COMPAT

Version 3.2

Direct Disk Interchange Utility

MYCROFT LABS INC

P.O. Box 6045
Tallahassee, FL 32314

Kaypro Installation Notes

It is never a good idea to use your master disk in the microcomputer when sending or receiving data. Through a mistake or equipment failure, this disk may be destroyed. Therefore, you should make a working copy of your Compat Master disk. Since your Compat Master disk comes in a single sided double, density Kaypro 2 format, do not try to use any of the other disk copy utilities available to you. This may damage the disk. To make copies, follow the steps below.

KAYPRO 10 WITH HARD DISK

1) Cold boot your computer from the hard drive by turning the Kaypro on without the floppy in the disk drive.

2) Exit to CP/M, if you are using the menu system you should be at the A0> prompt.

3) Put your Compat Master disk in the floppy drive.

4) Type PIP A:=C:*.*[OV] <CR> (OH-VEE, not ZERO-VEE)

5) You now have the files on the hard drive. If you wish to make a backup copy, do so. If you put a double sided disk in C:, reboot the system before using.

OTHER KAYPROS WITH DOUBLE-SIDED DRIVES

1) Put a write protect tab on your Compat Master disk.

2) To cold boot, turn your Kaypro on. Put your operating system master (or a copy of it) in drive A.

3) Put a blank disk in drive B.

4) Format the blank disk, and put an operating system image on the disk (make it bootable). The procedure for formatting a disk and making it bootable varies from Kaypro to Kaypro, so you will have to consult a manual for your computer.

5) Copy the PIP.COM program from the system's disk onto the bootable blank disk by typing the following at the A> prompt: PIP B:=A:PIP.COM[OV] <CR>. (OH-VEE, not ZERO-VEE)

6) The disk in drive B should now be bootable and contain the copy utility PIP. Now remove the system's disk from drive A and replace it with the disk from drive B.

7) Reboot the system by typing ^C (hold down the CONTROL key while typing C). If the system does not reboot and return an A>, you most likely forgot to put a system image on the disk.

8) Put the Compat Master disk in drive B.

9) At the A> prompt, type
PIP A:=B:*.*[OV] <CR> (OH-VEE, not ZERO-VEE)

04/22/85 TS JAT

Forward

COMPAT is a system utility developed by Mycroft Labs, Inc., for CP/M-80 that allows you to format, read and write diskettes in any of a wide variety of CP/M formats. It also supports transfer of files between any of the supported CP/M formats and any of the four most common MSDOS / PCDOS / ZDOS diskette formats. Finally there is a generalized file copy utility called MYCOPY, which is a powerful replacement for PIP. This last program is also quite useful outside of COMPAT.

COMPAT is available for several CP/M-80 based microcomputers, including the following:

System(s)	Drive Type	Supported Drives	Default Drive
Kaypro II/83, II/84, 2X	SS 48tpi	A,B	B
Kaypro IV/83, IV/84	DS 48tpi	A,B	B
Kaypro 10	DS 48tpi	C	C
Sanyo MBC 1000, MBC 1150	DS 48tpi	A,B	B
Sanyo MBC 1250	DS 96tpi	A,B	B
Zenith Z100 (under CP/M-85)	DS 48tpi	A,B	B
Zeinth Z100Q (under CP/M-85)	DS 96tpi	A,B	B
Zorba 7	DS 48tpi	A,B	B
Zorba QD	DS 96tpi	A,B	B
Memotech MTX 512	DS 48tpi	B,C	C

Table Of Contents

1. General Information

1.1 Capabilities

This package consists of several utilities that allow you to do the following things:

1) Initialize blank diskettes in numerous CP/M and MSDOS formats
2) Convert either (or both) drives into any of the CP/M formats
3) Do directories of, and copy files to/from various MSDOS formats
4) Transfer information between diskette files and/or devices

Once a drive has been 'converted' into a foreign format, you can use diskettes of that format as if they were on a real machine of that type. You can log into them, use 'built in' commands like DIR, execute programs (8 bit .COM files only - you can read and write CP/M-86 disks, but not run the programs), use programs such as PIP or MYCOPY to transfer files to or from them, etc.

The diskettes created by these utilities are indistinguishable in every way from ones actually created on a real hardware system of that type, except where specifically noted in this document, such as the Z100 formats, and 48 tpi disks written on 96 tpi drives.

For a list of formats supported by the current release, run through the menus in either utility. Section 7 contains a list of supported formats as of the writing of this manual. Four MSDOS formats are currently supported via MS. Note that only the root directory is accessible on MSDOS 2.x disks.

The Zenith Z100 formats (Z100S and Z100D) have been modified slightly for reasons of speed. On the original formats there was a 1:1 interlace, which the real BIOS can (just barely) keep up with. The COMPAT replacement BIOS will accept the original 1:1 disks, but runs very slowly (about half normal speed on disk accesses). By changing the interlace on Z100 diskettes created with our FORMAT utiltity to 2:1, there is little loss of speed under the normal CP/M-85 BIOS, but disk accesses are actually notably faster with our 2:1 disks under the COMPAT BIOS than with 1:1 disks under the original BIOS. Should you want to maintain exact compatibility with the original (1:1) format, format your disks with the Zenith FORMAT utility instead of ours. Everything else will work as usual, albeit somewhat slower when COMPAT is in effect.

1.2. System Requirements

COMPAT is available to run on several popular microcomputers. It is assumed that there have been no hardware modifications to the standard systems (additional or larger capacity disk drives, etc.) and no modifications to CP/M (replacement CCP, such as ZCPR, etc.). When COMPAT is running, the user program area (TPA) is smaller by about 5K than it is normally. If you want to run programs that require this room, use COMPAT to transfer the program(s) to a native format disk, remove COMPAT, and then the program(s) should run just fine.

