

VIRTUAL

The Virtual Memory Program for the Macintosh II Computer

CONNECTIX



This page left blank in original.



Virtual Memory Operating System Initializer
for Macintosh™ II Computer

**User's Manual
&
Installation Guide**

This page left blank in original.

Table Of Contents

Guide to Operations

User Summary (for people who don't read manuals).....	1
What it is.....	2
Tips on using Virtual™.....	3
Tips on using MultiFinder™.....	7
How Virtual Memory works.....	10
For Developers.....	14
Commonly Asked Questions.....	16

Installation Guide

Installation Summary (for people who don't read manuals).....	22
Hardware Installation.....	23
Software Installation.....	29
Installation Troubleshooting.....	32

This page left blank in original.

User Summary (for people who don't read manuals):

- ➔ Virtual requires both hardware and software installation. See the Installation Guide at the back of this manual.
- ➔ Hold down the ESCape key at startup to disable Virtual's operation.
- ➔ For optimum performance, use MultiFinder™ whenever possible, and turn off the "Ram Cache" on the control panel.
- ➔ Register your copy to receive update information!
- ➔ Read the manual — you might learn some valuable tips!

What it is:

Virtual™ is an operating system initializer for the Macintosh Operating System which allows standard applications to run on the Mac II using virtual memory. Virtual memory works by taking information which would normally be stored in RAM, and putting it in a file on your hard disk. This information is transparently retrieved from disk whenever it is required, giving the appearance of much more RAM than is actually installed on the machine. Use of this product requires a Mac II, a hard disk with at least 8 Megabytes of space available, and the Virtual hardware and software. Installation of the hardware and software is covered in the Installation Guide section of this manual.

Certain hardware and some specialty applications may not operate while running under the current version of Virtual. Hardware cards which are not usually compatible include any card that acts as an "Alternate Bus Master." These cards bypass the CPU when transferring information into memory, writing to memory directly. When these cards are running in a virtual memory environment, the CPU doesn't have a chance to intercept and translate the card's information to the proper virtual memory address.

Specialty software which may not be compatible includes low-level debuggers or disassemblers, and programs which bypass the Device Manager and perform operations directly on the hardware. For example, MacsBug 6.0 does not work under the current release of Virtual. This usually only affects software developers.

Finally, some SCSI devices which do direct SCSI input or output may not be compatible. This may include some tape drives and image scanners.

All of these potential conflicts are being worked upon for future releases. If you have any problems, please let us know.

Tips on using Virtual™:

Virtual expands your memory to the 8 Megabyte maximum amount allowed under the Macintosh Operating System. For installation instructions, refer to the **Installation Guide** at the back of this manual. While using Virtual, keep in mind the following tips:

- **Escape key at startup**

Even after the Virtual software is installed, disabling Virtual's operation is easy: Just hold down the escape key (ESC) at startup. If you decide for some reason that you don't normally want Virtual to load during the boot process, you can remove it from the System Folder or 'hide' it in a sub-folder within the System Folder. It's best not to try to rename the program.

You can use the escape key trick whenever you suspect that there is a problem running a particular application under virtual memory — it's easy to try running it with Virtual disabled. Also remember to check on the application's compatibility with MultiFinder.

When you hold down the escape key, you may often also want to bypass the loading of MultiFinder, and start up with the Finder™ instead. To do this, hold down the command key during system startup.

- **Use MultiFinder whenever possible**

MultiFinder and Virtual seem made for each other, and most users will find that MultiFinder helps them get the most out of their expanded workspace. Without MultiFinder, some applications actually exhibit degraded performance due to inefficient use of very large memory partitions, so avoiding the use of Finder can improve responsiveness as well as increase utility.

- **Don't quit things (until you have to)**

The first instinct of almost all new users of MultiFinder when they have finished with a program is to quit; an automatic, almost knee-jerk reaction for the "command-Q" key. By not quitting an application when you finish, not only is it quickly available in the future, but (more importantly) you do not fragment memory. The tendency for memory to

VIRTUAL™

become fragmented and less useable is one of the most frustrating aspects of the current release of the Mac's system software. So unless you are absolutely sure you won't be needing the program again, just switch back to MultiFinder (by selecting it under the Apple menu) instead of quitting, and start up your next application.

- **Turn off the RAM Cache on the control panel**

Any use of the RAM Cache actually slows down the machine when Virtual is enabled. The RAM cache works by storing frequently used disk sectors in RAM memory. With Virtual, RAM memory is stored on the disk. By using both, disk sectors are stored twice on the disk, slowing down operation. If you have a third-party disk cache program, use it only if it has a mode in which it will cache floppy disk accesses only.

- **To Re-install**

If you receive an update for Virtual, or if the software seems to have been corrupted by a virus or crash, you can easily re-install Virtual from the distribution disk: Restart the machine without Virtual running (by holding down the escape key when booting) and then drag Virtual's icon to the trash. Immediately put in the Virtual distribution diskette, copy Virtual from the floppy into the System Folder and restart the machine. [If you do other things after discarding Virtual and before restarting, the contiguous block of space on the disk which has been created and set aside by Virtual might get chopped up into smaller pieces and the next time the software tries to start up, it may not be able to find a single block large enough remaining on the disk. If the program comes up with the "Not Loaded" icon after re-installation, then there is probably not enough contiguous (all in one piece) space available. To remove the disk fragmentation, refer again to the Installation Guide of this manual.] At the next restart, the new copy of Virtual will be installed.

