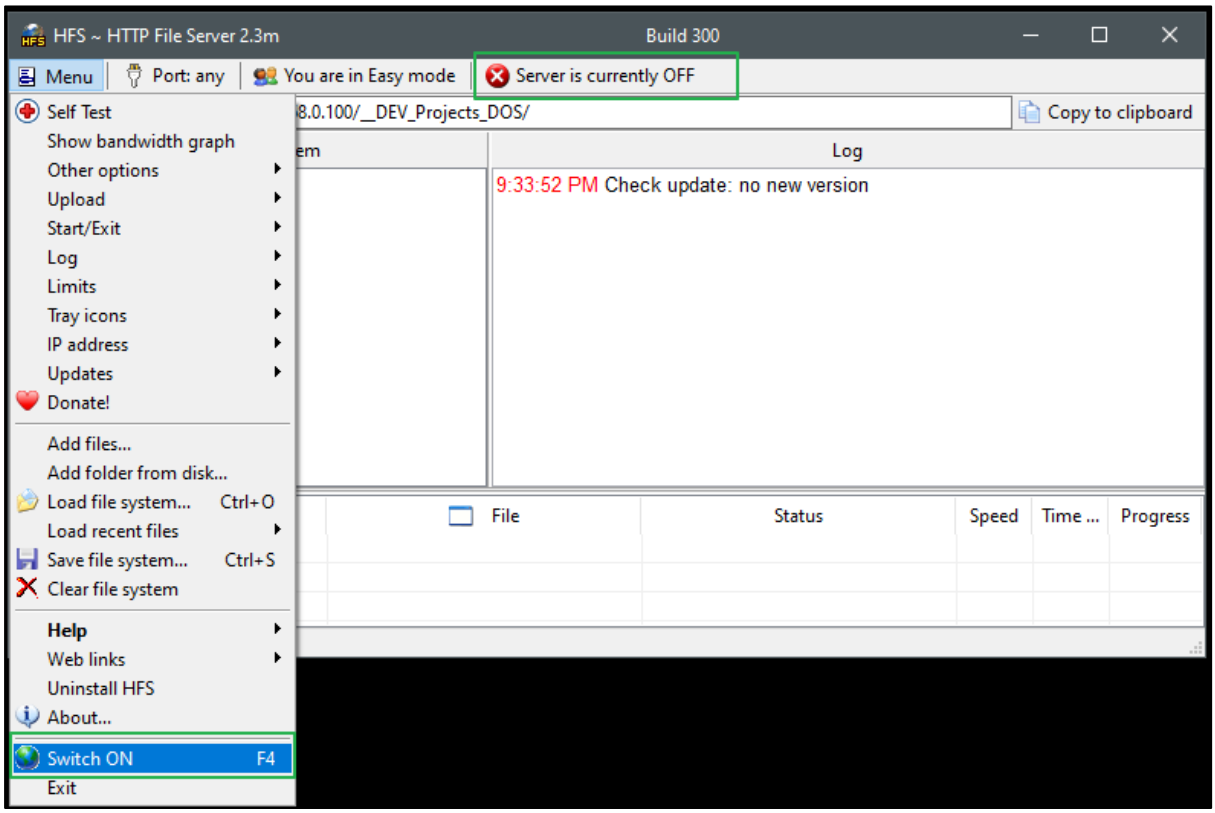


I typically use a real folder.



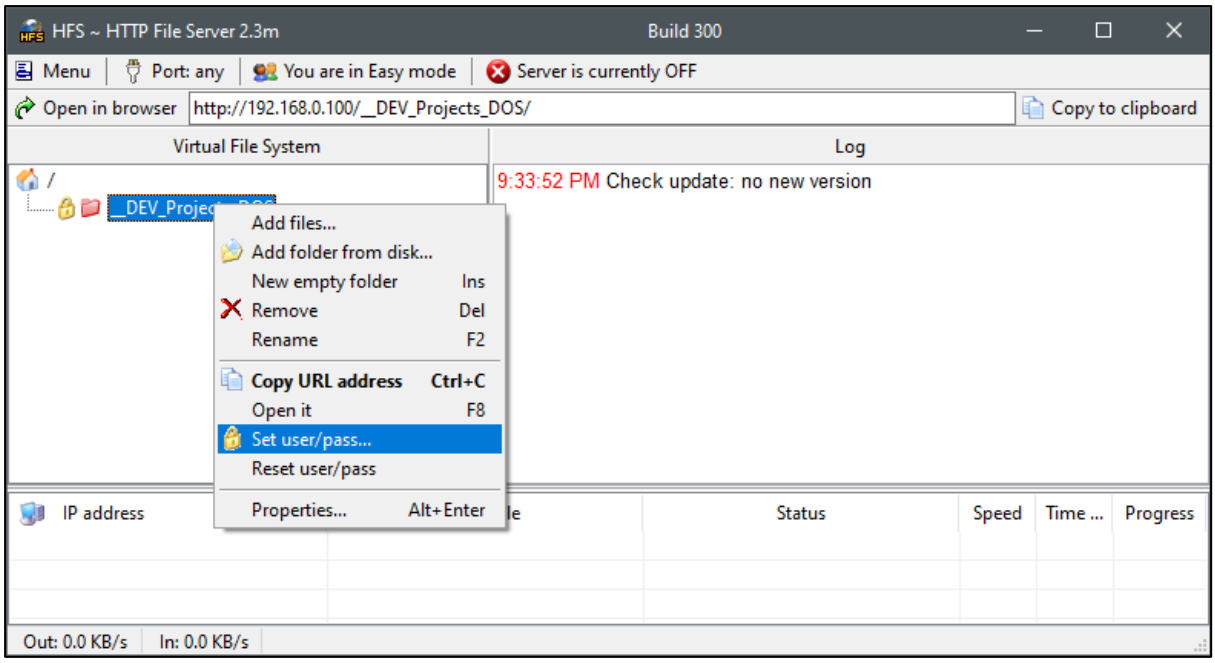


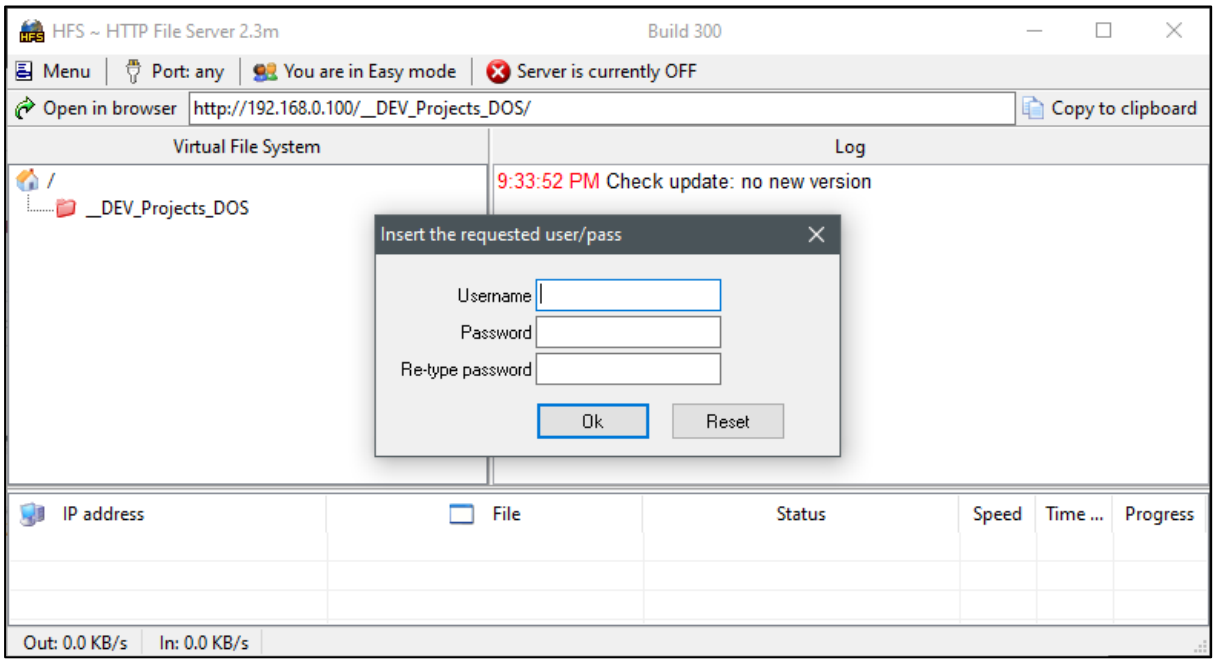
At this point any file or directory that is inside your selected directory will be available when you turn on the file server. The server is ON by default when first started. I suggest keeping it turned off unless you need to access it from another computer (FreeDOS client in VirtualBox) in your local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



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

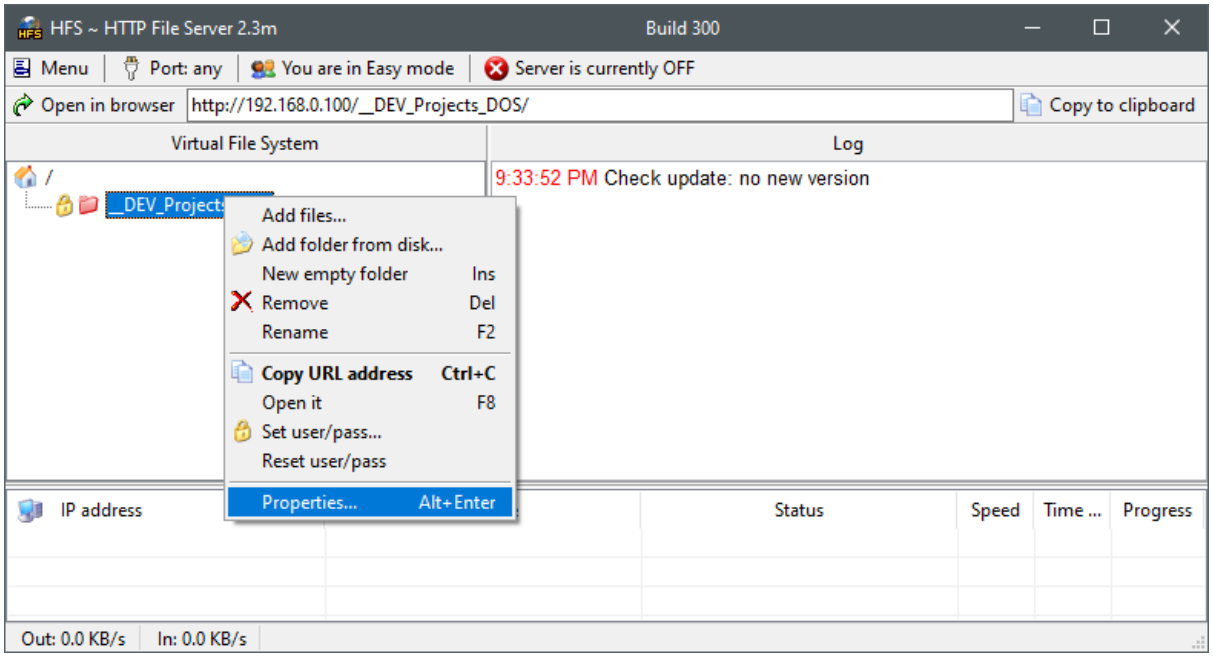
Right click on the root share directory and select “Set user/pass...”





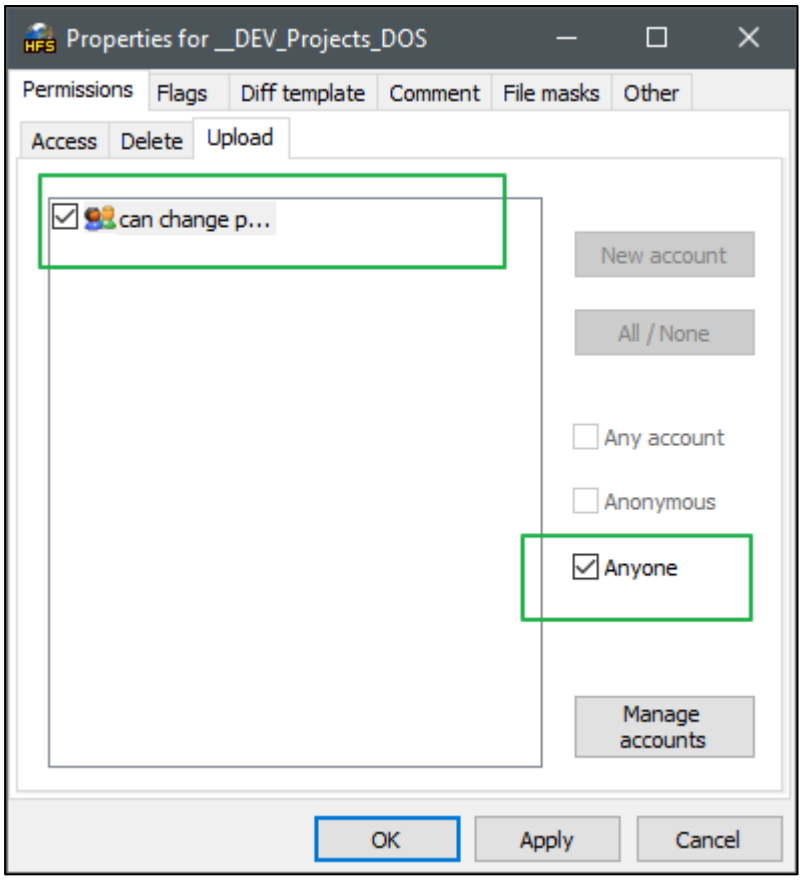
The lock will then show beside the file share directory.

Next become familiar with the directory and user share properties.

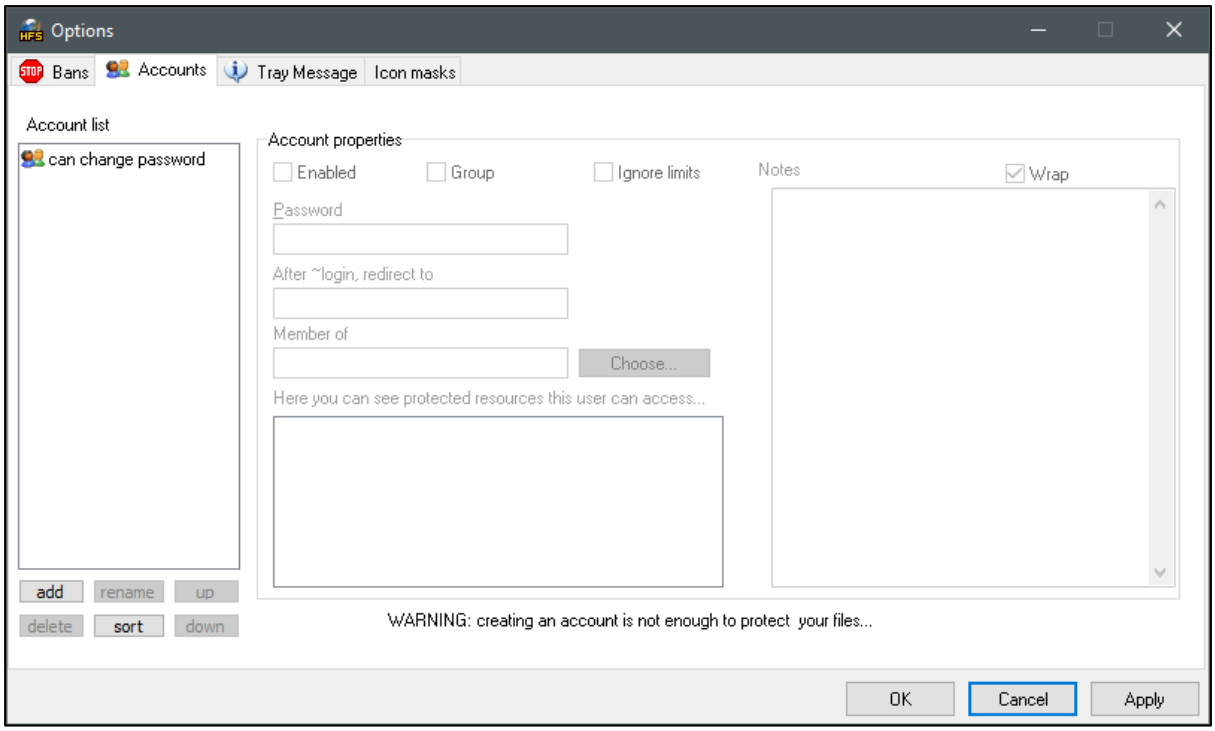


This is the root directory or Admin Account. You can add users and groups if you wish, but it is not required for simple file transfers. Become familiar with these setting just the same and also have a look at the “Manage accounts” button. These are the main server account and permission controls.

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

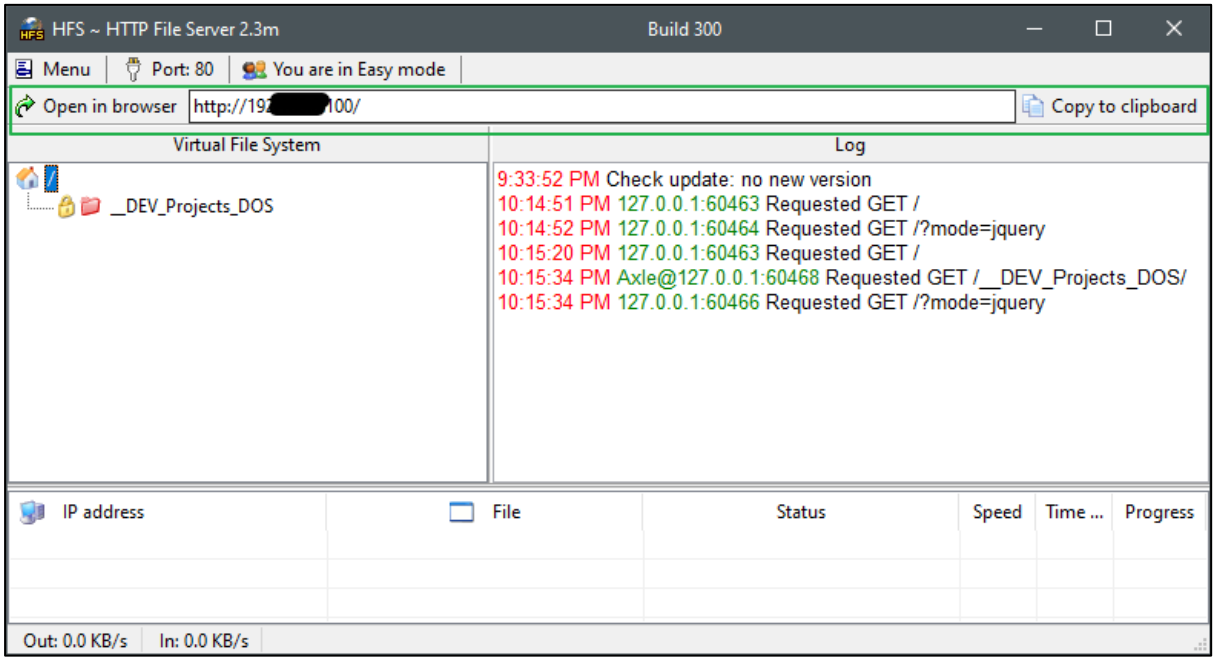


You can add Users and groups here later if you like.

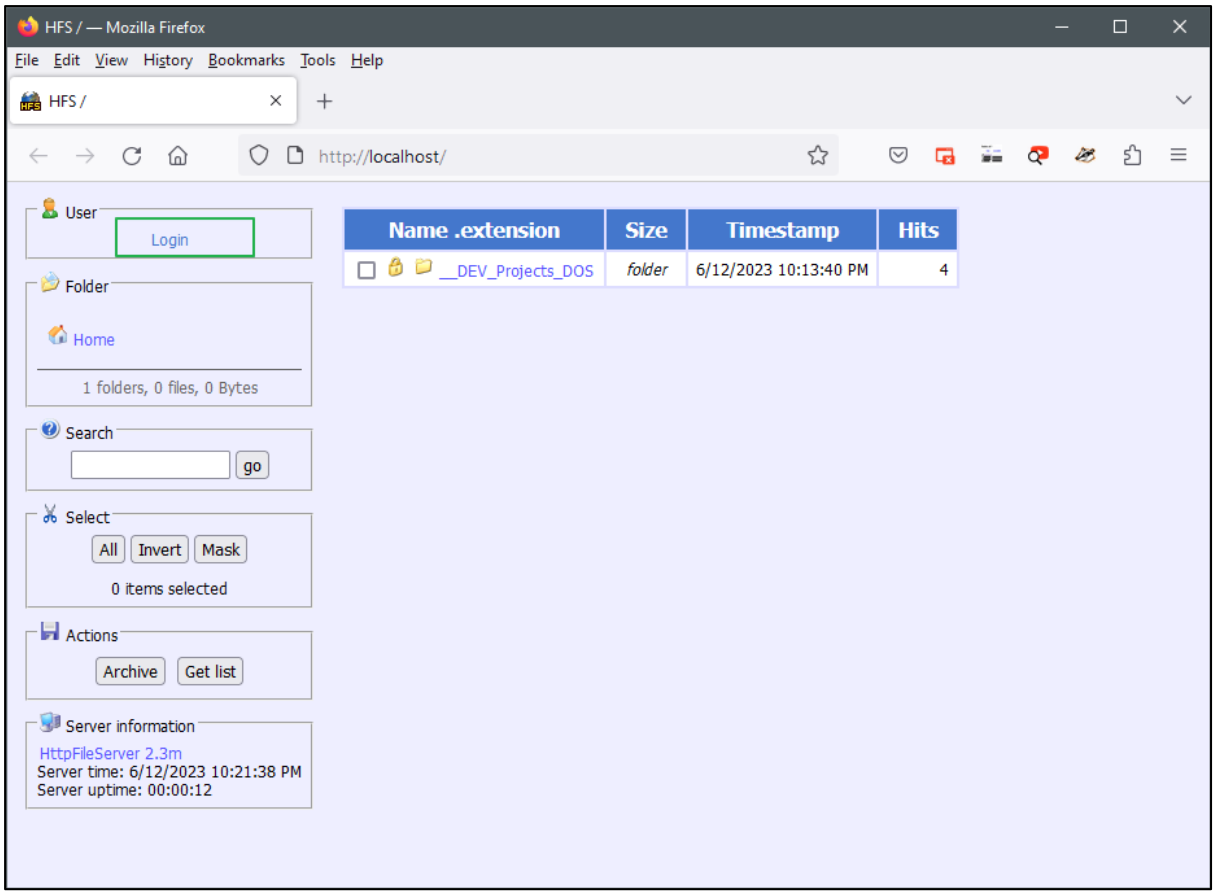


Open the location of your server root directory and add a sub directory and a test text file.

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. If your server is using a different port to 80 then add the correct port number to the endo of the IP address 127.0.0.1:8080

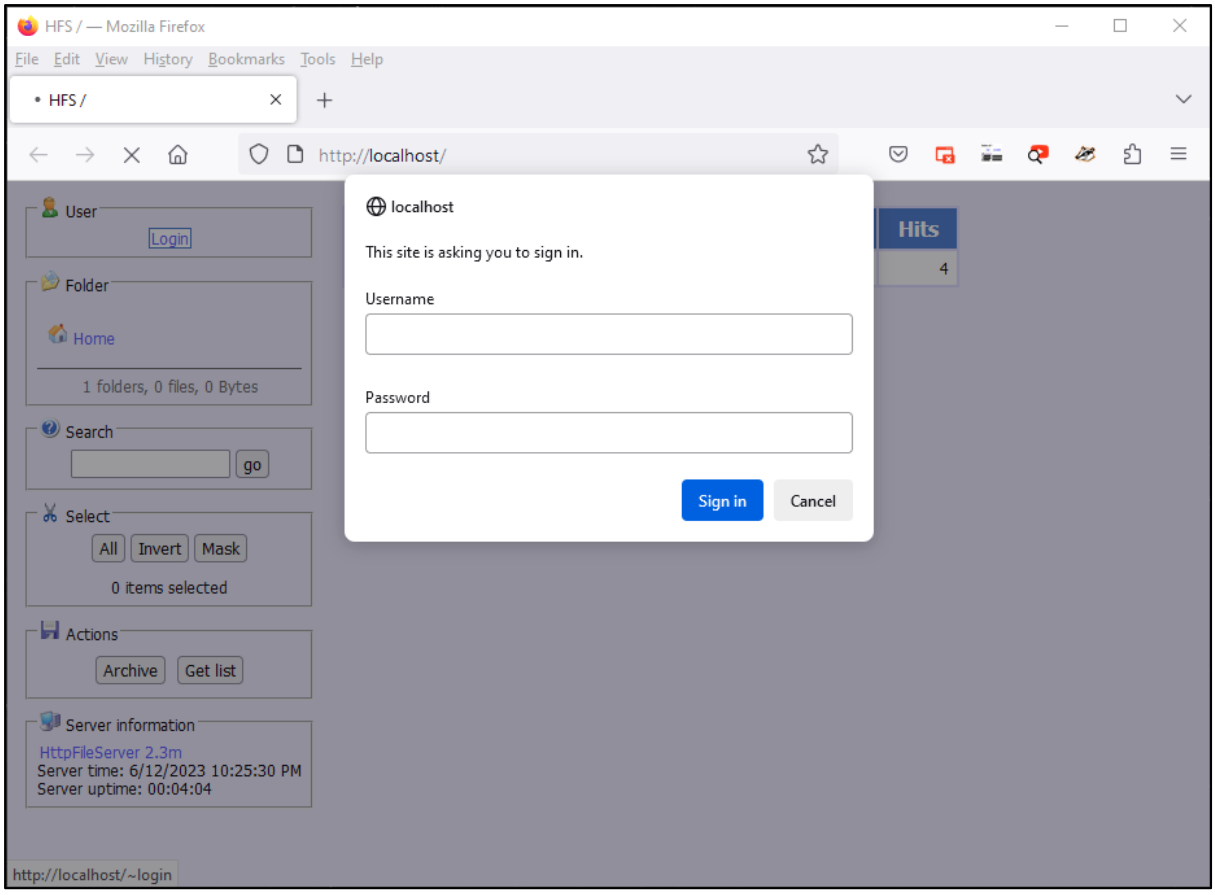


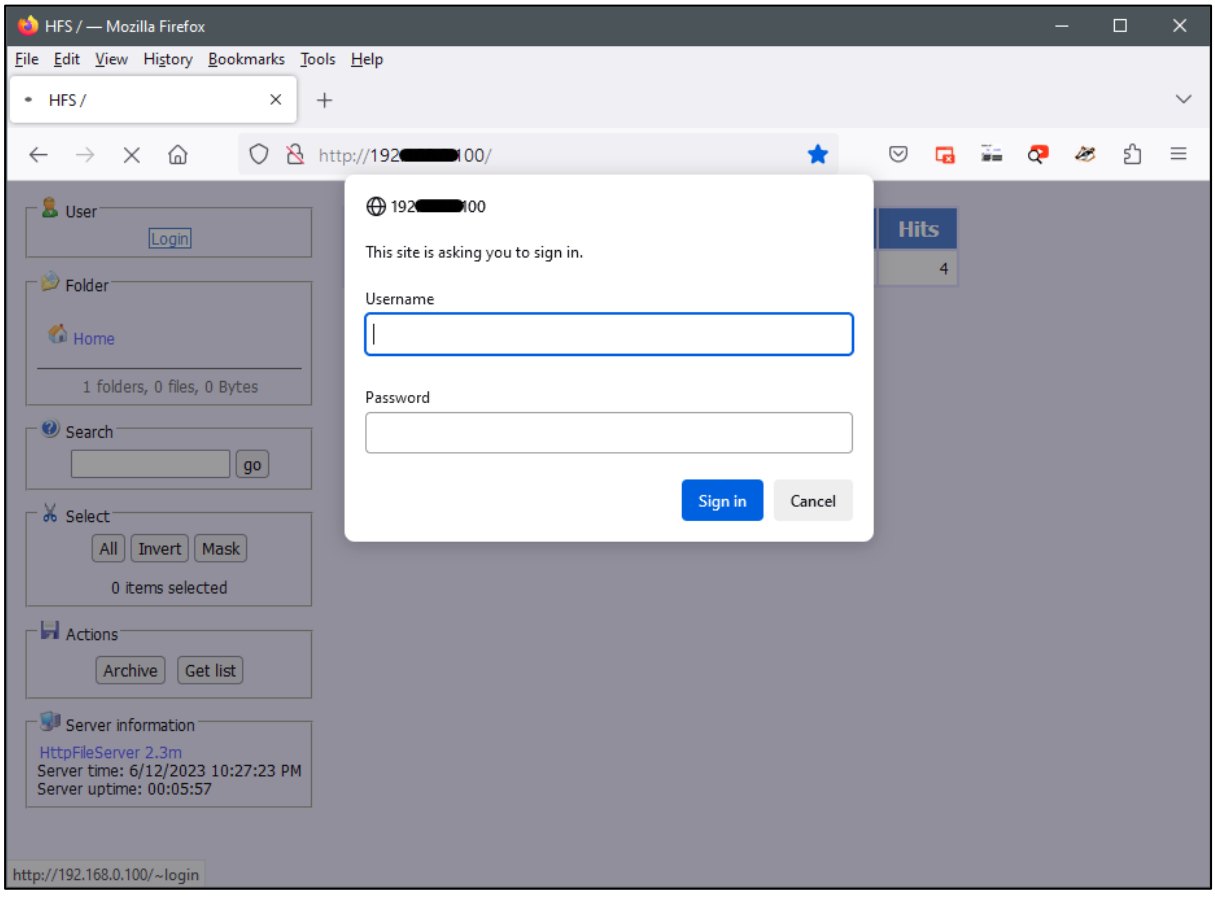
Click on the login link.

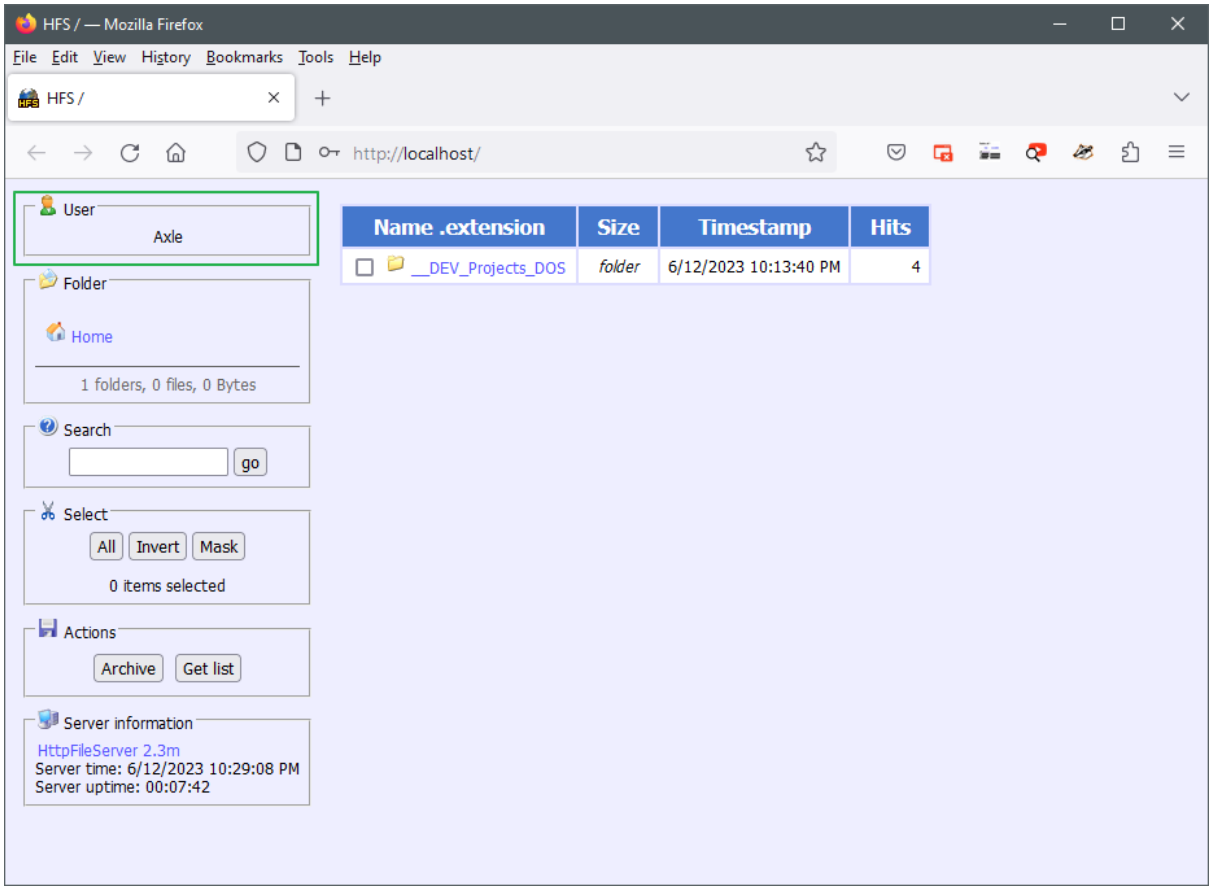


If the login does not show, close the browser and then reopen the link form the server.

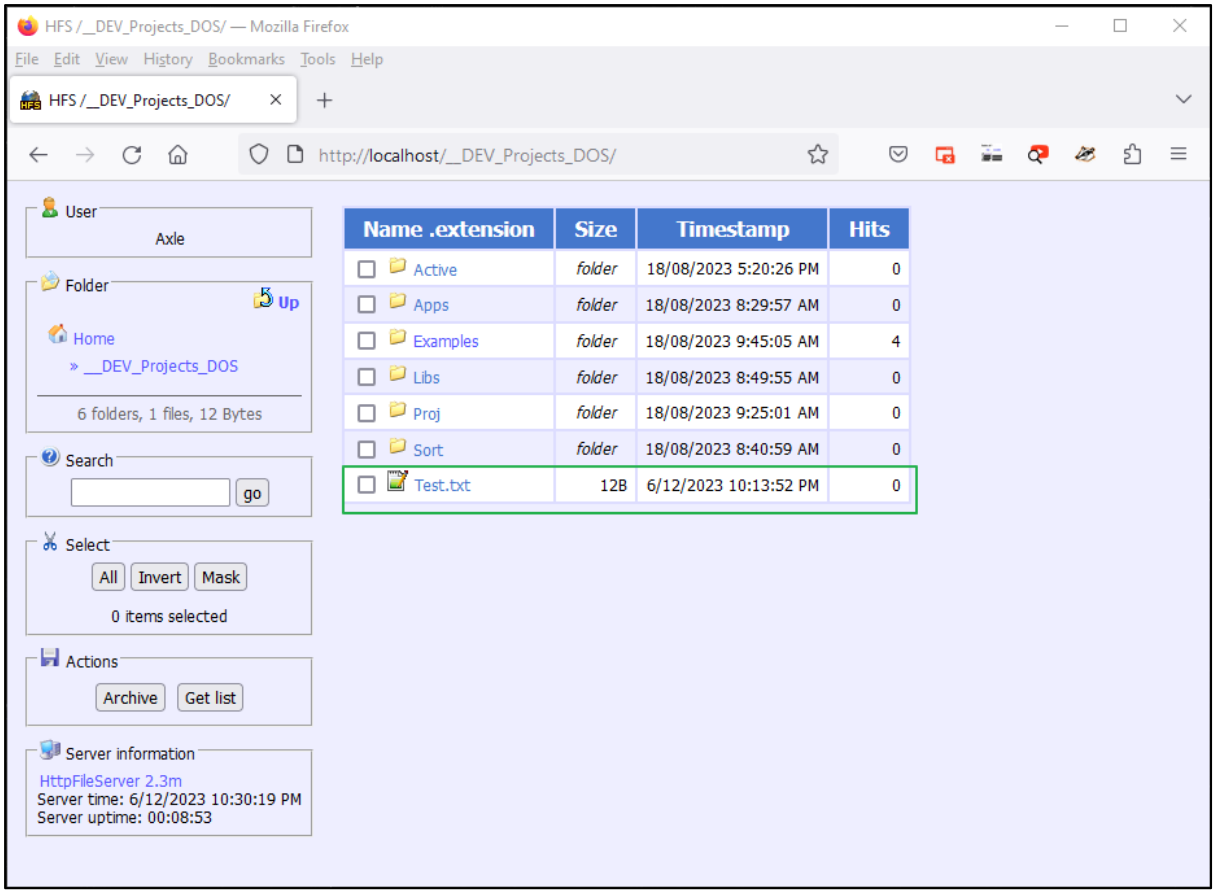
Enter the useer name and password you created earlier.


HFS / — Mozilla Firefox
File Edit View History Bookmarks Tools Help
• HFS /
http://localhost/
User
Login
Folder
Home
1 folders, 0 files, 0 Bytes
Search
go
Select
All Invert Mask
0 items selected
Actions
Archive Get list
Server information
HttpFileServer 2.3m
Server time: 6/12/2023 10:25:30 PM
Server uptime: 00:04:04
Hits
4
localhost
This site is asking you to sign in.
Username
Password
Sign in
Cancel
http://localhost/~login


HFS / — Mozilla Firefox
File Edit View History Bookmarks Tools Help
• HFS /
http://192[redacted]00/
User
Login
Folder
Home
1 folders, 0 files, 0 Bytes
Search
go
Select
All Invert Mask
0 items selected
Actions
Archive Get list
Server information
HttpFileServer 2.3m
Server time: 6/12/2023 10:27:23 PM
Server uptime: 00:05:57
Hits
4
192[redacted]00
This site is asking you to sign in.
Username
Password
Sign in
Cancel
http://192.168.0.100/~login

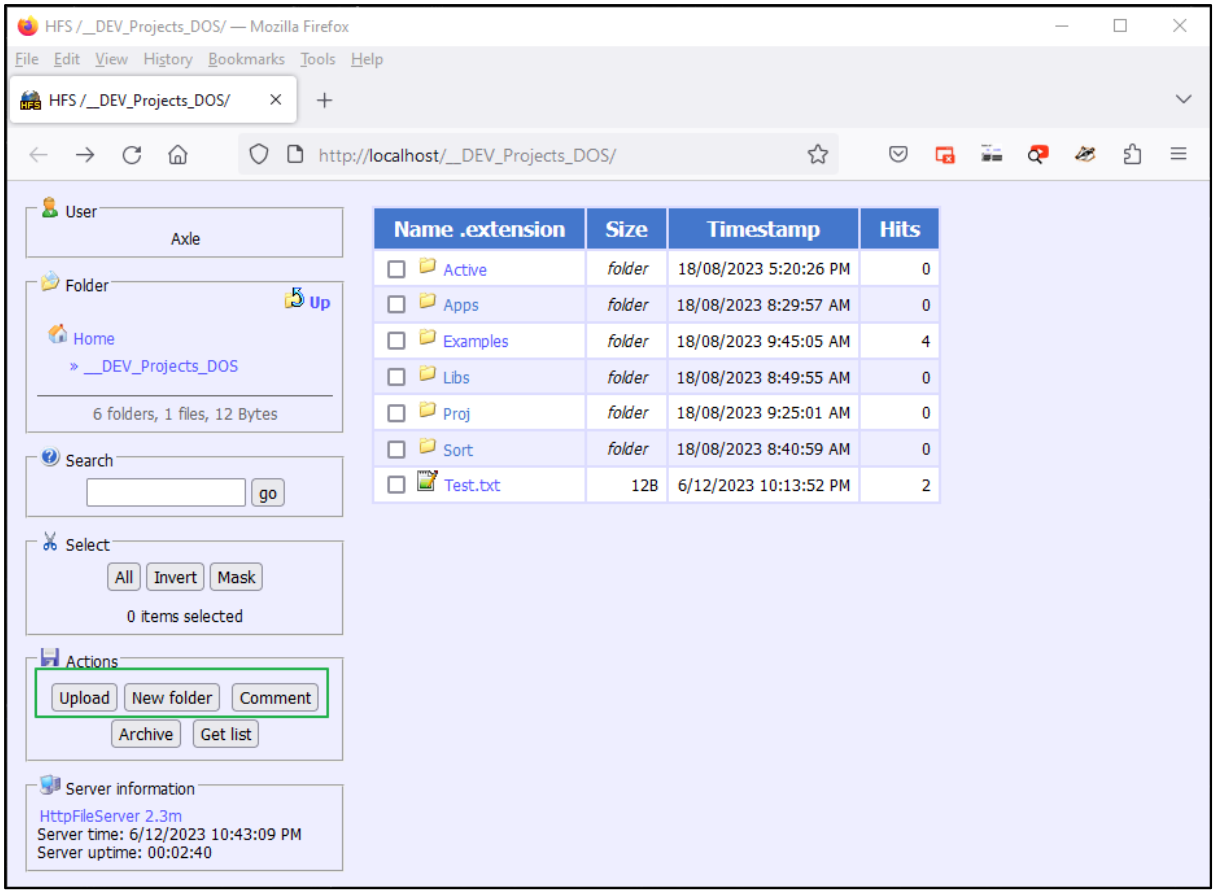


Click on the root directory listed in the browser to view sub directories and files.

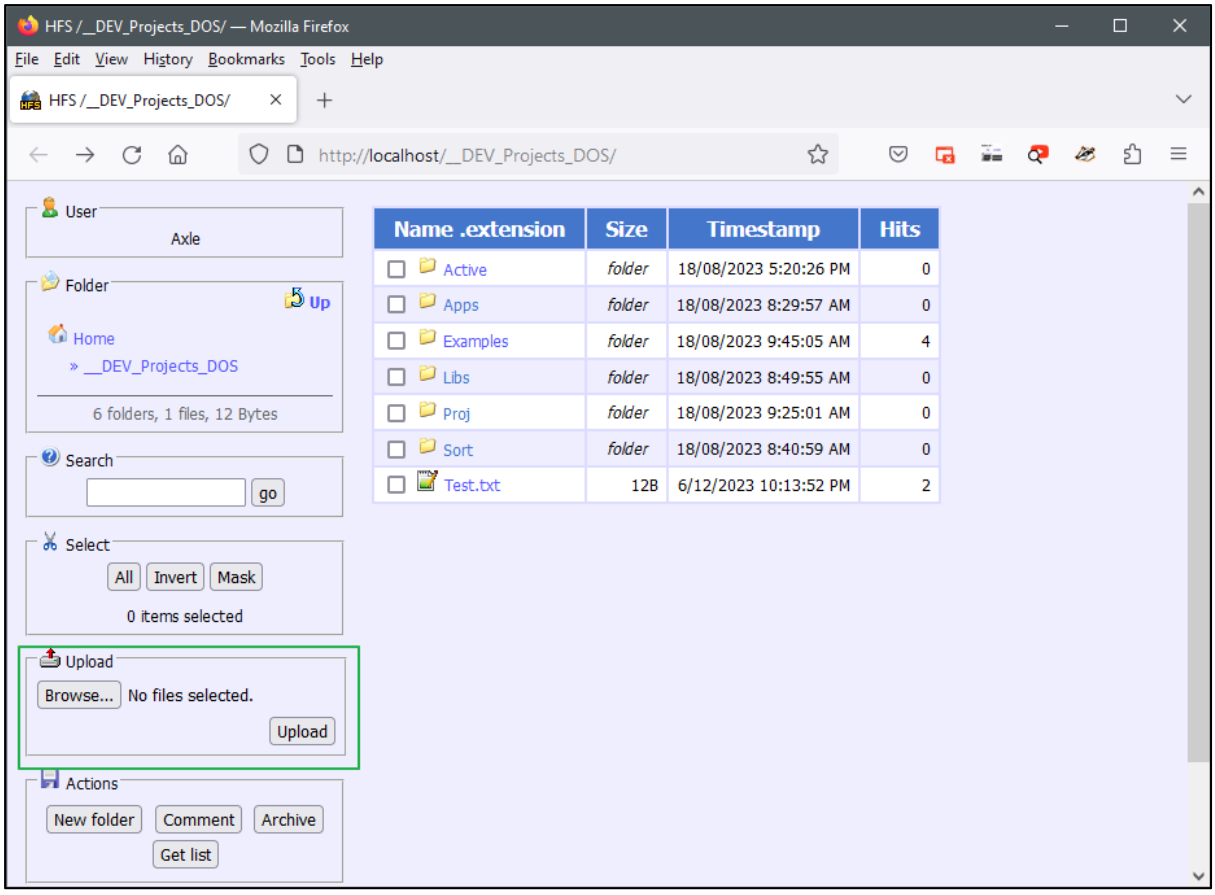


You should now be able to view and download your test text file form the server.