1.3. Caveats

Even though you may be able to transfer files to or from particular foreign diskettes, there are still certain situations that can prevent full (or any) use of at least some of those files. These situations are not the fault of COMPAT, rather they result from other incompatibilities (such as CRT screen codes), or intentional efforts to prevent use of these programs on other systems.

CRT Screen Codes

The first class of problems results from the fact that there is also no standardization of CRT screen codes, such as what character(s) erase the screen, what character(s) move the cursor to a given position, etc. If your system uses the same screen codes as the system the program(s) originally ran on, you have no problem. In the more likely case (incompatible screen codes), you have two options: re-install the program(s), or use a terminal emulation package. If an INSTALL option (or utility) is available for the program(s), just re-install for your CRT type. If not, there are several packages available for CP/M (such as Talisman from Disco-Tech) that can dynamically translate screen codes for any of about 90 different terminals into the codes your console display requires.

Software That Checks For A Specific Machine

The second class of problems results from situations where software (and or hardware) vendors have taken steps to prevent the transport of the program(s) to other computers. Much of the software that came bundled on the Osborne (such as dBASE II) was modified to check and see if it is running on a real Osborne, and if not, to abort. Other vendors that have been known to do this sort of thing are Heath/Zenith and Microsoft. Some software vendors may be willing to exchange the 'protected' software for a version that will run on your new system for considerably less than the list price of the package. If not, you may well be unable to use that software on anything other than the original system.

Hardware Dependent Code

A third class of problems results from hardware dependent code, such as direct screen access, serial I/O drivers, etc. One example of this would be our own MITE communications package. Even though all console, printer and disk I/O are done through standard CP/M mechanisms, there was not sufficient support in CP/M for the degree of control we needed over the communications channel. Each copy of MITE is customized to work with the specific I/O port(s) available on the machine it is sold for. Hence, even though a version of MITE for the Kaypro 10 could be transported to a Sanyo 1000, it will not function properly on the Sanyo.

Other areas in which this sort of problem might arise are: calls to ROM entry points in the original system, direct manipulation of the screen display memory (common on custom screen oriented editors), and anything doing unusually complicated input/output via serial or parallel I/O ports. There is generally not a lot you can do in these cases, other than see if the vendor sells additional copies for other machines at a reduced rate, once the first copy has been purchased. For example, if you bought (or owned) a copy of MITE for the Kaypro 10, Mycroft Labs will sell you additional copies of MITE for any other machine we support (disk only) at half price.

Copy Protected Programs

A fourth class of problems is associated with copy protection of programs. Virtually none of the commonly used techniques will allow you to run such programs on anything other than the original system the program(s) were purchased for. Our recommendation is to avoid copy protected software whenever possible. There are almost always equivalent packages that are not protected (at least under CP/M). Most reputable software vendors have recognized that the only one really penalized by copy protection is the legitimate owner.

Differences In Operating Systems

A final class of problems is related to differences between operating systems. Although it is possible with COMPAT to transfer any file between CP/M and MSDOS disks, only certain types of files are actually useful once transferred. This would be true regardless of how the files were transferred. The most obvious case is executable object programs. Programs compiled (or assembled) into 8-bit object code can not run on 16-bit systems, and vice versa. In fact, executable programs for MSDOS will not execute properly under CP/M-86 and vice versa. Note that with some 'pseudo compilers', such as CBASIC, Business Basic, etc, the intermediate object files (.INT, .BI, etc) can be transferred and executed properly if a version of the language interpreter (CRUN, BI, etc) is obtained for the new operating system.

A more subtle form of incompatibility is file structure. For example, Microsoft WORD and MicroPro WORDSTAR have very different conventions as to how to store text files. Two different spreadsheet programs may look very similar, but use very different means of storing data and templates. In general, if the same program is available under both operating systems, the data files will be usable once transferred. Even this, however, is not certain. Examples of things we have found to transport with no problems include: simple text files; BASIC source programs in ASCII form; CBASIC .INT files; dBASE II data bases, index files, programs, etc.; Multiplan data files; and Wordstar documents. Other forms of data files may well transfer ok, but the only way to know for certain is to try it.

Note that in certain cases, files transferred between CP/M and MSDOS disk formats may increase slightly in size. This is due to differences in the way the two operating systems allocate disk storage. It should never cause a problem, and in many cases, the next time the file is edited (or re-written in any way) the 'surplus' will disappear.

2. Technical Background

If you are not technically inclined, and are not curious as to why some disk formats are compatible with your system while others are not, skip this section, and just take our word for it. If you are curious, a serious attempt is made here to explain the technology of floppy disk formats so that anyone can understand the essentials.

2.1. Terminology

In order to understand the following discussion, a few technical terms must be explained:

DISKETTE - (DISK, FLOPPY, FLOPPY DISK, FLEXIBLE DISK, etc.). A flexible, interchangable disk type magnetic media intended for data storage and retrieval. Intermediate between cassette tape and hard disk with respect to storage capacity, access speed and cost. Its characteristics make it ideal as a mass storage device for a low cost, low performance computer.

TRACK - A circular region on a diskette that is swept out by the read/write head as the diskette rotates under it. Since the head can be accurately (and repeatably) positioned to any one of about 35 to 80 possible positions along a given radius, there are that many 'tracks' on a given surface of a diskette.