- **Hardware add-ons which may not work with Virtual**

As mentioned above, some hardware may not currently operate while running under virtual memory. This includes slot cards which act as Alternate Bus Masters and write information directly into motherboard memory, thus bypassing the CPU and the virtual memory address translation process. Other potential hardware incompatibilities examples include programs which do direct calls to hardware devices and bypass the Mac's Device Manager. This may include some SCSI scanners, tape

VIRTUAL™

drives, or other SCSI devices. If you have a device which will not operate under virtual memory, read the Commonly Asked Questions section to learn tricks which might be of some help.

- **Cautions**

Programs which depend upon real-time response may not work as desired in a virtual memory environment. The most common examples are **sound** and **animated video** programs which may "stutter" as disk accesses are performed from virtual memory. Real-time feedback or control systems dependant upon guaranteed-minimum or very fast response times should also not be run in a Virtual environment.

- **Programs which seem to have problems**

Specialty applications like low level debuggers may not operate under the current release of Virtual. Some debuggers take over low-level system vectors in a way which makes operation in a virtual memory environment very tricky. Connectix will be working to expand the areas of compatibility with future releases.

Some programs seem to be slow (especially at startup or when performing some particular function) while running under virtual memory. A common problem is that when they are given a huge block of memory, some programs grab and initialize the block. When running in RAM, this process takes just a few tenths of a second, but running under virtual memory, we have to go through and write initialization values to large blocks of the hard disk. This process can take much longer (more than 30 seconds). This is another way in which MultiFinder can sometimes be of significant help — try running the application under MultiFinder after reducing the size of the memory partition allocated to the program (when possible, reduce it to an amount less than the amount of RAM actually in your machine). See the section on MultiFinder Tips for how to set this value. This can increase performance dramatically.

- **Thrashing situations**

Very occasionally, you may find yourself in a thrashing situation where the disk seems to be doing non-stop accesses, and your Macintosh is unresponsive (although the cursor will still move on the screen). This process can continue for some time (a minute or more) before things calm down. If at all possible, its best not to interrupt the Mac in this state, but let it work it out. If the thrashing seems persistent, try reducing the size

VIRTUAL™

of the partition allocated to the program in MultiFinder. Some programs seem to generate such situations more often than others, and often it is just a particular operation in that application.

Tips on using MultiFinder:

The following is intended to summarize and complement the **MultiFinder User's Guide** shipped with your Macintosh. If you are a new MultiFinder user, you may want to first review that booklet to become familiarized with the new features available to you through MultiFinder.

- **To start using MultiFinder**

The easiest way is to use the "**Set Startup...**" feature under the "**Special**" menu in the Finder. You may also select the applications you want to have loaded upon restart (highlight them before selecting Set Startup). If the applications you want to start are not all in the same folder (and so cannot all be selected at once), you may either drag them to the same folder, or after starting MultiFinder, go through and start the applications one by one. Set Startup will then let you choose to bring up all of the running applications after restart.

Another way to start MultiFinder is to open your System Folder (best done with the option key down), and double-click the MultiFinder icon while holding down the option and command keys.

- **Command key to bypass the use of MultiFinder at startup**

If you have MultiFinder selected at restart, but want to start up in the Finder instead, hold down the command key (and the command key only) before MultiFinder is loaded in the boot process. This trick is especially helpful when you have a number of applications set for startup which you want to bypass loading, or if you have disabled the use of Virtual, and don't have enough memory to load many applications otherwise.

- **How to allocate space under MultiFinder**

After you are up and running under MultiFinder, the look of the "Get Info" box (which appears after you have selected an application and chosen "Get Info" under the Finder's File menu) has two new fields. The editable box (Application Memory Size) lets you select how much of memory you would like to allocate to this particular application when it is running under MultiFinder.

VIRTUAL™

We have found it to be a good idea to increase the partition that Finder itself receives when running under MultiFinder. Use "Get Info" on the Finder and increase the Finder's Application Memory Size from 160K to at least 320K (and preferably 512K). This enhances file duplication, copying, *etc.*

It is also a good idea to set larger size partitions for many other applications as well. A value too small can limit the utility of the application, while too high a value might force excessive and unnecessary virtual memory activity. Experiment to find the best setting for you. It takes some time to break the old mind-set of conserving memory and to adopt one of conserving your time and effort.

- **Don't quit programs (until you have to)**

To avoid memory fragmentation as mentioned above, don't quit programs until you find yourself getting low on memory. Who knows? Maybe you'll need the application later in the day (or the week!). Many applications will let you close all their windows, and can be almost transparent until needed. Starting up an application is often one of the most time-consuming tasks involved in using it, and not quitting will keep the number of starts required to a minimum.

When you are looking to free up a block of memory to start another program, check the "**About the Finder...**" selection under the Apple menu when running the Finder. Not only do you have to clear out enough space to load the new program (as specified in its "Get Info" box), but the blocks which you empty must be next to each other. About the Finder shows the current size allocated to each block, but also displays the programs order next to each other in memory.

- **Some programs must be loaded in special areas.**

The most notorious example here is Microsoft Excel™, which must run in the lowest 1 Megabyte of memory. MultiFinder knows about some of these exceptions, and tries to reserve these zones until they are required, but sometimes you can have lots of memory available and still not be able to load a program. If you use Excel, you may want to start it first (even if you don't need it right away) just to have it reserve its place. (Updated versions of Excel which address this problem are said to be forthcoming).