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



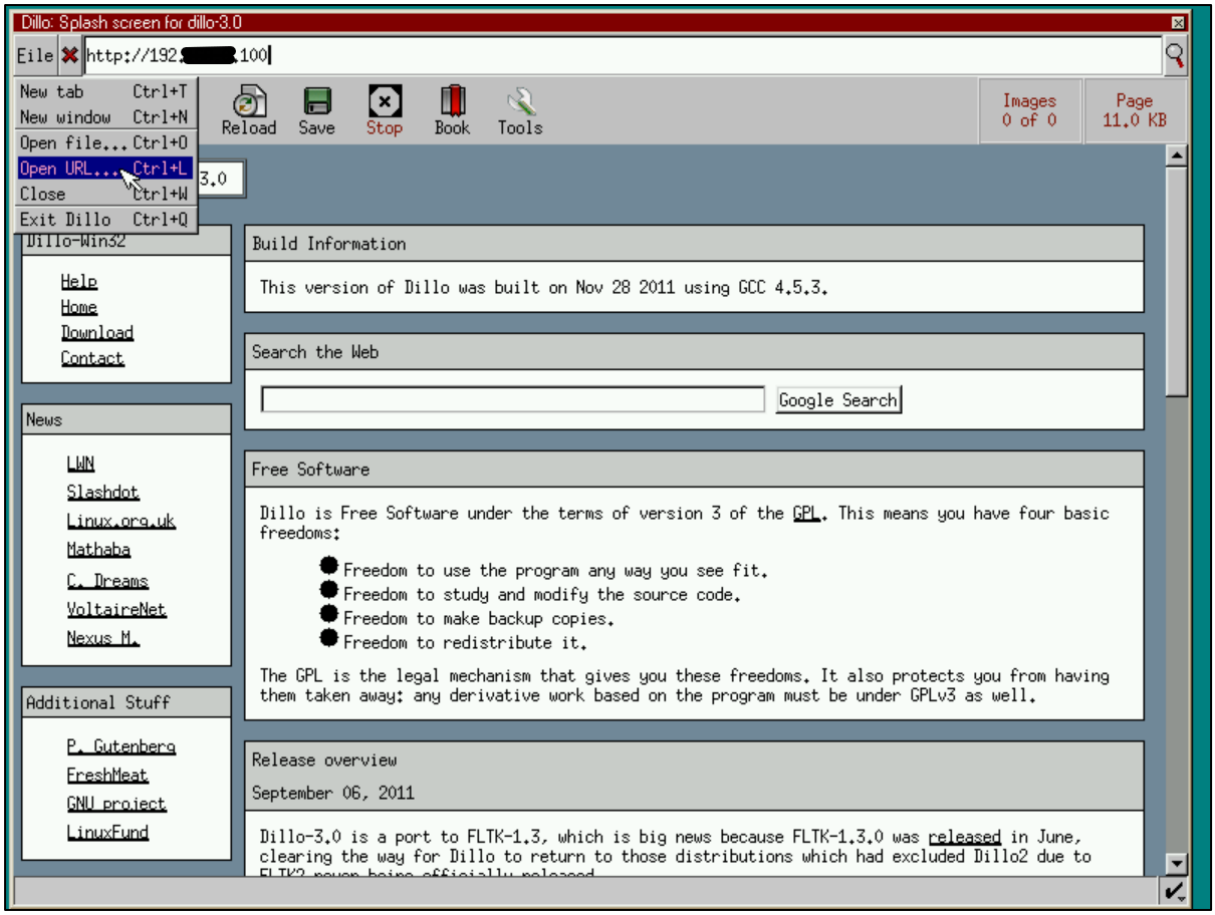
Upload will launch the file browser on your local computer and you can select the file to upload to the file server.



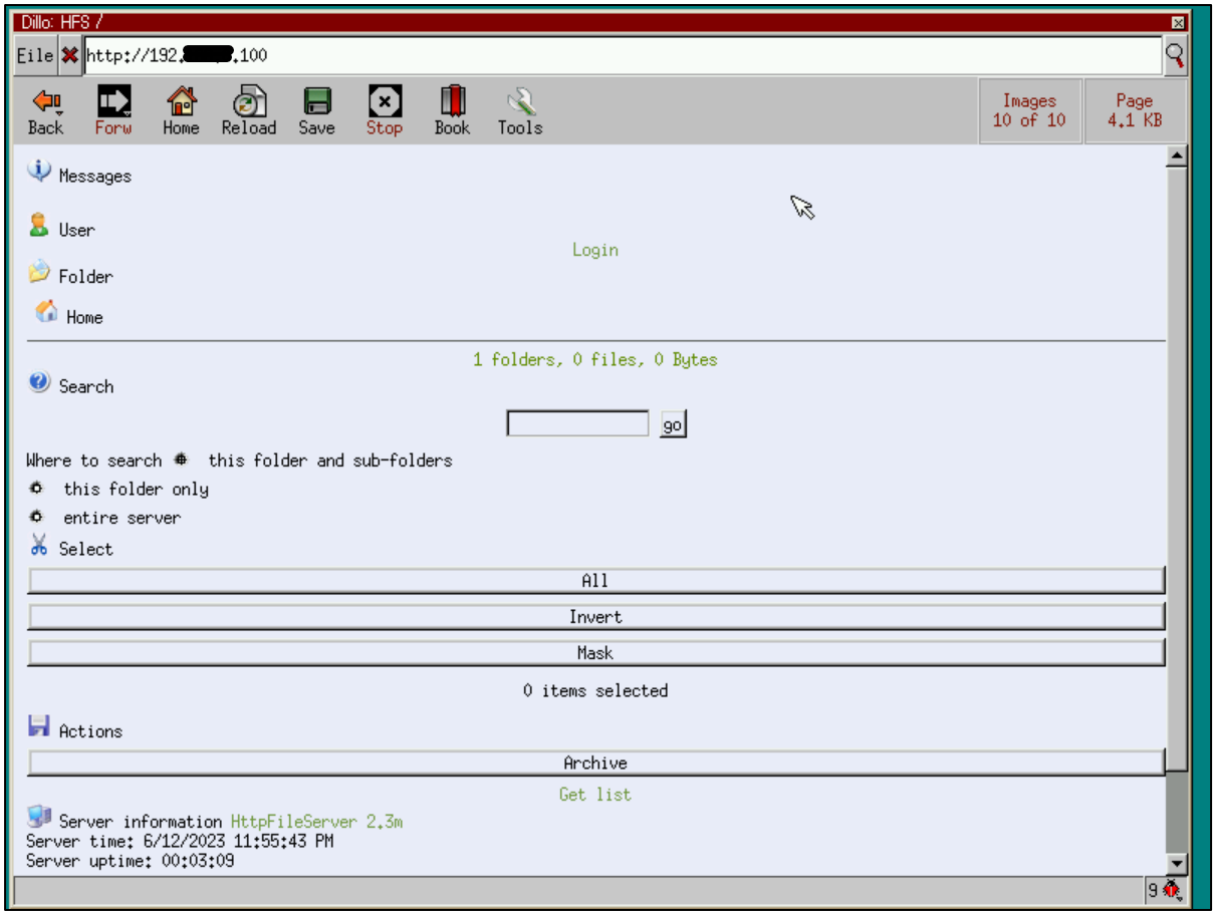
That's it for setting up the basic server and tests. Note that it is running in Admin mode which is fine for quick file transfers. If you wish to run the server for extended periods I would recommend reading the documentation and adding groups, user accounts and select file permission settings. This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

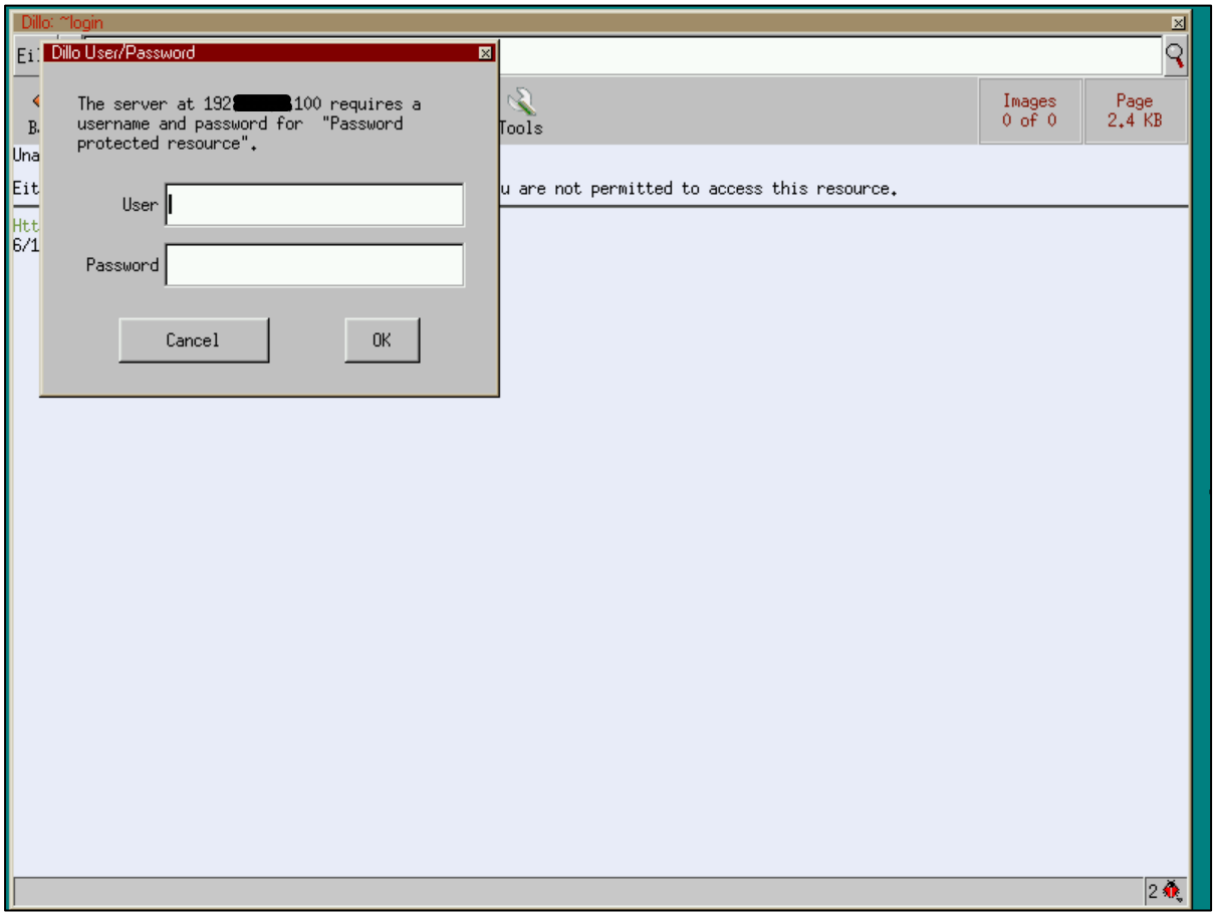
Navigating with Dillo in FreeDOS.

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

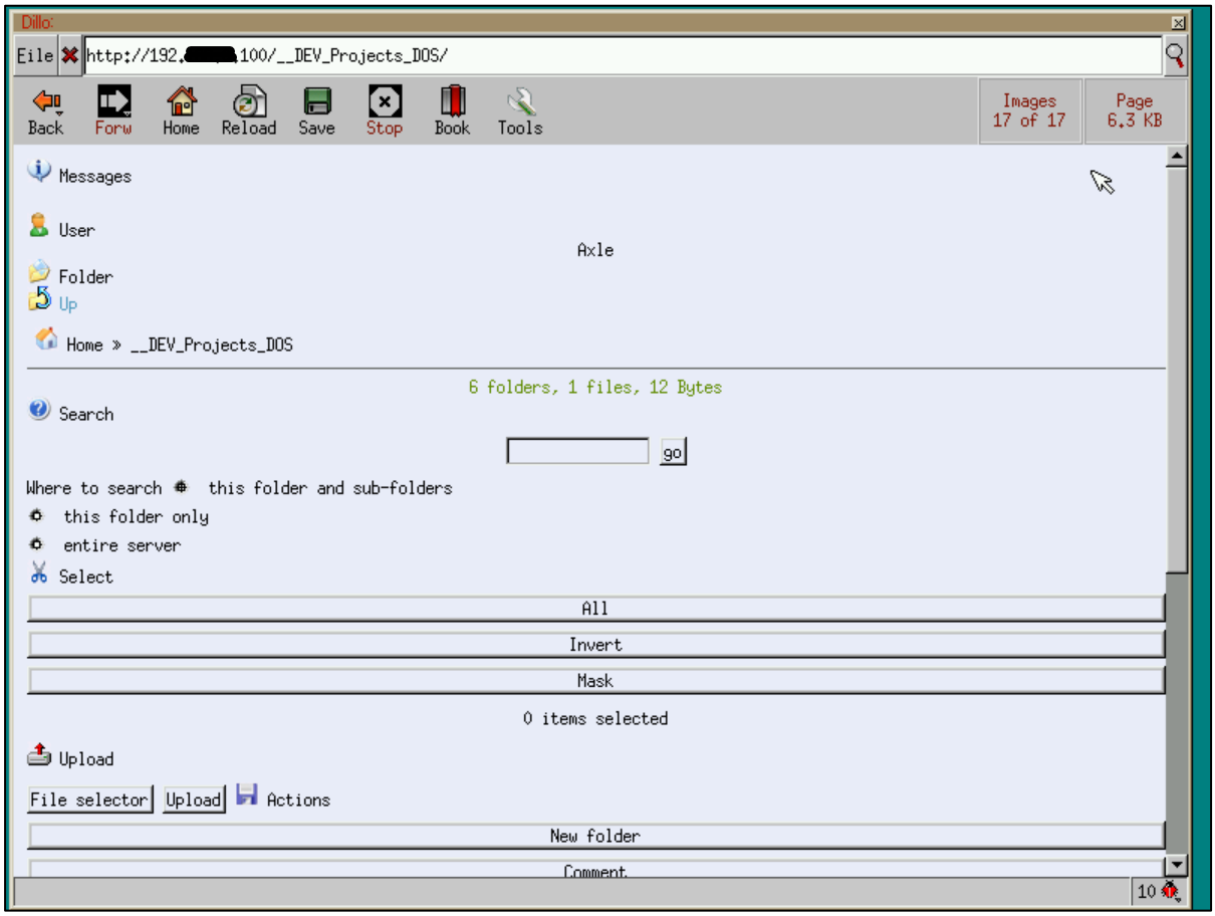


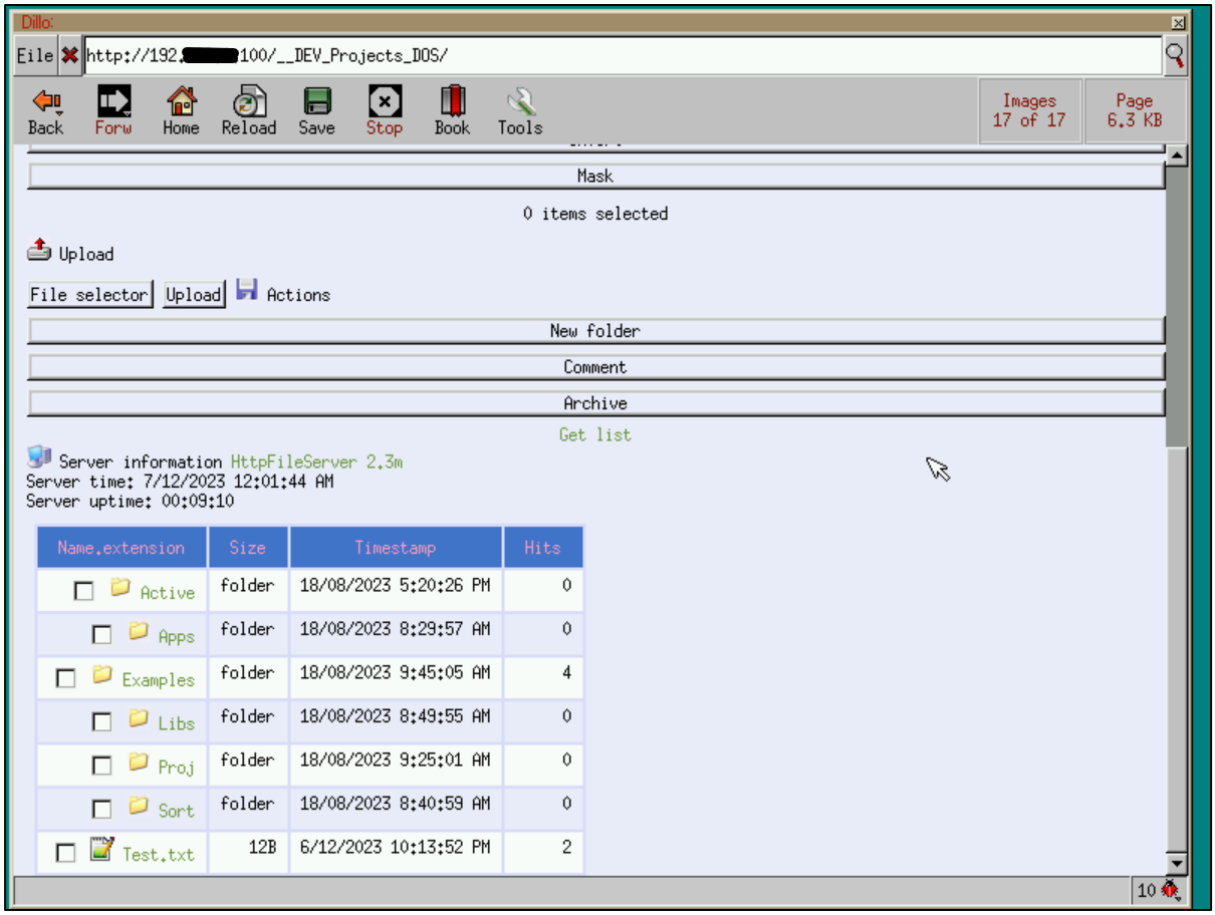
You are now at the HFS login page. Click on the login link and enter your user name and password.


Dillo: HFS /
File
http://192.[redacted].100
Back
Forw
Home
Reload
Save
Stop
Book
Tools
Images
10 of 10
Page
4.1 KB
Messages
User
Login
Folder
Home
1 folders, 0 files, 0 Bytes
Search
go
Where to search
this folder and sub-folders
this folder only
entire server
Select
All
Invert
Mask
0 items selected
Actions
Archive
Get list
Server information HttpFileServer 2.3m
Server time: 6/12/2023 11:55:43 PM
Server uptime: 00:03:09


Dillo: ~login
Dillo User/Password
The server at 192.[redacted].100 requires a username and password for "Password protected resource".
User
Password
Cancel
OK
Tools
Images
0 of 0
Page
2.4 KB
u are not permitted to access this resource.

You will find the layout a little different from your modern browser, but the functionality is still there.





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

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.

I typically use a separate directory from the `C:\APPS` 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 you IDEs. Create an image capture, copy of the directory structure on the host machine or in a note file that is easily viewable.

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.

QB 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 VisualBAISIC for DOS and Windows. QB 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 tradition 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 FreeDOS install drive. You can use any directory you want such as `C:\APPS`, 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.

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:\FREEDOS\LINKS`.

QBASIC.BAT

```
@ECHO OFF
REM Launcher for QBAISICv1.1
IF EXIST %dosdrv%\DEVEL\QBAS\QBASIC.EXE GOTO QBAS
GOTO NOFOUND
GOTO END
:QBAS
REM You can add environment paths or change working directory here.
SET PATH=%path%;C:\DEVEL\QBAS
```

```
REM Change current working directory
CD \DEVEL\QBAS
REM Launch QBAS
CALL %dosdrv%\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
```

You can also create an icon/link in Costa desktop for convenience. It may be helpful to place all of your development tools onto one (or Two) of the 5 available desktops in Costa to keep them separate from general apps or games.

Run the QBASIC.BAT. You can do this from the \LINKS directory, or any other directory as well as Costa if you have placed an icon on the desktop.