SECTOR - One of several (typically 5 to 20) regions spaced evenly around a given track. This is the smallest unit of information read or written by the disk system in normal operation. In a 'soft sectored' system, there are two parts in each sector, the header and the data parts. The header part contains the side, track, sector number and sector size code, while the data part contains the associated data. Both parts have error detection codes and other overhead bytes appended before and after the useful information. Both the header and the data fields are written when the disk is 'formatted'. Thereafter, only the data field is ever written.

FORMATTING - Writing the header and data fields for each sector of each track. Once this is done, the normal processes of reading a given sector, or re-writing the data part (in place of the original) can take place. In effect, this lays down a 'map' which the controller uses to determine exactly where to start reading or writing the data part of a given sector. Typically all of the data fields are filled with some kind of dummy data, such as all OE5H bytes. In certain cases specific information must be written in particular bytes of one or more sectors, to help a system that supports multiple formats (say both single and double sided) to determine which format a given disk is.

SINGLE SIDED - Some drives (mostly older or less expensive ones) have a read/write head only on one side of the media (the side opposite the label), and a pressure pad on the other side. This allowed writing on only one side of the media at any given time. The 'flippy disk' was developed so that the user could insert the disk either label side up (normal position) or label side down, and hence use both sides of the media, but not at the same time. It is possible to make a flippy disk out of a normal disk by punching one or more additional holes in the black envelope that contains the actual disk media. There are various magazine articles and kits with templates and tools that show how to do this.

DOUBLE SIDED - Later, more expensive drives added a second read/write head on the other side, in place of the pressure pad. This doubled the capacity of a drive with little increase in the cost of the drive or the media. Certain technical problems were introduced by this step that took a while to overcome. One was that a read/write head did not make a very good substitute for a pressure pad, and could not press the media against the active read/write head well enough for reliable signal transfer. Another problem was that the media tended to wear out faster due to passing between two hard read/write heads, and was especially prone to problems when the heads 'loaded'. One problem still present in double sided drives is that you need to insert a 'head protector' (the piece of cardboard that was inserted in the drive when you got your system) anytime the system is transported, and especially when it is shipped. If you have lost the original protector(s) that came with your system, an old discarded disk will work almost as well.

SINGLE DENSITY - Virtually all drives are capable of writing the same number of 'magnetic flux changes' per inch, hence there is no such thing as a single density or a double density 'drive'. The disk controller circuitry used determines whether the disk is written in 'single density' or 'double density'. Older controller designs used a highly reliable, very redundant encoding technique called FM (frequency modulation). In FM, every other flux change is always 'on', and the ones in between contain the actual data bits. Hence a stream of flux changes in FM would look like this:

```
    1    d1    1    d2    1    d3    1    d4    1    d5    1 ...
```

where the '1's represent the clock bits (always ON), and the 'di's represent the data bits (any of which can be ON for 1 or OFF for 0). The advantage is that you never go more than one bit without an ON signal, hence can always stay in synchronization. Another advantage is that if a few clock bits are lost, the circuitry can usually stay in synchronization anyway. Hence almost 50% of the area of the disk is relatively immune to a few 'hits'. The disadvantage is that half the flux changes on the disk cannot be used for data.

DOUBLE DENSITY - Later disk controller designs use a more complex technique for encoding information into flux changes, called MFM (modified frequency modulation). Using this technique, every flux change represents a data bit, but ON does not always represent a '1' (or vica versa). Basically, the mapping of OFF and ON onto 0 and 1 varies with time as a function of past history so that there is never a stretch of more than a few OFF positions in a row. Hence this encoding technique is 'self clocking'. The advantage is that all flux changes on the disk now represent data bits. The disadvantage is that there is now little room for error. Anywhere a flux change is lost, data is lost. Double density rated media must be much higher quality than single density rated media.

SINGLE TRACKED - Some drives (mostly older and less expensive ones) can position the read/write head(s) every 1/48 of an inch, yielding 48 tpi (tracks per inch). On 5.25" drives, the range of head movement is just under an inch. Very early 5.25" drives had 35 tracks, later ones had 40 tracks. On 8" drives, the range of head movement is about 1.5", for a total of 77 tracks.

DOUBLE TRACKED - Later 5.25" drives were modified to position the read/write heads every 1/96 of an inch, yielding 96 tpi (tracks per inch), or 80 tracks. The advantage is twice as much storage with little increase in the cost of the drive or the media. The disadvantage is that the tracks are now twice as close, hence there is 4 times as much interference between tracks (crosstalk) as with 48 tpi drives (and as we all know, 'crosstalk' is a TERRIBLE thing to have in your computer). To correct this, the signal level must be decreased by a factor of 4, and the read head must be 4 times more sensitive.

Since the tracks on a 48 tpi disk fall exactly on the 'even' tracks (0,2,4,...) of a 96 tpi drive, it is possible to read 48 tpi disks on a 96 tpi drive by 'double stepping'. It is possible to write 48 tpi disks on a 96 tpi drive, but the less sensitive read heads on the real 48 tpi drives will usually have a hard time detecting the weaker 96 tpi level signal. Certain tricks will usually allow transfer in this direction, but it is never very reliable. For details, see section 8 (Notes).