VIRTUAL™

- **Use the Apple menu to switch between applications**

When you have a lot of applications open, it is often much faster to shift to another application by selecting it from under the Apple menu as opposed to cycling through all open applications by clicking on the icon in the upper right corner of the screen. Suitcase™ can make this process even easier by suppressing the desk accessories listings when the option key is depressed before the Apple menu is selected.

- **Use function keys whenever possible.**

Function keys and command-key equivalents can really speed and ease the task of getting around when lots of windows are open. For example, if you are in the Finder and the desktop is hidden by a large window which prevents you from clicking on a Floppy disk you just inserted, try just hitting the command-O key (this will open the last item selected, which should be the disk you just inserted). Similarly, if you can't drag a diskette to the trash, try option-command-E to accomplish the same end. There are lots of shortcuts like this, and many more can be programmed or assigned using some third-party function-key software.

- **When possible, avoid arbitrary moving and resizing of windows.**

Every time you move or resize a window and expose other applications running underneath, all of the windows newly exposed have to be updated. Usually this is no big deal, but sometimes (especially when you are running lots of applications at the same time) under virtual memory, each application must swap itself in, update any small window zone just exposed, and swap itself back out again — the process becomes time-consuming. After a little practice, it is easy to learn how to set up your windows on the screen to minimize this problem. Use large windows to hide less frequently used applications in the back. Try to keep the right edge of your main display clear to have access to your disks and the trash when doing Finder operations. Use function keys or the Apple menu to switch between applications.

How Virtual Memory works:

This section is only for those who have some idle curiosity of how their new virtual memory software works. An understanding of these concepts is in no way necessary to enjoy the benefits of virtual memory.

Virtual memory has been used for many years on mainframe and minicomputers, but Virtual on the Macintosh really represents the first time virtual memory has been available to any program running on a popular personal computer operating system.

General Concept

When we want to expand the working memory space of a computer (to add more storage locations at which the computer can address values to be read or written with values) we have a wide choice of what kind of storage to add. There are lots of different ways to store information, all the way from very fast static RAM, to DRAM, to hard disks, to floppy disks, to tape storage, to writing things down on a piece of paper, and then the user later having to type it back in.

Normally a distinction is made between directly addressable locations, like RAM memory, and Input/Output (I/O) devices, like disk drives. Virtual memory works by blurring this distinction. Instead of adding expensive DRAM to expand memory, we can allocate some part of an alternate storage device — create a file on the hard disk — and use that file to store the information in our expanded memory space.

Later, when the computer is running, special hardware (provided by the PMMU or the CPU's MMU) detects every access to any location in this expanded virtual memory space. First it checks to see if the information is already stored somewhere in real DRAM memory. If so, the proper current address of that information (called the physical address) is applied to the memory hardware, and the memory responds with the appropriate value requested. Note: the physical address (where the requested information actually resides) may not be the same value as the 'logical' address (which is where the CPU thought the information had been stored). This represents a translation of addresses, and is called

VIRTUAL™

“remapping” an address from one location to another. Tables accessible to the MMU keep track of the latest translation values; a large fraction of the time, the MMU can perform this address translation lookup for the CPU in less than one-half clock cycle, which means that there will be no delay caused by the translation process.

Sometimes the requested location is not anywhere in the semiconductor memory, but is instead stored out on the hard disk. If this is the case, the processor saves its state on the stack, and looks for a zone of memory which it doesn't think will be needed for a while (perhaps one that hasn't been getting much use). It takes the information from that zone and writes it out to the hard disk, so it can use the space just vacated to read in the information which is stored out on disk. After the requested information has been retrieved from the disk, the address translation tables are altered so that the information appears to be at the 'logical' address where the CPU originally sought it. The processor is then free to restore its state from the stack and resume execution just where it left off. When it tries to access that information again, this time the information will be in DRAM.

Given a sufficiently fast hard disk, and sufficiently intelligent virtual memory management software (to minimize the number and frequency of swaps necessary) virtual memory performance can approach that of DRAM for many types of applications. This is because the vast majority of operations the computer performs are sequential in nature, and accesses to memory don't jump around in a totally arbitrary manner, but tend to progress across zones or clumps of memory. In the worst case, if every access to memory resulted in an access to the disk file, the computer would proceed at a snail's pace; more than a thousand times slower than normal. However, for the vast majority of real-world applications, this situation simply never occurs.

A Page swap

One standard method of minimizing the overhead of swapping locations in from external storage is to group a whole bunch of locations together into a unit called a “page.” After all, if we just requested an access to a particular location, there is a good chance we will need to access the location right after it in the near future. And besides, when we read from the disk it's lots faster to read a bunch of consecutive locations than it is to retrieve information a byte here and a byte there. The term “Page

VIRTUAL™

Swap" then refers to the action of writing one page out to disk to free up space in order to read another page back into memory. Common page sizes are 256, 512, 1K, 2K, 4K, and 8K blocks.

Demand Paging

There are several methods of deciding when is the best time to swap a page: One of the most obvious, (and a method often having excellent performance) is to swap a page only when it is needed, or required. This is called Demand Paging, since the page is retrieved on demand.