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
```

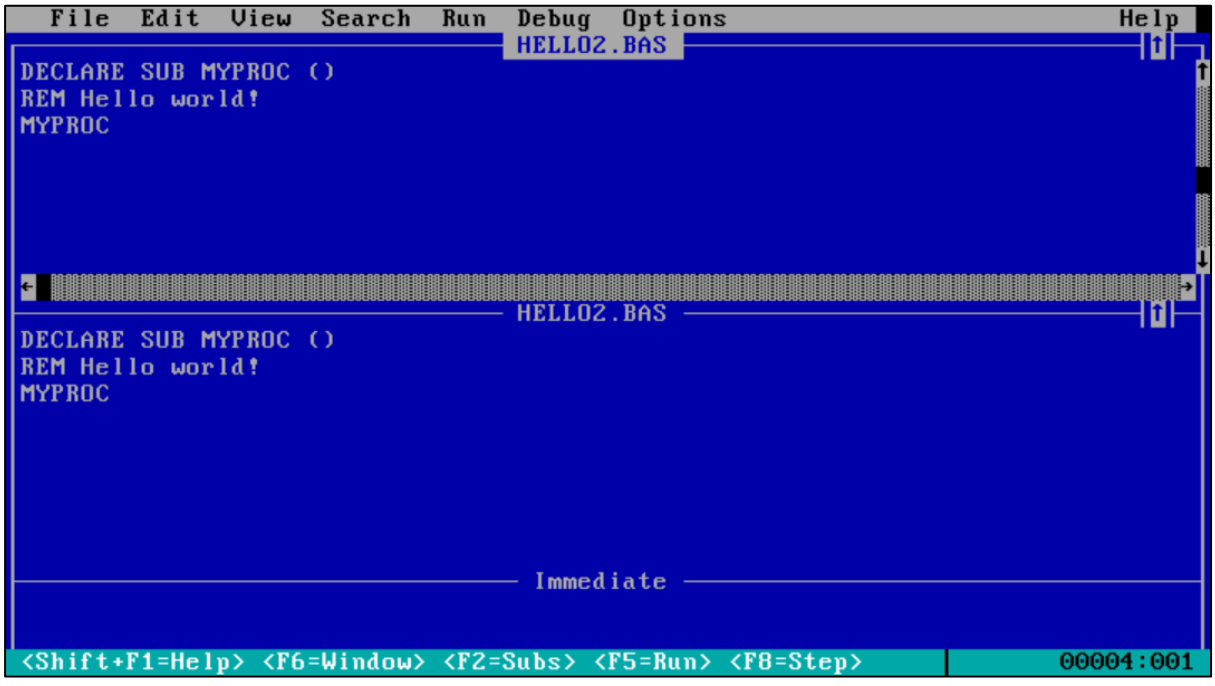
Select “Run -> Start”.

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. 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 when using FreeDOS_1.3 QB45 installers.

PDF Manual (Microsoft_QuickBASIC_4.5_2nd_Edition_Manual.pdf)

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 same directory structure as the archive downloaded from qbasic dot net on your DOS hard drive.

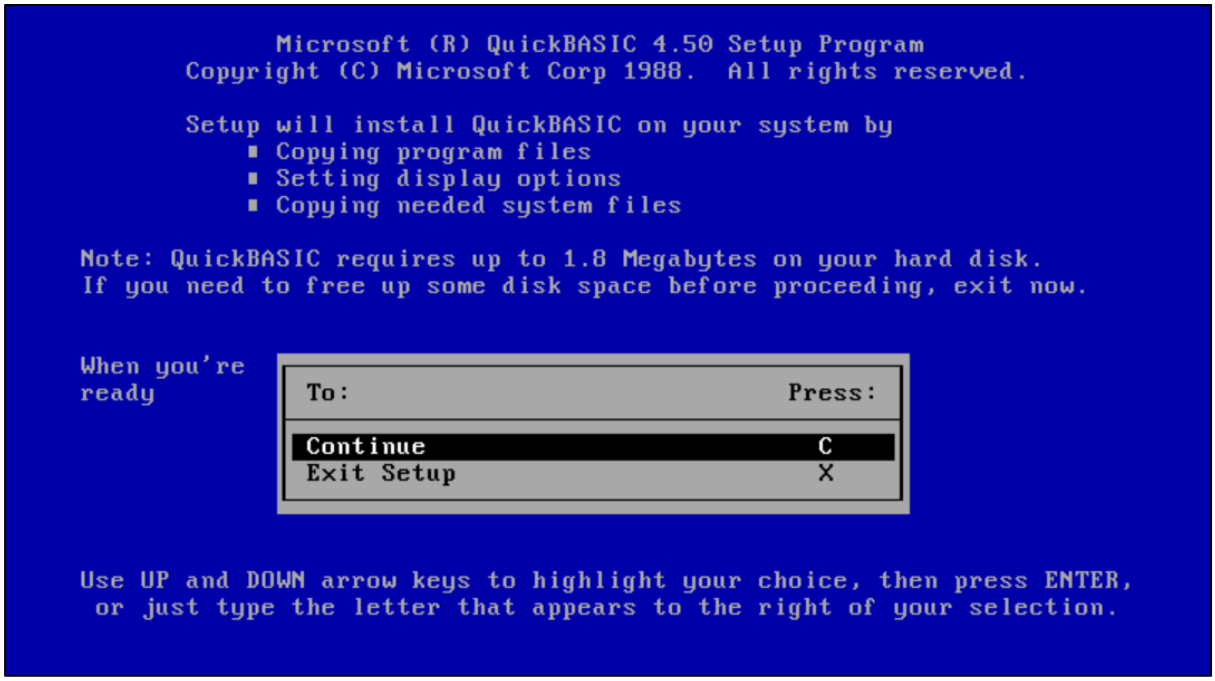
Either Install QuickBASIC by using the 5 floppy images mounted in VirtualBox (Recommended), copy the unpacked floppy images to a suitable location in your DOS drive with both directories combined and run the installer or copy the complete unpacked ready to go directory. I would recommend either the 5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

Install (003495_microsoft_quickbasic.7z) Full install set.

Mount "setup.img" in VirtualBox and navigate to the floppy drive. You can do this easily with FM3.

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.

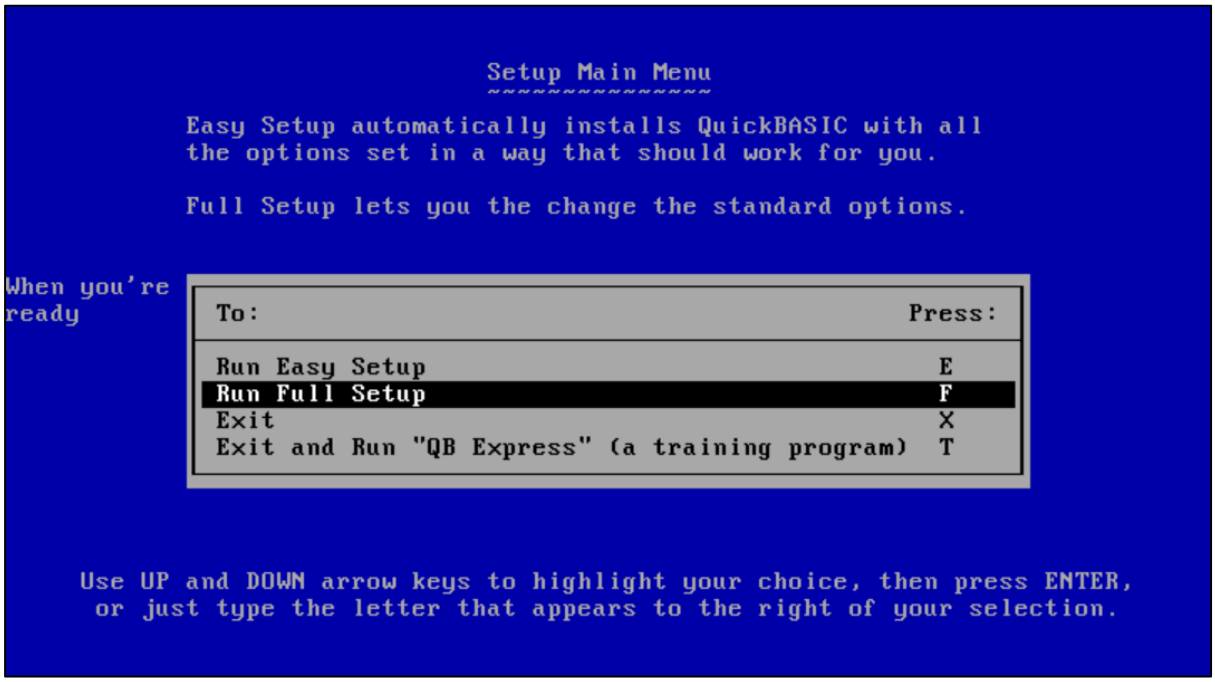
Type SETUP (SETUP.EXE) and Enter to begin the install.



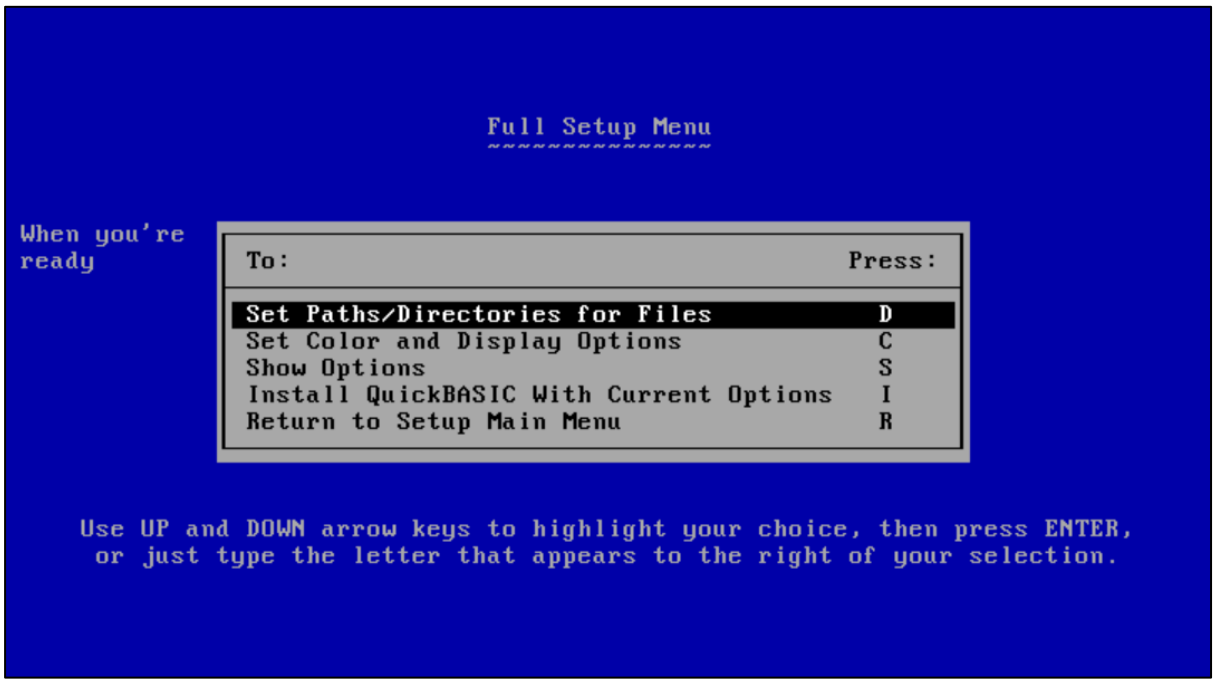
I would choose "Run Full Setup" so that we can choose the install location for QuickBASIC. If you are unsure about the directory structure you can use "qb45.zip" from qbasic net as a guide.

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 uncompelled library source files)
.\QB45\LIB (pre-compiled library files)
.\QB45\HLP (Help files)

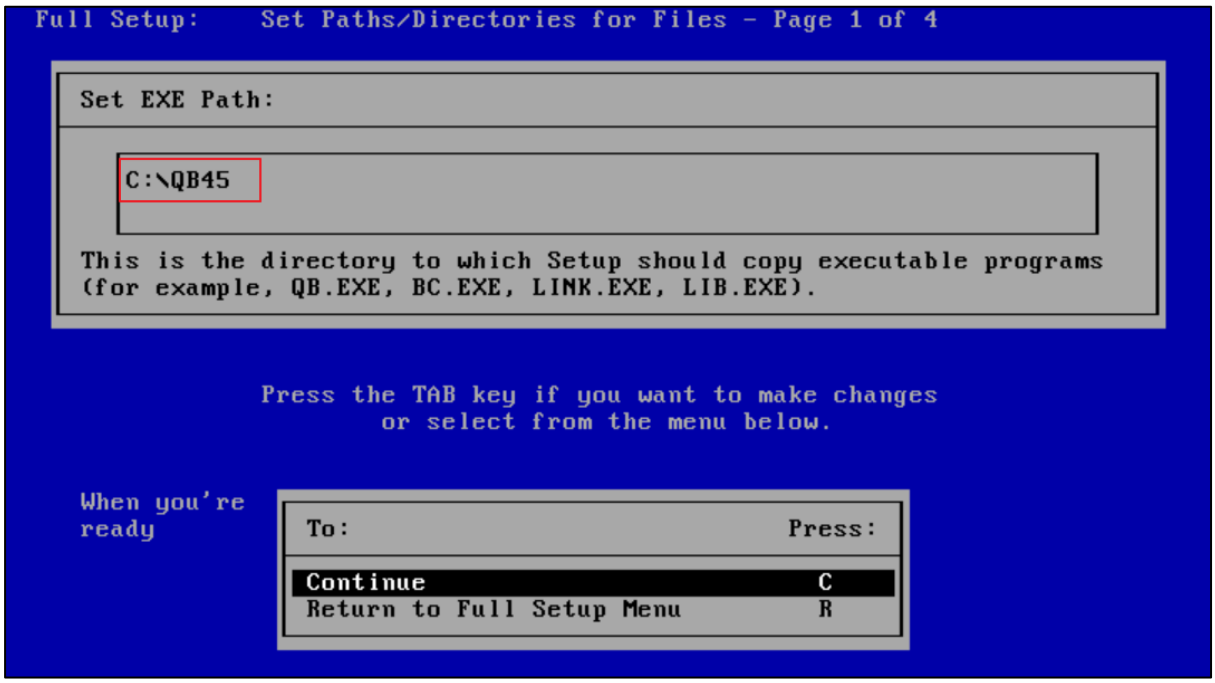
The “Run Easy Setup” will just install to a default location, usually C:\QB45 as well as default options which we do not want. We want to place out install in C:\DEVEL\QB45*.*

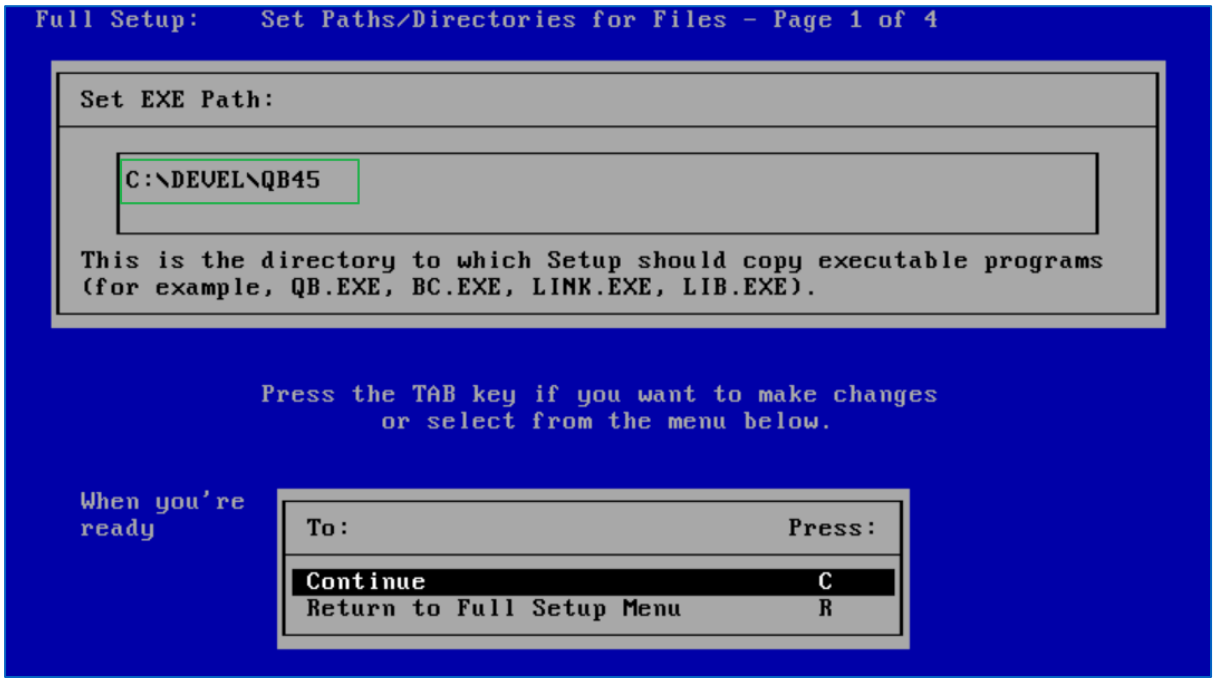


In the next screen change the “Set Paths/Directories for files” options to your C:\DEVEL\QB45 path.



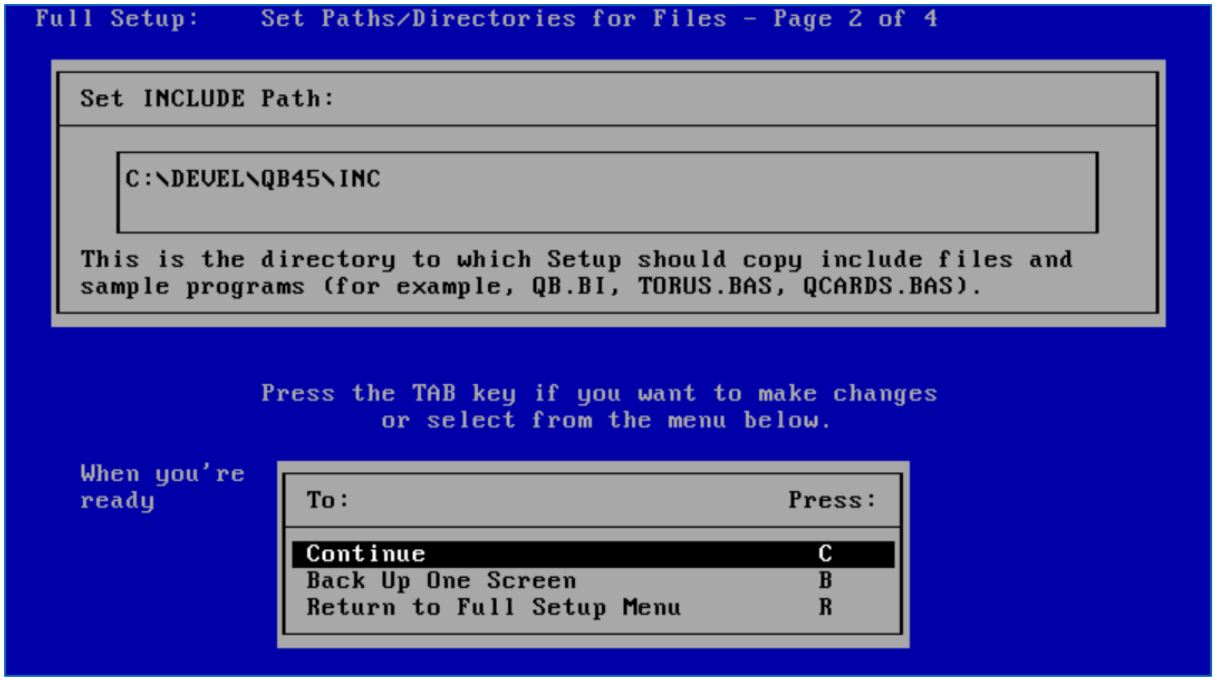
Make sure you select the correct directory name of your development directory if it is different to C:\DEVEL. I don't recommend using the default directory.





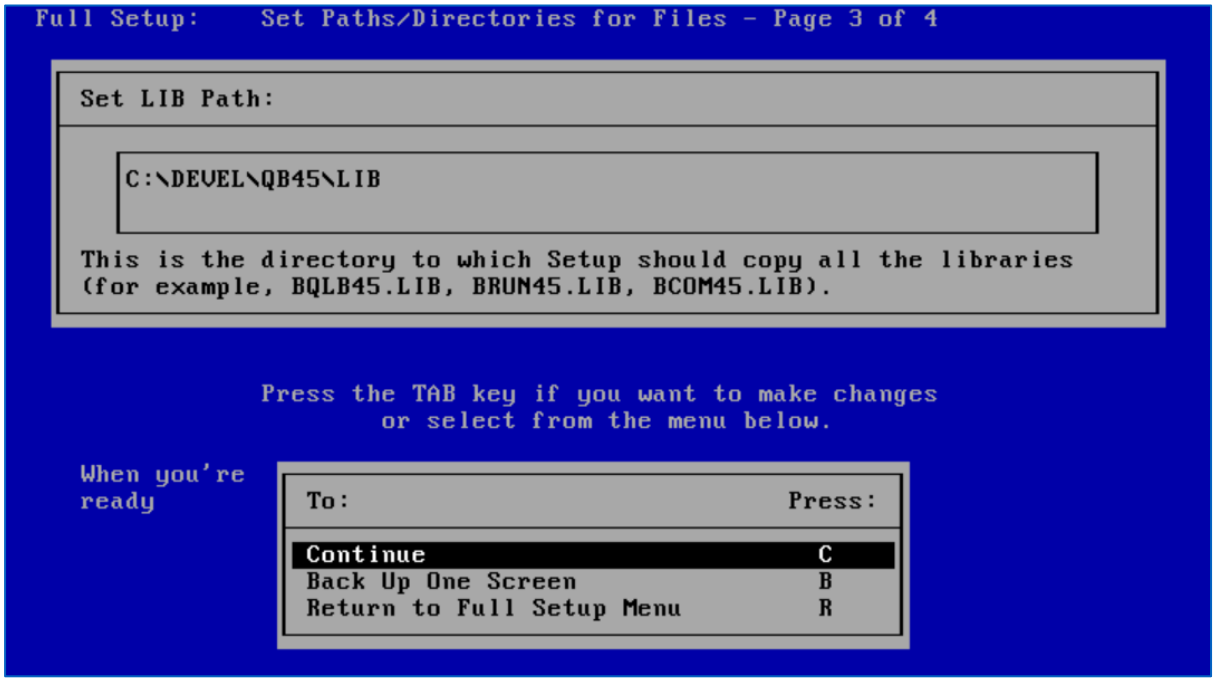
Use TAB to select C Continue.

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`

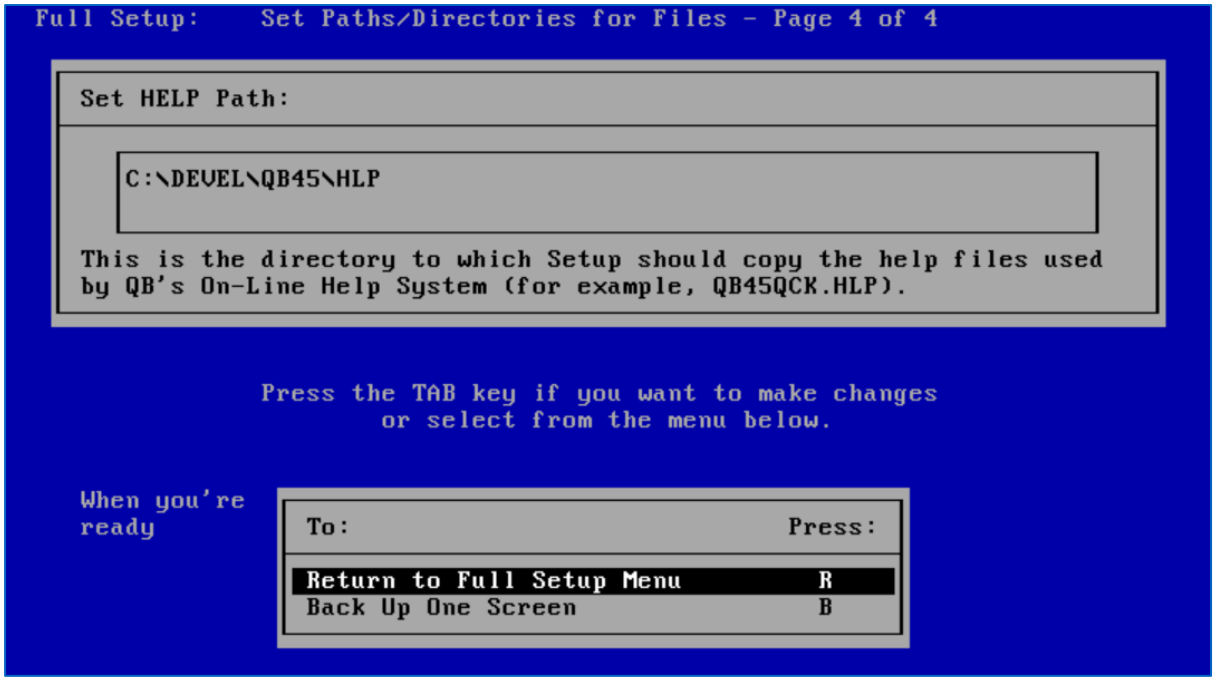


Select C Continue.

Change the library \LIB path as we did above for \INC.



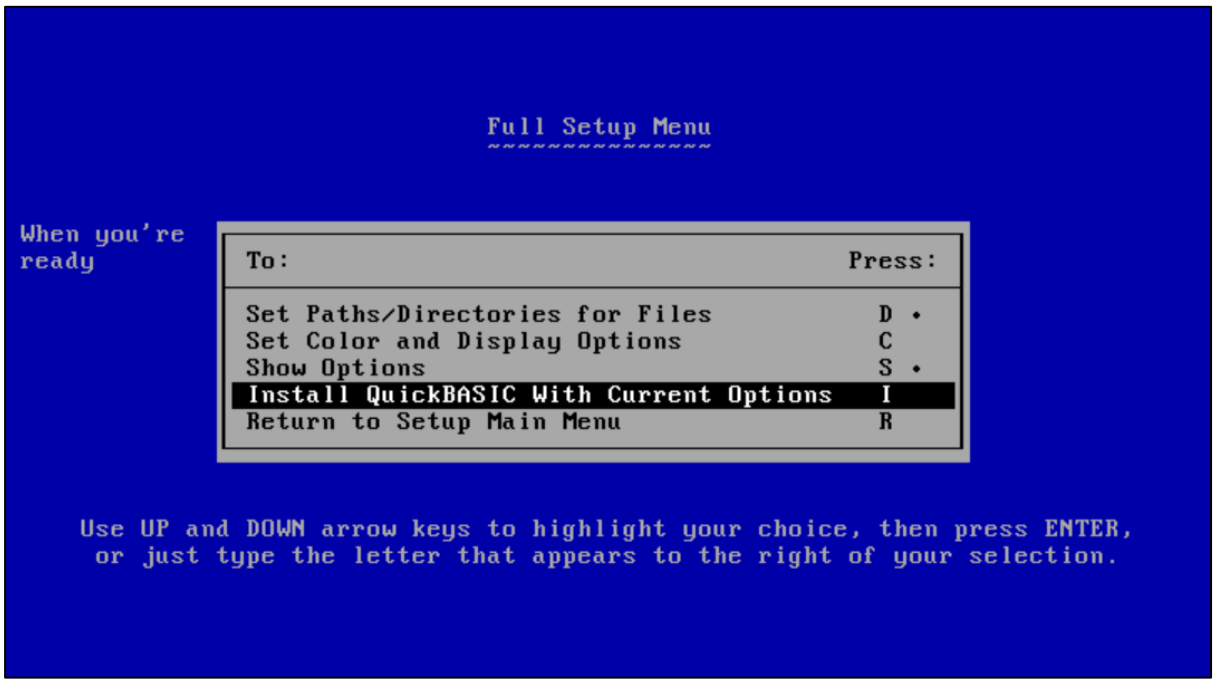
Select C Continue and set the path for the help files `.\QB45\HLP`.



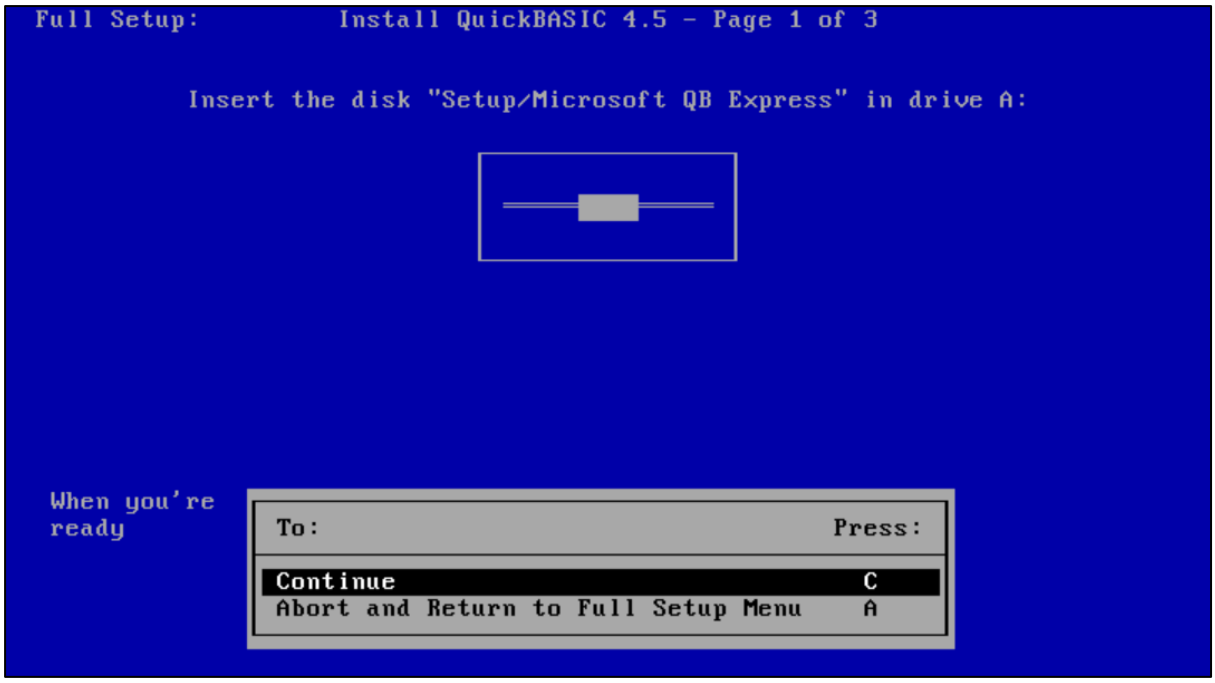
Select R "Return to the Full Setup Menu" as we are done setting the custom install paths.

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.

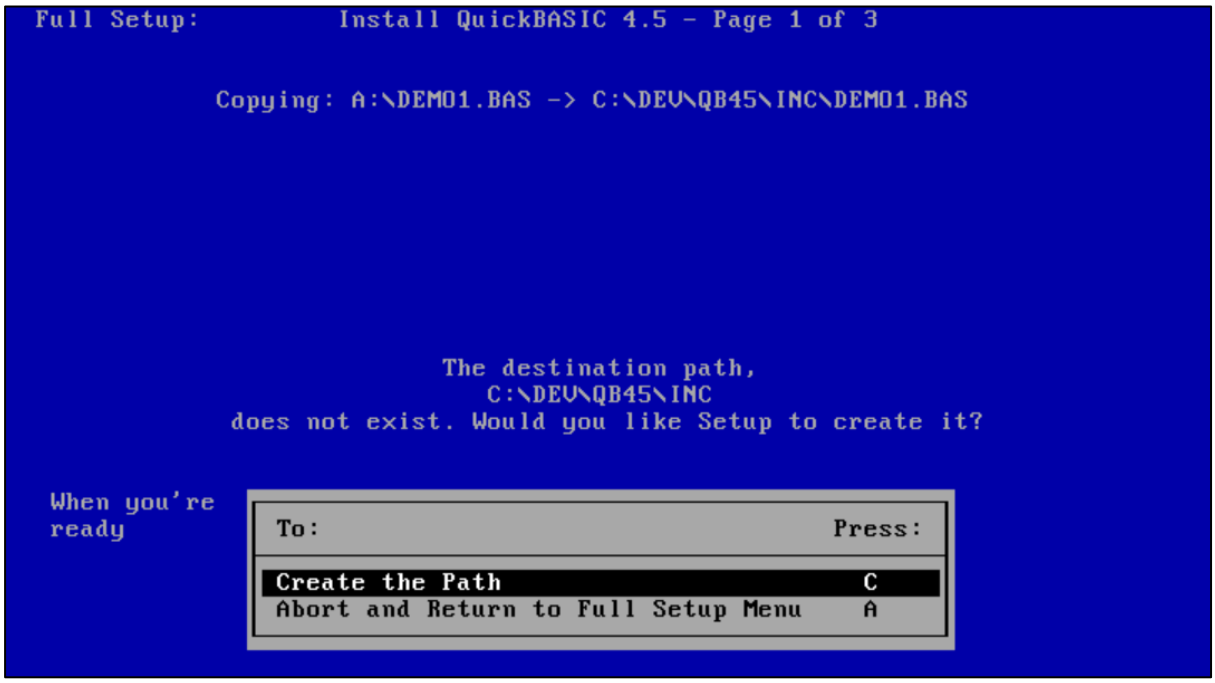
Next select "Install QuickBASIC With Current Options".



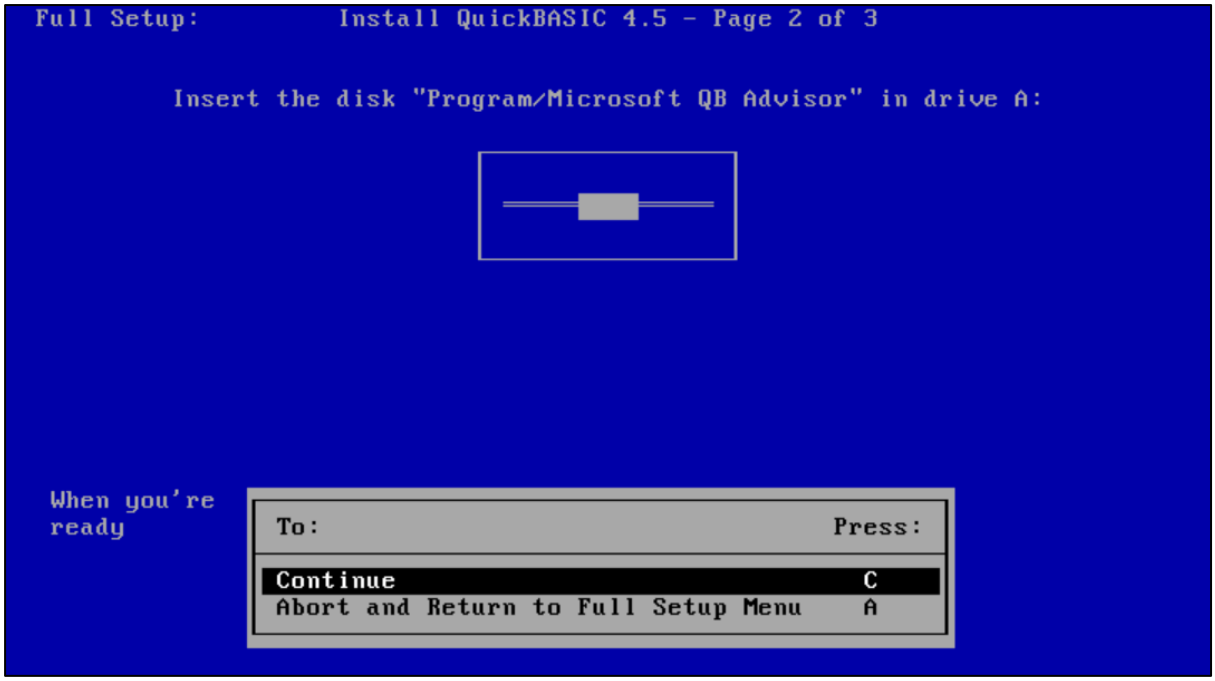
The next step relates to the current “setup.img” disk so it OK to just continue.



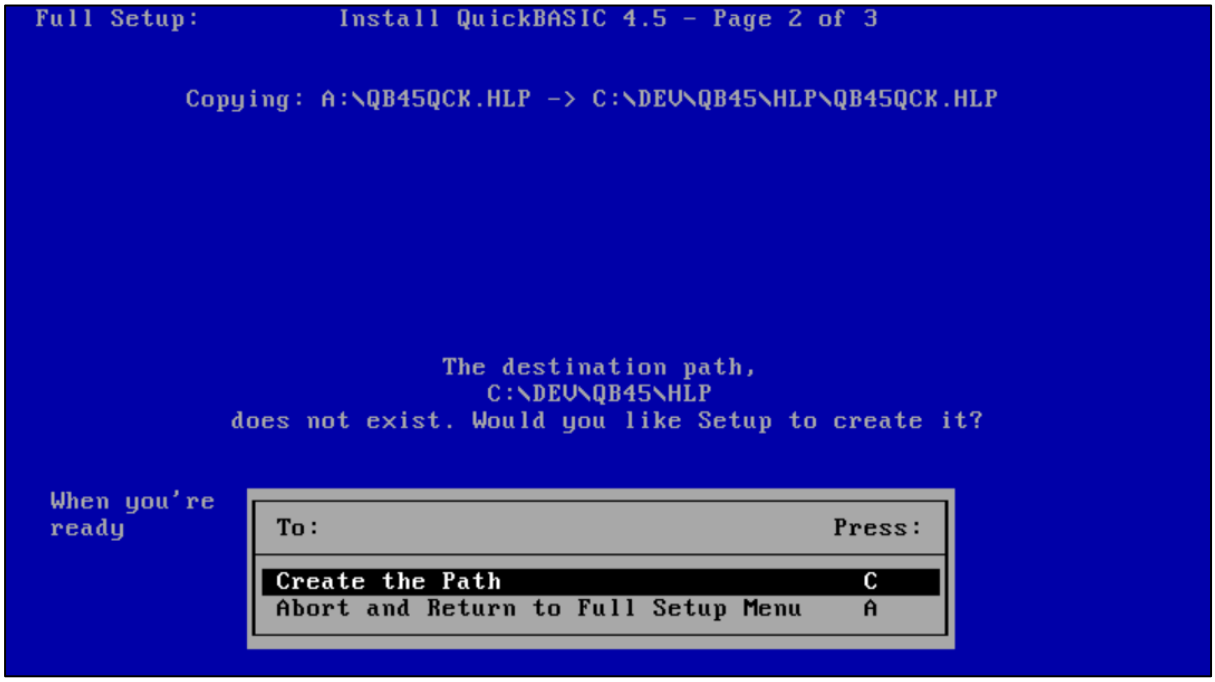
Select “Create the Path” for all of the next screens.



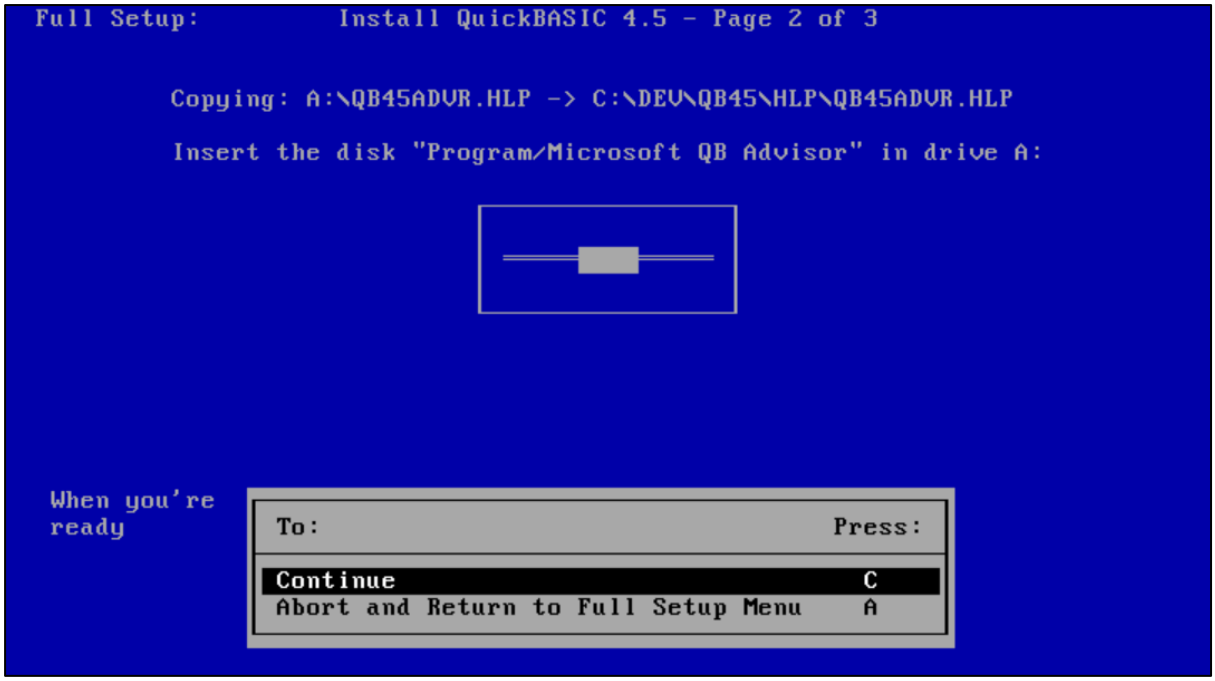
Next mount the “program.img” and select Continue.



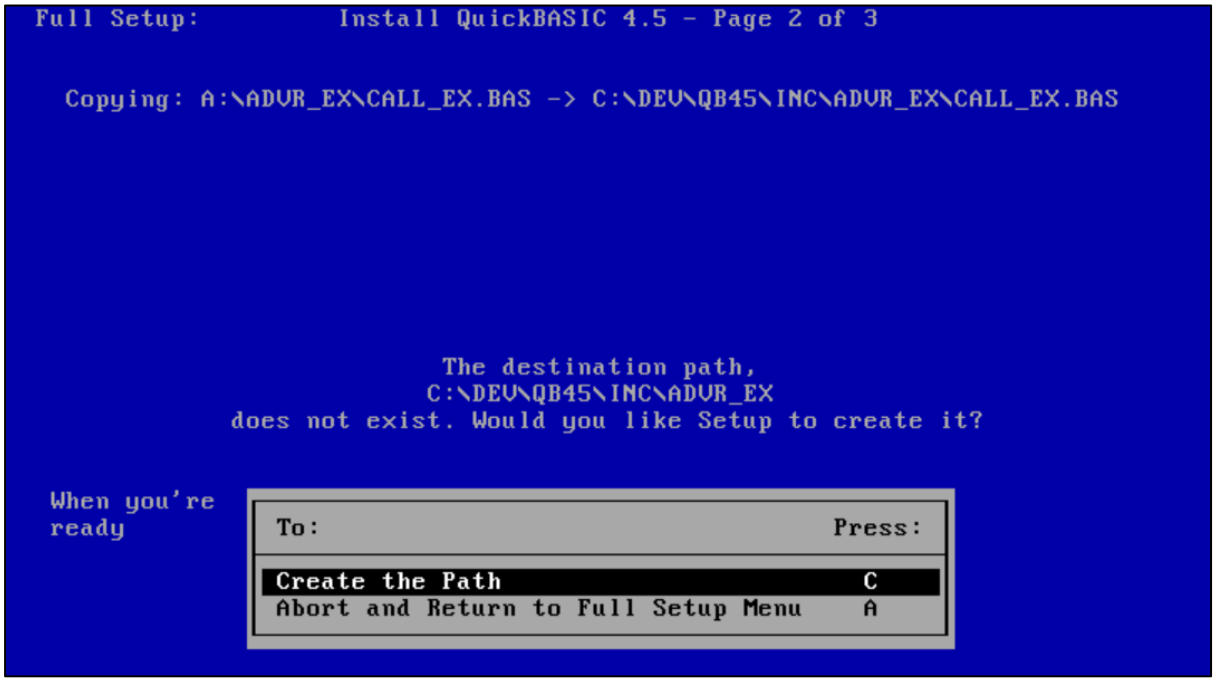
Select “Create the Path” and continue.



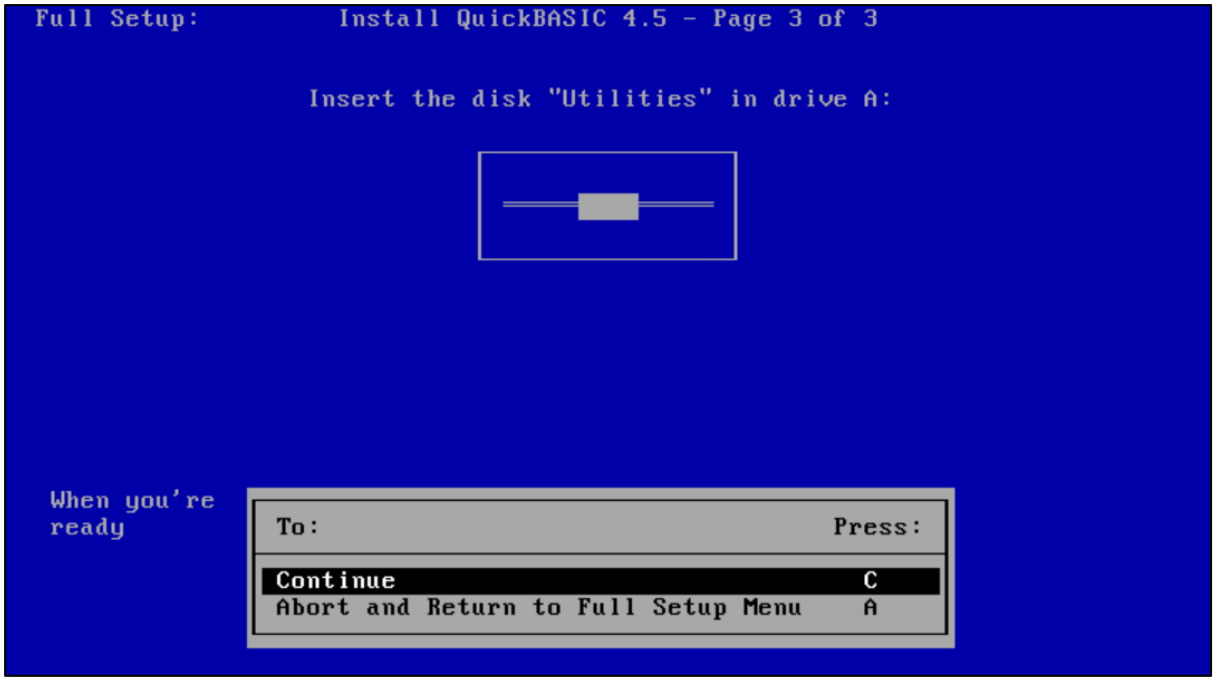
Next mount the “qb_advisor.img” and continue.



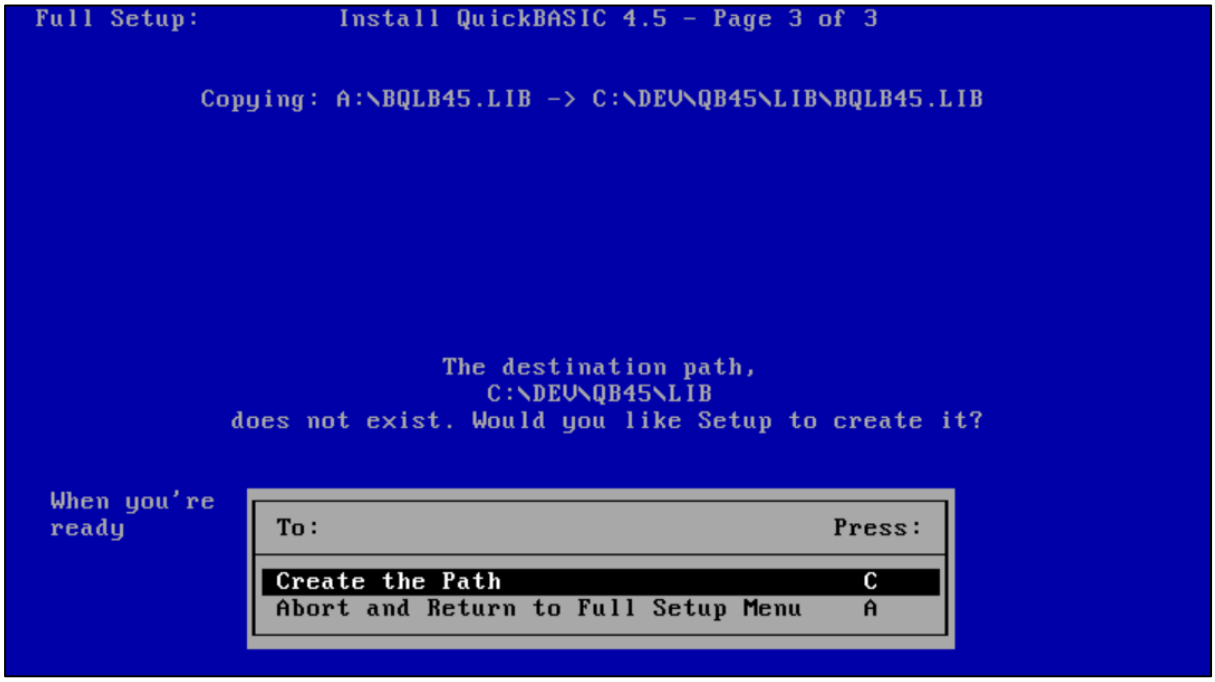
Select “Create the Path” to continue.



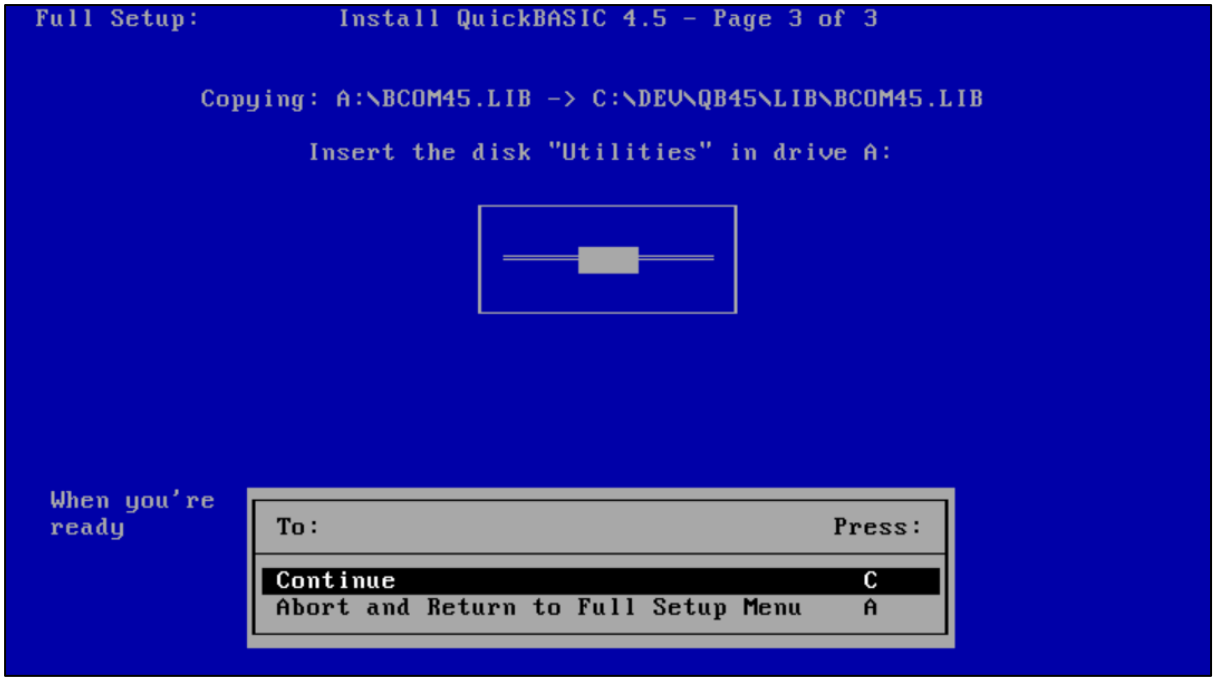
Next mount the utilities_1.img and select Continue.



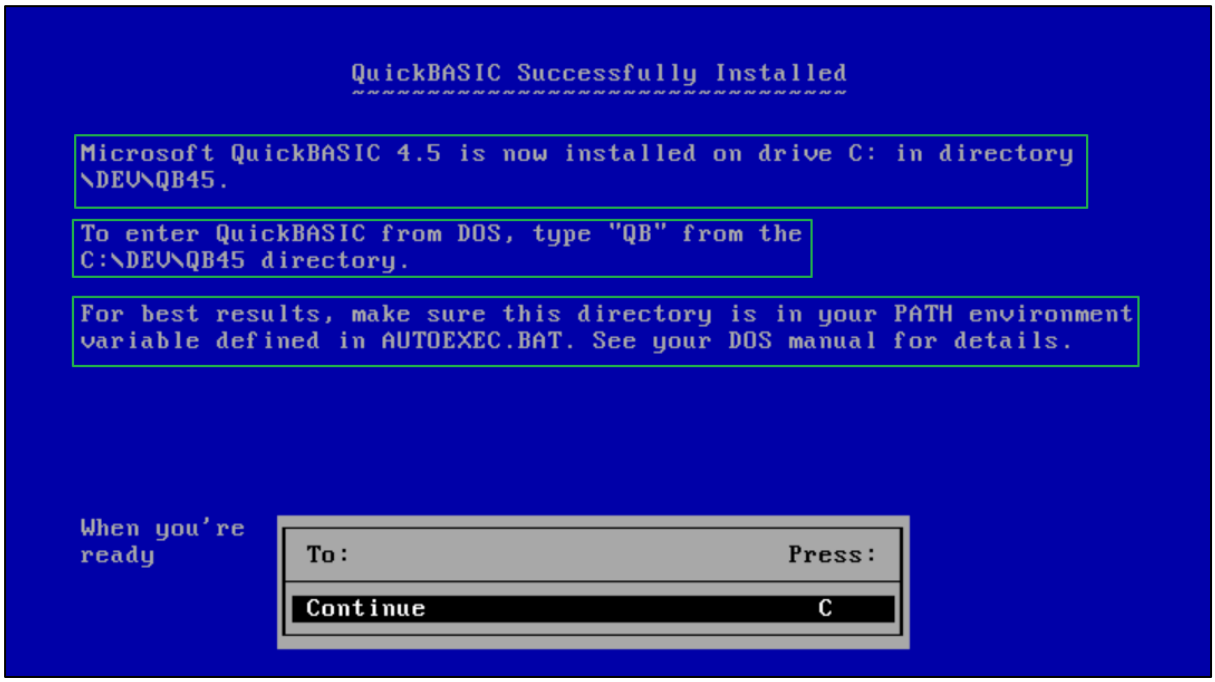
Select “Create the Path” to continue.



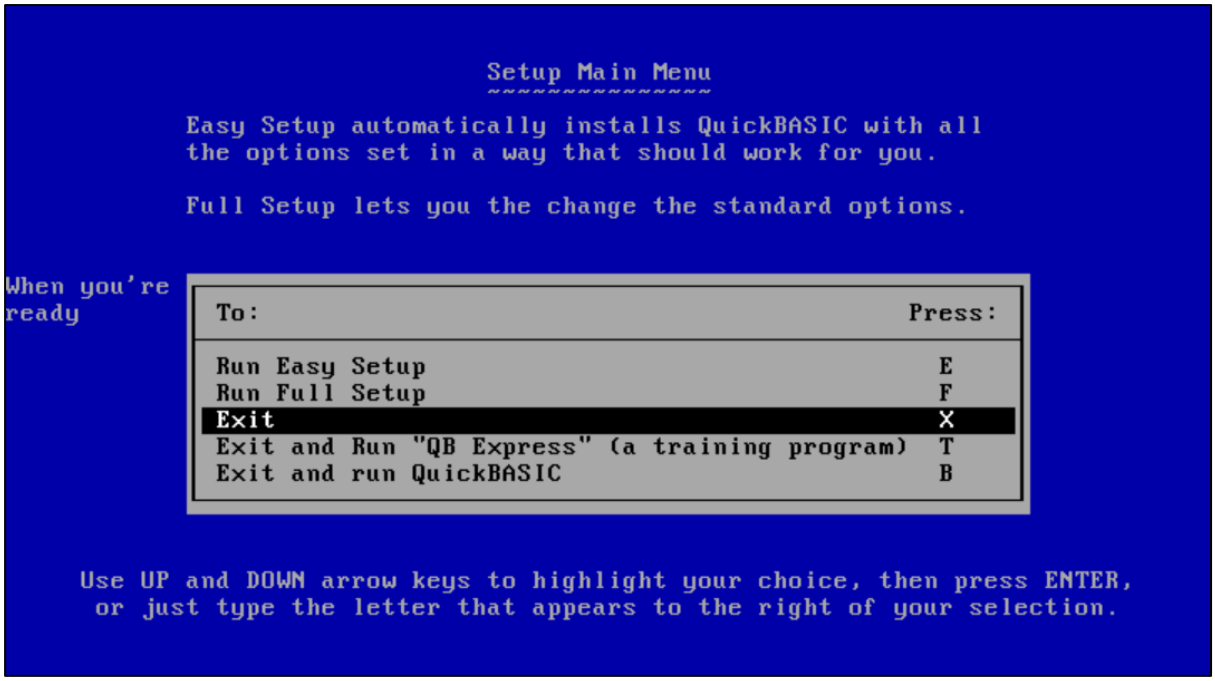
Next mount the utilities_2.img and select Continue.



The next screen shows that you have completed the install. After this we will next choose how to start the application. Don't set the path environment in AUTOEXEC.BAT just yet.



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 SETing 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, 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.

To use the AUTOEXEC.BAT (FDAUTO.BAT). Not recommended.

We can Place the additional path after the pre-existing SET PATH=C:\FREEDOS;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
PAUSE
CALL %dosdrv%\MENU.BAT
```

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:\FREEDOS\LINKS directory.

QB45.BAT

```
@ECHO OFF
REM Launcher for QB45
SET PATH=%path%;C:\DEVEL\QB45
CD \DEVEL\QB45
CALL %dosdrv%\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.

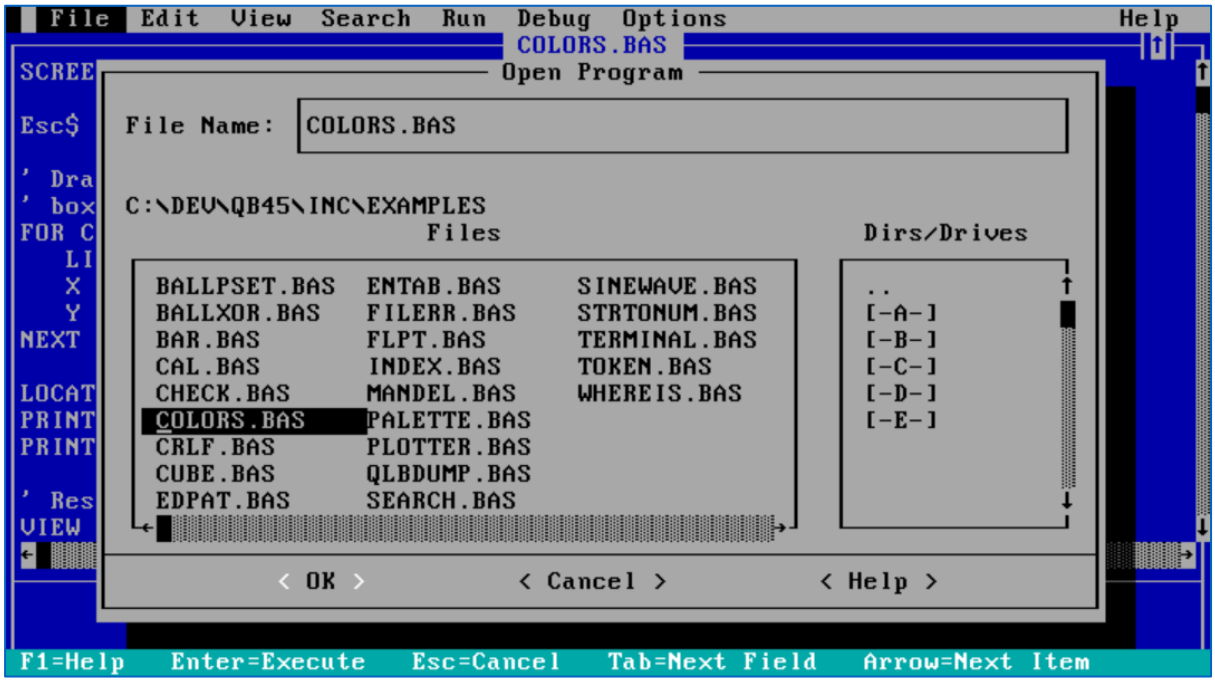
QB45.BAT