2.2. The Original 8″ SS/SD Standard

When CP/M was first introduced, back in 1976, there was really only one diskette format in common use: the IBM 3740 8", Single Sided, Single Density format. There were a few attempts on the part of design engineers to adopt hard sectored 8" disks, but these met with little success. Notable among these early vendors were MITS (the Altair folks), Processor Technology, and ICOM all of whom had proprietary hard sectored 8" formats (along with proprietary disk operating systems). The only early vendor to have the foresight to use a readily available soft sectored format (and more importantly, a transportable DOS from an independent software developer, i.e. CP/M) was IMSAI. By an amazing coincidence, IMSAI outlasted all three of the other early vendors, and CP/M is still firmly entrenched as the standard 8 bit DOS. IMSAI, by the way, is still very much with us. The original company split into two operations: a hardware company that continued to make S100 microcomputers (that shortly died); and one of the first retail computer store chains, Computer Shack. This new chain was soon challenged by Radio Shack, and changed its name to Computerland.

2.3. The Introduction Of 5.25″ Drives

The first vendor to use 5.25" drives in the microcomputer field was NorthStar, who introduced an S100 controller and one or two single sided, hard sectored minidrives. Costing much less than the existing 8" systems (which were around $1500 to $2500 for just a controller and one or two drives), it achieved a market niche. This was in spite of the fact that they ignored CP/M and introduced their own operating system (NorthStar DOS) that was not transportable and was vastly inferior to CP/M. Several third party software developers later released versions of CP/M for this hardware, but due to lack of support by the vendor it never caught on, and consequently NorthStar never became a major player in the hardware market.

Other hardware vendors, such as Osborne and Xerox, shortly released systems using 5.25" soft sectored drives and CP/M, and the race was on. In fairly short order, hundreds of vendors introduced similar systems and it looked like a new industry standard might be possible. Unfortunately, in attempts to outdo each other, and lock their customers into buying software just from them and their dealers, almost every system had a unique 5.25" soft sectored format. Another reason for this tower of Babel was simply that Digital Research had provided incredible flexiblilty in how CP/M could be installed, but made no recommendations as to interchange standards. Furthermore, there was little or no communication between developers in the different hardware companies, many of whom didn't really understand the software issues involved anyway.

2.4. Other Advances: Double Density, Double Sided, 96 TPI

While the 241k available on an 8" single sided, single density drive was barely adequate for normal CP/M operation, the newer 5.25" systems had only about 1/3 that much storage, usually about 80 to 90k. This was close to useless. One of the first attempts to correct this situation was the introduction of double density controllers, which brought the average storage of 5.25" systems up to about 160k to 190k. This was still on the low side, and double sided drives were introduced, which brought the average storage of 5.25" systems up to about 320k to 380k. By comparison, 8" double sided, double density disks had roughly 1200k capacity. Given this, it is amazing how few vendors chose to offer 8" systems. In fact, outside of the Xerox 820, virtually all 8" based systems were S100 designs.

Some of the vendors who got into the game relatively late (such as Sanyo) offered only double sided, double density systems. Others, like Xerox, found themselves having to support not only their double density and/or double sided formats, but also all previous formats they had released. Kaypro (which at the time was called Kaycomp) pulled off a real coup by supporting not only their own double density format, but also the Xerox SS/SD format. In fact, the video display and I/O ports of the Kaycomp were so similar to those on the Xerox 820, that virtually all Xerox 820 software ran without modification on the Kaycomp. Overnight, there was a massive amount of readily available software for an otherwise unknown, brand new system. This was perhaps the major reason for the immediate success of this portable. Another one of the truly staggering things about this field is that nobody else figgured this out, even after Kaypro demonstrated how essential compatibility was.

Meanwhile, 96 tpi 5.25" drives became available, and some vendors added high-end models (such as the Sanyo 1250) that used these drives, yielding as much as 780k per drive.

About this time, IBM reared its ugly head in our heretofore fairly reasonable field, and set progress back about 3 years by destroying virtually all standards that had been laboriously agreed upon. Among the casualties were: CP/M, the S100 bus, numerous good hardware systems (and in some cases entire companies), some 30,000 developed programs, and thousands of man-years of development expertise based on 8080/Z80 processors and CP/M. Most insiders were horrified to find that the state of the art had just regressed from 780k per drive to 160k per drive. Instead of a virtually bug free, widely adopted, well designed operating system we now had a nightmare that the original developer washed his hands of, and in which even proponents freely admitted there were over 2000 bugs. Compound this with a processor that requires a minimum of twice as much memory to accomplish the same thing as 8 bit processors, and you will get some idea of just how IBM 'legitimized' the field.

2.5. Apple: A whole Different Universe

Another real can of worms can be found under the cover of any Apple (or compatible). Here the reason is not predatory marketing, so much as simple ignorance of several important technical concepts. The rank amateurs who designed the original Apple just didn't understand digital recording techiques, and were more comfortable with the tricks they had invented to use audio cassette tape as a storage medium. So, when the time came to build a disk interface, none of the existing chips or techniques were employed, and the disk was thought of as a "random access cassette tape". In order to do this, they had to re-design much of the circuitry on a standard diskette drive. In the process, they gave up any hope of using standard components (like disk controller chips); access to further advances in density, etc, from other developers; and last but not least, any hope of compatibility with the rest of the world. Trying to read or write Apple disks on a standard drive and controller would be like trying to play a video disk on an audio record player. You can't get there from here.

Similar problems exist with CP/M disks on the Commodore 64, Atari and some other 'mass market' machines.

2.6. Physical Level Compatibility

In order for COMPAT to be able to read and write a given format, that format must be compatible with your hardware at the physical level. What this means is that your hardware must be able to work with the required density, number of sides, number of tracks and track density. The format must also be soft sectored, use industry standard FM or MFM encoding, and use certain conventions concerning those overhead bytes before and after the header and data fields on each sector.