An alternative approach is the "Working Set" theory. Here, the page management software attempts to figure out what pages are likely to be used in the near future, and swaps those pages in before they are needed. For example, if you just shifted under MultiFinder to a different application than the one you were just using, the software might swap in all of that application, rather than just the pages which were needed on demand. This can have both positive and negative effects: if you just popped into the application to do a quick copy, you could find yourself waiting to swap in a lot of pages you are never going to use. It is easy for the software to "outsmart" itself. On the other hand, while running that application, you may not have to wait for any further page swaps to occur.

Virtual uses the Demand Paging approach, which means that it will effectively load up only those sections of an application that you use. The first time you use a new function or pull down a menu, there may be a brief pause while code for that function is loaded. But thereafter, any subsequent use should find the code already loaded and ready for execution. It will keep just those sections of those programs you most need and use in memory.

pseudo LRU algorithm

To do this, the software uses what has been called a pseudo Least-Recently-Used function. Every time you use any memory location on any page, extra hardware on the PMMU chip remembers that page has been accessed. In addition, if the location is modified, that information is stored as well. (The amazing thing is that all of this information can be continually updated with virtually no degradation in performance!) When it comes time to do a page swap, the computer looks through all the pages, and just selects the one which hasn't been used in the longest

VIRTUAL™

time — the Least Recently Used page. As time goes on, all of the frequently used information stays in memory, while infrequently used information spends most of its time on the disk.

We can speed things up even more by taking into account whether a page has been changed at all while it has been in memory. If no information on the page has been changed, we can save ourselves some time by not bothering to write that page out to disk — the information already stored on the disk is the same as that in memory. Since the time to do a disk write is almost half of what it takes to do a page swap, we can increase our page swap performance by nearly a factor of two if we preferentially select pages which have not been modified. This is the reason that the Virtual file occupies a full 8 Megabytes on the disk, instead of just the amount needed to supplement the memory in your machine up to 8 Megabytes — the space is used to store the image of every page in memory, in case that page remains unchanged.

From another viewpoint

If all of this seems confusing, an alternative viewpoint on the whole virtual memory process is to imagine that the entire memory space of the computer has been moved from RAM out to the hard disk. Then a very fast, intelligent hardware disk cache is set up in memory accessible to the CPU, so that frequently used disk sectors (pages) are the only ones kept in RAM. The net effect of this process would be very close to the implementation of virtual memory.

For Developers:

Although Virtual is transparent for many applications, others require an intimate knowledge of the machine's memory map for either performance or alternate bus master use. Connectix is very interested in supporting other third party developers with the calls and documentation required to support such applications in a virtual memory environment.

Guaranteed zone of resident, 1:1 page mapping.

For applications which do not require large blocks of reserved memory, the first 64K of memory are guaranteed to have a 1:1 logical-to-physical mapping. The problem now is to somehow get hold of memory down this far in the system zone. Once a beach-head has been established, any transfer out of this memory which is performed by the CPU is of course able to support translation to the full virtual address space. In this manner, a rotating buffer scheme will allow continuous transfers from a slot card into memory.

Other calls.

Such transfers are not always possible to establish in a DMA environment. Connectix has a set of proposed memory management calls which will allow smoother operation in a virtual memory environment. These calls serve to:

- 1) establish whether virtual memory (any non-unity memory mapping) techniques are in use
- 2) Determine Page Size
- 3) Communicate the logical-to-physical address translation
- 4) Lock and unlock zones of pages for performing transfers

VIRTUAL™

More information

Such standards should not be created in a vacuum, and so we look forward to hearing from developers, both to distribute this information and to receive input on extensions and exceptions. Please address all correspondence to:

Connectix Corporation
125 Constitution Drive
Menlo Park, CA 94025

Tel: (415) 324-0727
Fax: (415) 324-2958

Commonly Asked Questions:

- **Can I set Virtual to emulate less than 8 Megabytes of virtual Memory?**

No. A design decision was made that the problems of creating and uncreating contiguous blocks of disk storage every time a different amount of virtual memory was desired, versus the simplicity and transparency of operations with a fixed 8 megabyte size leaned heavily toward filling the operating system to its limit. We think you'll agree (especially after some use) that too much memory is far superior to too little.

- **Why only 8 Megabytes?**

8 Megabytes is the maximum amount currently recognized by the Macintosh Operating System, so Virtual fills your memory to the maximum currently allowed. Although the Motorola 68000 family uses 32 bit addressing, in the current operating system release Apple uses only the lower 24 bits for address information. This means that under the Macintosh Operating system, 2^{24} addresses can be used, which is equivalent to 16 Megabytes. Of these 16, 8 are dedicated to system RAM, one megabyte is dedicated to each of the six slots, and the rest is occupied by ROM and motherboard I/O addresses. (A Mac II running A/UX™ can address the full 32-bits, but this is another operating system not much like the Mac's). Apple is rumored to be working on system software which will support the full 32 bits of addressing, at which time the Mac II's current memory map would allow a full Gigabyte (1000 Megabytes) of system memory. When this occurs, using the Memory Management hardware you now have available, the amount of virtual memory possible on your machine will not so much depend on the operating system, as on how much disk storage space you can afford!

- **If I have more than 1 Megabyte of DRAM on my machine, will Virtual take advantage of the extra memory?**

Absolutely. The more real memory, the better. Although it is possible to emulate 8 megabytes using just the original one Megabyte of RAM, a significant performance advantage is added each time you increase your DRAM memory. In particular, the first extra megabyte really helps, and we recommend you have at least two megabytes in your machine. Virtual memory allows you to do things which were not possible in the

VIRTUAL™

original memory available, but as you increase the semiconductor memory in your machine, you will see a performance increase.