```
@ECHO OFF
REM Launcher for QuickBAISICv45
IF EXIST %dosdrv%\DEVEL\QB45\QB.EXE GOTO QB
GOTO NOFOUND
GOTO END
:QB
REM You can add environment paths or change working directory here.
SET PATH=%path%;C:\DEVEL\QB45
CD  \DEVEL\QB45
REM Launch QB45
CALL %dosdrv%\DEVEL\QB45\QB.EXE %1 %2 %3 %4 %5 %6 %7 %8 %9
GOTO :END
:NOFOUND
ECHO QUICBAISIC not found!
ECHO Press any key to end...
PAUSE
:END
CLS
```

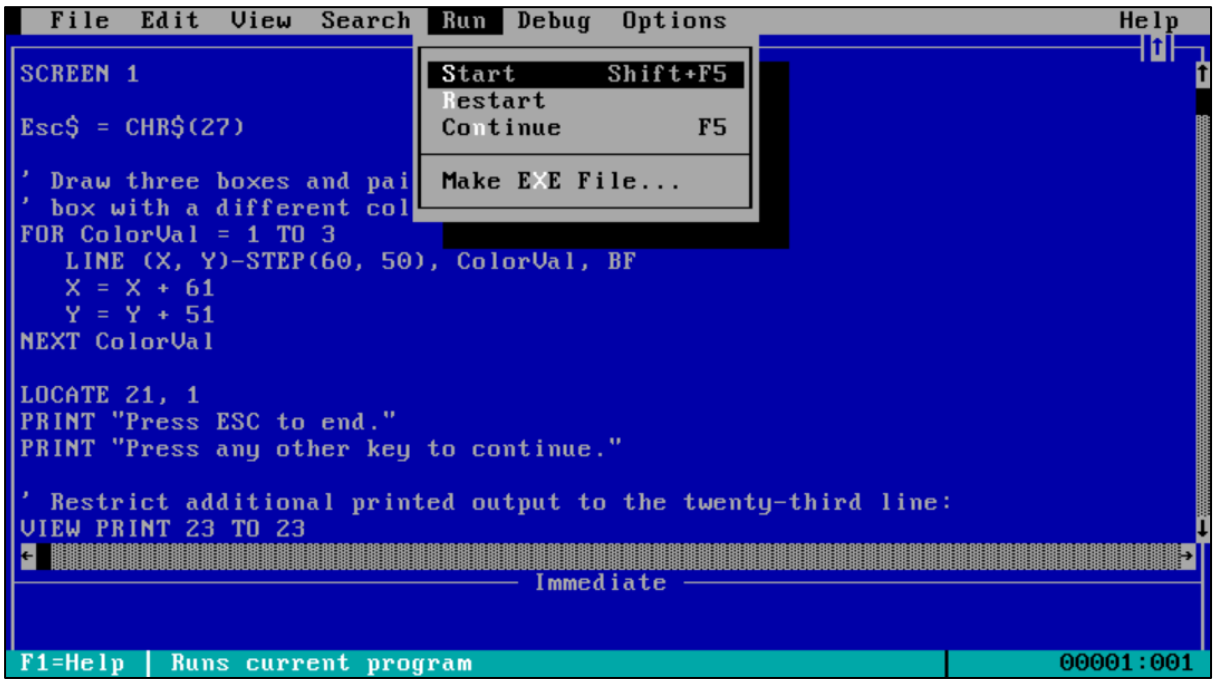
Run a test application.

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.

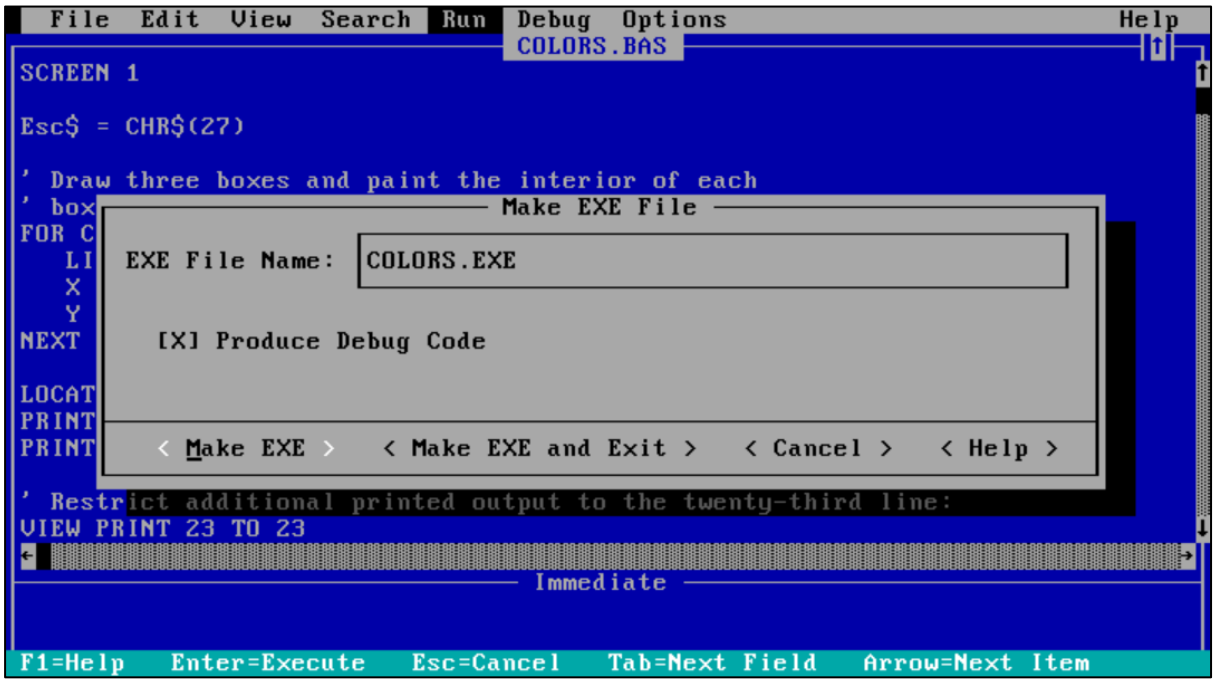


Next select “Run -> Start” to run and test the source code in interpreted mode.



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”

Install instructions – standard edition.

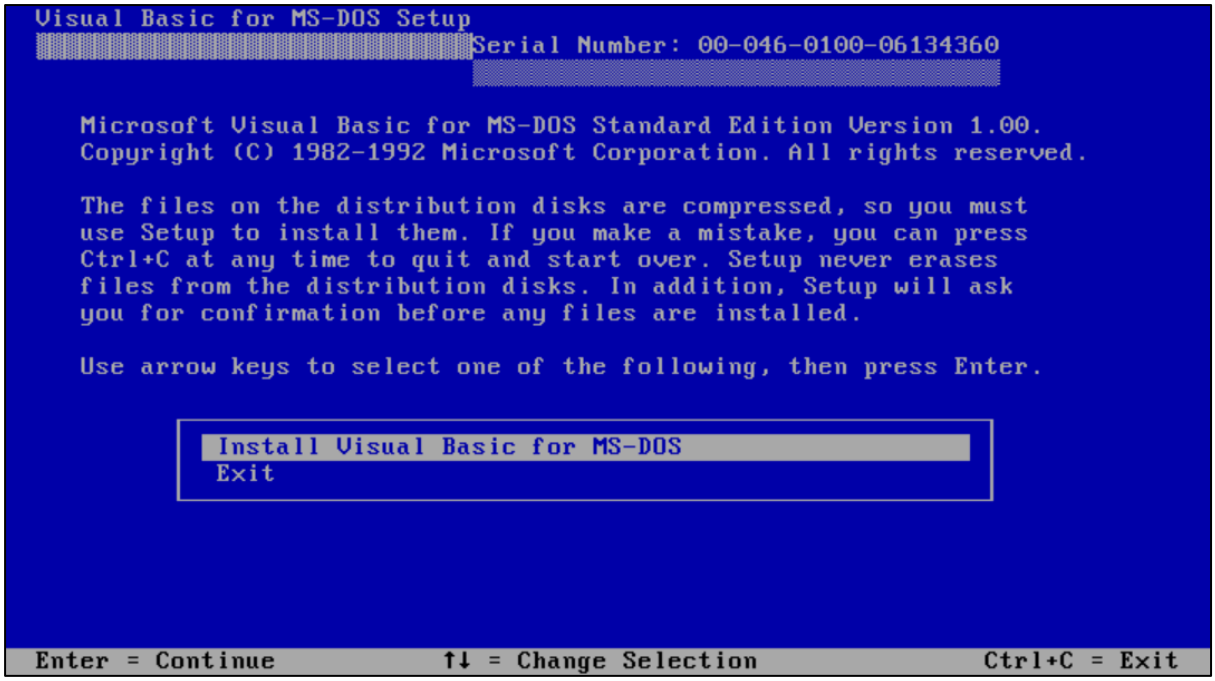
I will be using the 2 disk Standard Edition from WinWorld PC “Microsoft Visual Basic 1.0 Standard for MS-DOS (1992) (3.5-1.44mb).7z”

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.

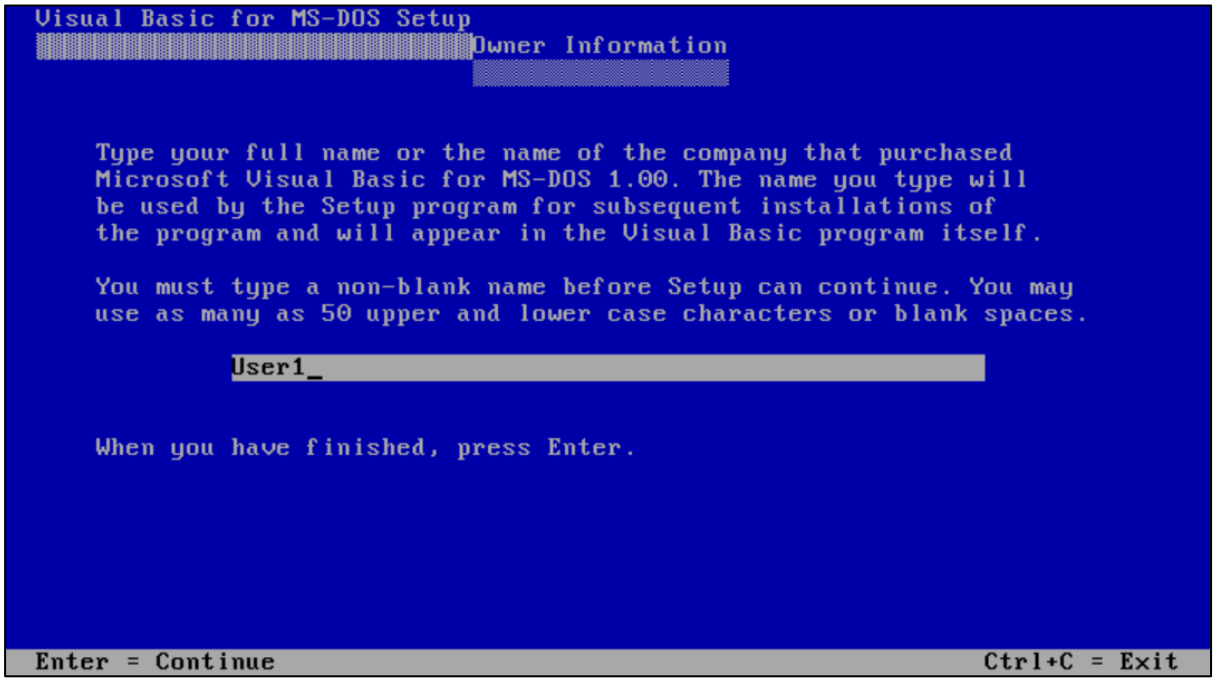
Unpack “Microsoft Visual Basic 1.0 Standard for MS-DOS (1992) (3.5-1.44mb).7z” and mount disk01.img in VitualBox.

From your FreeDOS client, navigate to drive A: and type SETUP[.EXE].

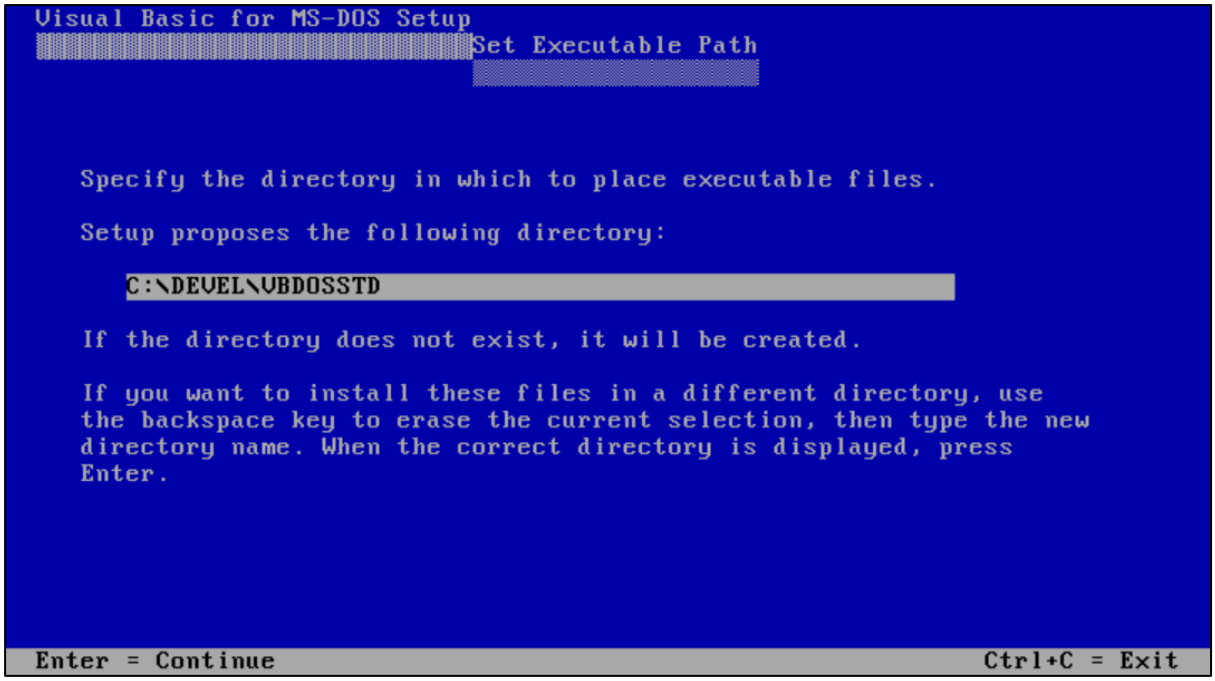
Select “Install Visual Basic for MS-DOS”.



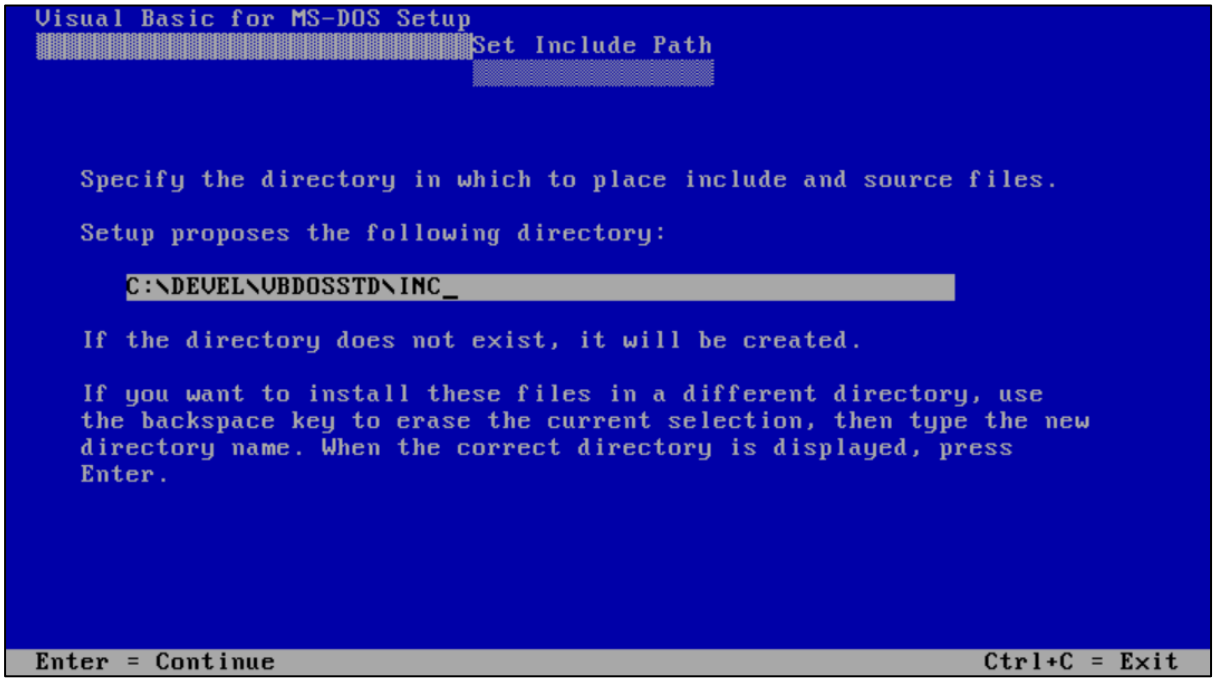
Enter the registered name of your copy of Visual Basic for MS-DOS, and verify the name.



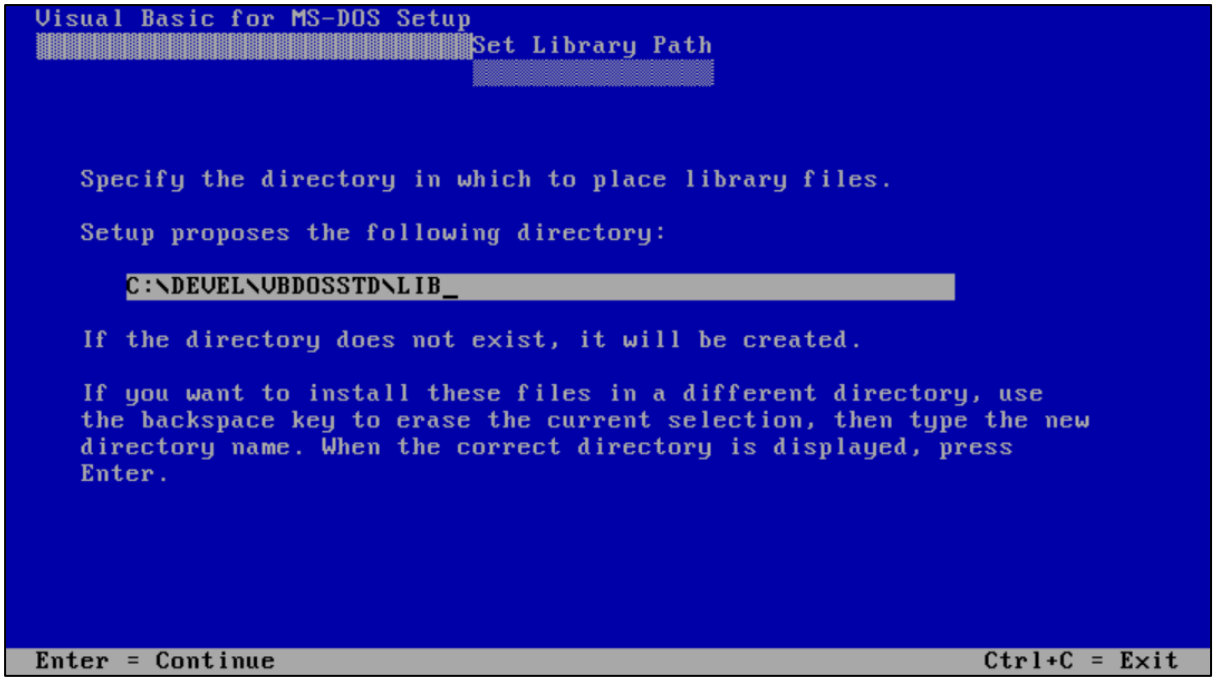
Next select the install directory. I am using `C:\DEVEL\*.*` for this guide which is the FreeDOS 1.3 default. Although the default directory is `C:\VBDOS` I have changed this to VBDOSSTD to so I will have a different directory name in case I choose to install the Pro edition later.



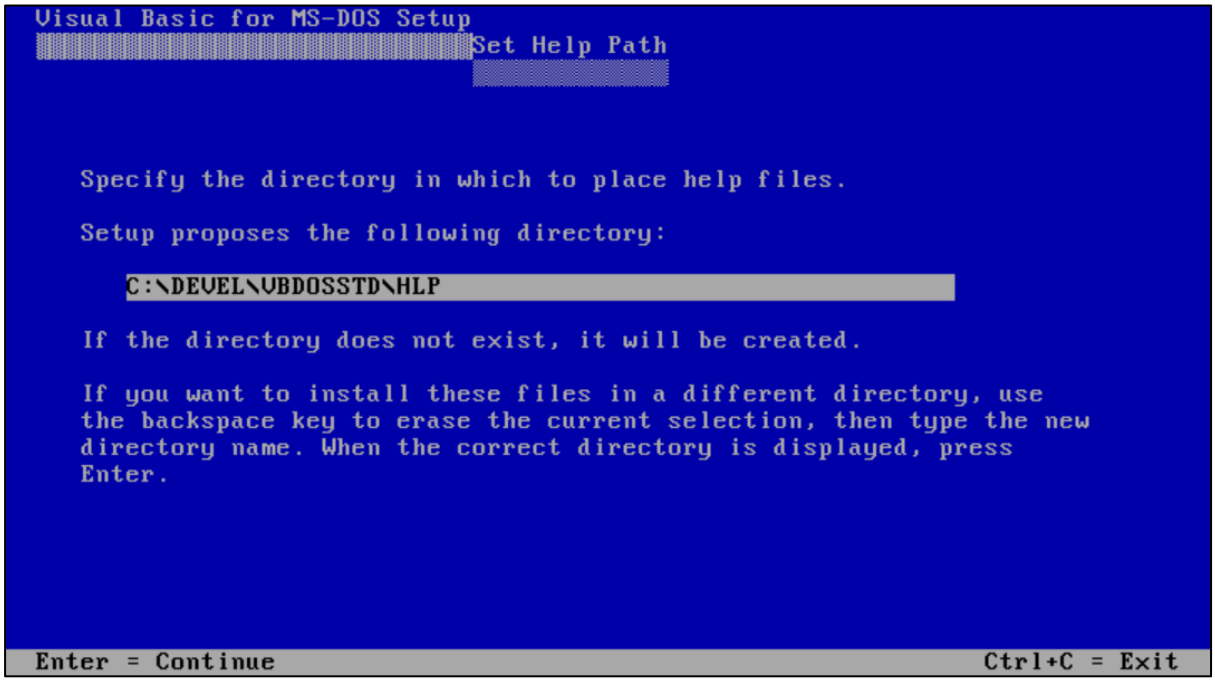
Enter the include path `C:\DEVEL\VBDOSSTD\INC`



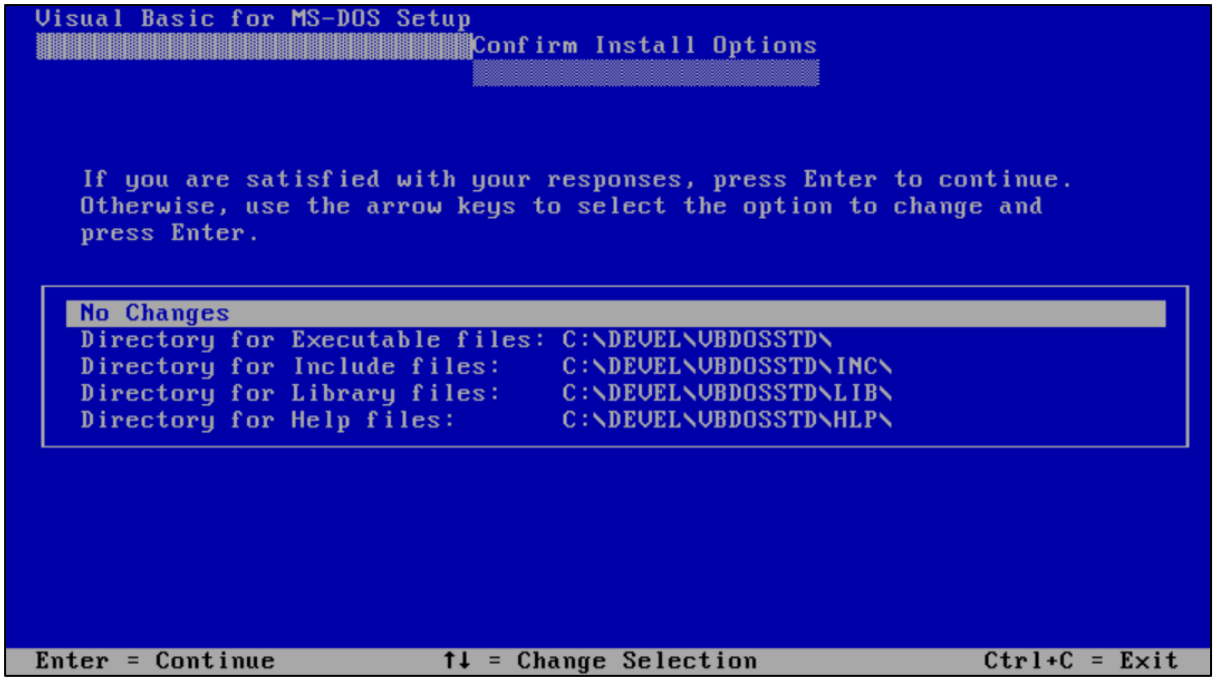
Enter the library path `C:\DEVEL\VBDOSSTD\LIB`



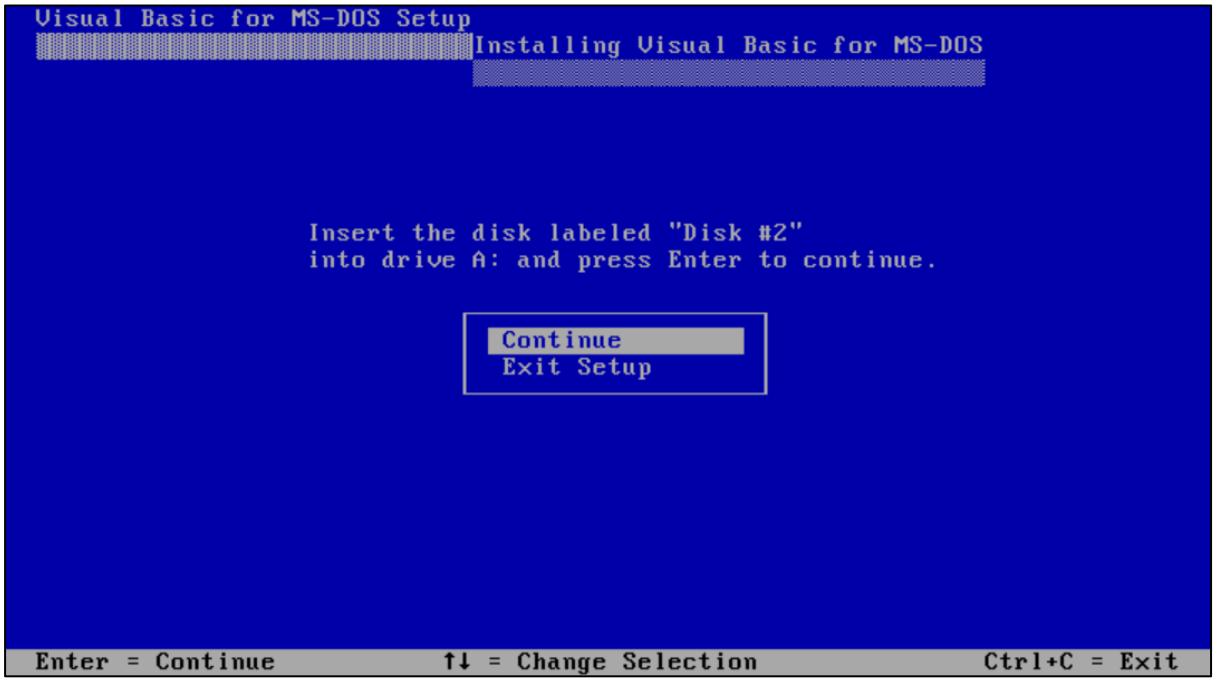
Enter the help path C:\DEVEL\VBDOSSTD\HLP



Check that the install paths are correct and press Enter to continue the install.

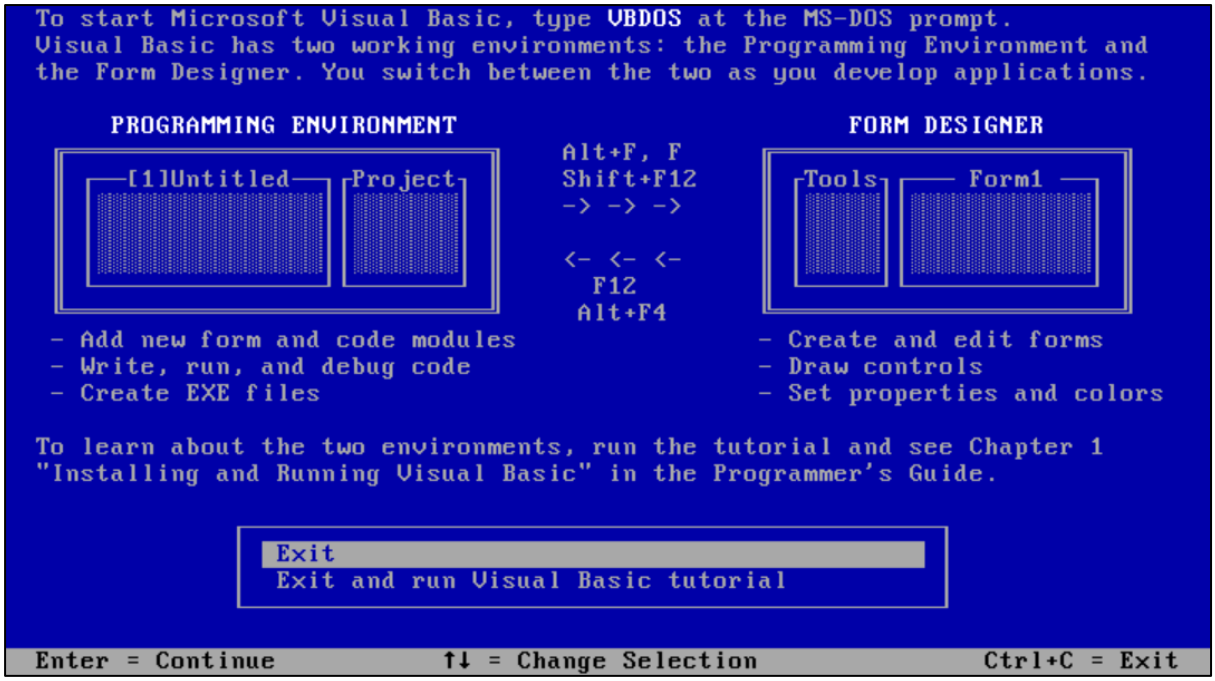


Mount the second disk in VirtualBox and Enter to continue the file install.



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.



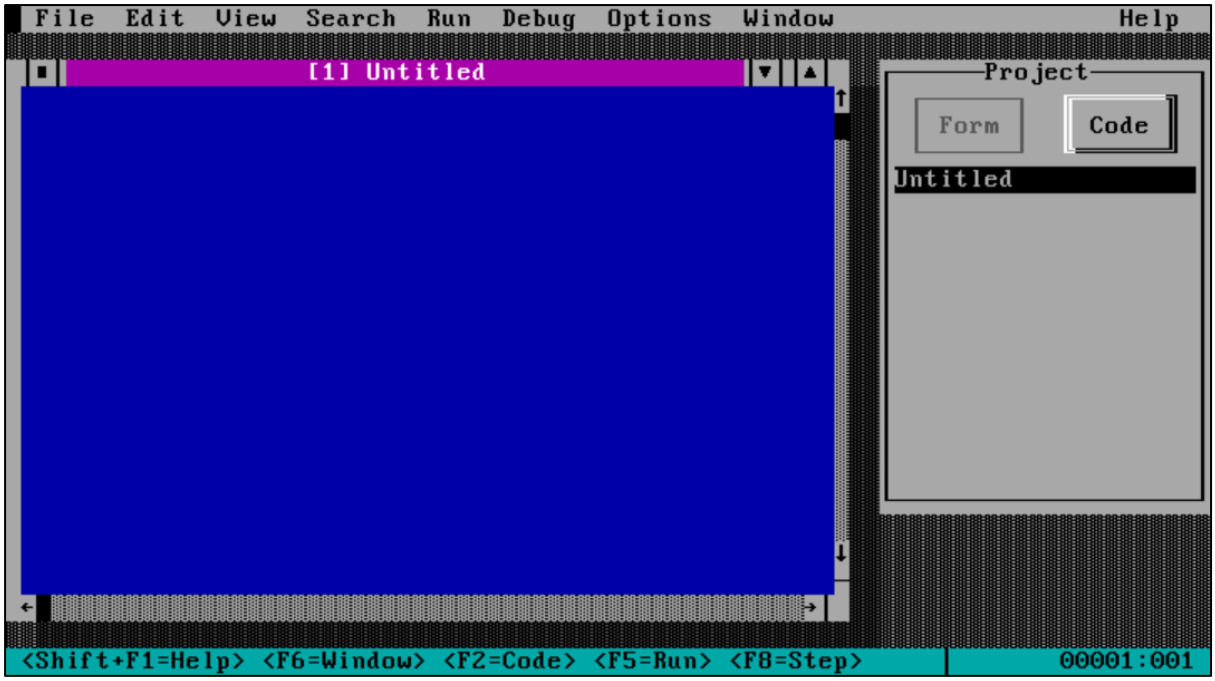
If you are installing to FreeDOS then the AUTOEXEC.BAT and other launch batch files will not be created.

Create a basic VBDOSSTD.BAT file from the guide "**Application launch BAT**" and place it in the \LINKS directory.

VBDOSSTD.BAT

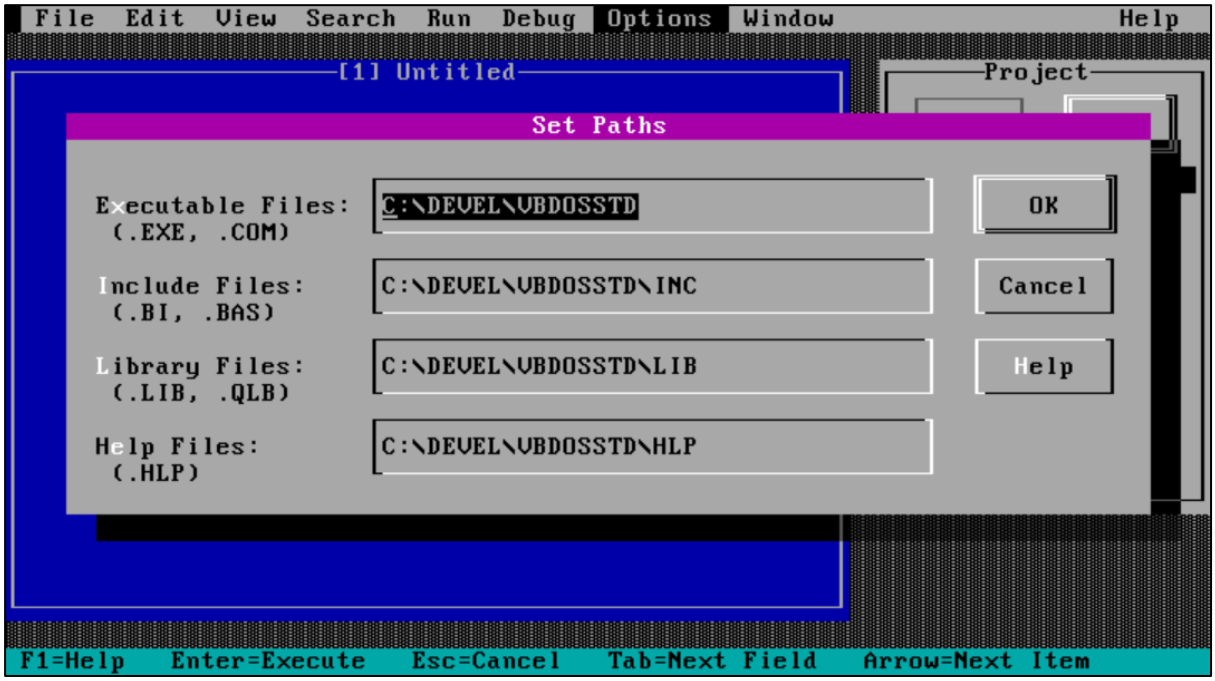
```
@ECHO OFF
REM Launch for Visual basic for MS-DOS Standard edition.
CLS
SET PATH=%path%;C:\DEVEL\VBDOSSTD
CD \DEVEL\VBDOSSTD
CALL VBDOS.EXE
CLS
```

Run VBDOSSTD.BAT



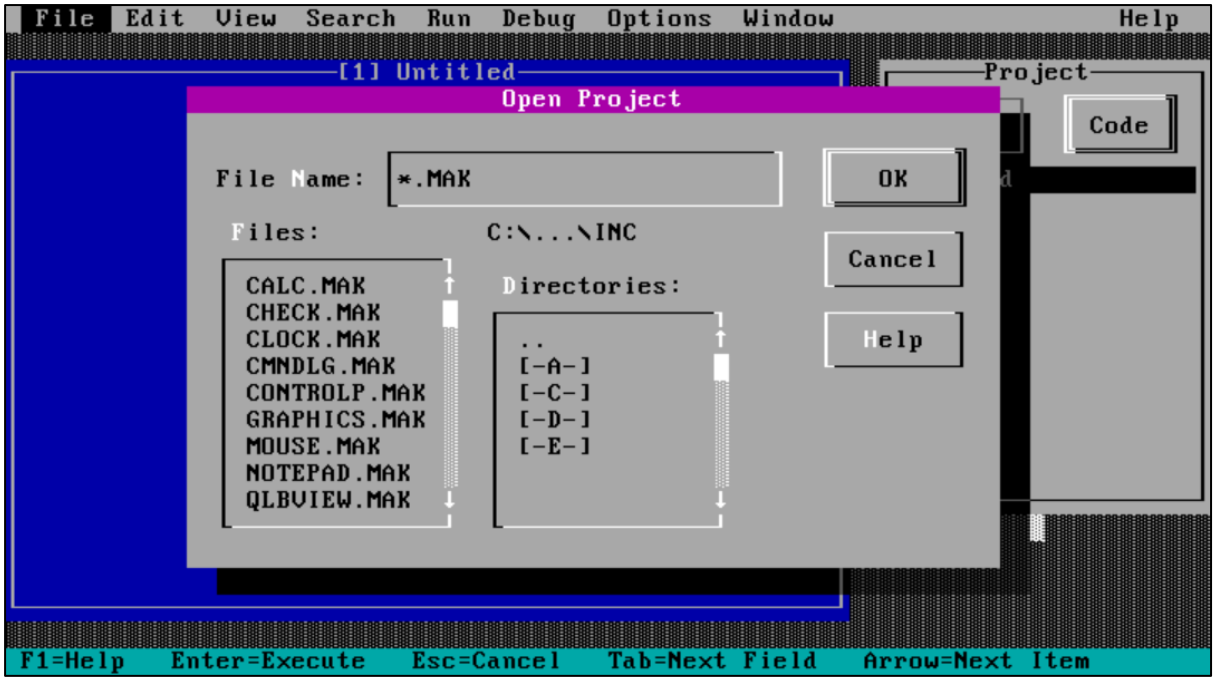
Next double check the paths for the VBDOS config file by selecting Alt + O.

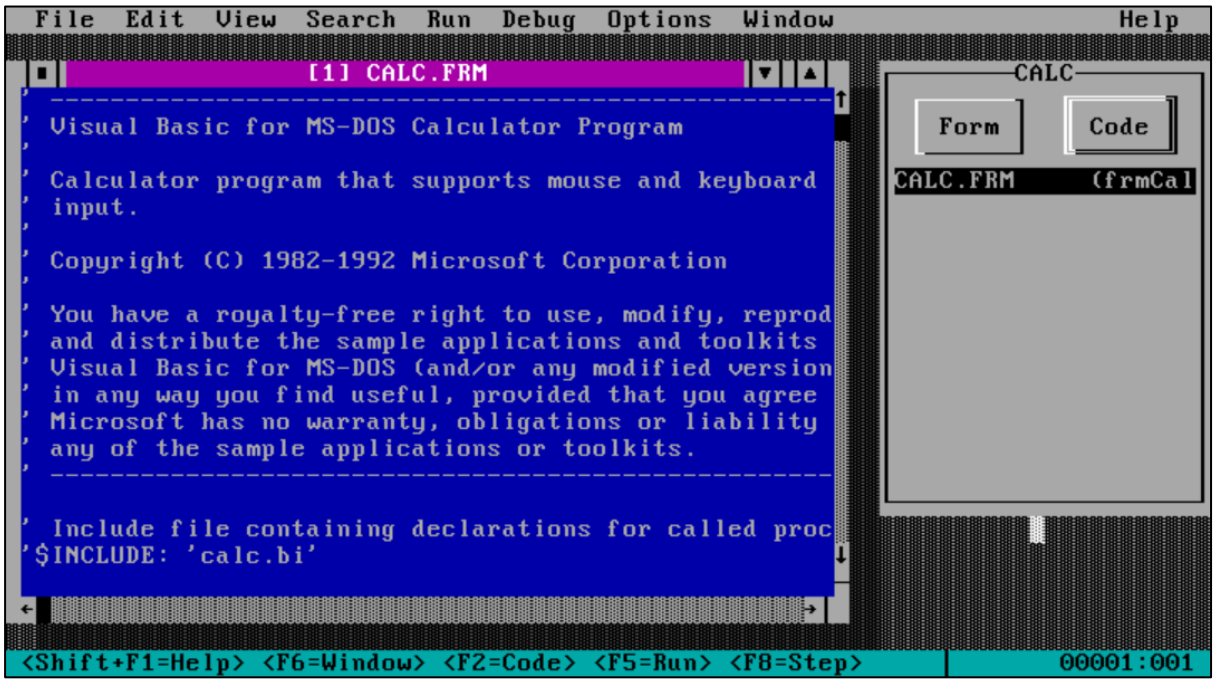
Under “Options -> Set Paths ...” check that the paths are correct. This was set during the install and can always be manually altered at a later time if needed.



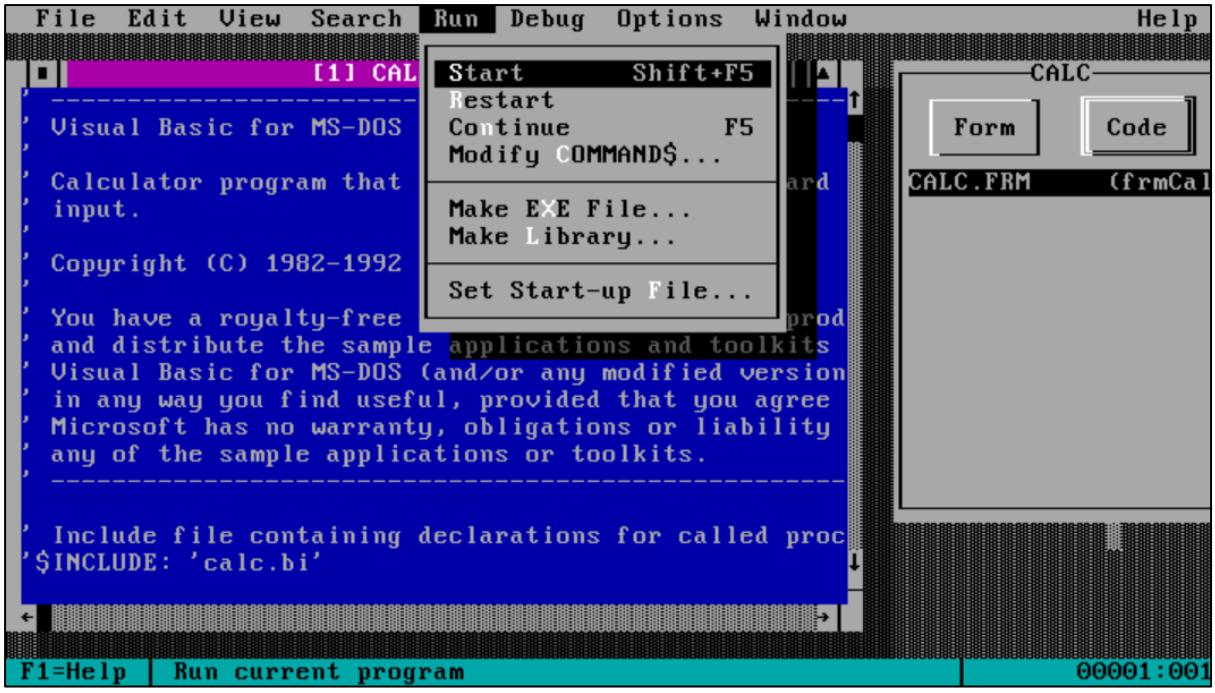
Next load the CALC.MAK example application to test the environment.

“File -> Open Project...” This will be found in the \INC directory.

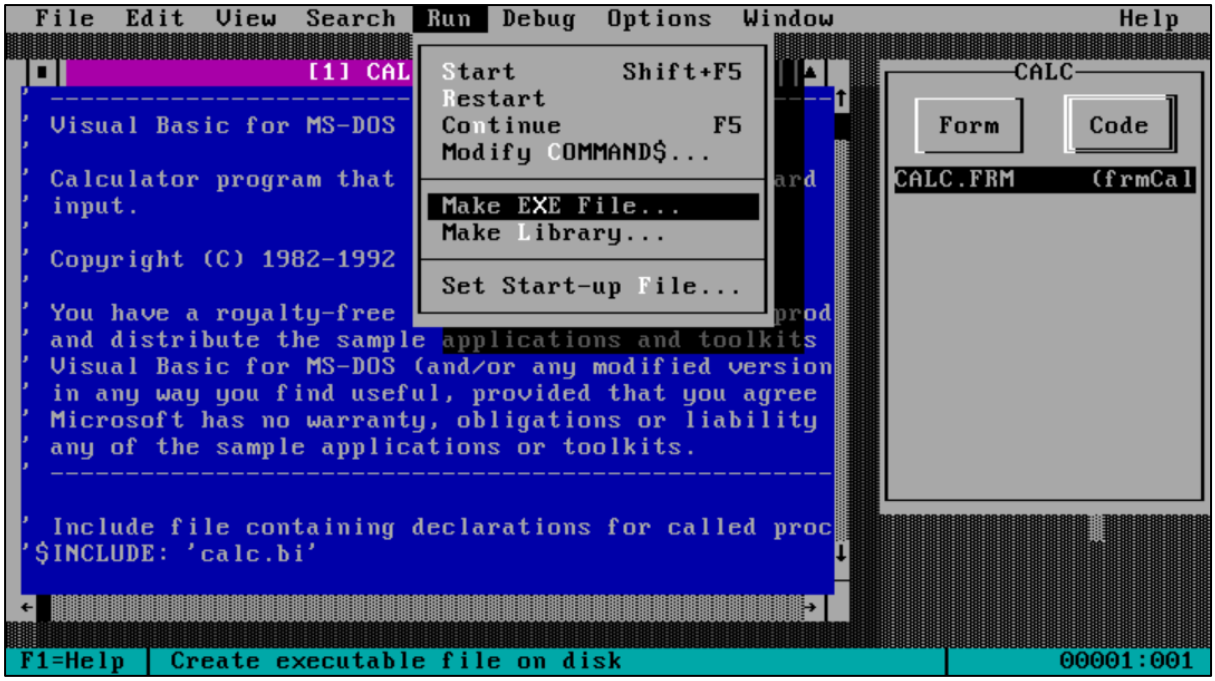




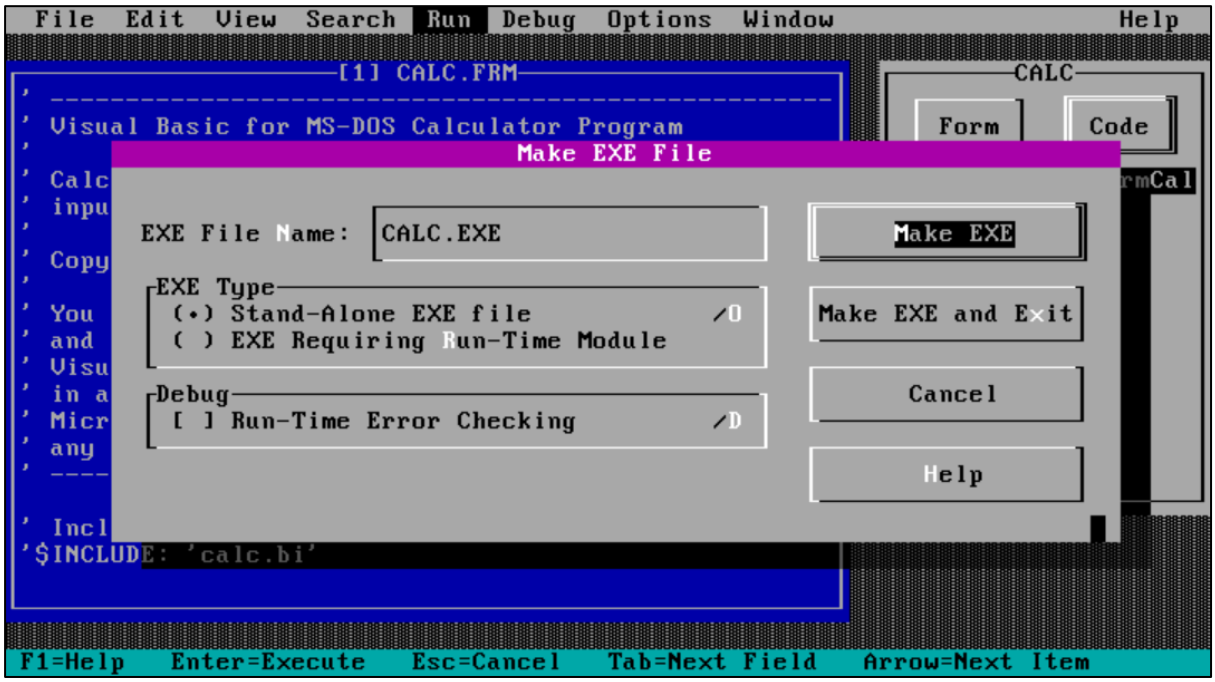
Next select Alt + R “Run -> Start” to test the calculator examples.



Next select Alt + R “Run -> Make EXE File...” to compile the source to an executable file.



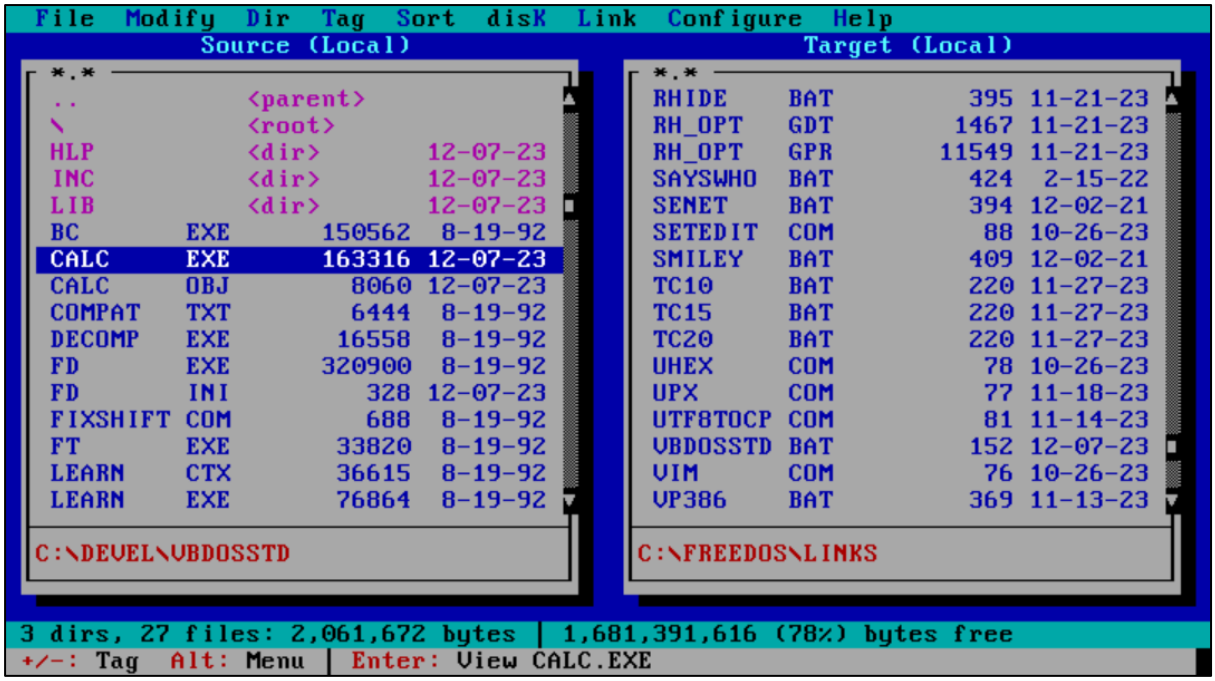
I have selected a “Stand-Alone” EXE so the runtime library will be static compiled into the executable file.



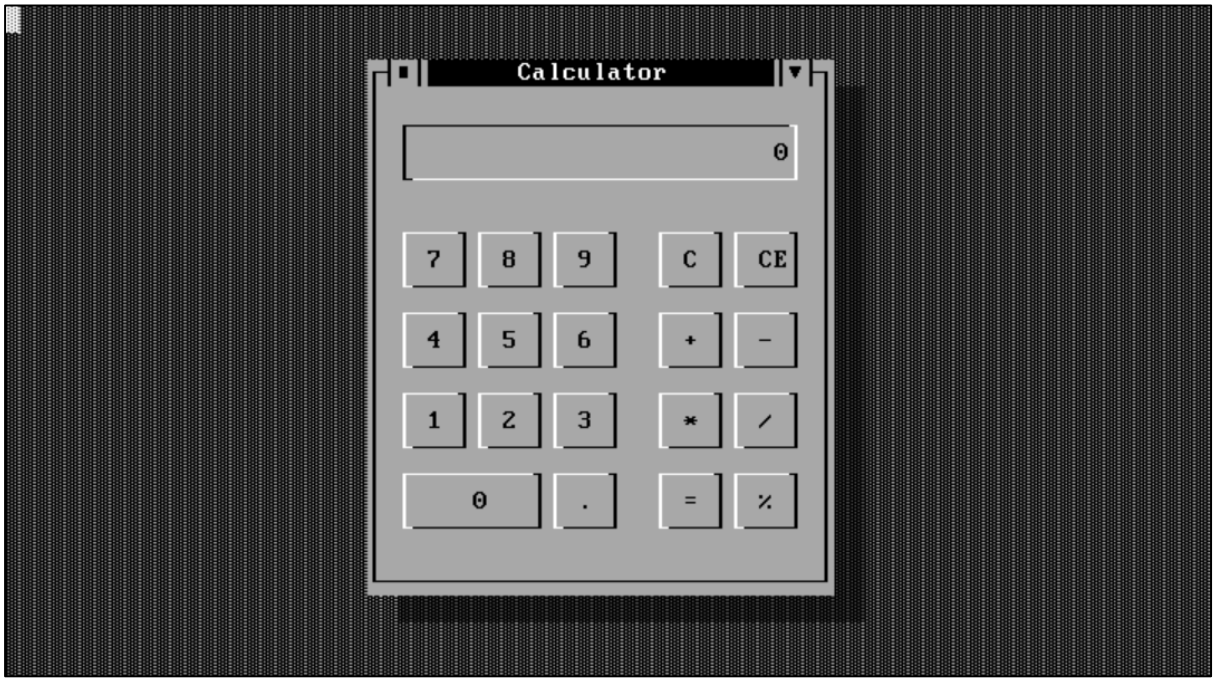
Exit VBDOS or open the MS-DOS Shell and navigate to the \VBDOSSTD root directory
C:\DEVEL\VBDOSSTD

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.



If you are using costa Desktop you can create an icon/link to the VBDOSSTD.BAT file in the \Links directory.

Visual Basic for DOS is now installed and ready for use.

DJGPP (C/C++)

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.

DJGPP can be installed on the system in a number of ways; Via the FDIMPLES Bonus CD package manager, via the FDNPKG manager from the repository or manually 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.

The simplest way to create the base install of DJGPP is via the Bonus CD and FDIMPLES. After installing DJGPP then update with FDNPKG update.

FreeDOS Bonus CD install

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

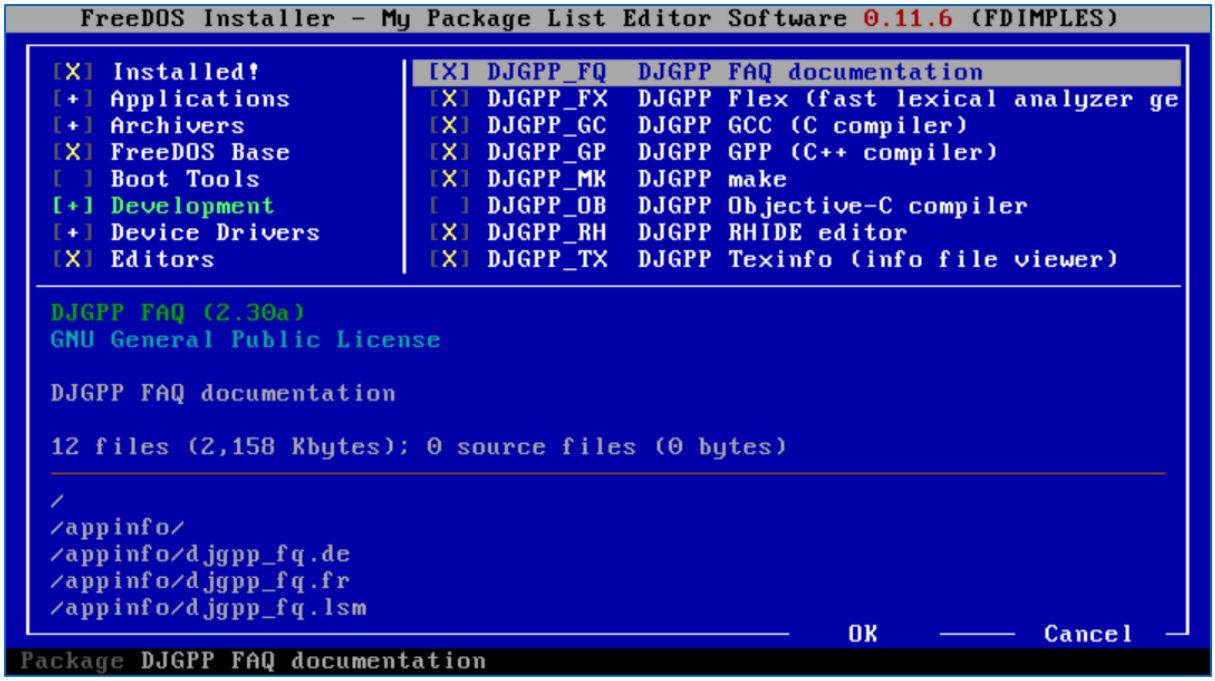
[X]DJGPP_MK

[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 lDebug which is not included on the Bonus CD. This will need to be downloaded and installed to C:\DEVEL\lDEBUG 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

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 DOS system drive C:\DEVEL\DJGPP*.*

Next copy .\APPINFO to the FREEDOS directory C:\FREEDOS\APPINFO*.*

If you are using a different version of DOS locate the .\APPINFO directory and merge the contents.

This will now have the manually installed DJGPP files in the same locations as is done by FDIMPLES from the Bonus CD.

Use FDNPKG package manager from the repository

You can also carry out the manual install from the repository using FDNPKG with the file list above.

Use **FDNPKG search DJGPP** to retrieve the file list and names.