An obvious example of incompatibility at the physical level would be trying to read a Televideo disk (double sided) on a Kaypro II (which has single sided drives). It is considerably less obvious why a Kaypro IV disk cannot be read on a Zenith Z100, but that is also due to an incompatibility at the physical level. In this case, the Kaypro IV disk format was designed slighlty wrong. The side number field in the header area of each sector is written as 'side 0' on both sides, with sector numbers 1 through 10 on the front, and sectors 11 through 20 on the back. Normally side 0 would have a 'side 0' code, and side 1 would have a 'side 1' code, with sectors 1 through 10 on both sides. It turns out that the disk controller chip used in the Kaypro IV (which is a WD 1793) can work with the Kaypro IV style disk anyway, whereas the disk controller chip in the Z100 (a WD 1797) cannot read (or write) a single sector on the back.

Things that have to match at the physical level include: disk size, density, number of sides, number of tracks, track density, sector size, and the 'soft sectoring' codes.

2.7. Logical Organization Of A CP/M v2.2 Disk

Even when there is compatibility at the physical level, there are still a number of different ways a CP/M disk can be organized with repect to allocation group size, number of directory entries, and so on. These concepts are covered in more detail in the Digital Research manual, "CP/M System Alteration Guide", and in even more detail in the book "System Programming Under CP/M-80" by L.E. Hughes (Reston, 1983).

Basically, though, the things that may vary from one format to another at this level are:

Track layout
Number of 128 byte 'logical' sectors per track
Allocation group size
Number of extents per directory entry
Total number of allocation groups
Number of directory entries
Number of checked directory entries
Number of 'logical' tracks before the directory starts

Even though a given disk can have sectors with either 128, 256, 512 or 1024 bytes, the CP/M BDOS can only handle sectors of 128 bytes. This means that the BIOS must be able to 'block and deblock' from 1 to 8 'logical' sectors per 'physical' sector.

2.8. Justifications For Different Disk Organizations

In practice, different disk controller and drive designs result in different maximum rates of reading and writing sectors. In many cases, the disk organization is fine-tuned to yield the greatest possible throughput on that system. In other cases, there may have been a desire to provide a large number of directory entries for some reason. In most cases, however, the incompatibilities resulted from a lack of knowledge about what else was going on in the field.

2.9. How COMPAT Works

COMPAT is a very complex program that dynamically modifies the way CP/M works. When you execute it, COMPAT determines where CP/M is located in your memory, and relocates a part of itself just below the CCP. It then plays some games with the BIOS jump vector that in effect insert its own BIOS routines between the BDOS and the real BIOS. Some BIOS calls are allowed to pass through untouched to the real BIOS entry point, others are modified before being allowed to pass through, while yet others are processed completely in the replacement BIOS routines.

The net effect of all this is that the native BIOS (the one supplied with your system) is greatly enhanced, and generalized to work with a wider variety of disk formats, from a table. The remaining trick is to load the correct values for a specific format into the table for the drive to be modified. If the COMPAT replacement BIOS has already been installed (from a previous execution of COMPAT), this loading of the table is all that is really done.

3. Using COMPAT

COMPAT.COM is the main utility program in this package. It is used to 'convert' either (or both) drives to any of the supported CP/M formats. Normally drive A: is left in native format and drive B: is changed to a foreign one. With COMPAT, however, it is possible to have both A: and B: set to foreign formats, to allow transfers of files directly from one foreign format to another, or to run a package that requires both drives to be set to the same foreign format. Note that some versions of COMPAT (such as the one for the Kaypro 10) operate on drive(s) other than A & B. The 'default COMPAT drive' is the last supported drive.

COMPAT may be used in either command mode or menu mode. Both modes are available at any time. In the menu mode, all prompts are basically self-explanatory. The command mode allows the more expert user a faster, less intrusive alternative. More technical feedback is given when the command mode is used.

To set the format of a drive to MSDOS, you can specify the 'format name' as MSDOS (on the command line, or in response to the first question after you ask to change the format), or select any of the formats MS8S, MS8D, MS9S or MS9D from the menus. The MS utility will automatically determine which of the four suported formats a given MSDOS disk really is, and adapt accordingly.

3.1. Ways To Run COMPAT

COMPAT

Bring up COMPAT in the menu mode. The current formats for all supported drives will be listed. Most versions support drives A: and B:. On such a system, the following options would then be displayed:

```
A - Modify format on drive A:, current = XXXXXX
B - change format on drive B:, current = XXXXXX

R - Remove COMPAT, return to the native BIOS
X - Exit to CP/M, using the currently selected formats
```

On some systems (such as the Kaypro 10, which supports changing only drive C:) , a different list of drives will be displayed.

If a drive name (A through D) is selected, you will be prompted to either enter a format name, or hit a <CR> (carriage return) to go to the menus. Once either of these is done, control will return to the main menu. Once the main menu displays the desired formats, exit to the operating system via the X option. The R option may later be used to remove COMPAT and return control to the native BIOS, using the native format on both drives.

COMPAT [d:]fmtname

Shortcut way to load the COMPAT BIOS (or just modify tables if it is currently loaded), and 'convert' the specified drive to the specified format. If no drive is specified, the default COMPAT drive is assumed.

COMPAT *

Shortcut way to remove the COMPAT BIOS and return control to the native BIOS, using native format on both drives.

3.2. Examples Of Using COMPAT

A>compat

Bring up COMPAT in menu mode (then follow directions).

A>compat kaypro

Set default COMPAT drive to Kaypro single sided format.

A>compat d:msdos

Set drive D: to MSDOS format.

A>compat *

Remove COMPAT and return to normal operation.