- **Can Virtual use the extra memory installed on an expansion slot board?**
Not in this release. Probably not until the 32 bit System is available. Under the current OS, only 1 Megabyte is allocated to each slot, and 8 megabytes is the limit. Using a board to reach this limit is not usually the most cost-effective route.
- **The Virtual file is 8 Megabytes big (8192K)! I have a 2 Meg machine, so why isn't the file holding the virtual memory information just 6 Megs big (2+6=8)?**
Performance. When swapping a page in memory that has not been modified, we can speed up the page swap time by a factor of two if we can count on there still being a disk image of the unmodified page. This requires that we be able to store the image of all pages in the virtual memory space, regardless of whether it is in RAM. (See the section on How Virtual Memory works for a more complete explanation.)
- **When I turn on the machine, it looks like it is starting normally, but then it resets itself halfway through the start sequence. What is happening?**
This "stutter-start" is absolutely normal. Right after Virtual expands the memory space, it has to move a bunch of code up to the new top of memory (some important code wants to be loaded at the uppermost memory addresses). This lets all of the memory in the machine be treated as one big block after a startup, instead of being broken up into pieces. The stutter takes place as these code segments are reset into their new locations.
- **I have 2 hard drives. Can I move Virtual off my boot drive?**
Not with this release. This is one of the planned upgrades — allowing virtual memory storage on any remote storage device. For Ethernet users, this means that virtual memory files could be stored on a remote server. One more good reason to send in your registration card!
- **Virtual seems much slower on my machine than on my friend's machine. What am I doing wrong?**
Virtual's performance is heavily dependant upon the access and transfer speeds of your hard disk drive. Those of you who made the investment in a fast, high capacity drive should find that your investment has paid

VIRTUAL™

off well. The amount of real memory is also of critical importance, and the more the better.

- **I still get "Not Enough Memory" errors when I try to start applications, even with 8 Megabytes of memory! What gives?**

Not unlike disk space, memory becomes fragmented. The current operating system is not able to relocate entire applications to coalesce free space in memory. After enough time, the zones get broken down into smaller and smaller pieces. At some point, it is usually best to restart the machine, which will clear the memory fragmentation problem.

- **How much does my machine slow down when using virtual memory?**

There is no fixed answer to this question. There is usually very little performance degradation until you start to use (in an active application) almost as much memory as is installed in your machine. As soon as you start actually using more memory than is installed, the number of page swaps required starts climbing quickly, and the performance decreases. How rapidly this occurs is very dependant on the particular application you are running. Theory can give us a worst-case answer (more than 1000 times slower) but in any real-world practice, this situation never occurs. The only real answer to the question is to try it, and see if the performance decrease on any particular application warrants temporarily running without virtual memory.

- **After I have loaded some programs, even if I later quit them, and I go to the Control Panel to shift monitor display characteristics, my machine crashes frequently. How can I avoid this?**

Some of the desk accessory memory checks are not as robust as those for applications, and control panel devices sometimes run out of memory. This usually results in a machine crash. Try setting your monitor and color characteristics only when you have lots of free memory available.

- **I've got an image scanner (or a backup tape drive) which almost always crashes while running under virtual memory. How can I make it useful?**

The easiest and surest way to get it to work is by disabling the use of virtual memory by holding down the ESC key at restart. If you have time to explore them, some alternate methods may also work: If possible, try reducing the size of the partition the program requests under MultiFinder to a value somewhat less than half of the ram installed in your machine. You can also try to do an operation which will force the

VIRTUAL™

loading of the required pages into memory, like reading in an old scanned image from disk and closing it before scanning a new one. As time goes on, patches to Virtual and to the drivers of such devices may become available to ease incompatibility problems.

This page left blank in original.

VIRTUAL™

Installation Guide

I
N
S
T
A
L
L
A
T
I
O
N

Installation Summary

(for people who don't read manuals):

- ➔ Virtual requires both hardware and software installation.
- ➔ We recommend that the hardware installation be done by a certified Apple technician. **DO NOT TRY TO INSTALL THE HARDWARE IF YOU ARE A NOVICE!** Connectix is not responsible for any damage done to the product or the machine caused by improper installation!
- ➔ Software installation is simple. Just drop the Virtual program into your System Folder, and then restart the machine.
- ➔ Register your copy to receive update information!
- ➔ Read the manual — you might learn some valuable tips!

Hardware Installation:

If you have an A/UX machine, a Mac IIx or any 68030 Macintosh, all of the required hardware is already incorporated into your machine and you can skip this section.[†]

We recommend that the hardware installation be performed by a certified Apple technician. These technicians already have training and experience in the proper installation of the Motorola 68851 PMMU into a Mac II. The installation process is easy, and can be done by appropriately trained personnel in just a few minutes. It is only slightly more difficult than removing or adding more memory to your Mac II, but if you have never inserted a chip in a socket before, *now is not the time to learn* — using an IC worth several hundred dollars, on a machine worth several thousand! Take the extra time and expense to have it done right. Your computer dealer should be glad to install this chip for a nominal service charge.

Do not open or play with the 68851 PMMU package until it is called for in the installation process. It is **sensitive to static electricity**, and has been specially packaged to avoid such damage occurring during shipping. As soon as you open the package, this protection will be lost, and any damage done to the chip will be your responsibility. Stay your curiosity until proper precautions have been taken, and preparations for installation made!