```
Loading http://www.ibiblio.org/pub/micro/pc-stuff/freedos/files/repositories/lat
est/net/...
Loading http://www.ibiblio.org/pub/micro/pc-stuff/freedos/files/repositories/lat
est/sound/...
Loading http://www.ibiblio.org/pub/micro/pc-stuff/freedos/files/repositories/lat
est/unix/...
Loading http://www.ibiblio.org/pub/micro/pc-stuff/freedos/files/repositories/lat
est/util/...

dj_bin - DJGPP binutils: linker, assembler, etc.
dj_bison - DJGPP Bison (a parser generator that is compatible with YACC)
dj_faq - DJGPP FAQ documentation
dj_flex - DJGPP Flex (fast lexical analyzer generator)
dj_gcc - DJGPP GCC (C compiler)
dj_gdb - DJGPP Debugger (GDB)
dj_gpp - DJGPP GPP (C++ compiler)
dj_make - DJGPP make
dj_objc - DJGPP Objective-C compiler
dj_rhide - DJGPP RHIDE editor
dj_txinf - DJGPP Texinfo (info file viewer)
djgpp - DJGPP environment
i16butil - GNU binary utilities, for ia16-elf target (on DJGPP host)
i16gcc - GNU C compiler for ia16-elf target (on 32-bit DJGPP host)
setedit - A programmer's editor for DJGPP, similar to BC++ IDE and RHIDE
C:\FREEDOS>
```

Use the command line install to install from the repository.

fdnpkg install [pkgname]

To update the DJGPP packages to the latest repository versions use **FDNPKG UPDATE [pkgname]**. This can be used to update the files installed from the FD Bonus CD to the latest version.

Setting up from the DJGPP website

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 an http site.

Next select “Build and run programs with DJGPP”

Select the DJGPP base components.

[/]C

[/]C++

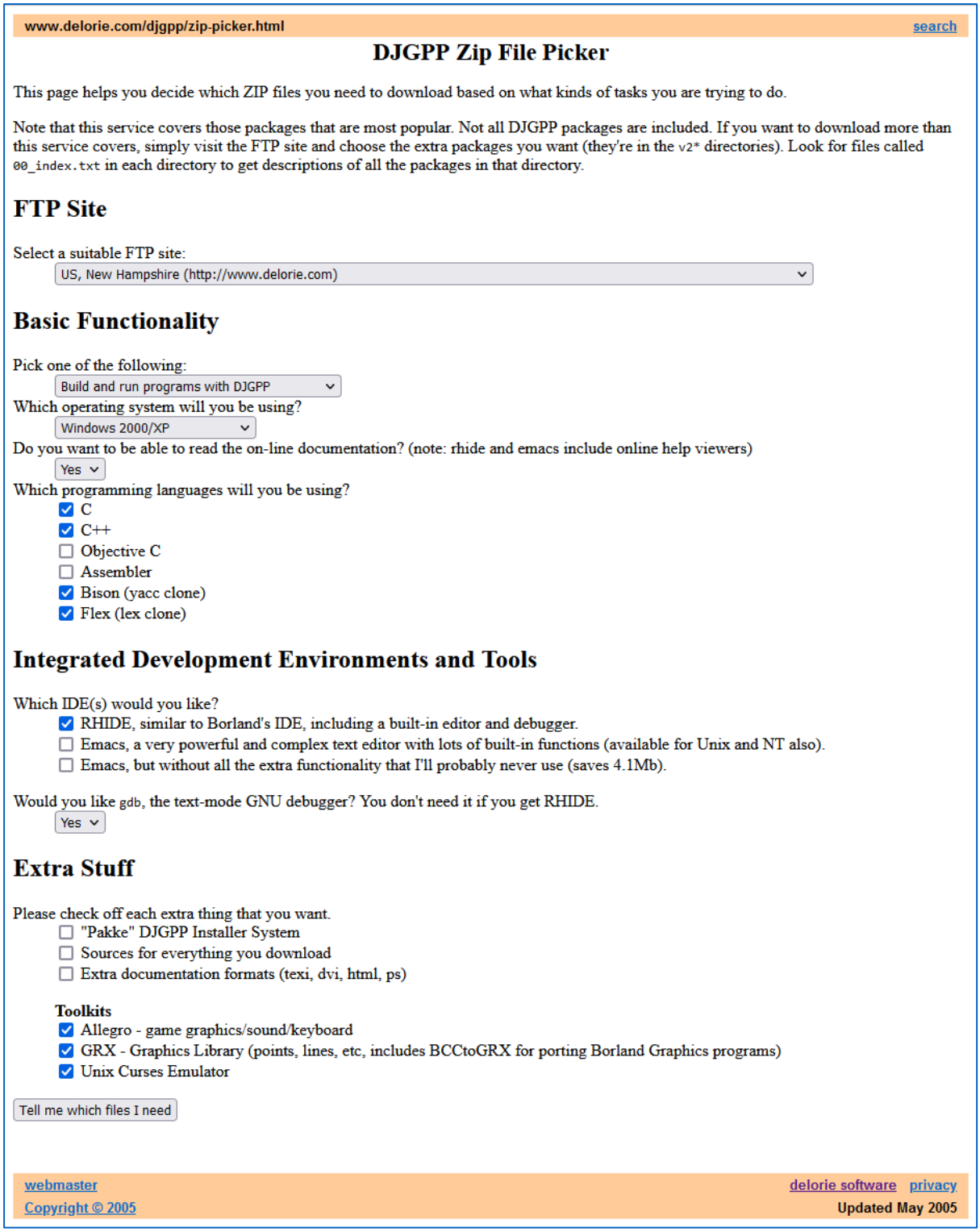
[/]Bison

[/]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 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?
- [x] C
- [x] C++
- [ ] Objective C
- [ ] Assembler
- [x] Bison (yacc clone)
- [x] Flex (lex clone)

Integrated Development Environments and Tools

Which IDE(s) would you like?
- [x] 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
- [x] Allegro - game graphics/sound/keyboard
- [x] GRX - Graphics Library (points, lines, etc, includes BCCtoGRX for porting Borland Graphics programs)
- [x] Unix Curses Emulator

Tell me which files I need

webmaster delorie software privacy
Copyright © 2005 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.

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. It 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:

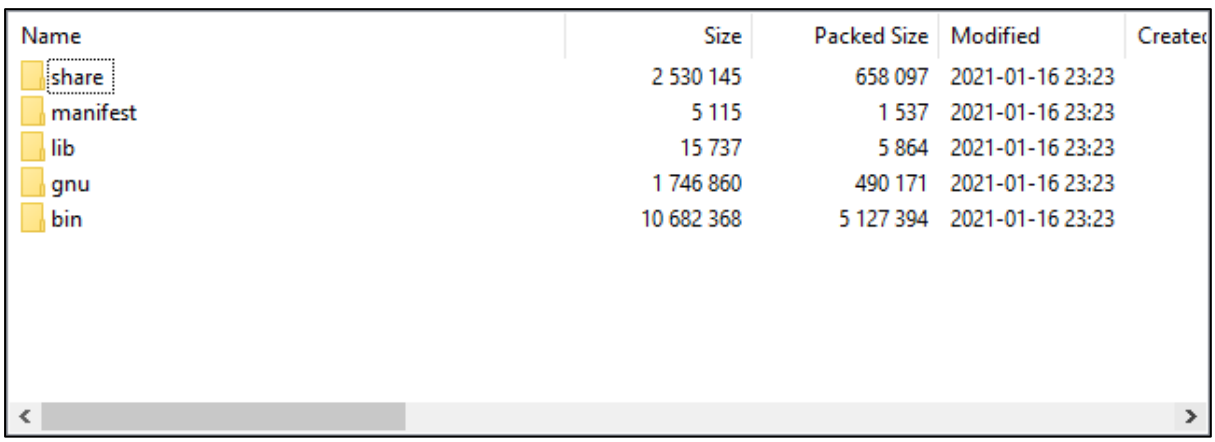
For DJGPP Version 2.01:

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.

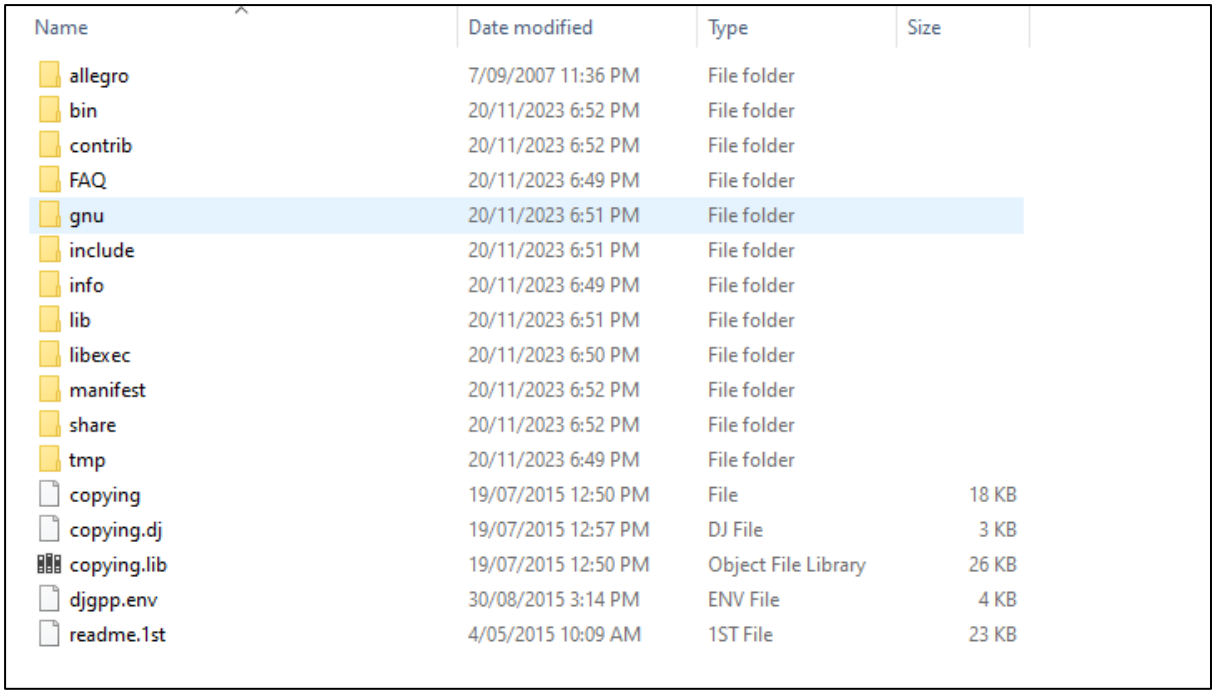
http://www.delorie.com/djgpp/v2faq/faq4_4.html

Unpack all of the zip archives into a common directory such as DJGPP. Keep the directory structure of the internal files.



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 FreeDOS 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

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*.*

I will explain the FED install in the section after RHIDE.

If you already have FED installed as part of the FreeDOS 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 .\LINK 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 and a FED instance. Place all 3 into the .\LINKS 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:\FREEDOS\LINKS directory preferred.
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.
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
```

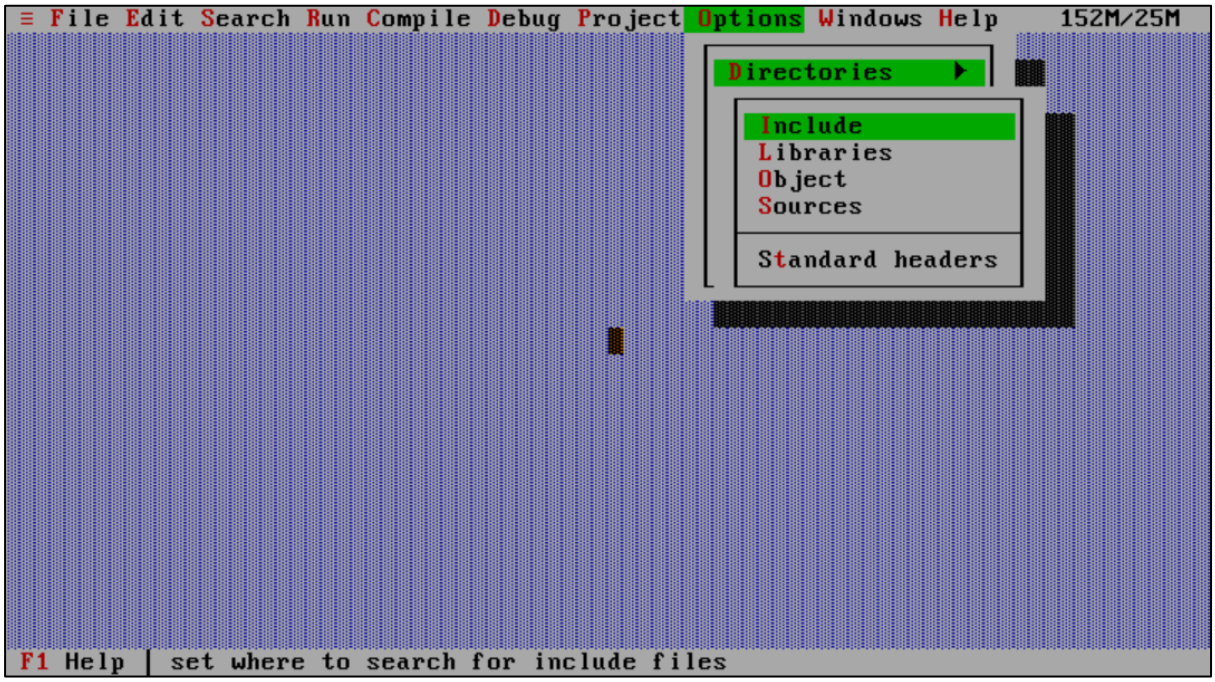
DJFED.BAT (FED Development instance)

```
@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.
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
```

RHIDE

Launch RHIDE from the batch 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.

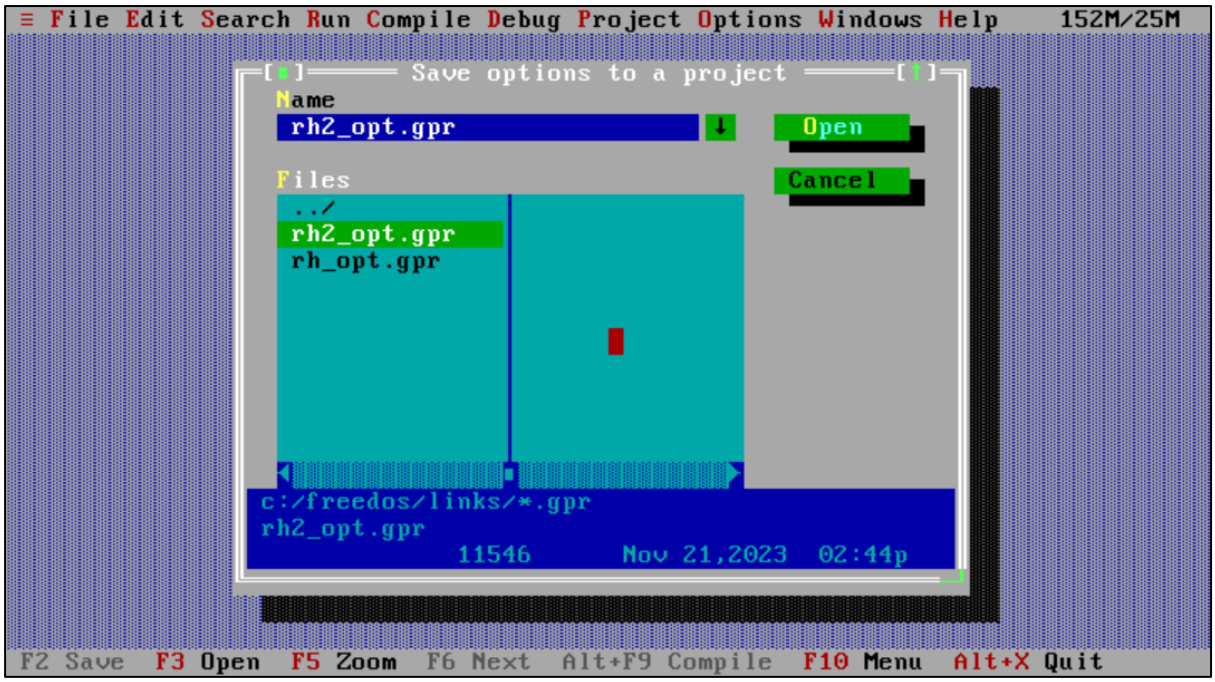
C:\DEVEL\DEVEL\DJGPP\INCLUDE

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

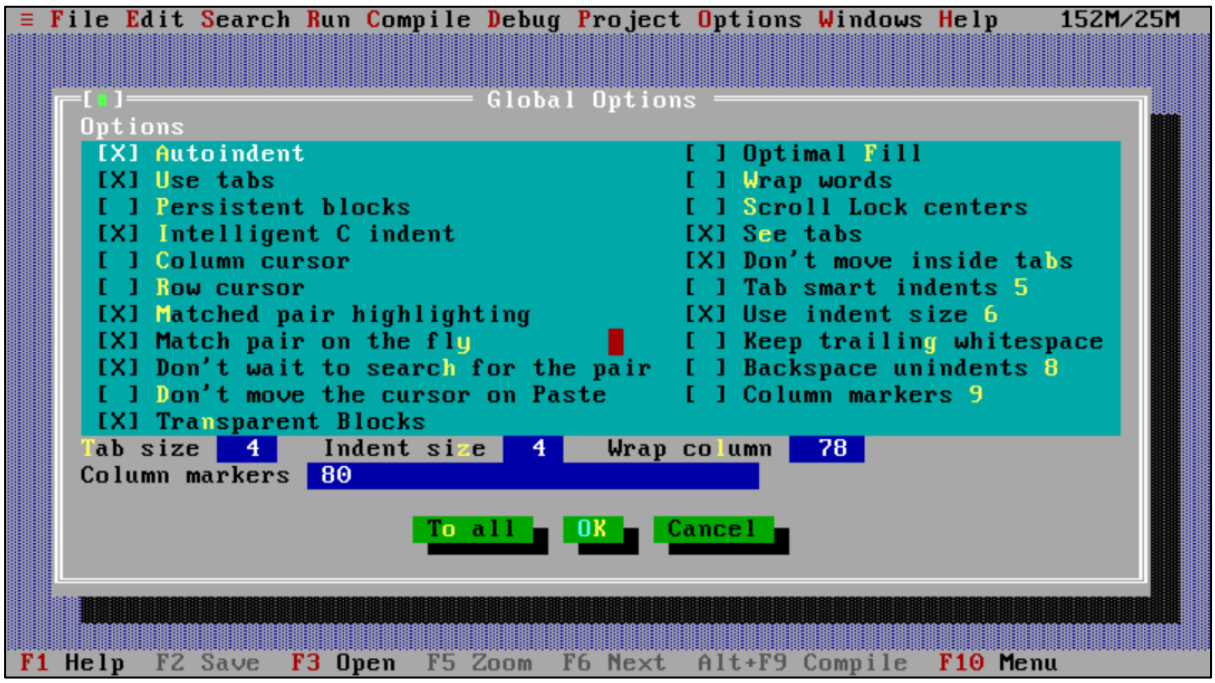
After setting your path or any other options I suggest saving a backup of the RHIDE configs with "Save Options". Take care with the path in the file name. The current directory path is listed in the lower part of the panel.



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.

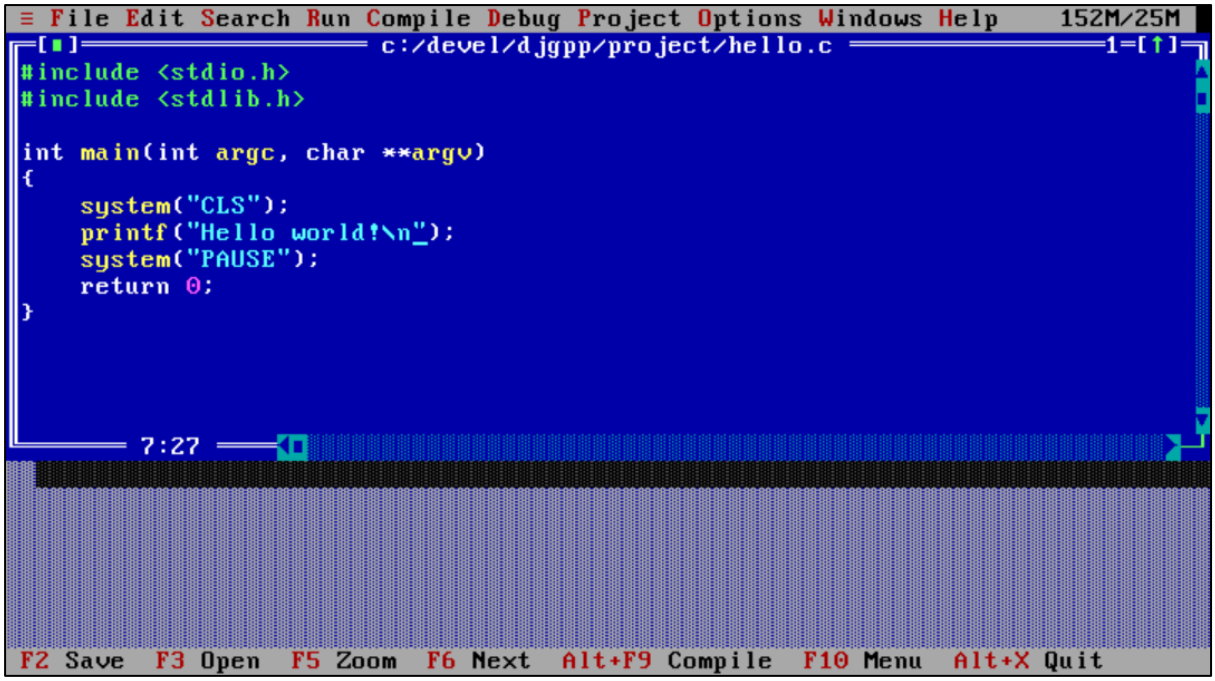


Create a project directory to test the IDE with a “Hello world!”. I will use `C:\DEVEL\DJGPP\PROJECT\hello.c`

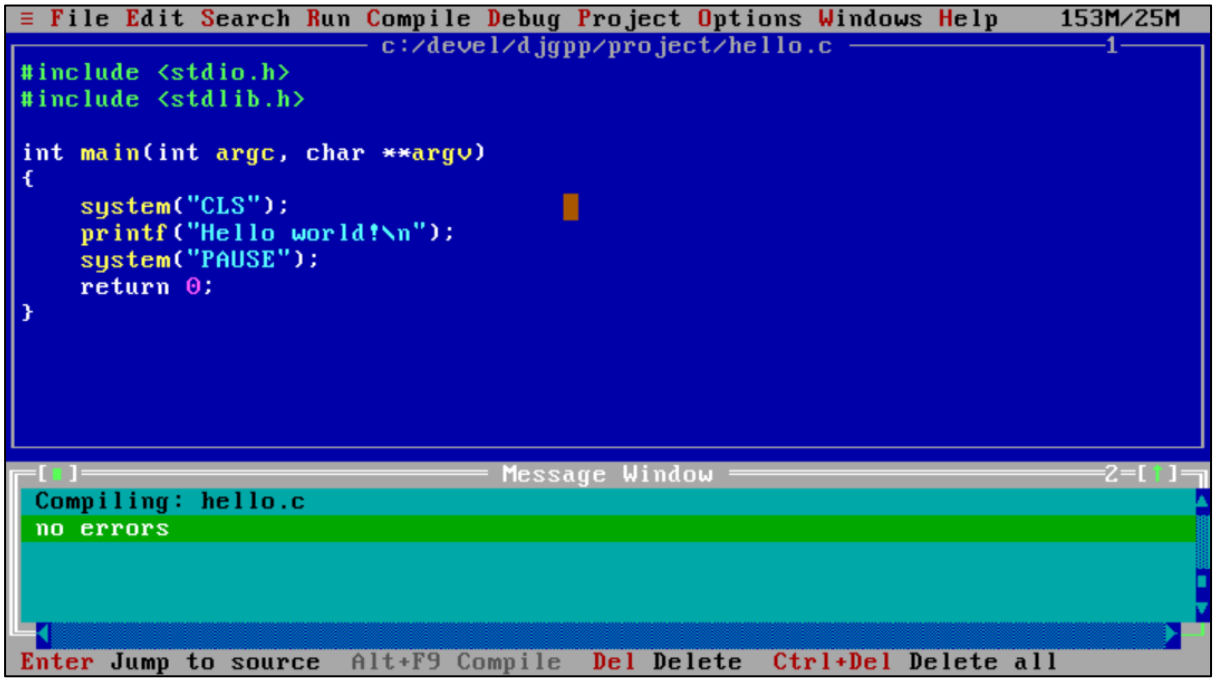
**Note that RHIDE uses Unix paths with forward slash / instead of the DOS backslash **
So inside the editor file dialog you will use C:/DEVEL/DJGPP/PROJECT/hello.c
This can be a point of confusion when working with GNU Unix tool sets.

From RHIDE create a “New” file and save it to the project directory.

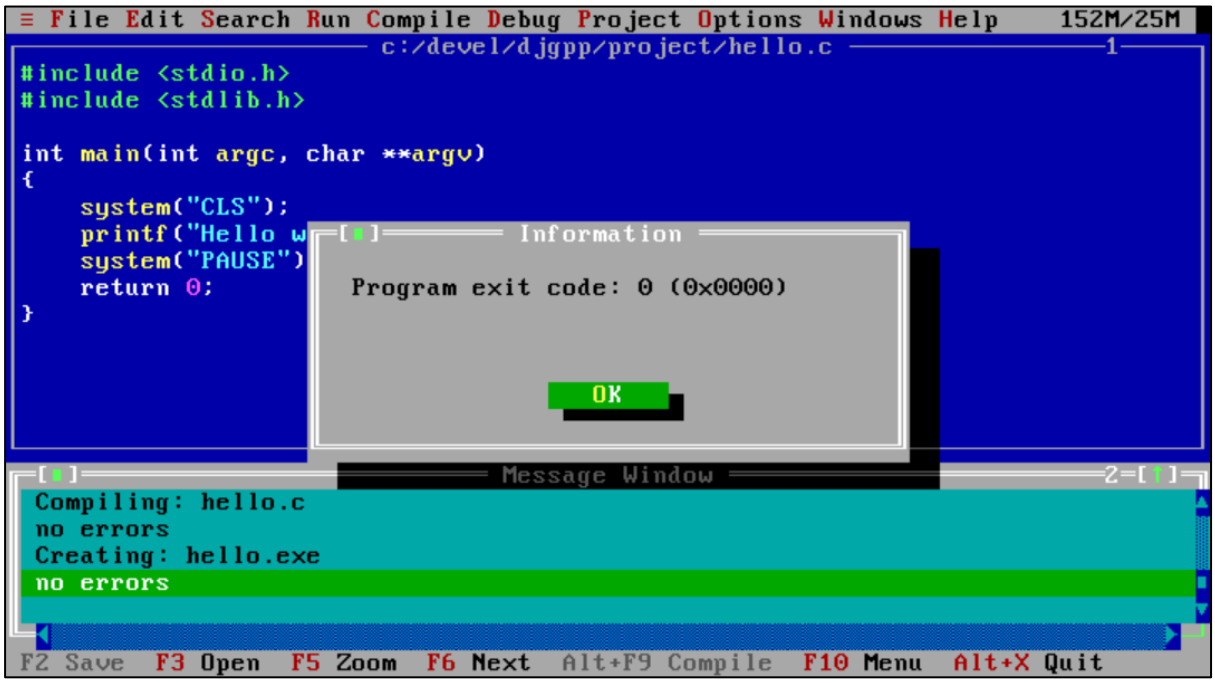
Add you “Hello world” code and then save.



Next select “Compile -> Compile”. The message window will appear in the lower pane.



Next select “Run -> Run” and you should see the Hello world application in the console. Note: Run will also compile the source code.



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 basics 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.

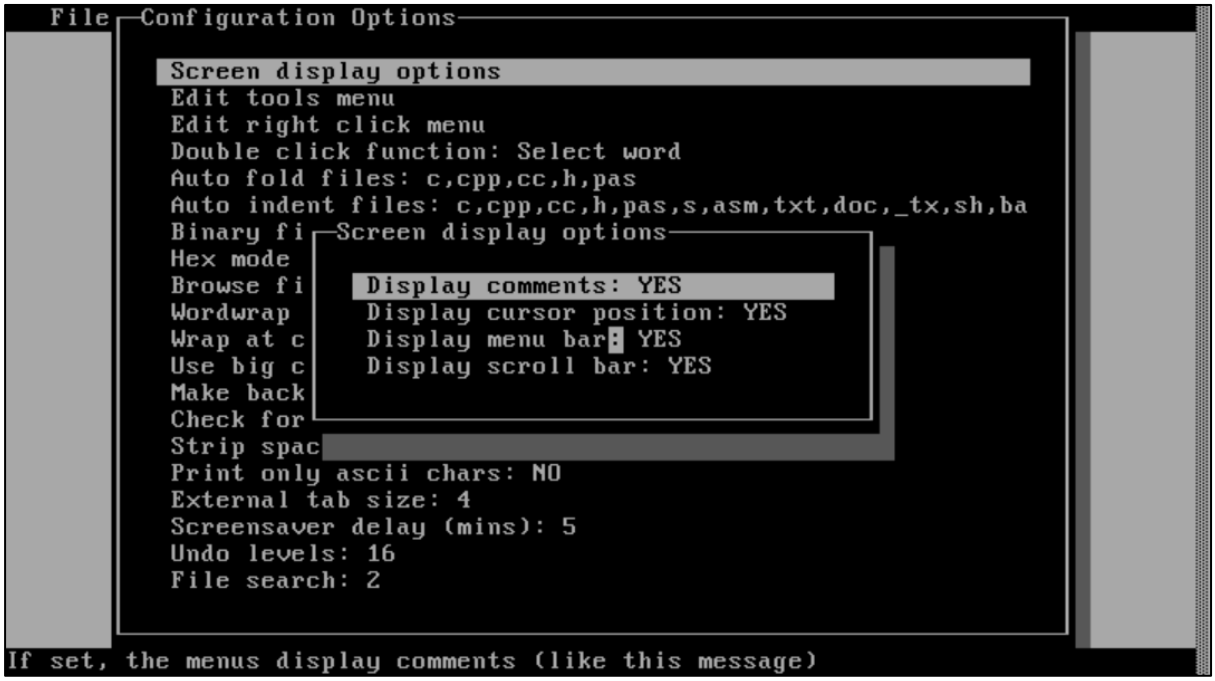
You can use the FED version that exists in the .\APPS directory in FreeDOS if you wish but I prefer to use a separate copy within the Development directory.

Copy C:\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 .\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 .\LINKS to correspond.

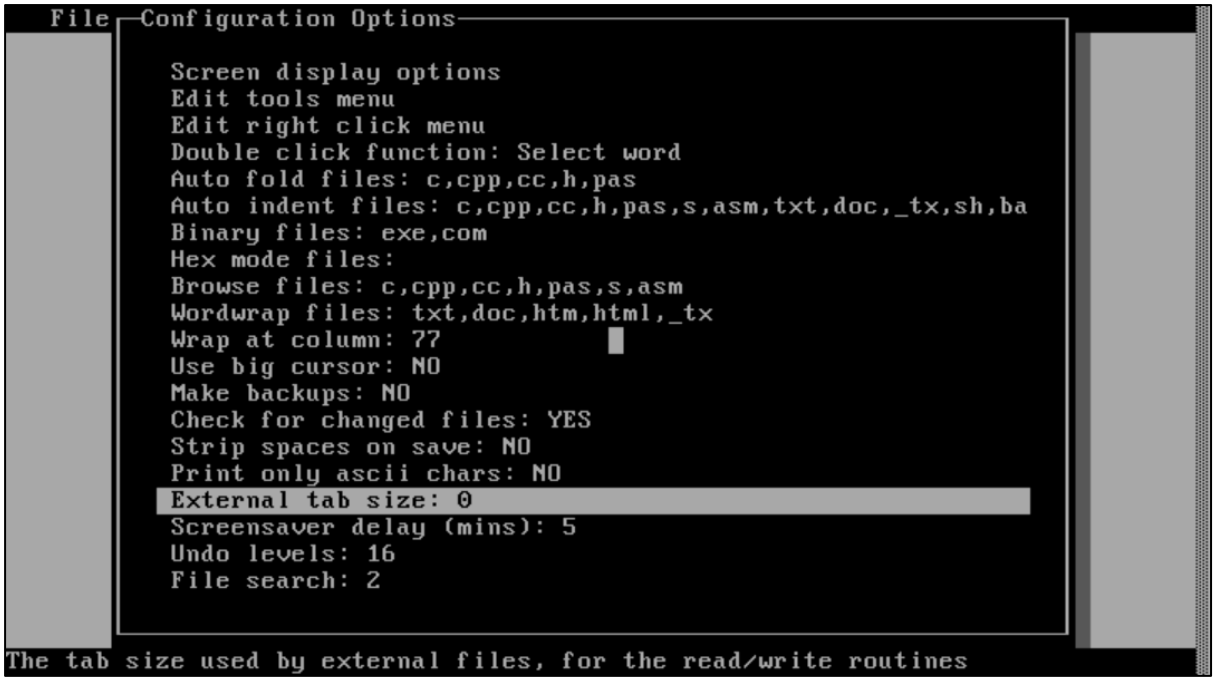
Open FED224 using the batch file in .\LINKS or from the command line with DJFED.BAT

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 menu bar 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.

You will need an extra command line application named “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.bttr-software.de/freesoft/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 DJGCC 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*.*

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
REM https://ftp.sunet.se/mirror/archive/ftp.sunet.se/pub/simtelnet/msdos/batchutl/
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
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
```

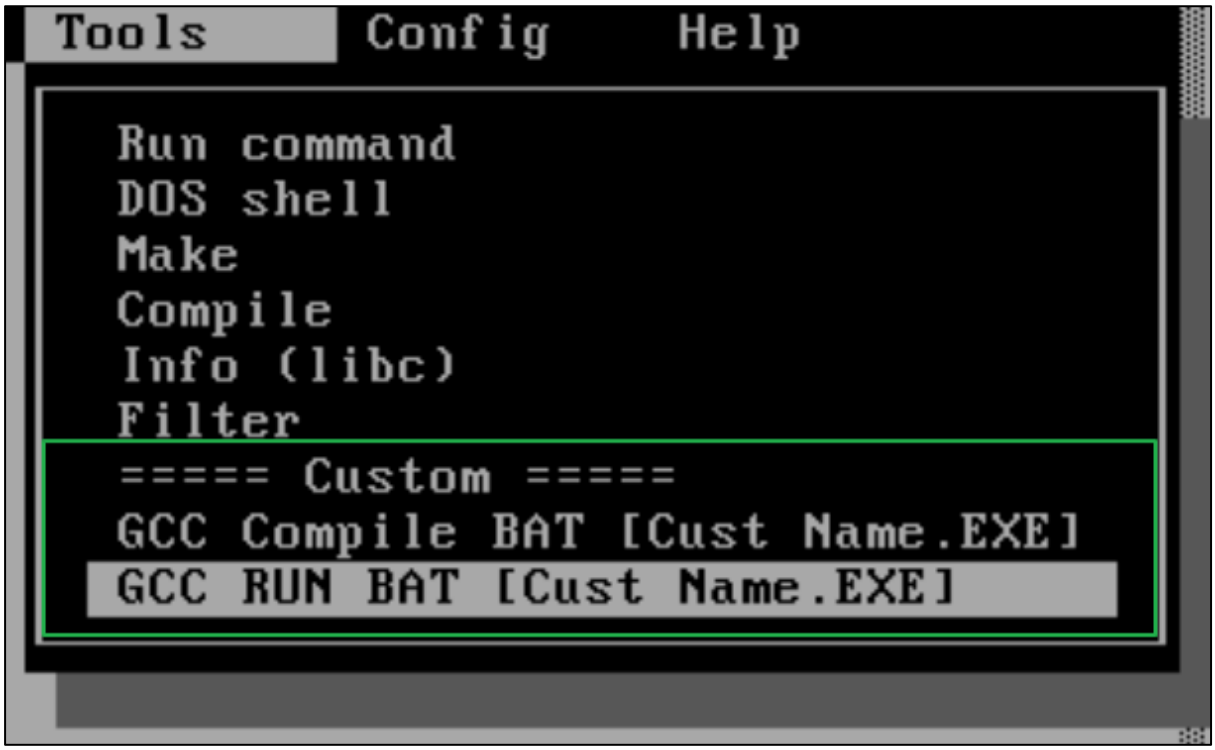
RUN.BAT

```
@echo OFF
REM This is a generic command line runner for FED and 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
REM https://ftp.sunet.se/mirror/archive/ftp.sunet.se/pub/simtelnet/msdos/batchutl/
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
```

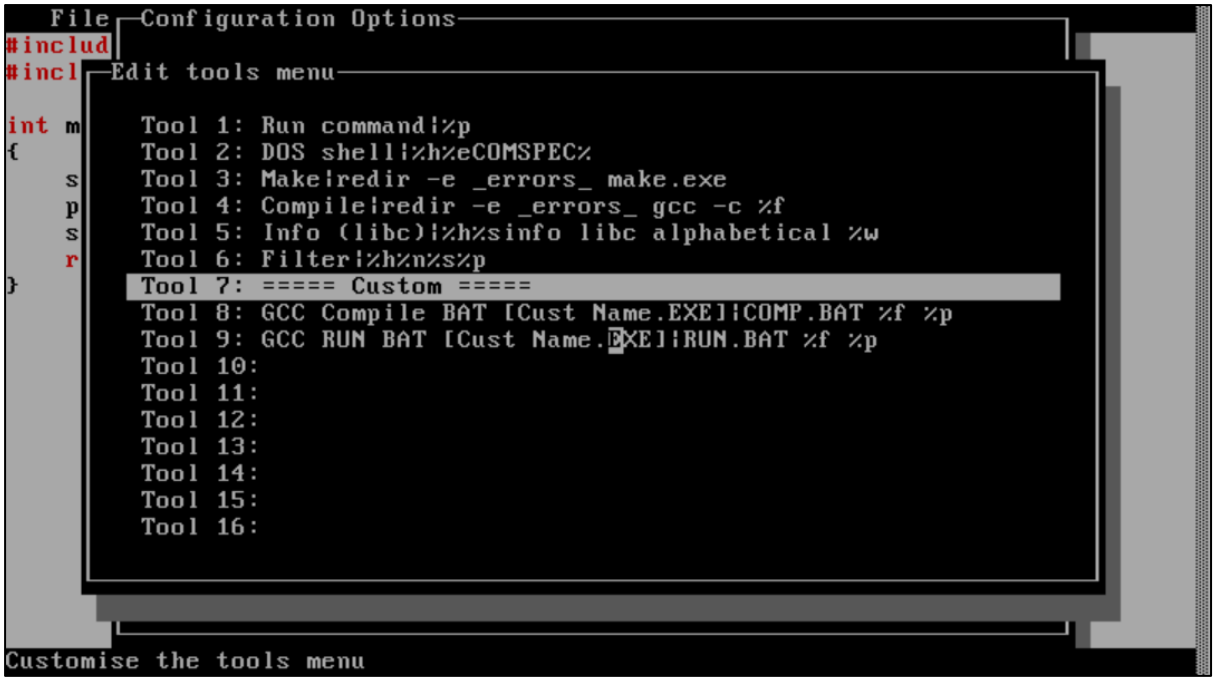
```
goto :ERROR

: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%%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.



Open from the menu "Config ->Options -> Edit tools menu" and add the following 2 lines.



Tool6====== Custom =====

Tool7=**GCC Compile BAT [Cust Name.EXE]|COMP.BAT %f %p**

Tool8=**GCC RUN BAT [Cust Name.EXE]|RUN.BAT %f %p**

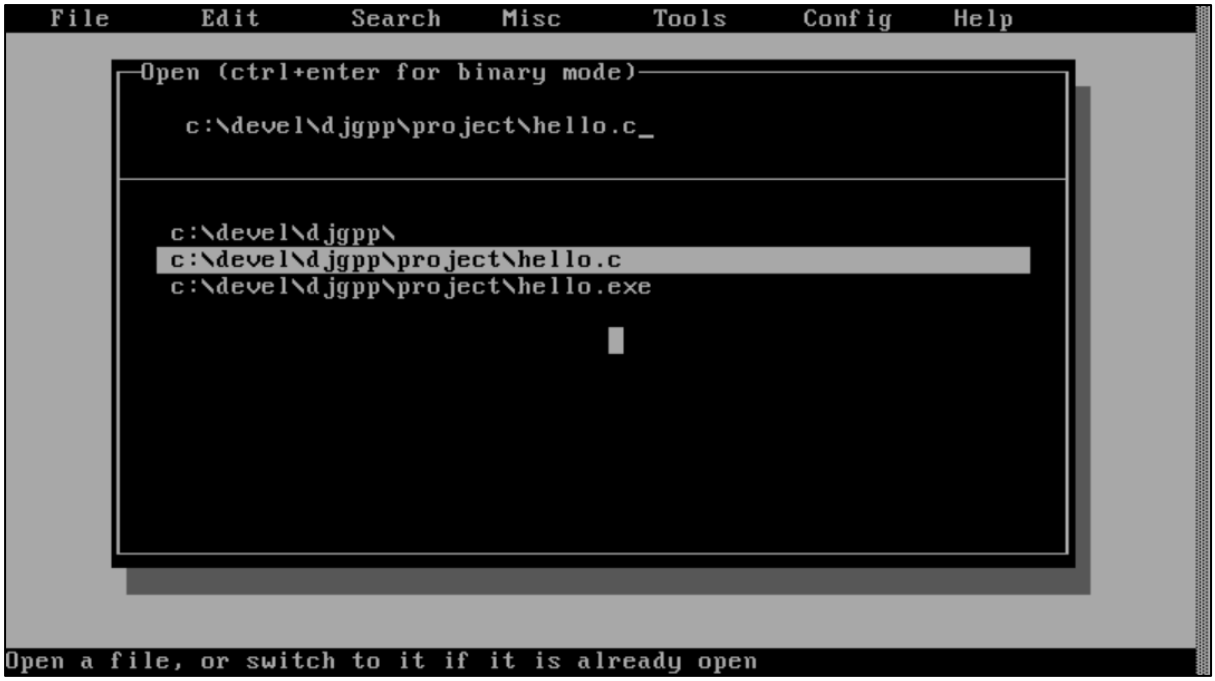
The first part before'|' is the tool description. The section after the '|' is the command line argument **COMP.BAT SourceName UserInput** where f% is the internal FED variable containing the full qualified path and source file name in the editor window and p% is a variable to take a custom output name 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.

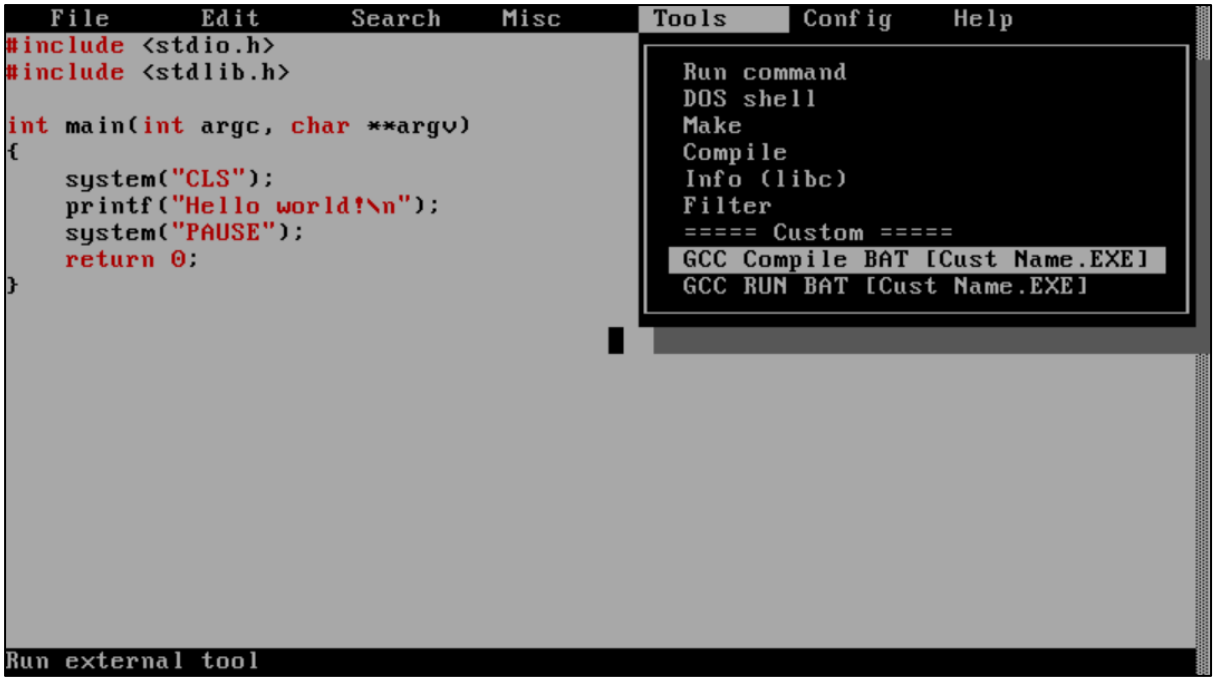
Hint you can copy the FED.CFG to other instances of FED to keep your FED customizations.