4. Using FORMAT

FORMAT is used to initialize blank diskettes to any of the supported CP/M or MSDOS disk formats. This writes the 'soft sectoring' information onto the disk to fragment it into managable 'chunks' (sectors) that can be individually read or written later. Each type of CP/M hardware system has its own unique way of doing this, and almost every system comes from the factory with the ability to initialize diskettes in at least its own (native) formats. Mycroft Labs' FORMAT utility can perform this same function, but for all supported disk types, not just one or two.

To format a blank disk to any of the four supported MSDOS formats, use the following format names (on the command line or from the menus):

MS8S	Single Sided, 8 sector/track (Usually DOS 1.x)
MS8D	Double Sided, 8 sector/track (Usually DOS 1.x)
MS9S	Single Sided, 9 sector/track (Usually DOS 2.x)
MS9D	Double Sided, 9 sector/track (Usually DOS 2.x)

Once an MSDOS disk has been formatted, the MS utility will automatically adpat to the correct MSDOS format, but do not forget to use COMPAT to set the physical parameters of the drive to MSDOS before trying to access it with MS. Note that the MSDOS bootstrap sector is NOT written by our format utility, hence disks formatted by our utility cannot be made bootable.

Our format utility does both formatting and validation in each pass. On each track, it tries to format the track, then tries to read it. If even a single error is detected (which is reported as a soft error), FORMAT will try again up to 2 times before giving up and reporting a hard errror for that track. If no errors are reported, the entire disk can be assumed to be error-free.

4.1. Ways To Run FORMAT

FORMAT [d:]

Bring up FORMAT in the menu mode. If no drive is specified, the default COMPAT drive will be assumed. To page through the menus, just type <CR> (carriage return). Once you are on the menu that contains the desired format, select the letter of that format. If no drive is specified, the default drive for that system will be used. You will be prompted to mount a disk in the target drive and type another <CR>. At this time, the new disk will be formatted. NOTE: this will destroy any previous data on that disk. One dot will be printed for each cylinder. Every ten cylinders a digit will be printed. Most 48 tpi disks have 40 cylinders. On double sided disks, each cylinder has two tracks (front and back). This process may be aborted at any time by typing a <CR>. Once the format is done (or aborted), you will be given the three options:

R   to repeat the current format (format another disk in the same format)

X   exit to the operating system

<CR> return to the menus

FORMAT [d:]fmtname

Shortcut way to format a disk in the specified format. If no drive is specified, the default COMPAT drive is assumed. You will be presented with the same options as before.

FORMAT [d:]fmtname G

Same as above, but do not pause for confirmation, and exit to the operating system when done. Useful for including in procedure files, etc.

FORMAT [d:]*

Shortcut way to format a disk in the native format. If no drive is specified, the default COMPAT drive is assumed.

4.2. Examples Of Using FORMAT

```
A>format
```

Bring up FORMAT in menu mode and follow directions. The disk in the default COMPAT drive will be formatted.

```
A>format kaypro
```

Format the disk in the default COMPAT drive in Kaypro II (single sided) format.

```
A>format d:ms8s
```

Format the disk in drive D: to the Single Sided, 8 sector MSDOS format.

```
A>format *
```

Format the disk in the default COMPAT drive to the native disk format.

5. Using MS

There is a utility (MS) provided to allow you to do directories of and transfer files TO or FROM various MSDOS formats. The following formats are currently supported:

8 sector per track, Single Sided (usually DOS 1.x)
8 sector per track, Double Sided (usually DOS 1.x)
9 sector per track, Single Sided (usually DOS 2.x)
9 sector per track, Double Sided (usually DOX 2.x)

Note that under MSDOS 2.x, only the root directory is accessible. To use this utility, you must use **COMPAT** to make one of your drives (typically drive B: on a two drive system) compatible with the correct physical parameters: COMPAT MSDOS. Note that PCDOS and ZDOS disks are standard MDSOS format. Files such as text files, Wordstar documents, dBASE II databases, etc. will be usable once transferred. Other binary file formats may not be particulary useful, but can be transferred nonetheless. In general, if the same program (e.g. Multiplan) is available under both operating systems, the data files will be useful once transferred.

MS may be used in direct mode (one command on command line, e.g. 'MS DIR *.ASM'), or in interactive mode (any number of commands, one in response to each prompt, until terminated with ctrl-Z). There are several subcommands available. In direct mode, the subcommand is specified immediately after the command name, followed by any parameters. In interactive mode, just the subcommand and parameter(s) are entered. To start interactive mode, just type MS with no subcommand.

All subcommands in MS automatically determine which format the specified MSDOS disk is, and adapt accordingly. The SPACE subcommand will tell you which format a given disk is.

Available subcommands are as follows:

HELP	display list of available commands
DIR d:afn	request directory of files on MSDOS disk
GET dst=src	get files from MSDOS disk to CP/M disk
PUT dst=src	put files to MSDOS disk from CP/M disk
TYPE d:ufn	display file from MSDOS disk on console
ERA d:afn	erase files from MSDOS disk
SPACE d:	request info on used/available space on MSDOS disk
/DIR afn	request directory of files on CP/M disk
/TYPE ufn	list file from CP/M disk to console
/ERA afn	erase files from CP/M disk
/SPACE d:	request info on available space on CP/M disk
/SIZE afn	request size of files on CP/M disk
/RESET	make all CP/M disks R/W again

5.1. DIR Subcommand

Syntax: DIR [d:][afn]

Display directory of files on an MSDOS disk. If more than one MSDOS drive is supported on your version of COMPAT, you can specify the desired MSDOS drive with a drive name (d:). If the drive name is not specified, the default MSDOS drive for your version of COMPAT will be assumed. If no ambiguous filename (afn) is specified, the filename '*.*' will be assumed (all files). Note that both the name and size of all files matching the specified afn will be listed.