[†] Please note that if you have an '030 Macintosh or already have a 68851 PMMU chip installed in your machine (for example, a Mac II configured for A/UX) you should have purchased this product in the software-only version. If this is not the case, talk to your dealer before proceeding! Also note that a different version of Virtual is required for operation on 68030 machines than for the 68020s; check to make sure you have the proper version for your machine.

The following is included as a guide and/or review for the technically competent technician installing the 68851 PMMU chip. Read through the entire hardware installation process before beginning work. Although installation is not difficult, CONNECTIX CANNOT BE HELD RESPONSIBLE FOR DAMAGE TO EITHER THE PMMU OR YOUR MACHINE CAUSED BY IMPROPER INSTALLATION!

- **The proper environment**

Before you start, take a moment to examine your clothing and environment. This installation should not be performed in an area where static electricity is a potential problem. Wearing a woolen sweater, or any clothing material commonly associated with static shocks is not a good idea (cotton clothes are best). You may want to remove your shoes, especially when working on a carpet. Remember that it takes far less to kill an IC than it does to get even a small shock from static charge.

- **Unplug the Machine**

This is such a simple step, that even old pros can forget! As long as the machine is plugged into the wall, there is the danger that full power could be applied to the logic at any time. An accidental drop of a screw might cause permanent machine damage. Be sure to unplug the machine.

- **Exposing the old MMU**

After opening the case, remove the 4 screws securing the disk-drive platform. We have found that it is not usually necessary to remove *any* of the disk drive cables; if caution is exercised, the entire platform may be gently rested upon the motherboard, leaning against the power supply or video card. This will expose the old memory management unit (MMU), at the location as shown in Figure 1. The old MMU is a large black square IC in a socket just in back of the main processor, and is usually labeled with something like 343-0002-1 ©APPLE'86. There is a raised metal dot at the location of pin 1.

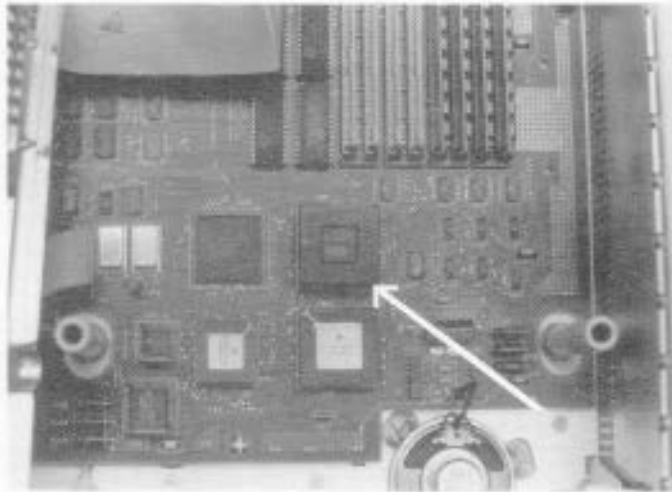


Figure 1

- **Extracting the old MMU**

One of the best, and certainly the most convenient extraction tools we have found to help remove the old MMU is right inside your Mac II — the metal Expansion Cover Shields located at the back of each expansion slot. After you lift a cover shield from its position, the lower portion of the piece makes a nearly ideal “right-angle lever” with which to help gently pry the old IC out of its plastic socket. Wedge the open end of the “L” between the IC and the socket, with about half the width of the shield extending beyond the socket’s edge (see Figures 2 and 3). Gently pry to lift up the corner of the IC. After you feel the first corner just starting to move, switch to the next corner. Once each corner has started lifting, the remainder of the extraction usually proceeds very easily. After the old MMU has been removed, check to make sure that none of the old pins have stubbornly remained in the socket, so that all of the socket’s holes are now clear.