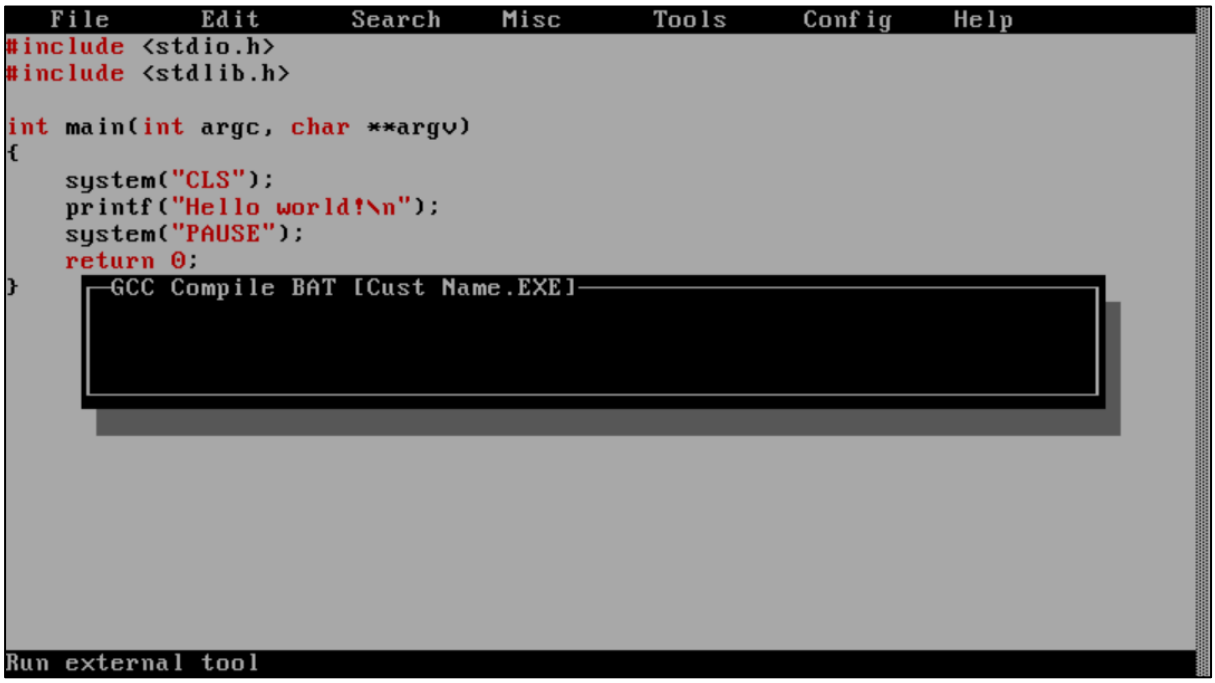
The FED website also makes available some alternative pre made syntax highlighting schemes FED.SYN. It is a good idea to keep a backup of your FED configs in case of mistakes or for recovering you settings.

Open FED for your DJGPP environment and create a "Hello World!" HELLO.C source file.



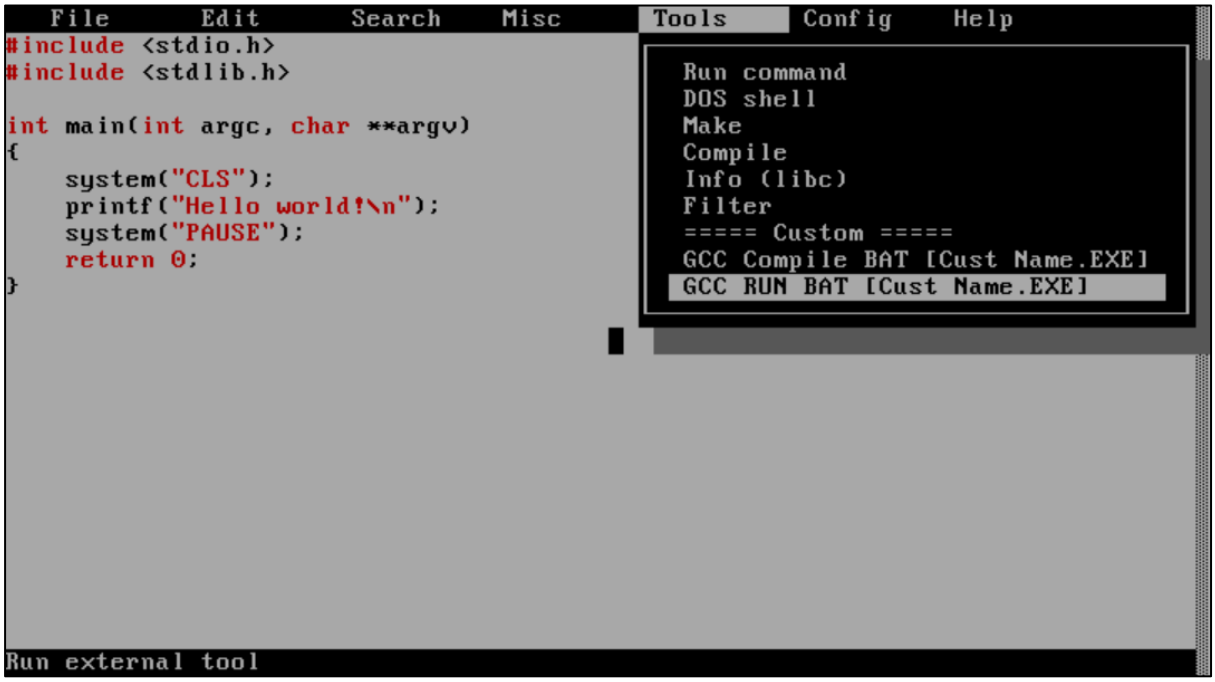
From the menu open “Tools -> ... ” and select your menu entry to COMPILE the source code followed by your menu entry to RUN the source code. I would also test both menu entries using the custom name for the output file. You will find that if you compile with a custom name the variable p% will be auto filled when using the Run option.

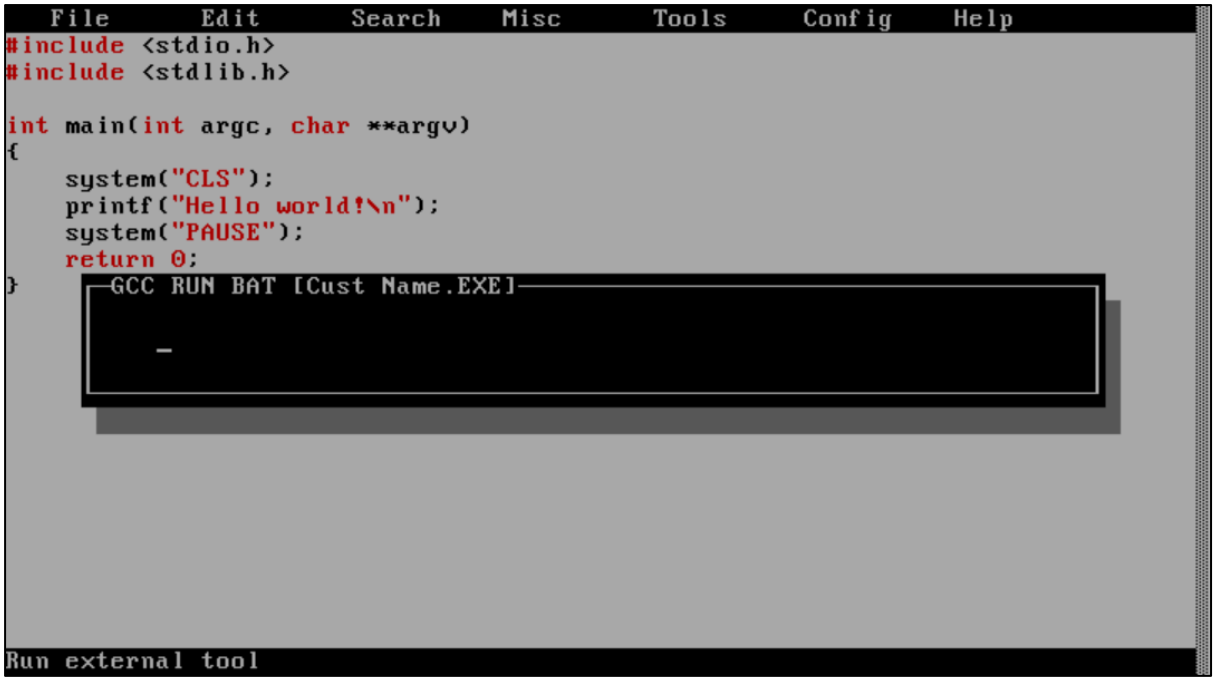

File Edit Search Misc Tools Config Help
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char **argv)
{
system("CLS");
printf("Hello world!\n");
system("PAUSE");
return 0;
}
Run command
DOS shell
Make
Compile
Info (libc)
Filter
===== Custom =====
GCC Compile BAT [Cust Name.EXE]
GCC RUN BAT [Cust Name.EXE]
Run external tool


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]
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/ccVdGWXC.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.



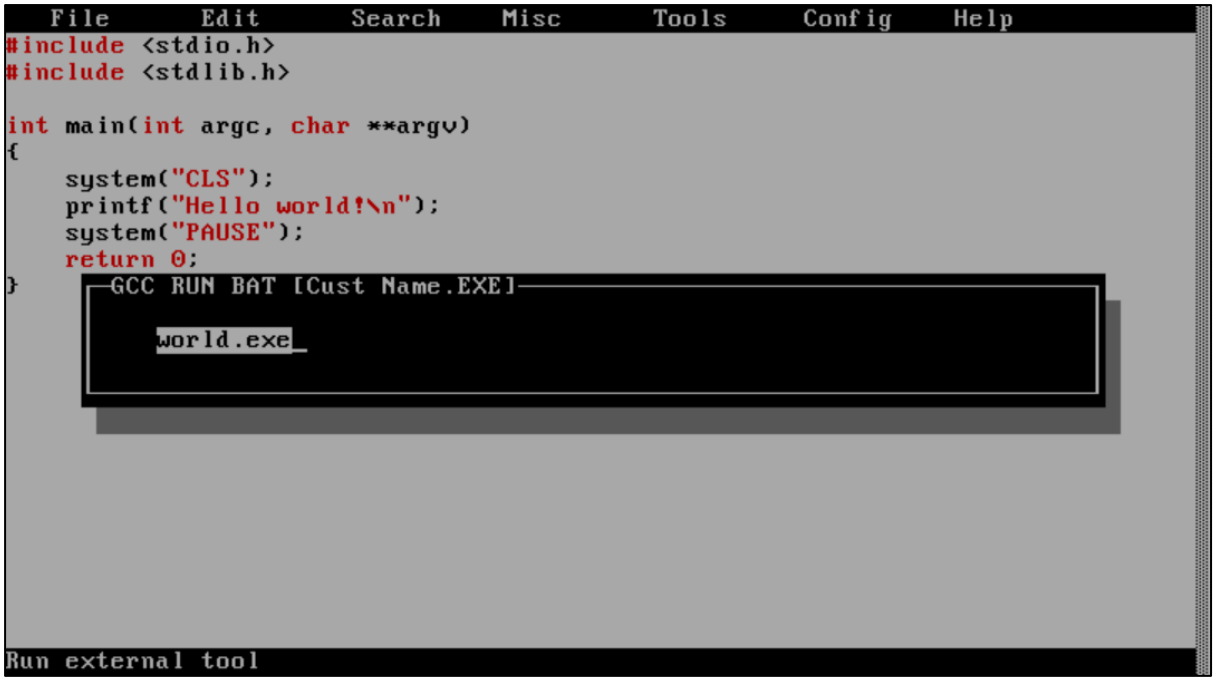


Using the alternative output name (Make sure to add the extension “world.EXE”)...

```
   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.exe' '-mtune
=pentium' '-march=pentium'
 c:/devel/djgpp/bin/as.exe -v -o c:/devel/djgpp/tmp/ccEpli0K.o c:/devel/djgpp/tm
p/ccICzFmA.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\world.exe' '-mtune
=pentium' '-march=pentium'
 c:/devel/djgpp/bin/../libexec/gcc/djgpp/4.71/collect2.exe -o c:\devel\djgpp\pro
ject\world.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/ccEpli0K.o
-lgcc -lc -lgcc
COLLECT_GCC_OPTIONS='-v' '-Wall' '-o' 'c:\devel\djgpp\project\world.exe' '-mtune
=pentium' '-march=pentium'
 c:/devel/djgpp/bin/stubify.exe -v c:\devel\djgpp\project\world.exe
stubify for djgpp V2.X executables, Copyright (C) 1995 DJ Delorie
stubify: c:\devel\djgpp\project\world.exe -> 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...



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.

If you are using Costa desktop you can create an Icon/link to FED .\LINKS\DJFED.BAT in one of the 5 desktops for convenience.

Using Libraries

The DJGPP contains many additional libraries and there are many more in 3rd part repositories.

You will find some common source and pre-compiled LIB files at:
https://www.delorie.com/pub/djgpp/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 "libpdcurses.a" will be truncated to "libpdc~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 sites 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 WinBGIm, 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.

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/tcinstall.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
      COx.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 files 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
|  SETENVP.ASM
|  TC.EXE
|  TC10.BAT
|  TCC.EXE
|  TCCONFIG.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
|      C0C.OBJ
|      C0H.OBJ
|      C0L.OBJ
|      C0M.OBJ
|      C0S.OBJ
|      C0T.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
|   |   C0C.OBJ
|   |   C0H.OBJ
|   |   C0L.OBJ
|   |   C0M.OBJ
|   |   C0S.OBJ
|   |   C0T.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
|           SETENVP.ASM
|           STARTUP.ARC
|           TC1p.html
|           WILDARGS.OBJ
|
\---UTILITY
        CINSTXFR.EXE
```

```
CPP.EXE
GREP.COM
MAKE.EXE
OBJXREF.COM
TC1p.html
TCC.EXE
TCCONFIG.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.

TCCONFIG.TC is not created by default.

Check that your FDCONFIG.SYS file has the following 2 lines. They are usually set by default in FreeDOS V1.3

```
!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 they would be found in the config file.

Don't forget to name the TC directory according to each version.

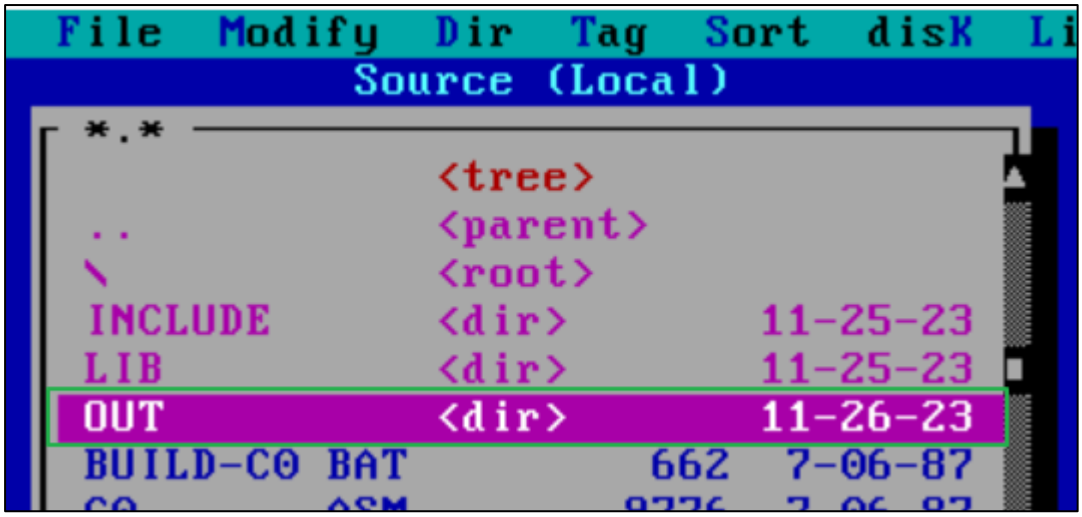
TC10.BAT

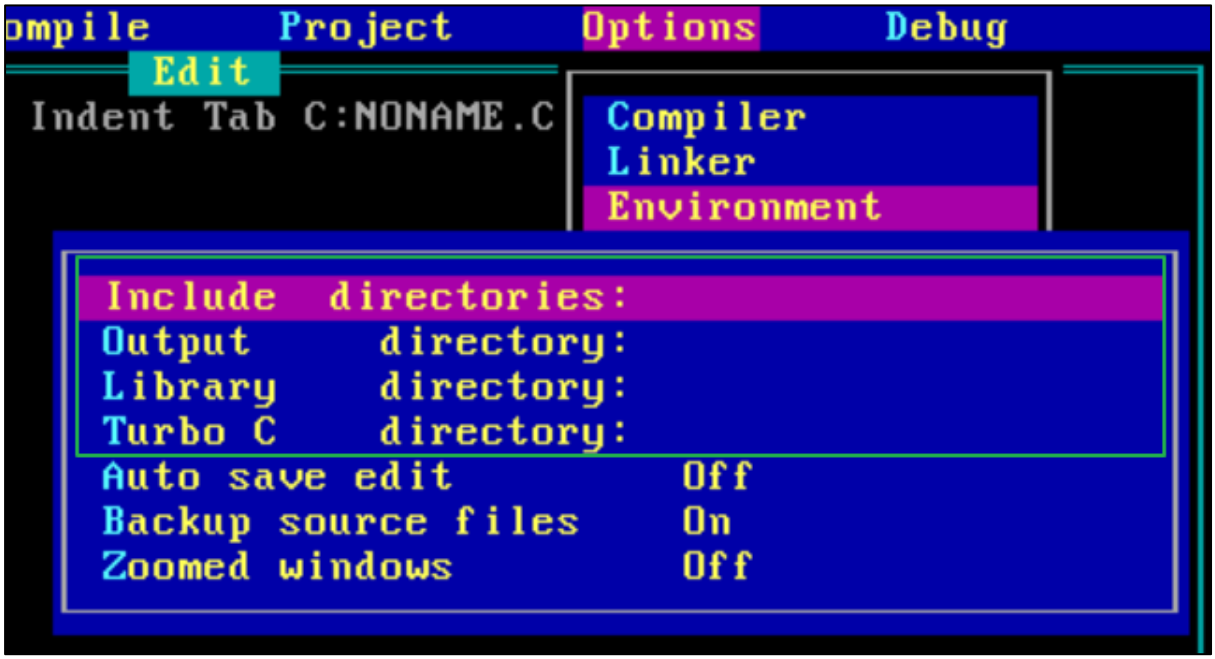
```
@ECHO OFF
CLS
set PATH=%path%;C:\DEVEL\TC10;C:\DEVEL\TC10\INCLUDE;C:\DEVEL\TC10\LIB
set INCLUDE=C:\DEVEL\TC10\INCLUDE
set LIB=C:\DEVEL\TC10\LIB
set CLASSPATH=C:\DEVEL\TC10\LIB
CD \DEVEL\TC10
call C:\DEVEL\TC10\TC.EXE
CLS
```

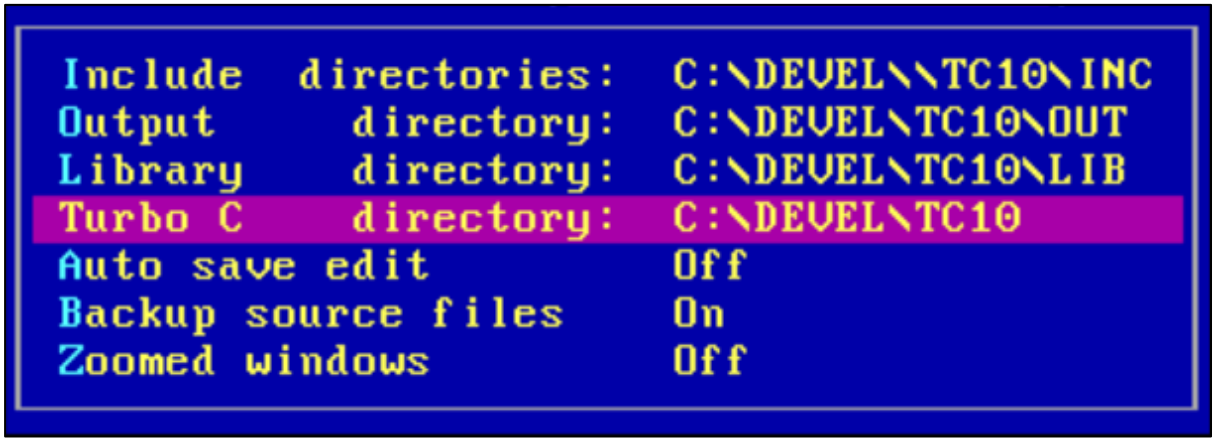
Launch TC.exe from the batch file TC.BAT

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 compiled 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.

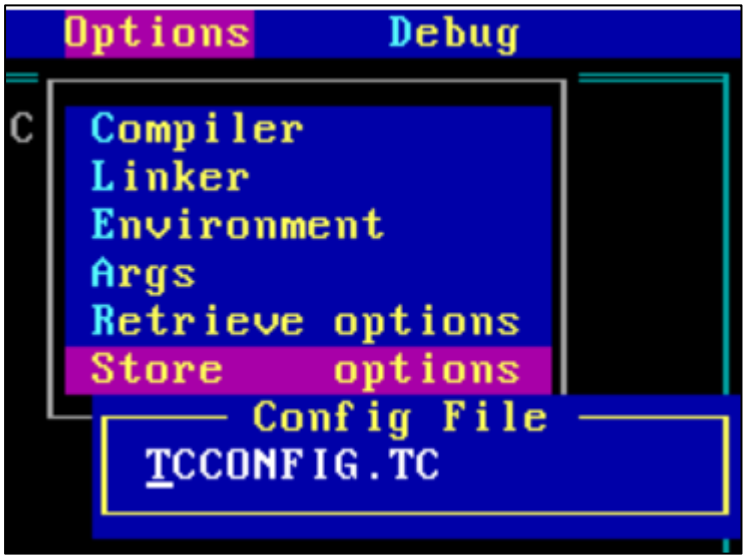




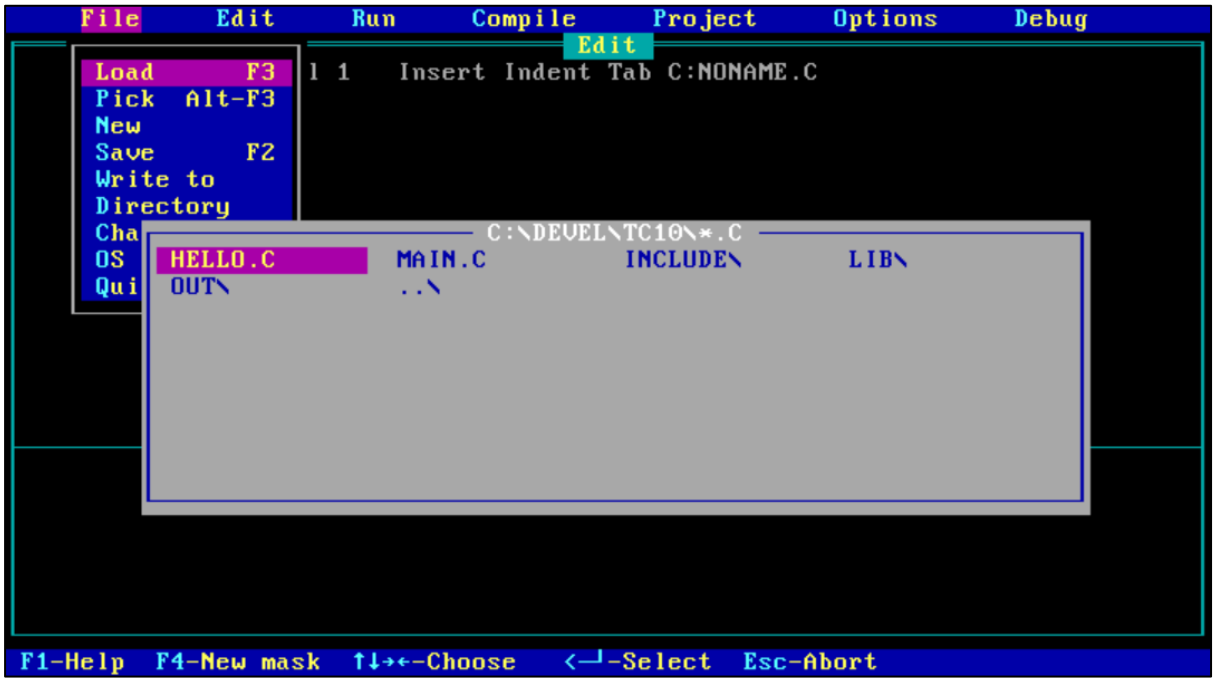


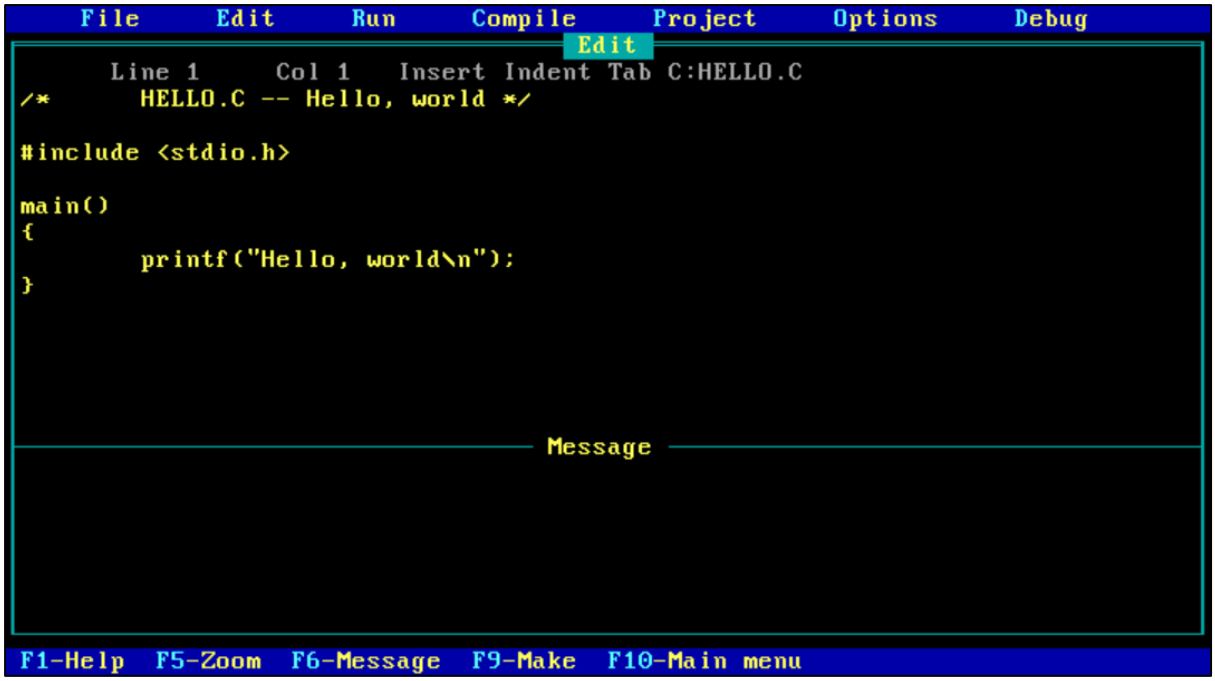
Also take note of the TAB size (4 recommended) and other editor options.

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.

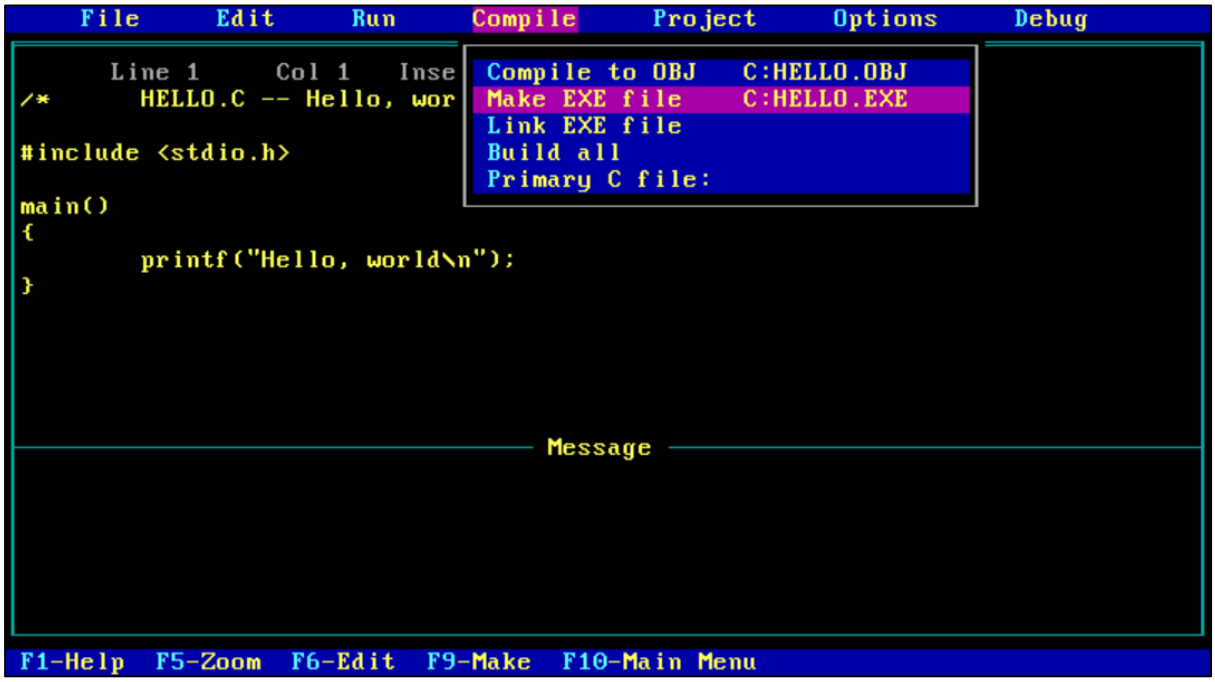


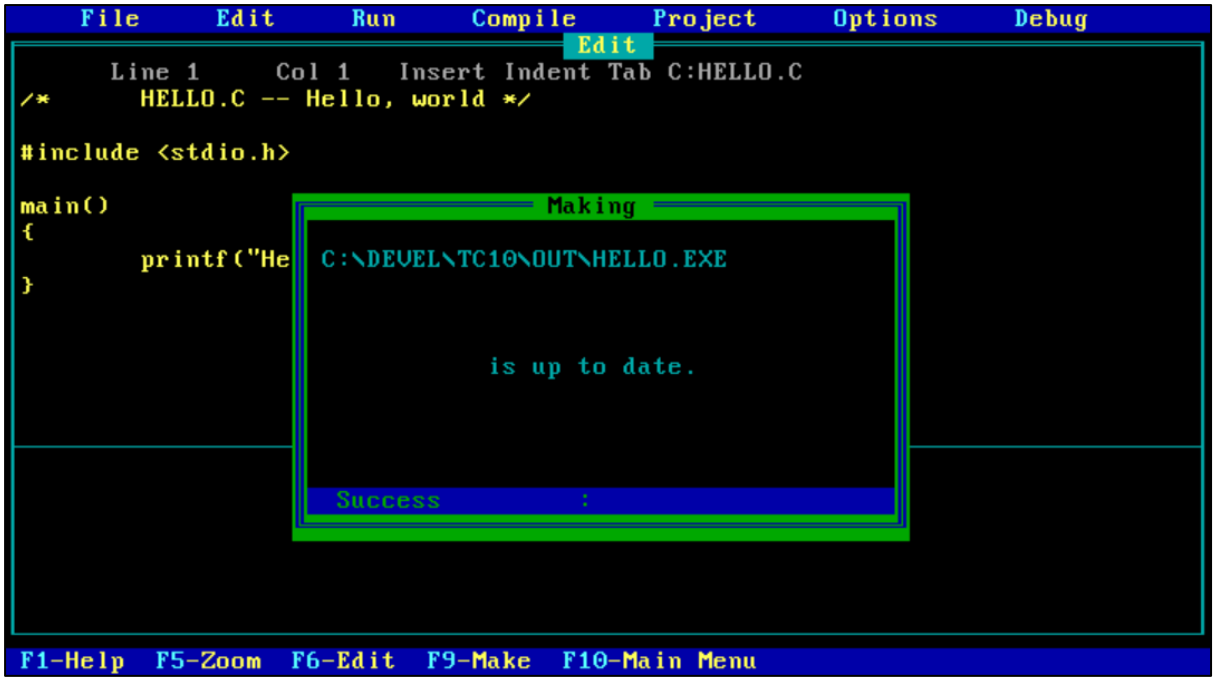
Next use “Alt + F” to open “File -> Load” and select the HELLO.C example in the root directory.

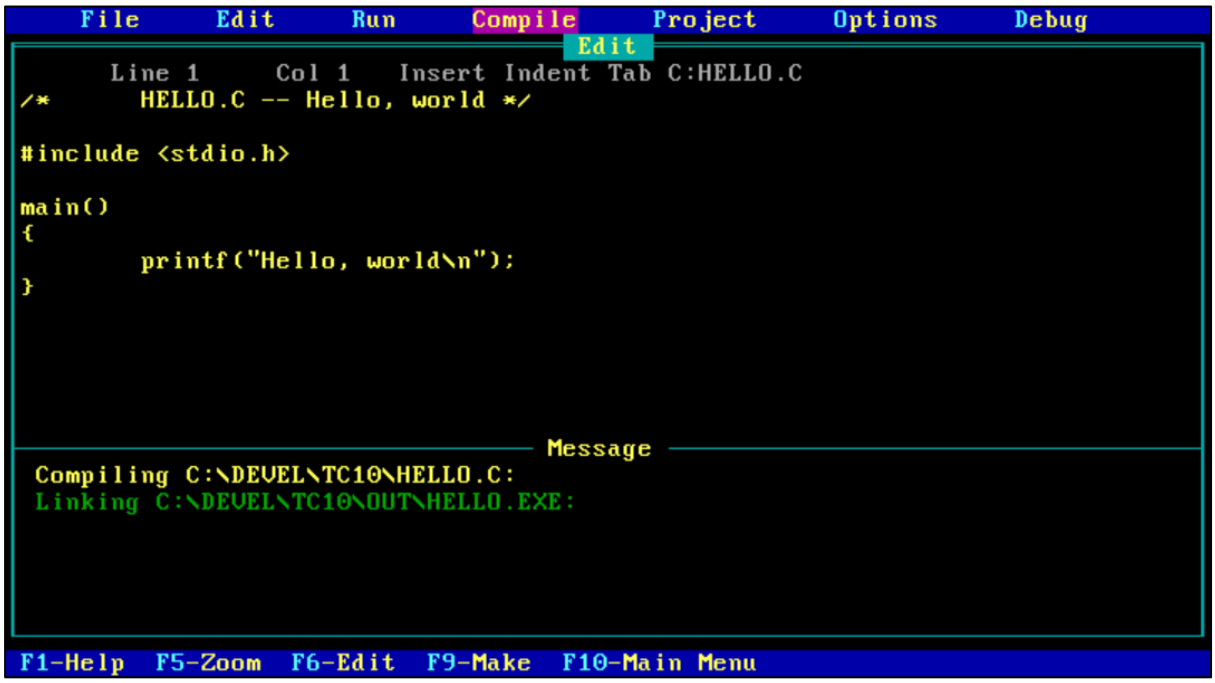




Next use “Alt + C” to select “Compile -> Make EXE file”, and then Enter.

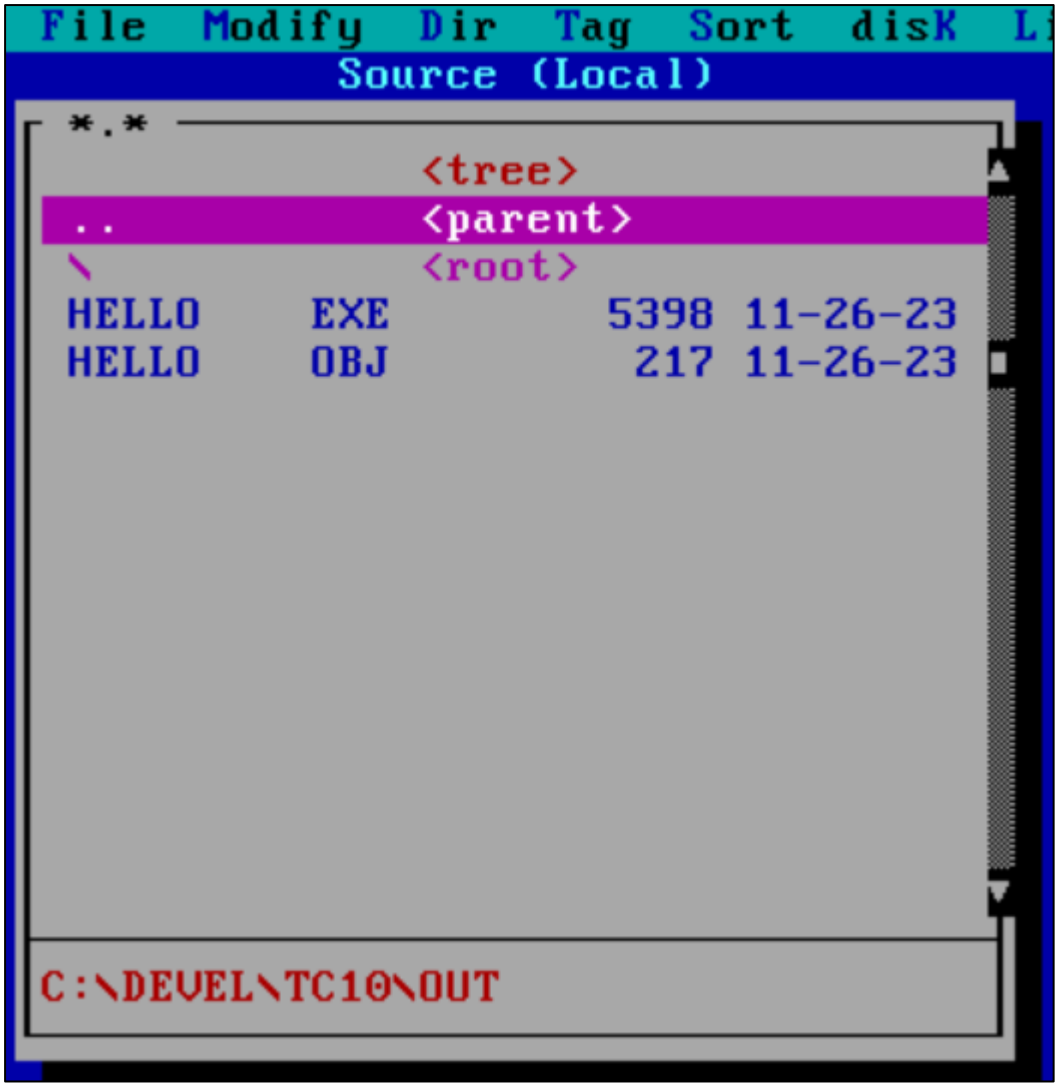






Next select “Alt + R” to Run the compiled executable “HELLO.EXE”

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.


File Modify Dir Tag Sort disK Li
Source (Local)
.
<tree>
..
<parent>
\
<root>
HELLO EXE 5398 11-26-23
HELLO OBJ 217 11-26-23
C:\DEVEL\TC10\OUT


Hello, world
Press any key ...

You will need to add extra linker options (Options -> Args) to link 3rd 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.

I have also added a link to my Costa desktop for convenience.

Turbo C 1.x did not ship the GBI graphics.h but if you look at the "tc.zip from https://eecs.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 projects. 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.

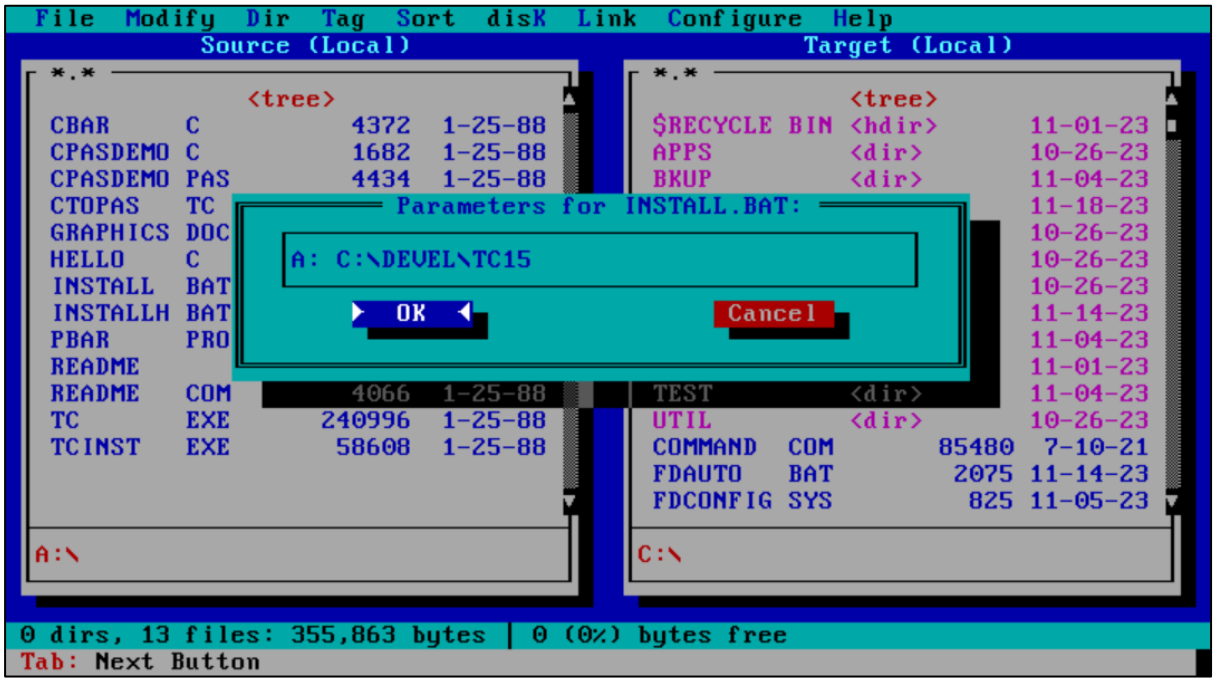
TC15

Mount "Disck01.img".