5.2. GET Subcommand

Syntax: GET [dst=]src

Get file(s) from an MSDOS disk to a CP/M disk. The destination (dst) is where the files should be copied to, and may be a drive (e.g. B:=*.ASM), or an unambiguous filename if only one file is being transferred (e.g. B:JOE.A86=FRED.ASM). Optionally a user number may be specified after the drive name (e.g. B10:=*.ASM). If the destination is not specified, the current logged CP/M drive and user number are assumed. The source (src) is an ambiguous filename that specifies which files to get, and may optionally be preceded by a drive name (if more than one MSDOS drive is supported on your version of COMPAT). If no drive name is specified, the default MSDOS drive for your version of COMPAT will be assumed.

Examples:

GET *.ASM - Get all files of type ASM from the default MSDOS drive to the current CP/M drive and user area.

GET A10:=D:*.* - Get all files from the MSDOS disk in drive D: to user area 10 of the CP/M disk in drive A:.

GET JOE.A86=FRED.ASM - Get the file FRED.ASM from the default MSDOS drive to the current CP/M drive and user area, under the name JOE.A86.

5.3. PUT Subcommand

Syntax: PUT [dst=]src

Put files to an MSDOS disk from a CP/M disk. The destination (dst) is where the file(s) should be copied to, and may be a drive name (if more than one MSDOS drive is supported on your version of COMPAT), and if only a single file is being copied, an unambgiuous filename (e.g. FRED.ASM=JOE.A86). If no destination is specified, the default MSDOS drive for your version of COMPAT will be assumed. The source (src) is an ambiguous filename that specifies which files to put, and may be preceded by a drive name (e.g. B:*.ASM) and optionally a user number (e.g. A10:*.DBF).

Examples:

PUT *.ASM - Put all files of type ASM from current CP/M drive and user area to the default MSDOS drive.

PUT D:=A10:*.* - Put all files on drive A:, user 10 to the MSDOS disk in drive D:.

PUT FRED.ASM=JOE.A86 - Put file JOE.A86 from current CP/M drive and user area to the default MSDOS drive, under the name FRED.ASM.

5.4. TYPE Subcommand

Syntax: TYPE [d:]ufn

Display contents of file from MSDOS disk to console. If your version of COMPAT supports more than one MSDOS drive, the desired drive name can be specified before the filename (e.g. D:FRED.ASM). If no drive name is specified, the default MSDOS drive for your version of COMPAT is assumed. The display may be paused with Ctrl-S (and restarted with any character), or aborted with any other character while it is scrolling to the screen. The top bit of all characters is stripped, and all control characters other than CR, LF, BEL and BS are filtered before being displayed on the console.

Examples:

TYPE FRED.ASM - Display the file FRED.ASM from the default MSDOS drive.

TYPE D:DEMO.TXT - Display the file DEMO.TXT from the MSDOS disk in drive D:.

5.5 ERA Subcommand

Syntax: ERA [d:]afn

Erase file(s) from an MSDOS disk. If your version of COMPAT supports more than one MSDOS drive, the desired drive name can be specified before the ambiguous filename (e.g. D:*.ASM). If no drive name is specified, the default MSDOS drive for your system is assumed.

Examples:

ERA *.ASM - Erase all files of type ASM from the disk in the default MSDOS drive.

ERA D:FRED.ASM - Erase the file FRED.ASM from the MSDOS disk in drive D:.

5.6. SPACE Subcommand

Syntax: SPACE [d:]

Display current used, current available, and total file space and directory entries for an MSDOS disk. If your version of COMPAT supports more than one MSDOS drive, the desired drive can be specified (e.g. D:). If no drive name is specified, the default MSDOS drive for your version of COMPAT is assumed. The information displayed is as follows:

```
MSDOS Disk Type = Single/Double Sided, 8/9 Sector

Total Directory Entries      = nnn
Used Directory Entries       = nnn
Available Directory Entries  = nnn

Total Space on Disk          = nnnK Bytes
Used Space on Disk           = nnnK Bytes
Available Space on Disk      = nnnK Bytes
```

6. Using MYCOPY

MYCOPY is an extended file copy utility intended to replace the standard Digital Research PIP utility. It supports all standard features of PIP, except those concerned with HEX files, and those concerned with excerpting portions of files. The primary enhancements are:

- command strings may be read from an 'indirect command file', in addition to interactively.

- groups of files may be specified in a more flexible manner using ambiguous filename expressions, allowing such groups as 'all files of type ASM or SRC, except those whose names begin with XYZ'.

- user numbers may be specified after the drive name on any filename (e.g. A7:*.*) to allow access to or from user areas other than the current one. If no user number is specified, the default is the current user area.

- CCP-like 'system commands' (DIR, ERA, REN, TYPE, USER, etc.) may be entered anywhere a copy directive is valid (which is especially useful in interactive mode).

- concatenation may be specified for any collection of files and/or devices (e.g. *.HEX), and is much faster than with PIP.

- a form feed may optionally be appended to the end of each file as it is copied, primarily for use in listing a group of files to a printer.

- the default mode is 'copy until physical end of file'. Termination at the first Ctrl-Z is automatic on logical devices, and when concatenation is specified.

- if either the input or output file is not specified, the default 'file' is the console (CON:).