VIRTUAL™

I
N
S
T
A
L
L
A
T
I
O
N

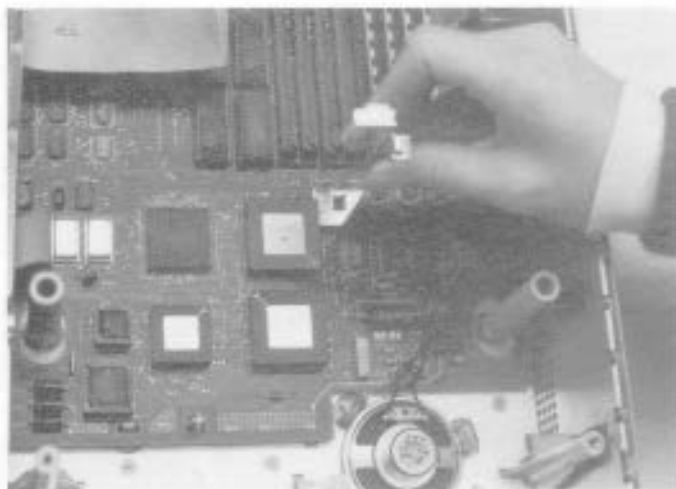


Figure 2

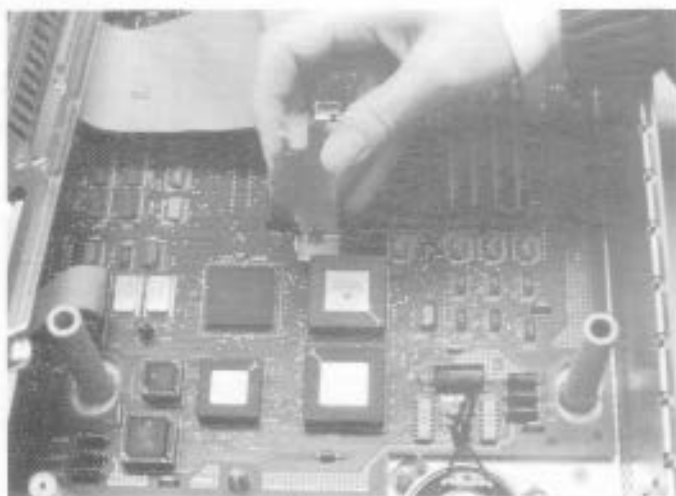


Figure 3

- **Getting to the same potential**

After the old MMU has been removed, bring the packet containing the new 68851 PMMU over to the machine. Touch the bare metal of the Mac II power supply's outer case while gently holding the PMMU packet with your other hand. This will put you, the computer, and the PMMU all at the same potential, so the danger of static damage will be minimized. Try to finish the installation before moving away from the machine. If you must move away, touch the metal case of the power supply again when you first return.

- **Inserting the new 68851 PMMU**

Now you can carefully cut open the static protection bag and remove the PMMU. Remove the protective rubber foam from the PMMU's legs, and take a moment to visually assure that none of the legs are severely bent out of alignment before proceeding. (If such is the case, take great care in gently straightening them using a set of precision needlenose pliers.) Place the IC on the socket from which the old MMU was removed, with the gold arrow on the top of the chip pointing to the "1" in white silkscreen on the motherboard (Figure 4). **Make sure this alignment is correct before proceeding!** The PMMU has 132 pins, and so the insertion force required to push it into the socket may seem rather large as compared to smaller ICs. With *firm pressure from directly above* with the thumb, try a gentle rocking motion until the chip is firmly seated in the socket. Try not to let one side of the chip get too much further down than any other side. If you are unsure of the depth of insertion required, compare the position to that of the main processor. **Do not pound the chip into place!** It is possible to crack the motherboard if too much force is applied! Gentle, firm pressure should seat the IC until the ceramic body rests slightly above the socket.

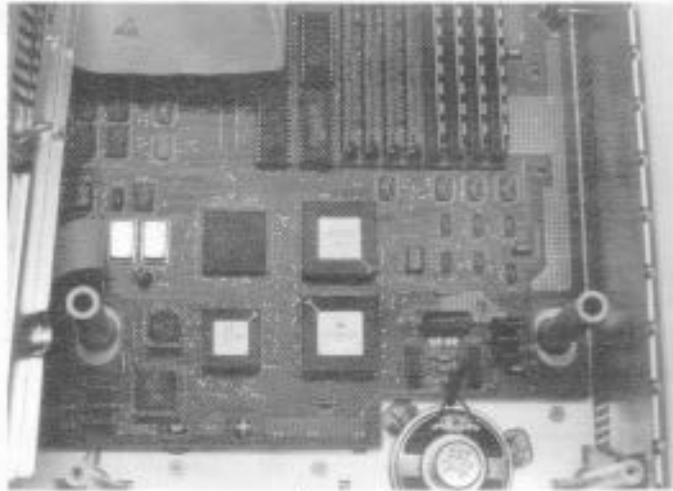


Figure 4

- **Closing up**

Lay the disk shelf back onto its normal resting position, and resecure the 4 screws. Replace the metal slot-shield used for extraction of the old MMU back into its appropriate position. After the machine is closed back up, take a moment to plug it in and test for normal operation; the machine should boot and operate normally.

Your machine will run fine without Virtual™ software installed

Addition of the PMMU chip does not change the standard functionality of your Mac, and you'll be able to run applications just as before whether or not you install the Virtual software.

Software Installation:

Drop Virtual into the System Folder

To install Virtual in your machine, start up your Macintosh, put the Virtual program diskette into the floppy drive and copy the Virtual program into your System Folder. Go to the Control Panel, and make sure that the RAM Cache is turned off (this will help increase performance — don't use the RAM Cache while running in virtual memory.) The software will automatically install the next time you restart your machine!

Restart Machine

When you restart your machine, one of the icons shown below in Figure 5 should appear on the screen to let you know whether Virtual has successfully installed. [If you don't see either of the icons after restarting, see the section entitled "Troubleshooting" — you probably have more than one System Folder on your disk, and Virtual is not in the proper folder.] If you see the "Successful Loading" icon, you are already up and running with virtual memory and you can skip the next two sections. If you see the "Not loaded" icon, don't worry! Just keep following the directions.



Figure 5

Virtual requires 8 Mbytes of contiguous disk space

The first time Virtual loads, it requires 8 Megabytes of space (8192K available) on the hard disk (the reasons for this are discussed elsewhere in this manual). This disk space must be *contiguous*, which means not broken up into little pieces scattered across the disk surface, but rather in one entire 8 Megabytes block. [Creating and deleting files on a disk

causes the space on the disk to undergo *fragmentation*, which means that although there may be many blocks of open space on the disk, all of which add up to a large amount of free space available, they are not together in one big block. Since keeping all of the virtual memory file together in one place on the disk can significantly speed up the access and response time of virtual memory, *Virtual* will not install itself unless it sees one big, contiguous block it can fit into. This means that even if your hard disk shows that there is enough space (more than 8192K available), *Virtual* may not be able to find a single large enough block unless the disk free space is sufficiently unfragmented.] If your software did not automatically install during the first restart, there is probably not a big enough block available on your disk, and you must *unfragment* or *coalesce* the free space on the disk.