Navigate to drive A: from the command line and run the following install line.

```
INSTALL.BAT A: C:\DEVEL\TC15
```

Do NOT use FM3 for installs, or copy INSTALL.BAT and INSTALLH.BAT to a separate location from the floppy drive A and do not access the floppy drive with FM3.



```
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:\TCCONFIG.EXE =>> C:\DEVEL\TC15\TCCONFIG.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:\SETARGV.ASM =>> C:\DEVEL\TC15\LIB\SETARGV.ASM
A:\SETENVP.ASM =>> C:\DEVEL\TC15\LIB\SETENVP.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:\C0T.OBJ =>> C:\DEVEL\TC15\LIB\C0T.OBJ
A:\C0S.OBJ =>> C:\DEVEL\TC15\LIB\C0S.OBJ
A:\C0L.OBJ =>> C:\DEVEL\TC15\LIB\C0L.OBJ
A:\CS.LIB =>> C:\DEVEL\TC15\LIB\CS.LIB
A:\MATHS.LIB =>> C:\DEVEL\TC15\LIB\MATHS.LIB
A:\CL.LIB =>> C:\DEVEL\TC15\LIB\CL.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:\C0T.OBJ =>> C:\DEVEL\TC15\LIB\C0T.OBJ
A:\C0S.OBJ =>> C:\DEVEL\TC15\LIB\C0S.OBJ
A:\C0L.OBJ =>> C:\DEVEL\TC15\LIB\C0L.OBJ
A:\CS.LIB =>> C:\DEVEL\TC15\LIB\CS.LIB
A:\MATHS.LIB =>> C:\DEVEL\TC15\LIB\MATHS.LIB
A:\CL.LIB =>> C:\DEVEL\TC15\LIB\CL.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:\C0C.OBJ =>> C:\DEVEL\TC15\LIB\C0C.OBJ
A:\CC.LIB =>> C:\DEVEL\TC15\LIB\CC.LIB
A:\MATHC.LIB =>> C:\DEVEL\TC15\LIB\MATHC.LIB
A:\C0M.OBJ =>> C:\DEVEL\TC15\LIB\C0M.OBJ
A:\CM.LIB =>> C:\DEVEL\TC15\LIB\CM.LIB
A:\MATHM.LIB =>> C:\DEVEL\TC15\LIB\MATHM.LIB
A:\C0H.OBJ =>> C:\DEVEL\TC15\LIB\C0H.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:\EGAVGA.BGI =>> C:\DEVEL\TC15\EGAVGA.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 your FDCONFIG.SYS file has the following 2 lines. They are usually set by default in FreeDOS V1.3

```
!BUFFERS=20
```

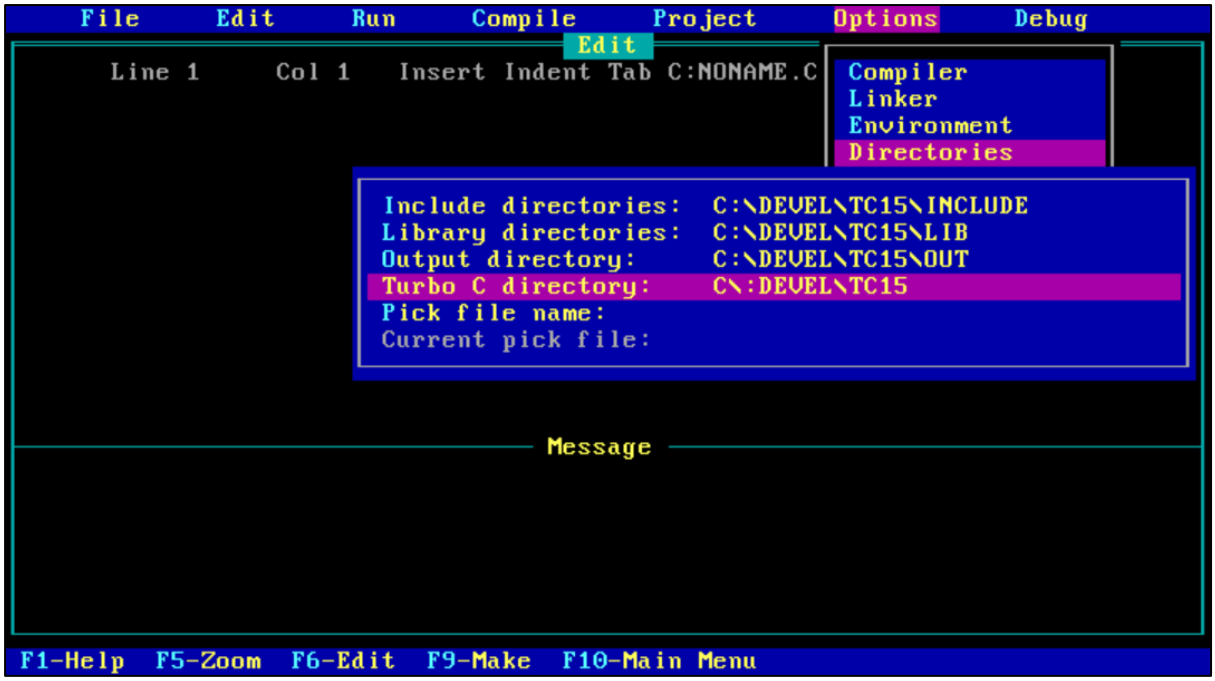
```
!FILES=40
```

Create a batch file to launch V1.5 as is shown for TC10.BAT

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.

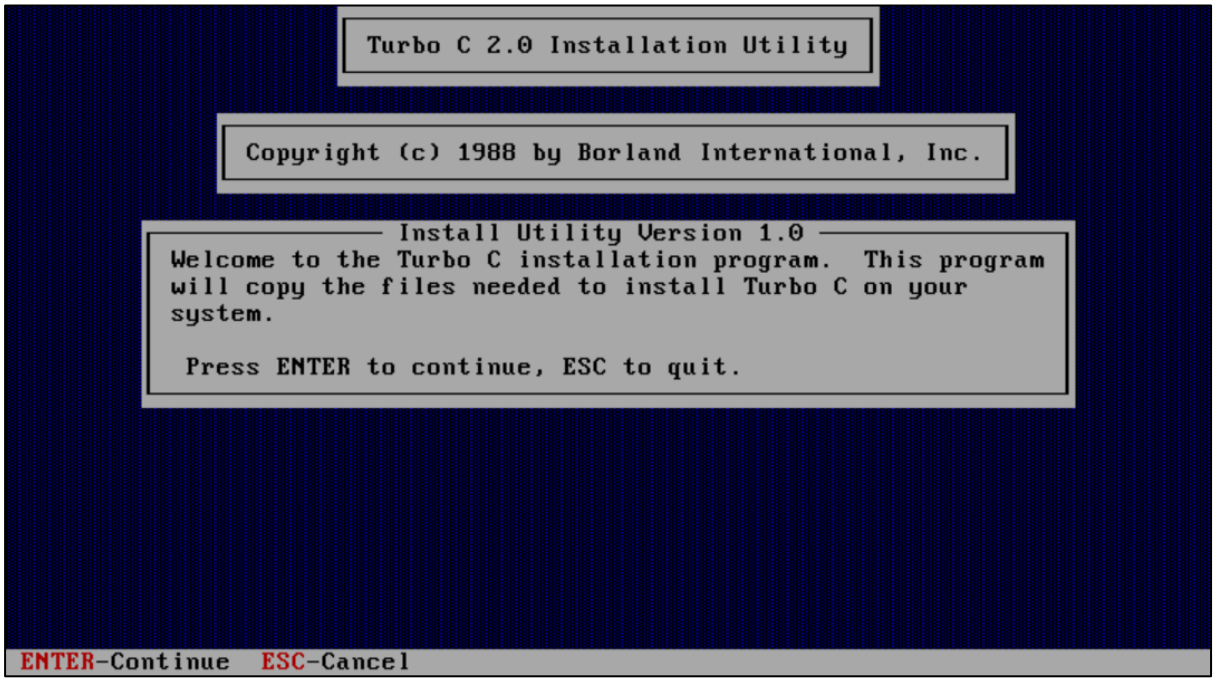
Also take note of the TAB size (4 recommended) and other editor options.

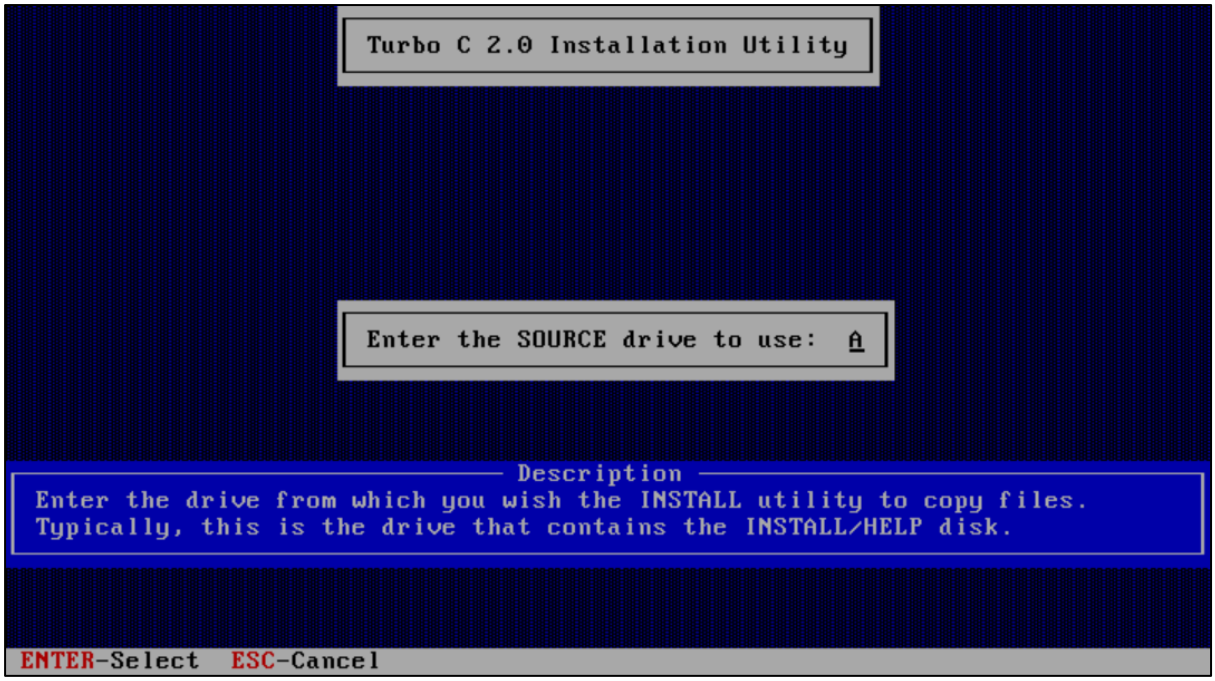


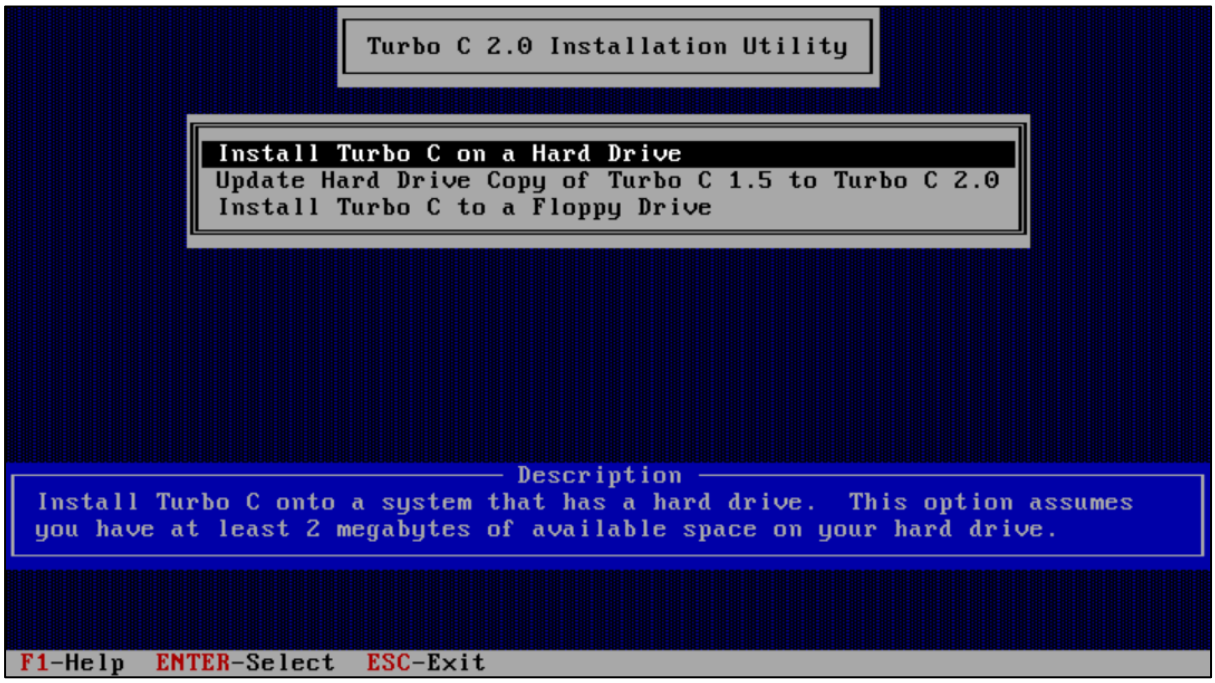
TC20

Mount the “disk01.img”.

Navigate to A: and run the “INSTALL.EXE”.

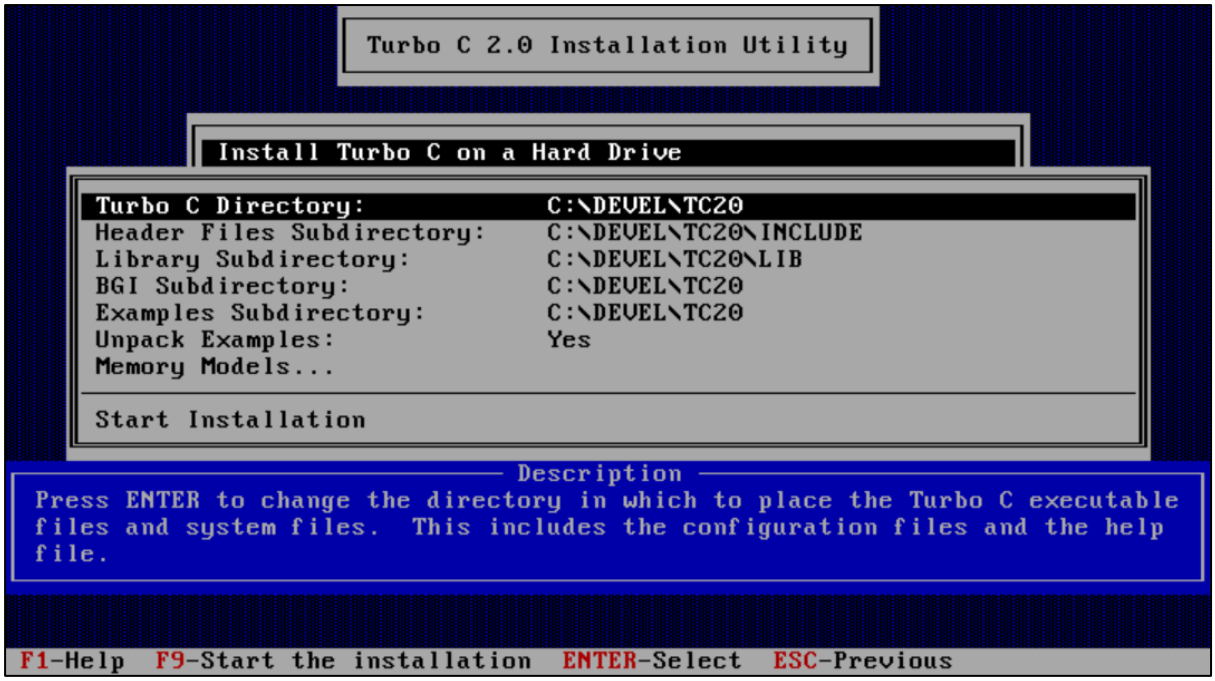




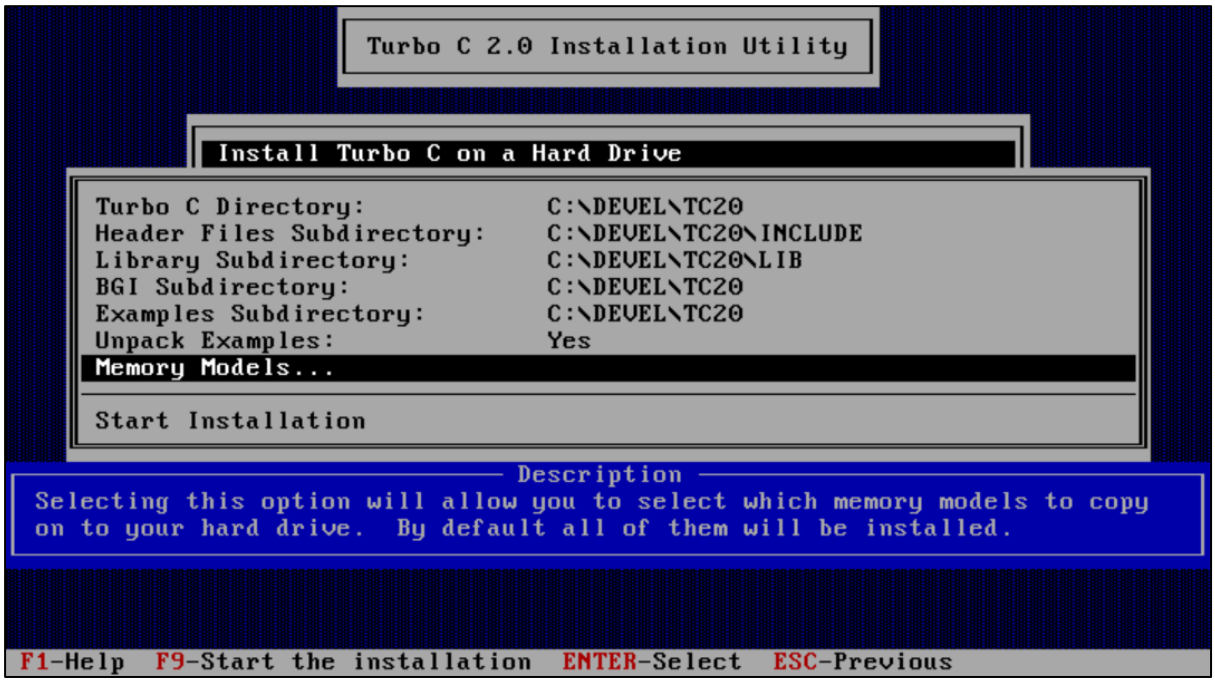


Enter the correct path for each. If you correct the “Turbo C Directory” the others are automatically updated.

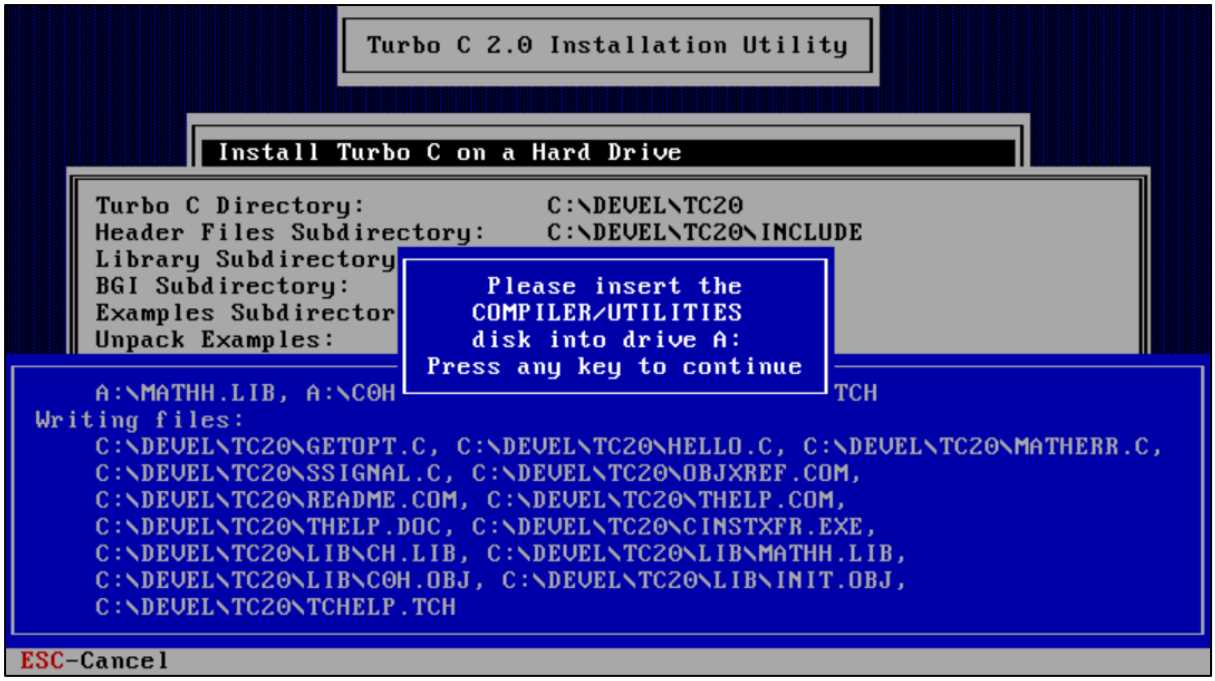
```
C:\DEVEL\TC20
```



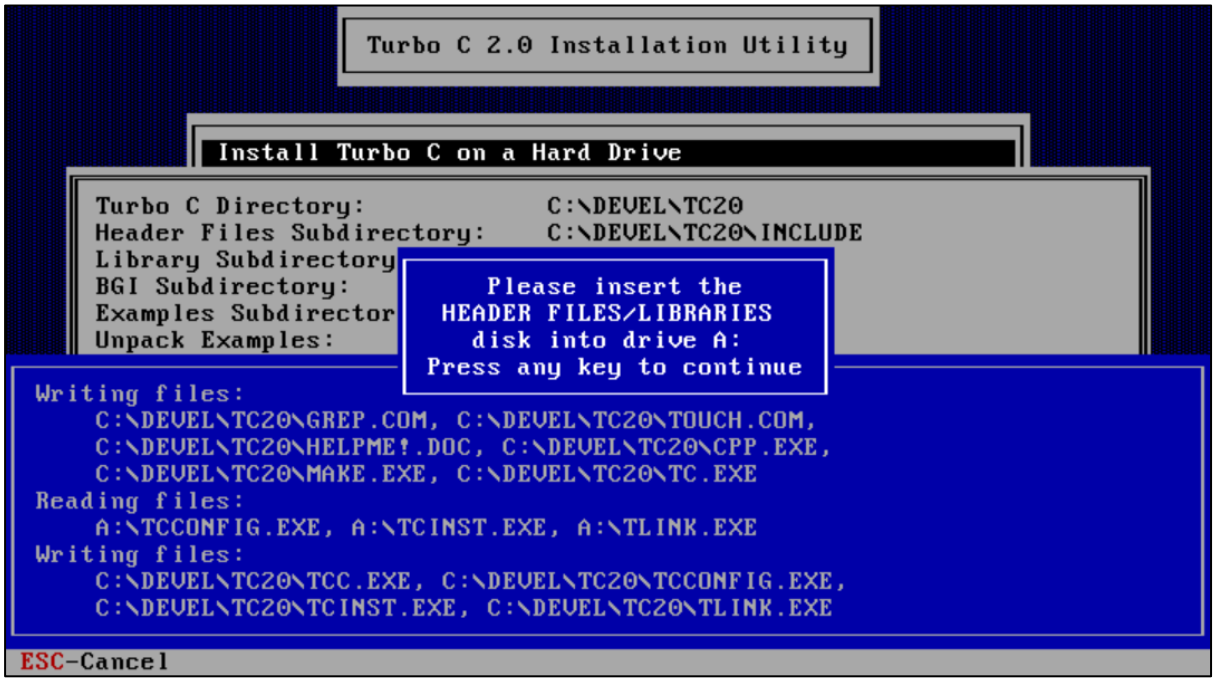
Leave the “Memory Models...” as the default where all are installed. Memory models can be selected at compile time.



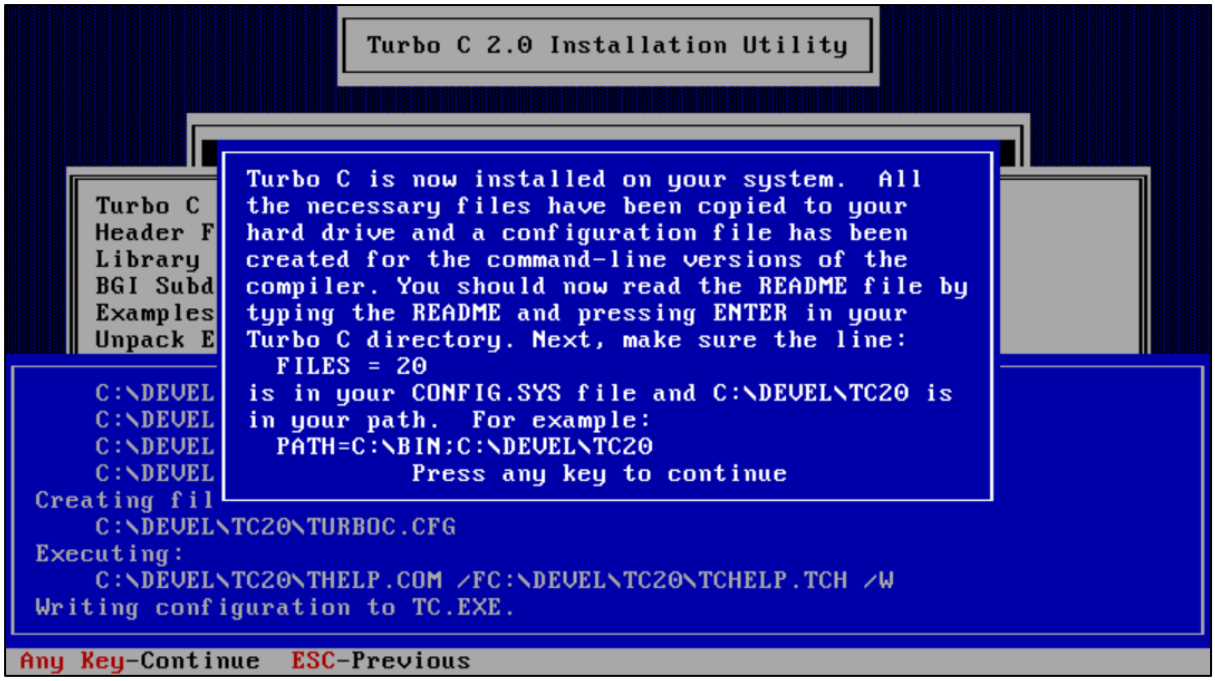
Select “Start Installation” and Enter to continue.



Insert disk02.img



Insert disk03.img

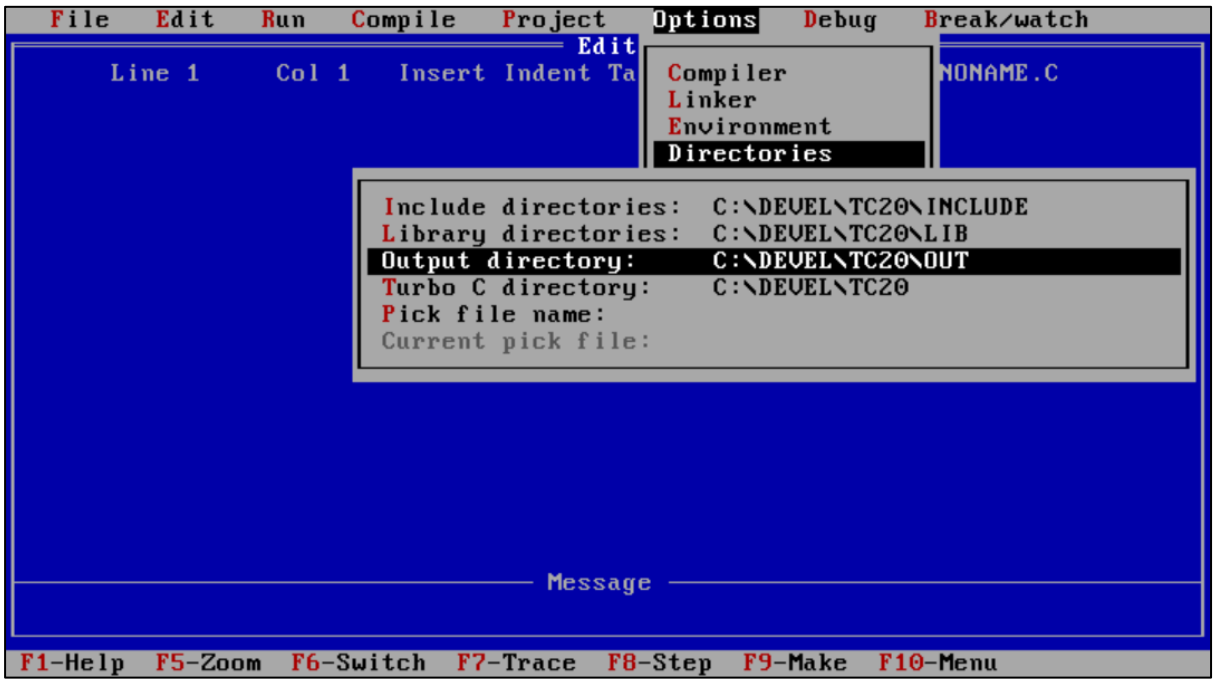


The Turbo C V2.0 install is complete.

FreeDOS sets !FILES=40 by default so don't change it to a lower value.

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

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.

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 repository 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.

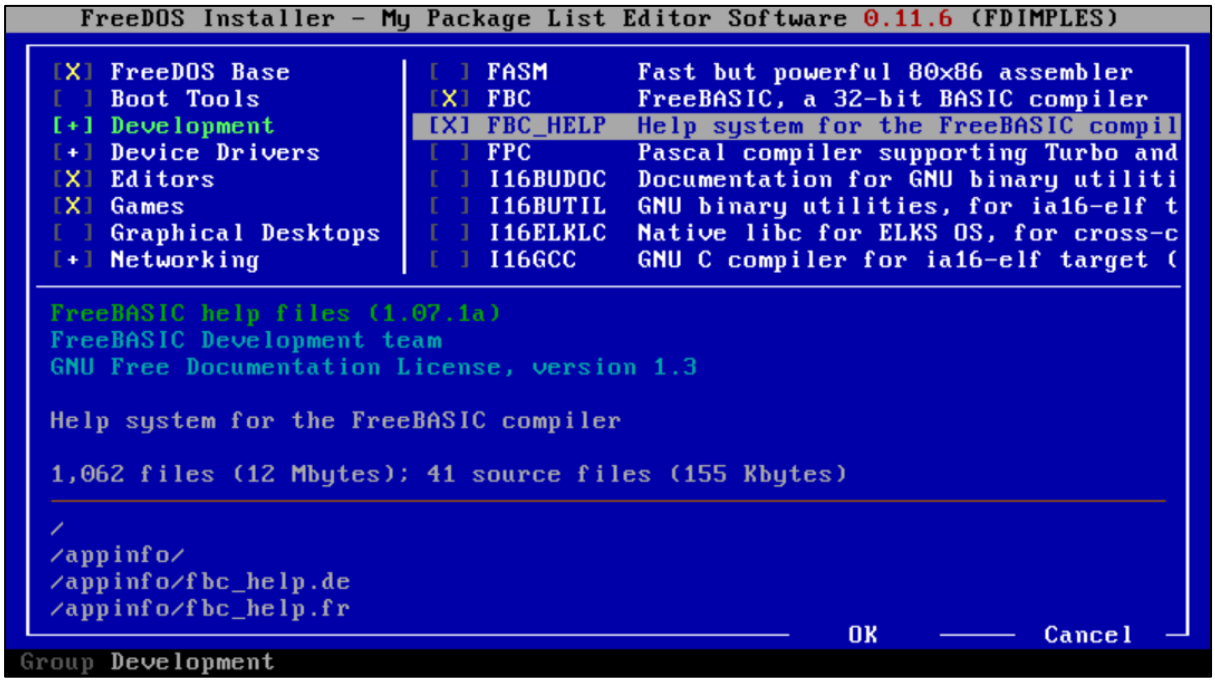
FreeDOS Bonus CD V1.07.1a

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.

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\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\lfnfiles.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\lfnfiles.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 las 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/repo
sitories/latest/devel/fbc.zip...
Downloading fbc.zip... 1050664 bytes [5%]
```

```
devel\fbc\inc\gsl\gsl_fft.bi -> c:\devel\fbc\inc\gsl\
devel\fbc\inc\gsl\gsl_sum.bi -> c:\devel\fbc\inc\gsl\
devel\fbc\inc\gsl\gsl_math.bi -> c:\devel\fbc\inc\gsl\
devel\fbc\inc\gsl\gsl_dht.bi -> c:\devel\fbc\inc\gsl\
devel\fbc\inc\gsl\gsl_sort.bi -> c:\devel\fbc\inc\gsl\
devel\fbc\inc\gsl\gsl_test.bi -> c:\devel\fbc\inc\gsl\
devel\fbc\inc\gsl\gsl_qrng.bi -> c:\devel\fbc\inc\gsl\
devel\fbc\inc\gsl\gsl_sys.bi -> c:\devel\fbc\inc\gsl\
devel\fbc\inc\png.bi -> c:\devel\fbc\inc\
devel\fbc\inc\libintl.bi -> c:\devel\fbc\inc\
devel\fbc\inc\libexslt\exslt.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_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 version1.07.1 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\putgfx1.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\lfnfiles.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 FDCONFIG.SYS or FDAUTO.BAT. Place the BFCCMD.BAT in the .\LINKS directory.

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
set PATH=%PATH%;C:\DEVEL\FBC
REM Optional paths:
REM set PATH=%PATH%;C:\DEVEL\FBC\BIN; C:\DEVEL\FBC\INCLUDE; C:\DEVEL\FBC\LIB\DOS

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.bttr-software.de/products/fed/

Install FED to C:\DEVEL\FBC\FED224B*.*

Create a batch file to launch FED in C:\FREEDOS\LINKS\FBFED.BAT

The PATH is so the system can find FBC.EXE, STINGS.COM and the compiler batch files.

FBFED.BAT (Or any other appropriate name)

```
@ECHO OFF
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.bttr-software.de/freesoft/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`

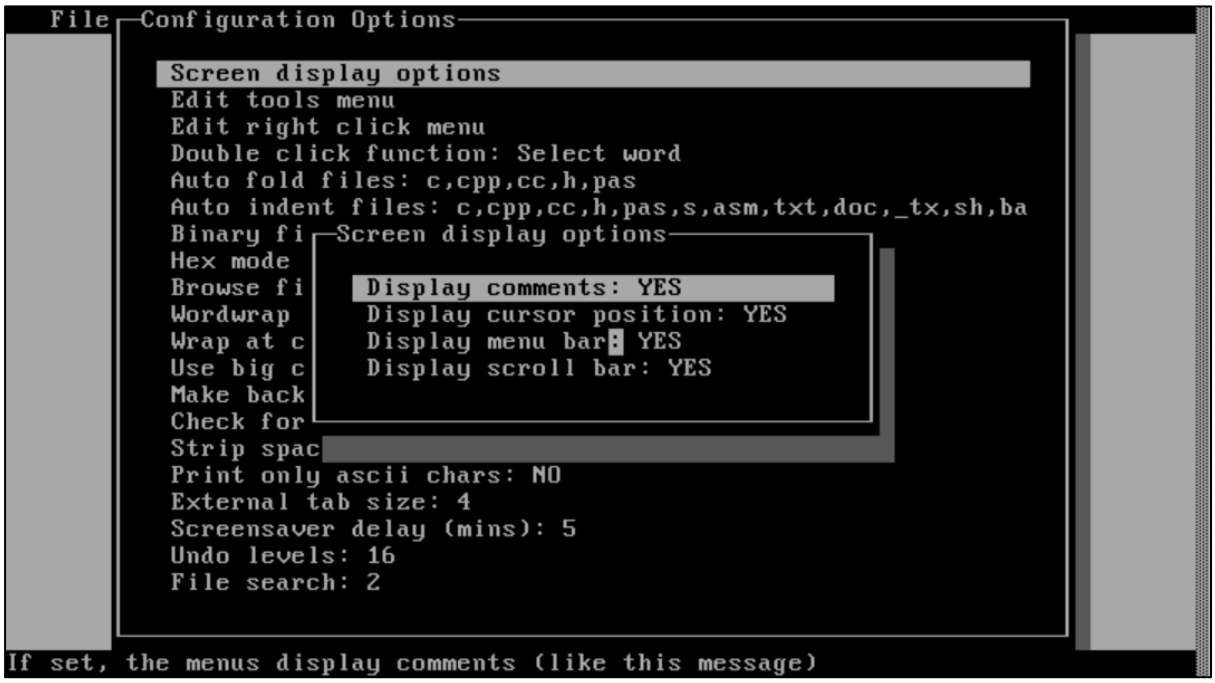
`C:\DEVEL\FBC\STRINGS.COM`

DOS does not have string manipulation tools for use in the command line or batch files, so STRINGS.COM fills this gap.

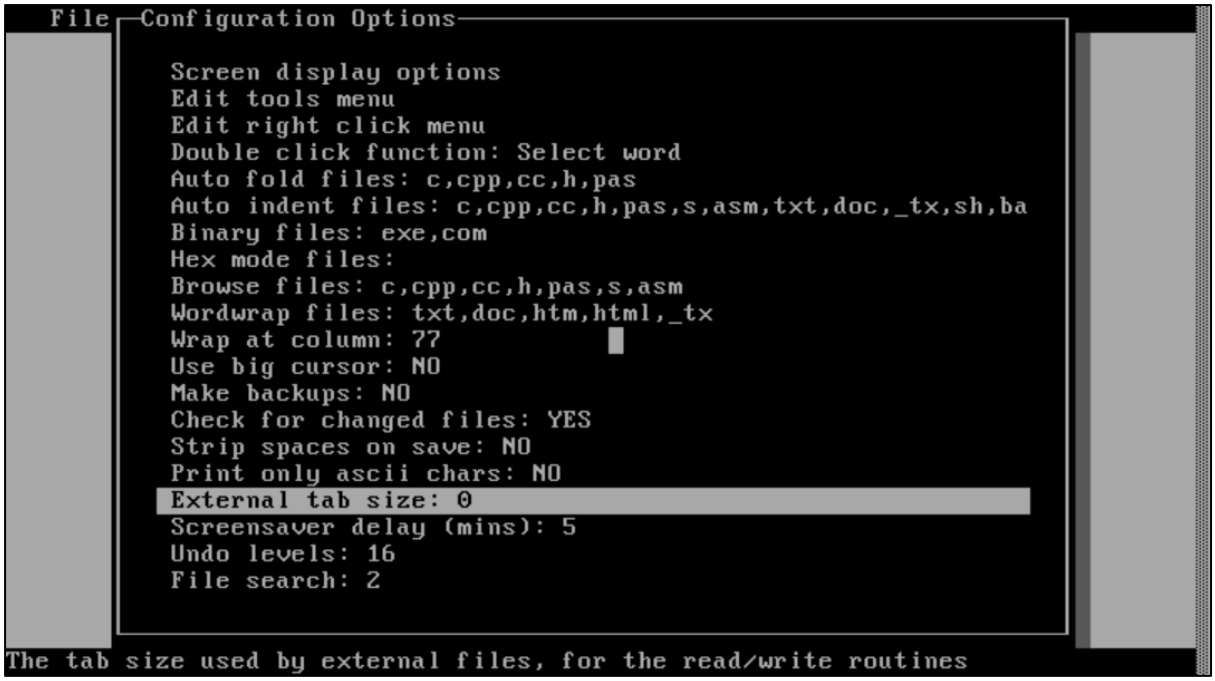
Open FED224 using the batch file in .\LINKS or from the command line with FBFED.BAT

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
REM https://ftp.sunet.se/mirror/archive/ftp.sunet.se/pub/simtelnet/msdos/batchutl/
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!
```

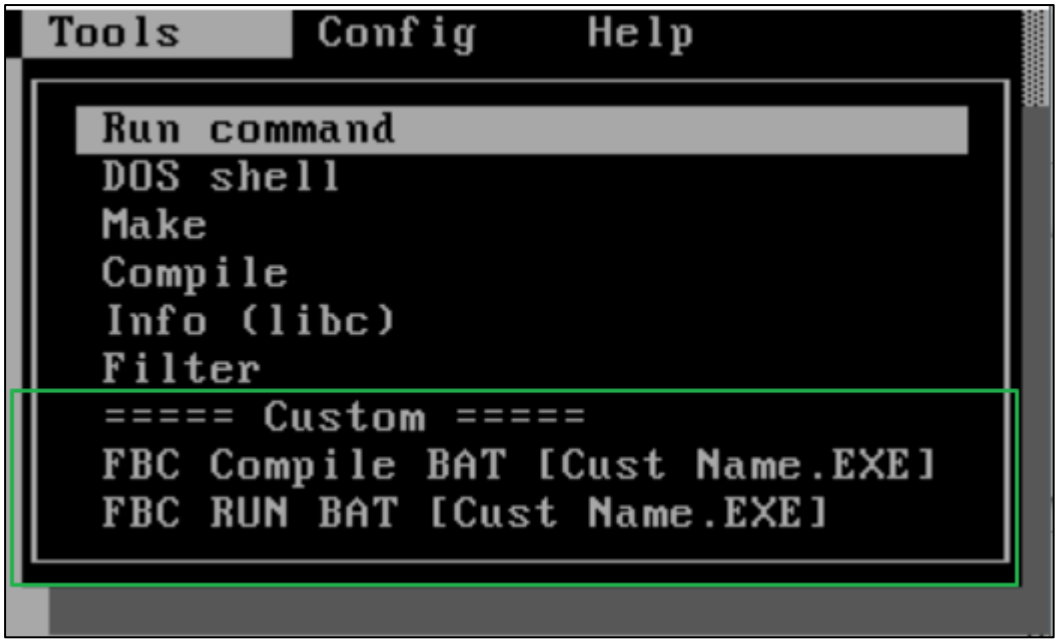
```
goto :END