Unfragmenting a hard disk

There are several ways that we can get all of the free space on a disk together into one big block. Several commercial disk utility programs (most notably Disk Express™ by AlSoft) have easy to use and well documented features for removing disk fragmentation. Getting rid of such fragmentation can have the side benefit of speeding up disk accesses, and you may even seem to have a faster, more responsive hard disk drive afterwards. If you do not have such a utility, there is the "brute-force" method: back up the contents of your hard disk in a file-by-file manner (not the "image copy," possible with some backup programs or devices), erase or reformat the entire hard disk, and then restore the files to the hard disk from backup — the newly restored files will leave all of the volume's free space in one block at the end. Once your disk is unfragmented, restart the computer and *Virtual* should load successfully.

Keep the original diskette!

It will be difficult to save the virtual file from the hard disk after it is installed because of the file's large (8192+K) expanded size. Keep your original floppy in a secure location as a backup (you may want to move the write-protect plastic slider on the diskette to the "see-through" position, which will lock the diskette from being altered until you move the slider back). This diskette is not copy protected; we're trusting your honor to make no copies other than for your own archival purposes.

VIRTUAL™

Send in the registration card to receive information on software updates!

We can't send you update information on new software releases if we don't know where you are! Apple may eventually update the operating system to work with full 32-bit addressing, at which point we can release a version of Virtual which will allow up to a full Gigabyte (1000 Megabytes) of virtual memory capability (or at least as much hard disk space as you can afford)! Additionally, bug fixes or significant feature enhancements may prompt an updated version prior to that time. Take just a moment now to fill out the registration card, to keep on the cutting edge of technology in personal computers.

Installation Troubleshooting

Hardware Installation Problems:

A complete treatment of all the potential problems incurred while installing the PMMU is beyond the scope of this manual. Again, let us recommend that even though the installation process can be very easy, since expensive hardware is involved, you should have a qualified technician perform the hardware installation.

If operation is intermittent, make sure that the PMMU is fully seated in its socket (compare again to the main processor). Check that none of the disk drive cables have worked loose during installation or transport.

Software Installation Problems:

Most software installation problems can be traced to one of the following problems:

The software is not resident in the "blessed" System Folder.

If you have copied Virtual into the System Folder on your hard disk, and neither of the icons shown in figure 5 appears when you restart your machine, the software has probably been placed in the wrong folder. Although it is not a wise idea, there can be several folders on a hard disk with the name "System Folder." For that matter, it is not even necessary that the folder which contains the Finder and System you are using be called "System Folder;" many people call it "System Stuff" or "Startup" or some other custom name. It is not usually a good practice to keep more than one System on your hard disk, but it is much more important that you at least identify which System you are using. Even if there is more than one 'System' on your disk (you can use the desk accessory "Find File" to see if there is) only one of the Systems is used during the boot process. The folder of this system is known as the "blessed" folder, and it is normally the last folder into which both a System and a Finder have been deposited. Recent versions of the Macintosh Finder display a small Macintosh in the icon of the blessed folder. Virtual is an INIT, and must reside in the same folder as the blessed System and Finder if it is to be installed at startup time.

There is not enough contiguous space available on the hard disk.

After Virtual has been deposited in the proper folder, there still must be enough room left on the hard disk drive for the 8 Megabyte file required for Virtual to operate. (When you are running your Macintosh, the information stored in the expanded virtual memory you are using must have some place to reside. Instead of being stored in expensive semiconductor RAM, this information is placed into a file on the hard disk drive. This file must be large enough to store all of the information which could ever be placed in virtual memory. Therefore, to make a space for 8 Megabytes of virtual memory, there must be at least 8 Megabytes (8192 K) of space available on your hard disk.) In this version of the software, there is no means of decreasing this requirement.

If your software still displays the "Not-Loaded" icon after enough space has been cleared on your hard disk, re-read the section on "Unfragmenting a Hard Disk."

Too many INITs or conflicting INITs are being loaded at startup.

Unfortunately, not all INIT files work together. However if your machine boots with Virtual disabled (hold down the ESC key at restart to check this) it probably can be made to work under virtual memory. The full virtual memory space is not available until Virtual has been started, and so after virtual memory has been set up, Virtual must re-load all of the INITs (and debugger, if any) which were loaded before it. In order to reduce this problem and speed the boot process, Virtual tries to be loaded as soon as possible in the boot sequence. It does this simply by a judicious choice of name: Since INITs are loaded in alphabetical order, the program is named `_Virtual` ('space'Virtual). Some other programs pull the same trick, and may get loaded before Virtual, but so long as they do not grab excessive amounts of memory this should not be a problem. If you suspect the loading order may be at issue, try renaming the files to change the order.

There could also potentially be problems in the loading of debuggers and/or disassemblers, not all of which currently run under Virtual. If you have a third party debugger (usually renamed as MacsBug) or disassembler, try dragging them out of your System Folder.

VIRTUAL™

Lastly, remember to make sure the "RAM Cache" has been turned off on the Control Panel, and that nothing is leaning on your ESC key during reboot.

This page left blank in original.

This page left blank in original.

This page left blank in original.

CONNECTIX CORPORATION
125 Constitution Drive
Menlo Park, California 94025
(415) 324-0727