: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

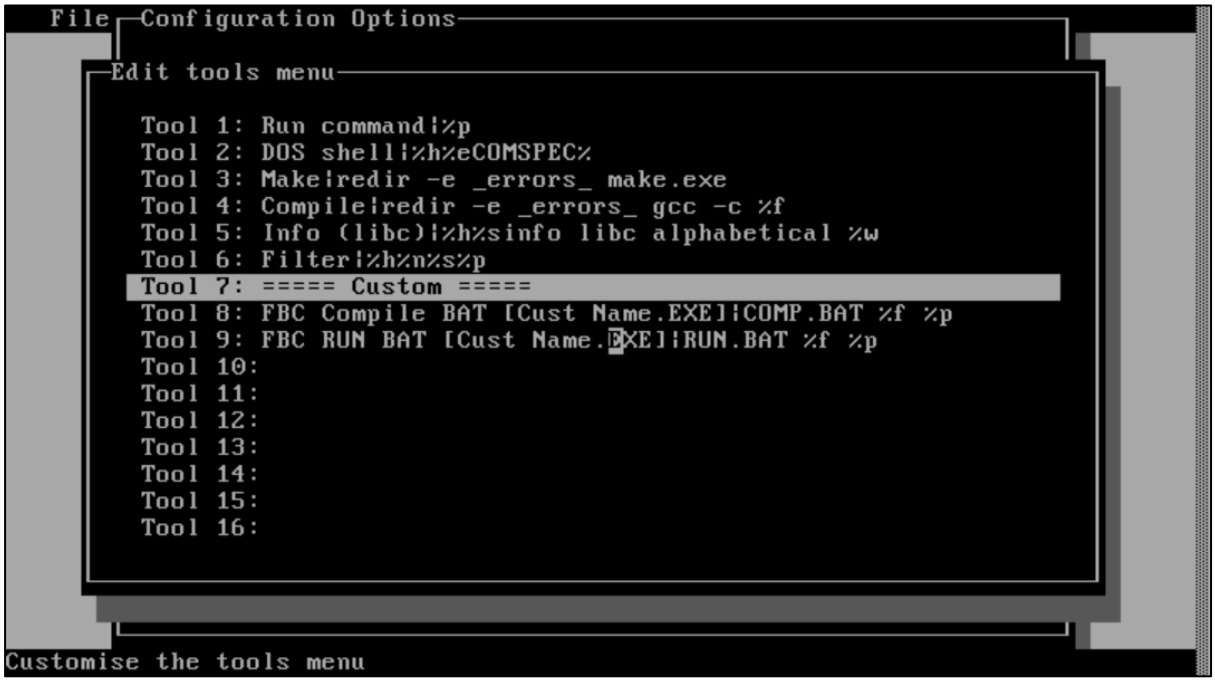
```
@echo OFF
REM This is a generic command line runner for FED and FreeBASIC_DOS.
REM With small modification the compiler command line call can be used with
REM FED for other compilers or languages.
REM (Other compilers) Change str=EXE to EXE/COM etc.
REM SEE: Accompanying COMP.BAT for FED
REM Requires STRINGS.COM  (Version 2.5) Copyright (c) 1991, 1992 Douglas Boling
REM ftp.sunet.se/pub/simtelnet/msdos/batchutl/string25.zip
REM https://ftp.sunet.se/mirror/archive/ftp.sunet.se/pub/simtelnet/msdos/batchutl/
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.

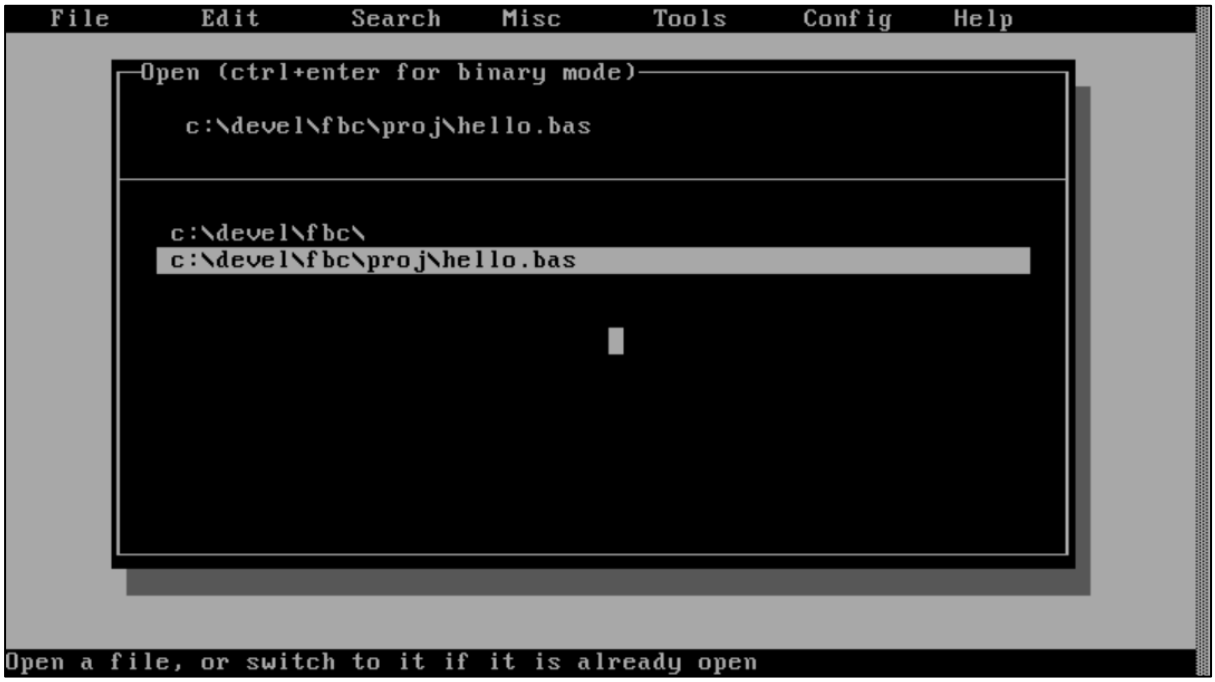
To remove a line just leave it blank.

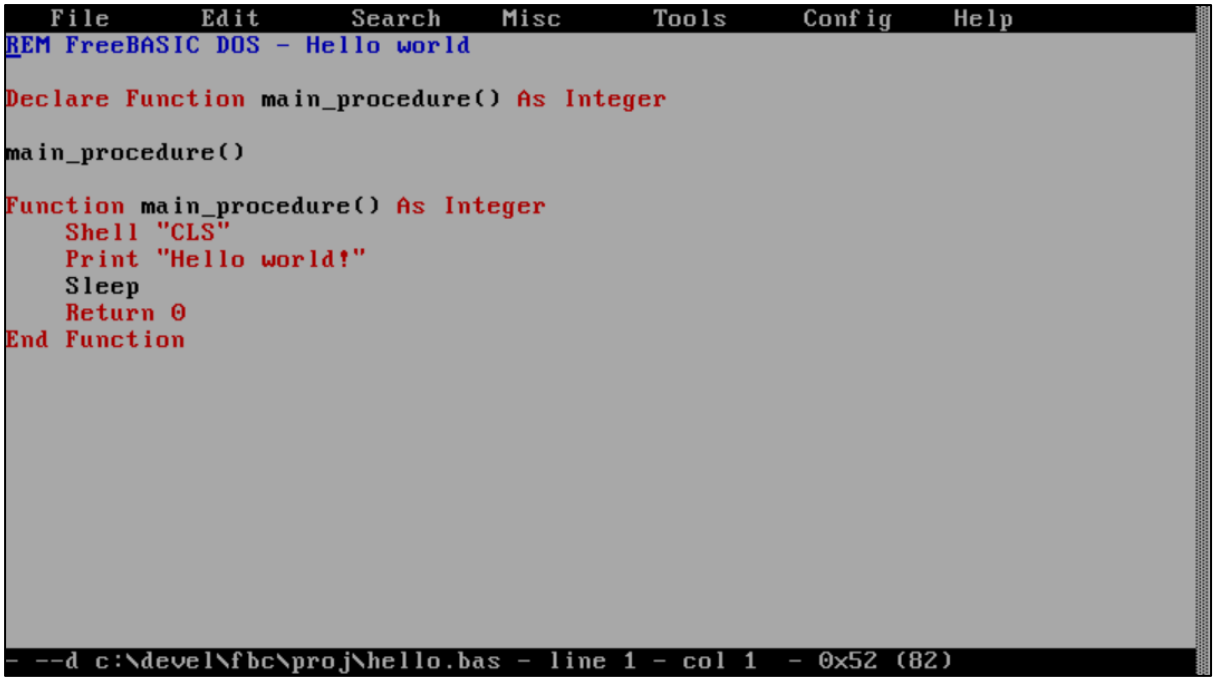
Make sure you “Save config” before closing FED.

Hint you can copy the FED.CFG to other instances of FED to keep your FED customizations.

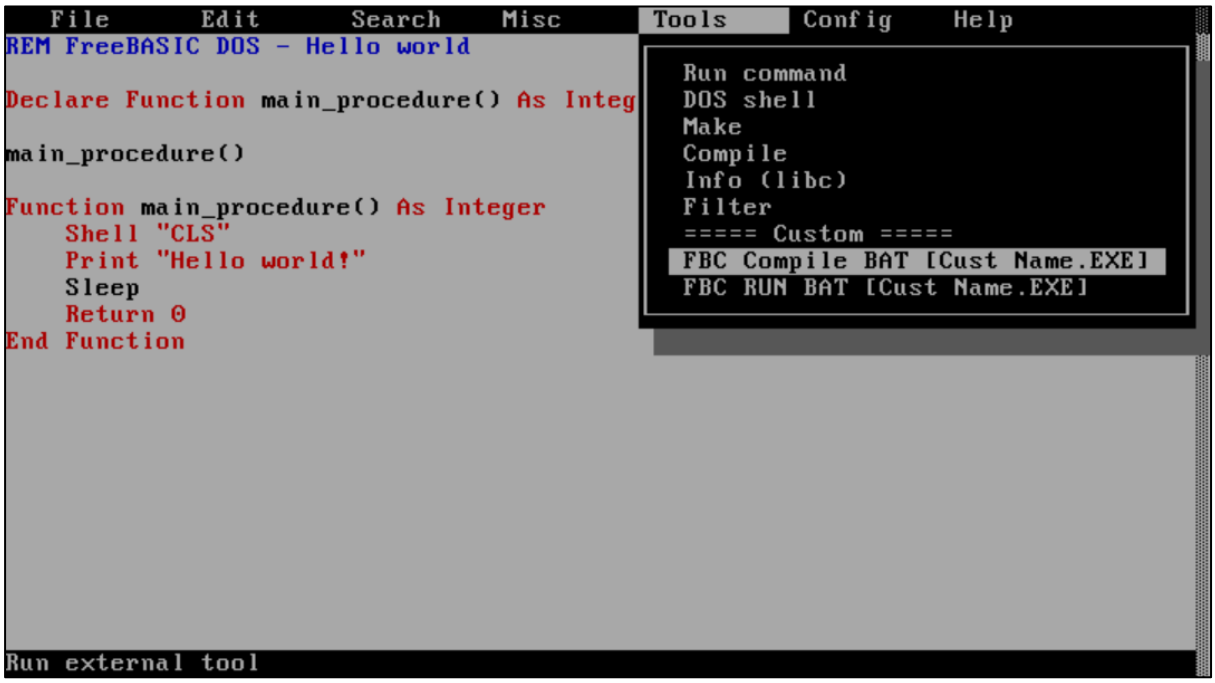
The FED website also makes available some alternative pre made syntax highlighting schemes FED.SYN. It is a good idea to keep a backup of your FED configs in case of mistakes or for recovering you settings.

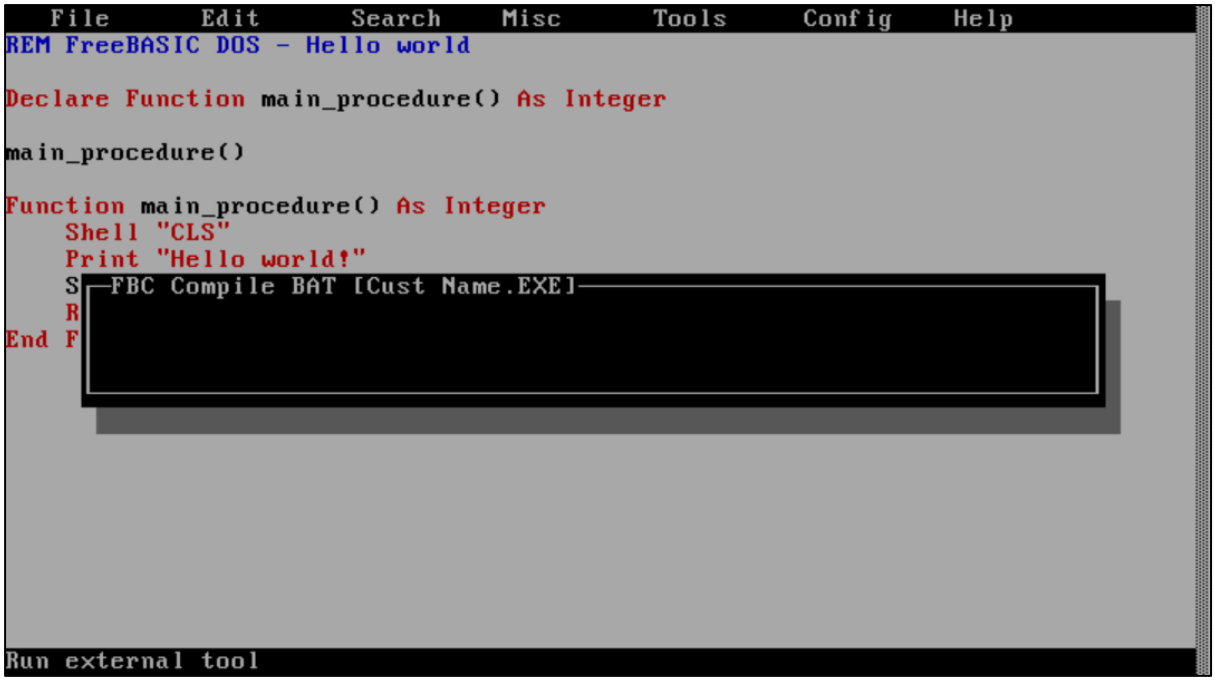
Open FED for your FBC environment and create a “Hello World!” HELLO.BAS source file.

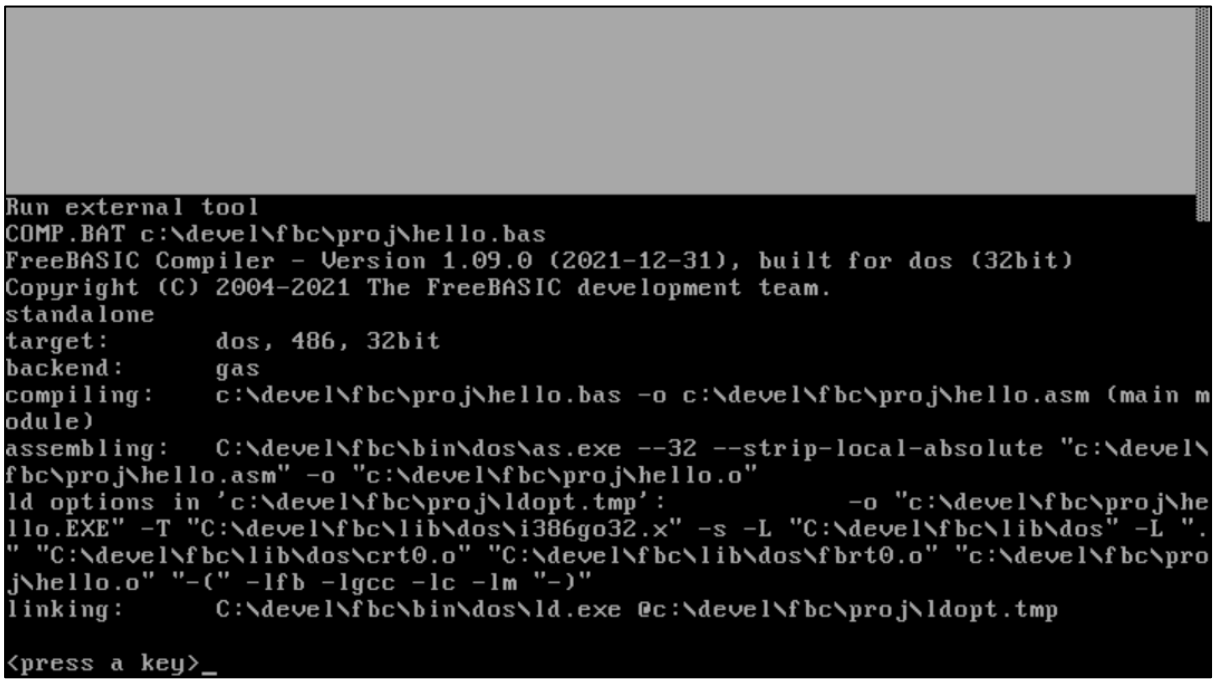




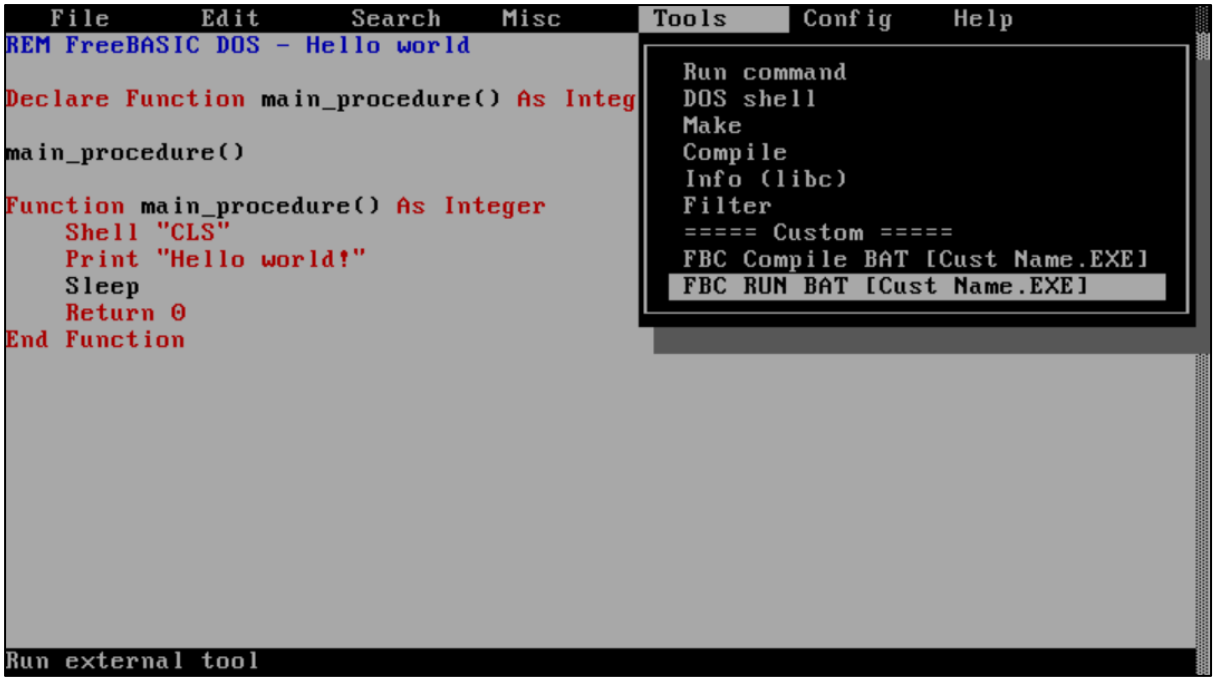
From the menu open “Tools -> ...” and select your menu entry to COMPILE the source code followed by you menu entry to RUN the source code. I would also test both menu entries using the custom name for the output file. You will find that if you compile with a custom name the variable p% will be auto filled when using the Run option.

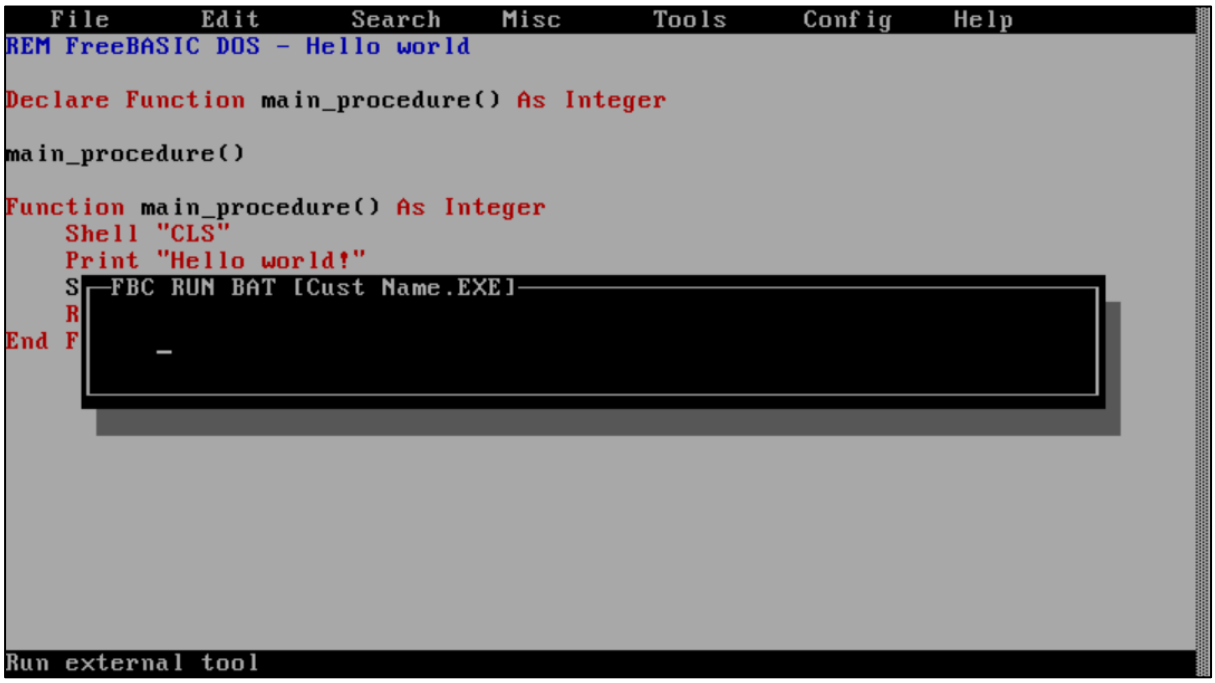






Now select run the output executable.


File Edit Search Misc Tools Config Help
REM FreeBASIC DOS - Hello world
Declare Function main_procedure() As Integ
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!"
FBC RUN BAT [Cust Name.EXE]
Run external tool

```
Hello world!
_
```

Using the alternative output name (Make sure to add the extension “WORLD.**EXE**”)...

```
   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┌─GCC Compile BAT [Cust Name.EXE]──────────────────────────┐
    R│                                                          │
End F│     WORLD.EXE_                                           │
     └──────────────────────────────────────────────────────────┘

Run external tool
```

```
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 m
odule)
assembling:   C:\devel\fbc\bin\dos\as.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\WO
RLD.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\pro
j\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...

```
   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_                                                       │
     └──────────────────────────────────────────────────────────────────────┘

Run external tool
```

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.

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 Costa desktop you can create an Icon/link to FED .\LINKS\FBFED.BAT in one of the 5 desktops for convenience.

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% > frofile.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 libpdcurses.a to make pdcurses.a, (Not recomended) or change the include line in the .\FBC\INC\PDCURSES.BI from

#inclib “pdcurses” to

#inclib “pdc~1”

“pdc~1” will find the 8.3 named library file .\FBC\LIB\DOS\LIBPDC~1.A (libpdcurses.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 an 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.

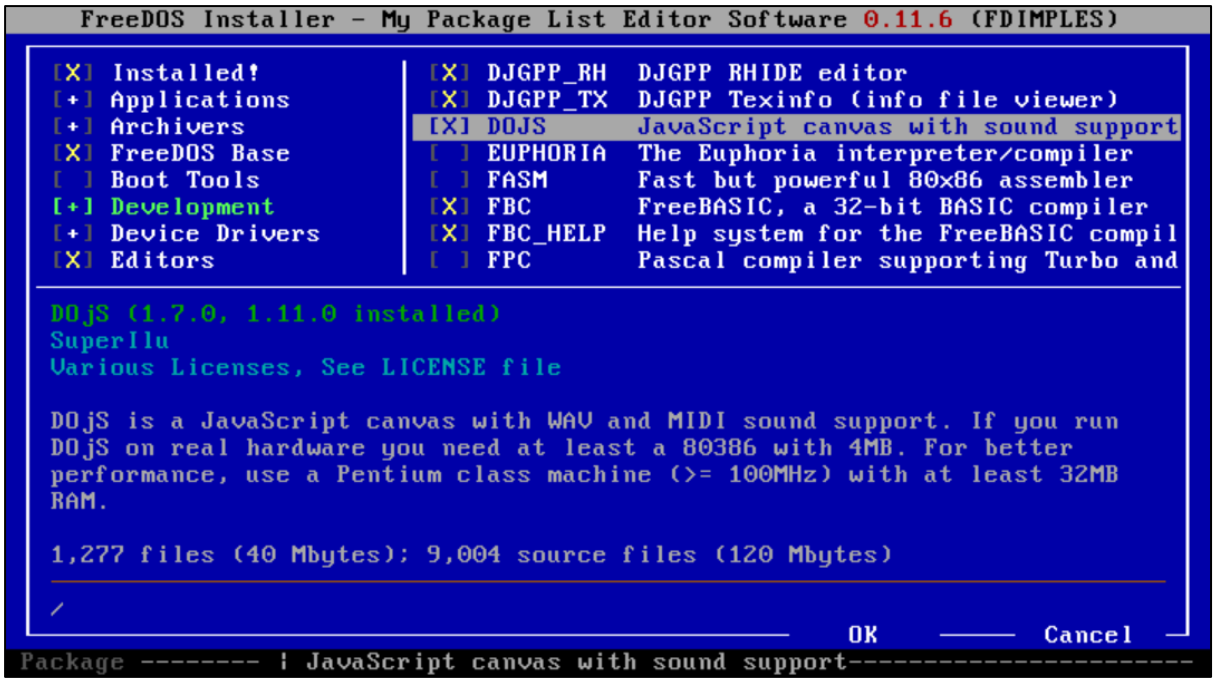
DOjS is a work in progress and evolving so be sure to check the GitHub page for bugs and updates.

Install

DOjS can be installed from the FreeDOS bonus CD via FDIMPLES, or via FDNPKG.

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/lat
est/edit/...
Loading http://www.ibiblio.org/pub/micro/pc-stuff/freedos/files/repositories/lat
est/emulator/...
Loading http://www.ibiblio.org/pub/micro/pc-stuff/freedos/files/repositories/lat
est/games/...
Loading http://www.ibiblio.org/pub/micro/pc-stuff/freedos/files/repositories/lat
est/gui/...
Loading http://www.ibiblio.org/pub/micro/pc-stuff/freedos/files/repositories/lat
est/net/...
Loading http://www.ibiblio.org/pub/micro/pc-stuff/freedos/files/repositories/lat
est/sound/...
Loading http://www.ibiblio.org/pub/micro/pc-stuff/freedos/files/repositories/lat
est/unix/...
Loading http://www.ibiblio.org/pub/micro/pc-stuff/freedos/files/repositories/lat
est/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/repo
sitories/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/repo
sitories/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.

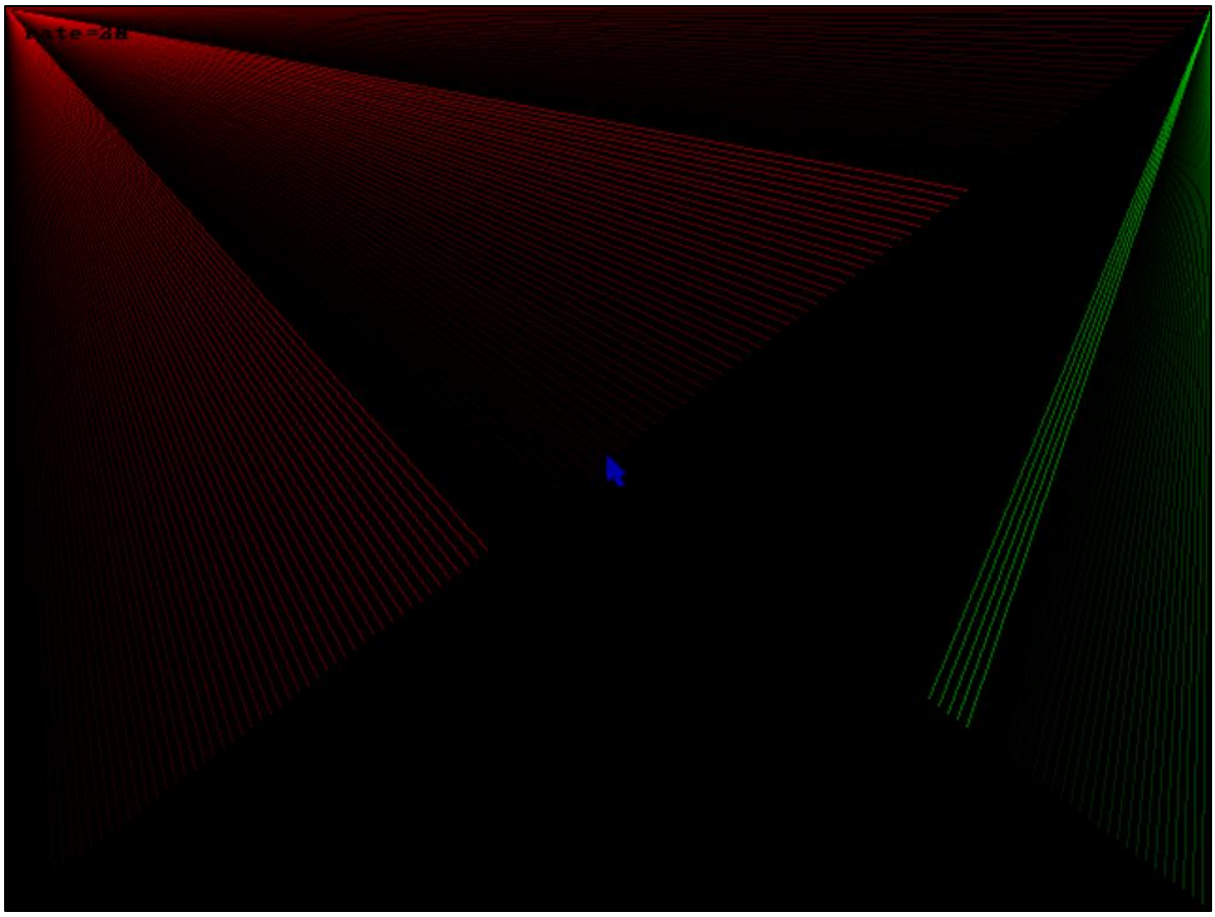
Navigate to `C:\DEVEL\DOJS` and enter the following into the command line.

```
DOJS.EXE C:\DEVEL\DOJS\EXAMPLES\DEMO.JS
```

This will open the source in the DOjS built in editor with the DEMO.JS source.

```
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.



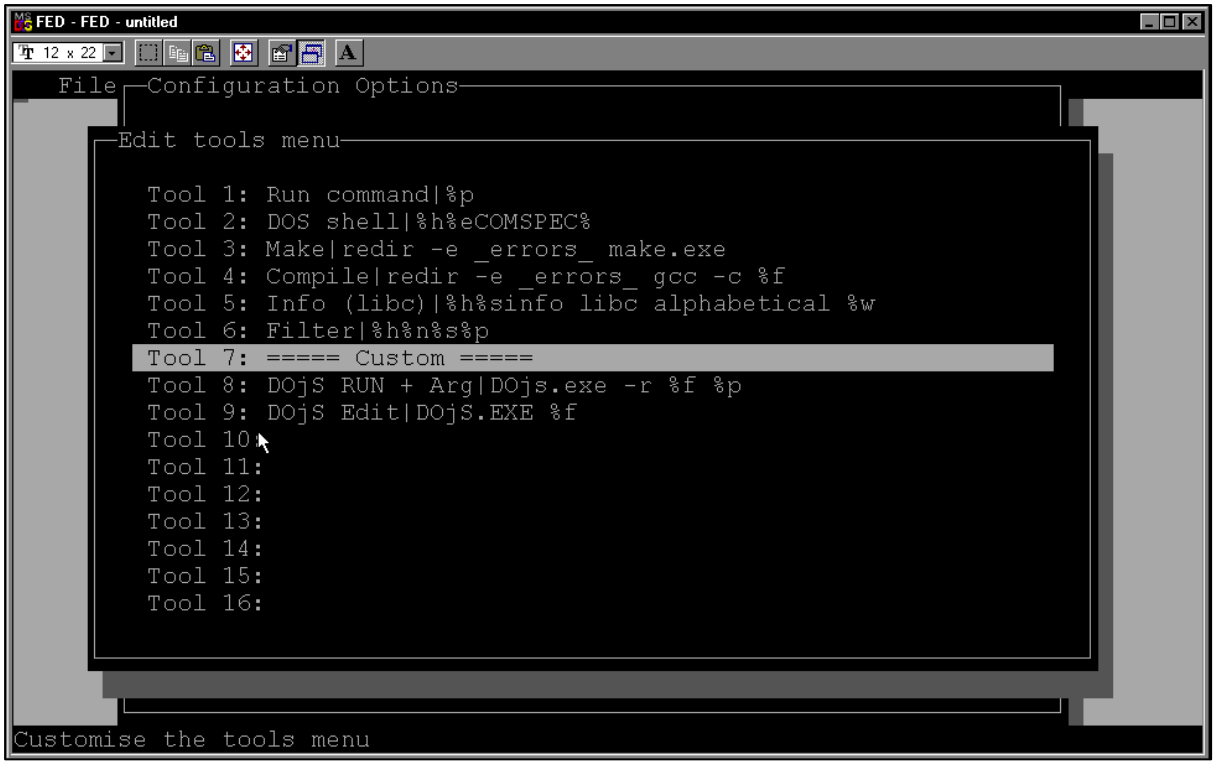
Take careful note of the placement of your source code as it needs to be in or below the root directory. You will need to keep the directory structure of your source related to the directory of DOjS.

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.

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.

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

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

Libraries

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

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

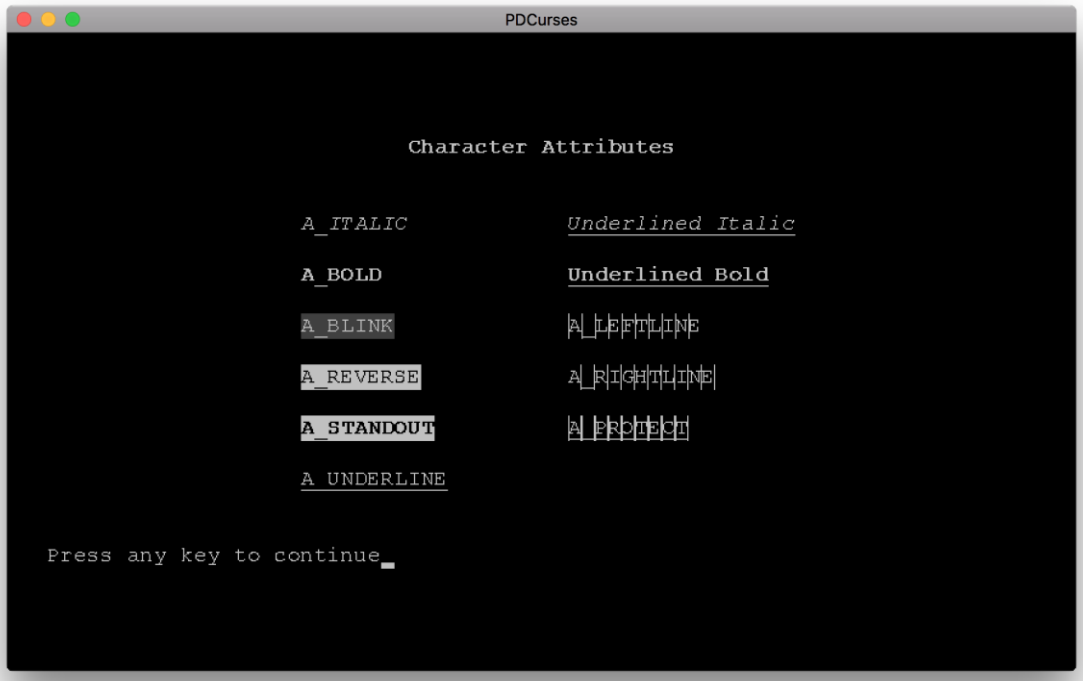
- Allegro has Audio playback APIs.
- Housemarque Audio System
  http://www.s2.org/hmqaudio/
- Sound Blaster
  http://www.shdon.com/dos/sound#mixing
- Apogee Sound System (Open Watcom)
  https://github.com/jimdose/Apogee_Sound_System
- Judas Sound Library
  https://github.com/volkertb/JUDAS
- Miles Sound System

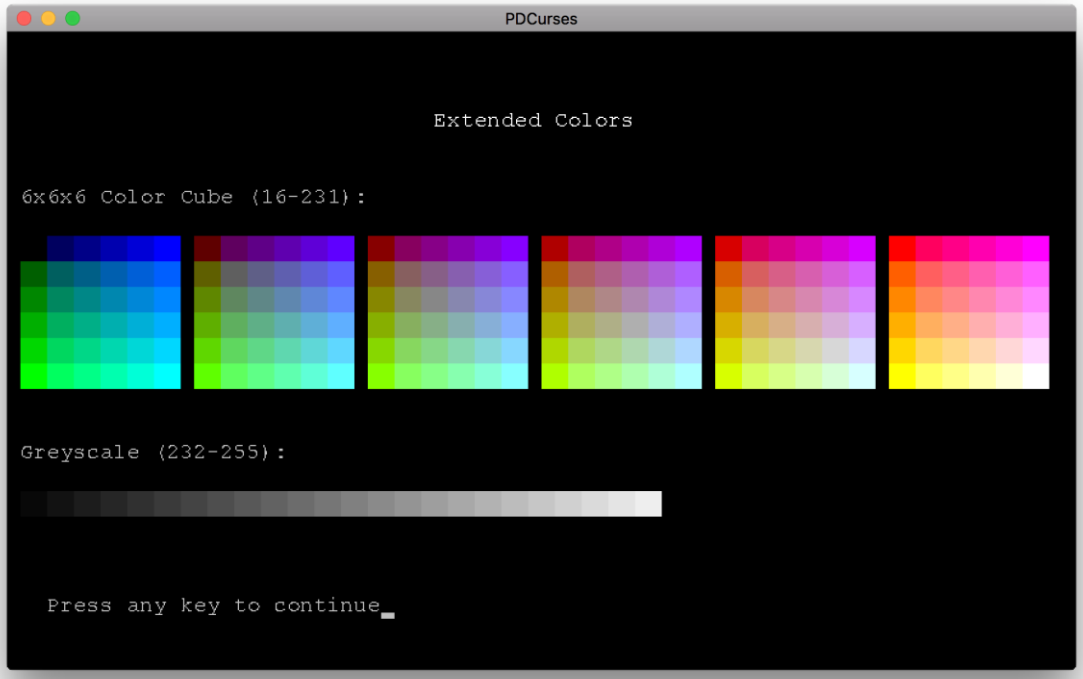
http://www.ke5fx.com/
http://www.thegleam.com/ke5fx/misc/AIL2.ZIP

- 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 versions)

This repository has the PDCurses V3.4 which is usable in most DJGPP and FBC versions.

https://sourceforge.net/projects/fbc/files/Older%20versions/0.90.1/Binaries%20-%20Windows/Libraries/

Source code V3.9 – All supported platforms.

https://pdcurses.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

GRX is a 2D graphics library originally written by Csaba Biegl 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.

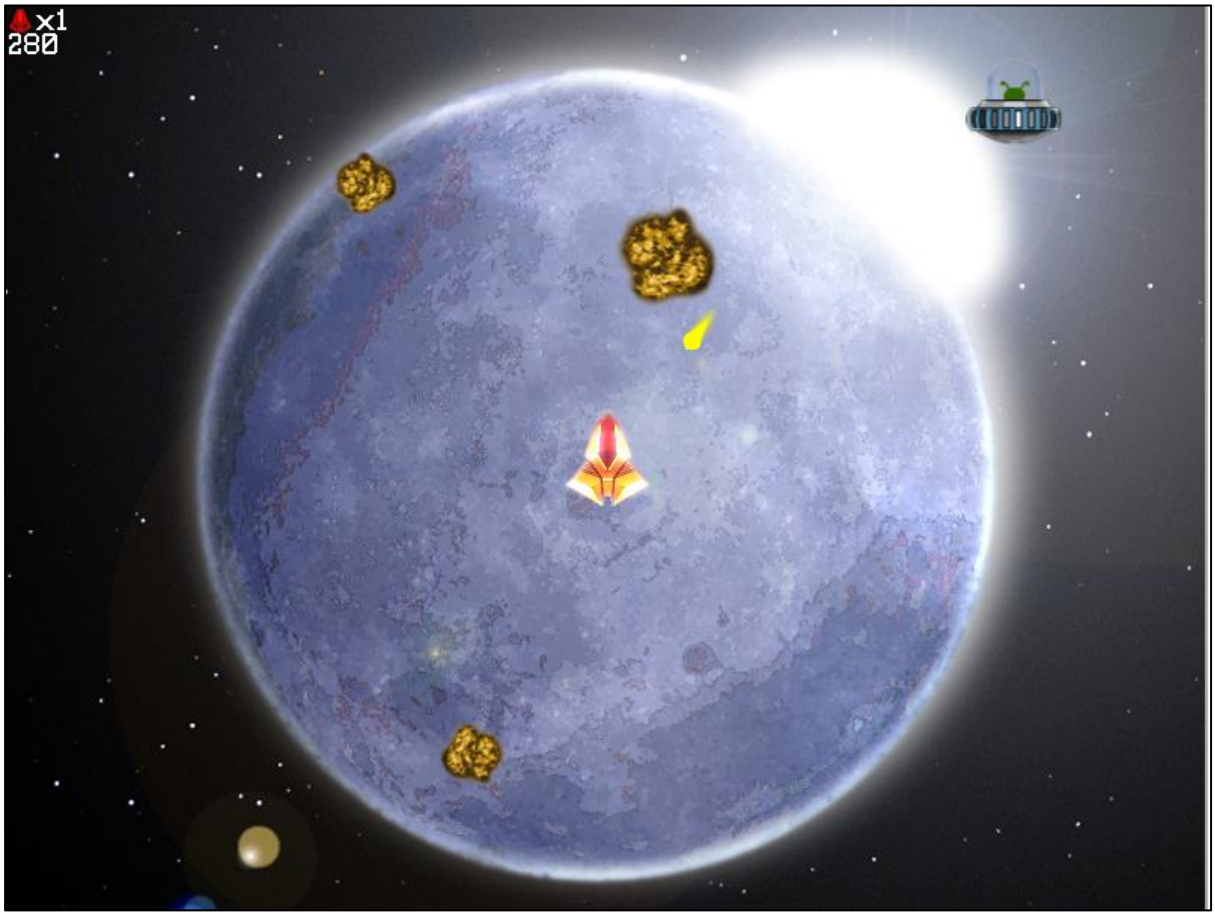
MGRX supports four platforms: DOS (DJGPPv2), Linux console, Linux X11 and Win32 (TDM-GCC). On DOS it supports VGA and VESA compliant cards. On Linux console it comes with framebuffer and KMS/DRM drivers. On X11 and Win32 it runs in a window.

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]

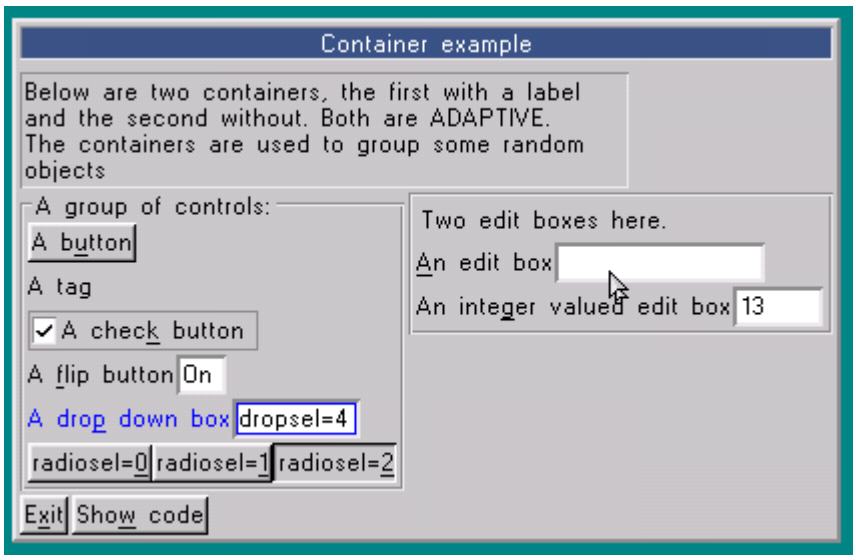
Allegro.bi V 4.4.2

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 libary 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/jmgui.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 you DOS system drive.

Set up your Virtual machine environment for DOS.

Keep a log of install tasks and customisations for use in later DOS 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.

Install FreeDOS V1.3 with additional components.

Use the FreeDOS Bonus CD to install additional applications including some software development environments using FDIMPLES.

Use the network package manager FDNPKG to install as well as update and additional applications from the FreeDOS repository.

Make intermittent backups of the FreeDOS virtual hard drive for roll backs in case of an error.

Make backup copies of the FDCONFIG.SYS (CONFIG.SYS) and FDAUTO.BAT (AUTOEXEC.BAT) before each install or change to the system.

Become familiar with the DOS 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 and video. Adjust the mouse speed in FDAUTO.BAT after installing a GUI environment such as Costa desktop.

Install Filemaven 3 and Costa desktop for ease of navigation.

Create an optional boot (start-up) menu.

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.

Set up the custom path and environment variable for each software development environment using batch files. Include your preferred editor/IDE as part of the batch file.

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 Costa desktop to organise your work environment, add the icon/links to the Costa desktops.

Become familiar with the DOS environments 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 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 others 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://files.mpoli.fi/software/

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://www.freedos.org/download/

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

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

https://freedos.sourceforge.io/wiki/index.php/FDNPKG

https://www.vogonswiki.com/index.php/DOS_memory_management

https://en.wikipedia.org/wiki/DOS_extender

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

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

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

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://www.freedos.org/books/get-started/7-bat-files/

http://wiki.freedos.org/wiki/index.php/Games

https://costa.jacobpalm.dk/

https://support.creative.com/downloads/searchdownloads.aspx?nLanguageLocale=1033&filename=SB&nPage=39

https://support.creative.com/downloads/searchdownloads.aspx?filename=CTCMBBS

https://www.philscomputerlab.com/creative-labs-drivers.html

https://www.mmsp.ece.mcgill.ca/Documents/AudioFormats/WAVE/Samples.html

https://mauvecloud.net/sounds/index.html

http://wiki.freedos.org/wiki/index.php/Games

http://www.multimediaware.com/qv/index.html

https://mpxplay.sourceforge.net/

https://www.rarewares.org/rrw/dosamp.php

https://www.rarewares.org/rrw/programs.php

https://www.bttr-software.de/products/sbmix/

http://help.fdos.org/en/hhstndrd/ctmouse.htm

https://www.bttr-software.de/products/vp386/

https://winworldpc.com/product/pc-paint/3x

https://winworldpc.com/product/pc-paintbrush/3x-dos

https://www.ibiblio.org/pub/micro/pc-stuff/freedos/files/util/file/dospdf/

https://winworldpc.com/product/pc-paintbrush/iv-dos

http://grafx2.chez.com/

https://clasqm.github.io/freedos-repo/Graphics.html

https://sourceforge.net/p/freedos/news/2022/04/grafx2-port-to-freedos-alpha-release/

https://www.ibiblio.org/pub/micro/pc-stuff/freedos/files/repositories/unstable/apps/

https://www.ibiblio.org/pub/micro/pc-stuff/freedos/files/repositories/unstable/apps/grafx2.zip

http://www.pictview.com/

https://clasqm.github.io/freedos-repo/Graphics.html

https://www.ibiblio.org/pub/micro/pc-stuff/freedos/files/util/file/dospdf/

https://drv.nu/vbediag/

https://www.elhvb.com/webhq/download/index.htm

https://www.os2site.com/sw/dos/util/video/index.html

https://www.elhvb.com/webhq/download/index.htm

https://www.os2site.com/sw/dos/util/video/index.html

https://www.navsoft.cz/products.htm

https://www.bttr-software.de/freesoft/system.htm

https://www.uselesssoftware.com/download/infoplus-zip

https://www.hwinfo.com/download/

http://web.archive.org/web/20000302035130/http://www.scitechsoft.com/down_dos.html

https://www.ibiblio.org/pub/micro/pc-stuff/freedos/files/repositories/1.2/pkg-html/dillo.html

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

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://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/djgpp/zip-picker.html

http://www.delorie.com/pub/djgpp/current/

https://www.freebasic.net/wiki/DevBuildDos

http://www.delorie.com/djgpp/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

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

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/freesoft/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.elhvb.com/webhq/download/index.htm

https://www.elhvb.com/webhq/download/index.htm

https://www.bttr-software.de/freesoft/system.htm

https://archive.org/details/msdos_shareware_fb_INFO

https://www.hwinfo.com/download/

https://mklasson.com/hexit.php

https://www.bttr-software.de/freesoft/filutil2.htm

https://www.sac.sk/search.php

https://www.bttr-software.de/freesoft/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

http://www.delorie.com/djgpp/v2faq/faq22_4.html

https://www.freebasic.net/wiki/ExtLibTOC

http://www.s2.org/hmqaudio/

http://www.shdon.com/dos/sound#mixing

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/djgpp/current/v2tk/

https://www.delorie.com/pub/djgpp/current/v2tk/

https://sourceforge.net/projects/fbc/files/Older%20versions/0.90.1/Binaries%20-%20Windows/Libraries/

https://pdcurses.org/

https://github.com/wmcbrine/PDCurses

http://grx.gnu.de/

GRX library

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

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/djgpp/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://www.seriss.com/people/erco/fltk-videos/tutorial-fltk-hello.html

https://github.com/ghaerr/microwindows

https://www.execulink.com/~coder/freebasic/jmgui.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/

Additional URLs with DOS downloads and information:

https://www.freedos.org/

https://github.com/balintkissdev/awesome-dos

https://www.bttr-software.de/freesoft/

https://clasqm.github.io/freedos-repo/

http://reimagery.com/fsfd/

https://www.rarewares.org/rrw/programs.php

https://www.japheth.de/

https://www.elhvb.com/webhq/download/index.htm

https://4dos.info/dlinks.htm

http://www.chebucto.ns.ca/~ak621/DOS/Websites.html

https://freelinks4u.tripod.com/dos.htm

https://archive.org/details/softwarelibrary_msdos

https://oldos.org/

http://www.retroarchive.org/dos/

https://www.uselesssoftware.com/

https://www.freedos.org/books/cprogramming/part1/

http://www.resoo.org/docs/dos/free_software/keyb.htm

https://www.vogons.org/viewtopic.php?t=23789

http://qbasicgui.datacomponents.net/

http://theguiblog.com/guipedia.php

https://github.com/cout/vbdos

https://dosgames.com/utilities.php

https://www.plop.at/en/dostools.html

https://www.pc-tools.net/dos/dosutils/

https://archive.org/details/DOS_Power_Tools_Paul_Somerson

https://www.dosdays.co.uk/index.php

https://www.elhvb.com/webhq/download/index.htm

http://www.retroarchive.org/cdrom/index.html

https://clasqm.github.io/freedos-repo/

https://github.com/clasqm/freedos-repo

https://web.archive.org/web/20221129201724/http://jumpjet.info/Application-Software/DOS/dosfiles.htm

https://dosprograms.info.tt/

https://www.devdungeon.com/wiki/doku.php?id=operating_systems:freedos

https://archives.scovetta.com/pub/simtelnet/msdos/fileutil/

https://psychocod3r.wordpress.com/2019/02/19/adventures-in-virtualbox-dos-windows-3-1-and-other-blasts-from-the-past/

http://web.archive.org/web/20080514152935/http://www.devedia.com/dosghost/dos/utildos.htm

https://www.hiren.info/downloads/dos-files

https://archives.scovetta.com/pub/fehq/DOSUtils/00_index.htm

https://multiboot-windows.dscloud.me/All_Pages_for_DOS_GUIs/DG00_My_Pick_Of_DOS_GUIs_Worth_Trying.html

http://www.lanet.lv/simtel.net/msdos/editor-pre.html

http://www.manmrk.net/tutorials/bootdisk/index.htm

http://omolini.steptail.com/mirror/dcee/Files/

http://startup.retropc.se/dos.html

http://files.mpoli.fi/software/PROGRAMM/

https://www.fgrim.com/mgrx/

https://dosprograms.info.tt/apps.htm