6.1. Running MYCOPY

MYCOPY may be invoked in any of three ways, the first two of which (direct and interactive) are similar to those for PIP. The third (indirect) is somewhat more powerful:

Direct: one command string on the execution line

```
A>MYCOPY i=*.asm,o=b:
```

Interactive: any number of command strings entered one at a time from the console in response to ? prompts

```
A>MYCOPY
?i=*.asm,o=b:
?/dir b:
?^Z
```

Note that for this mode, MYCOPY is invoked with no parameters, and is terminated with a Ctrl-Z as the first character of a command string. (Ctrl-C will also abort this mode.)

Indirect: any number of command strings specified on a file of type .IND, one per line.

```
A>MYCOPY @BACKUP
```

In this example, the file BACKUP.IND is assumed to have one or more command strings, one per line, created with ED, MYCOPY, or whatever.

We recommend that you rename MYCOPY.COM to some shorter name, such as C.COM or MC.COM, as you will find it to be one of the most common commands you use.

6.2. Ambiguous Filename Expressions

An 'ambiguous filename expression' (afnx) may be a string of one or more ambiguous or unambiguous filenames and/or device names separated by + and/or - signs. Any filename may be preceded by a drive name (A through P) and an optional user number (0 through 31), followed by a colon (e.g. A:FRED or B7:*.ASM, etc.). The first file or device name is assumed to be preceded by a +. All files which satisfy an afn preceded by a + will be added to the overall list if they are not already there. Likewise, all files that satisfy an afn preceded by a - will be deleted from the overall list if they were present.

Hence, to specify all files of type ASM or SRC except those whose name begins with XYZ, use the string '*.ASM+*.SRC-XYZ*.*'. These operators are evaluated from left to right. Note also that it is not possible to specify more that one copy of a particular file (e.g. 'FRED.SYM+FRED.SYM' is equivalent to 'FRED.SYM'. The characters + and - are not special-cased when between double quote marks. Hence to specify the file HANG-UP.COM, use the string "HANG-UP.COM".

Legal device names are as follows:

CON:	console	input/output
PUN:	punch	output
RDR:	reader	input
LST:	list	output
NUL:	eof/bit bucket	input/output

The output name can be a specific 'ufn' if only a single file is being copied, or if concatenation was selected. Otherwise, a logical device or drive name should be specified.

Even on single file transfers, if the filename AND the filetype fields are left blank on the output filename, the corresponding fields of the input filename will be used (e.g. i=fred.asm,o=b: is equivalent to i=fred.asm,o=b:fred.asm).

6.3. MYCOPY Parameters

Each command string may be either an actual file copy directive, (e.g. i=*.asm,o=b:), or a system command (e.g. /dir). System commands have a leading slash, and are quite similar in syntax and effect to the standard CCP 'direct' commands. Actual copy directives may have one or more keyword parameters, (separated by commas) some of which are equivalenced:

Parameter	Argument	Default	Function
I	xafn	CON:	input device and/or file(s)
O	dev: or ufn	CON:	output device or file
V	----	----	output file(s) verified
X	----	----	tabs expanded to blanks
U	----	----	all alphas forced to U.C.
L	----	----	all alphas forced to L.C.
C	----	----	input files concatenated
F	----	----	add form feed after each file
Q	----	----	have user confirm each file
P	n	0	set page length to n lines
A	----	----	preserve file attributes
S	----	----	copy files with SYS attribute
RO	----	----	copy even if R/O file exists
LP	----	----	equivalent to O=LST:,F,P=60
SF	----	----	sort filenames before copying
VO	----	----	verify only, input with output
EF	----	----	erase first if file(s) exist
RS	----	----	Reset System to Read/Write

The V parameter will cause the output file to be verified (after each 8K block is written) against the memory buffer from which it was written.

The X parameter allows tabs to be expanded to blanks, assuming tab positions every 8 columns, as is standard in most of CP/M.

The U and L parameters allow alphabetic characters to be forced into all upper case, or all lower case, respectively, as the file is copied.

The C parameter specifies that all files satisfying the input file specification are to be written to a single output file or device.

The F parameter specifies that a form feed (page eject on most printers) is to be written to the output file or device, every time an end of an input file is encountered. This is primarily intended to allow multiple files to be printed, with each new file starting at the top of a page.

The Q parameter causes COPY to ask the user to confirm each file in a multiple file transfer (with or without concatentation). Each filename will be printed, followed by a question mark, in response to which the user should type a Y or N (no <cr> required, upper or lower case).

The P=n parameter allows the user to set the 'page length' to n lines, which will cause a form feed to be written to the output file after every n lines. This is useful for skipping page breaks. Recommended values are 60 for six lines per inch and 80 for 8 lines per inch (assuming 11" paper). Note that n=0 (default) is equivalent to infinite length pages (no form feeds).

The A parameter causes file attributes (SYS,DIR,R/W,R/O) to be preserved if the output filename is not specified (e.g. O=B:). The default is to clear attributes as files are copied.

The S parameter allows files with the SYS attribute to be copied. Normally, these would not be listed or copied.

The RO parameter specifies that if any previous Read/Only file should exist on the output disk of the same name as a new file being copied there, to automatically kill it and replace it with the new file. Normally, a check is made for this situation, and the user will be asked if he wishes to delete such a file.

The LP parameter is equivalent to specifying the three parameters: O=LST:,F,P=60. It is useful in printing files.

The SF parameter causes the list of specified filenames to be sorted into ascending alhpabetic order before the actual copying is done.

The VO parameter allows one or more pairs of files to be compared. The "input" file(s) will be compared against the "output" file(s). If any diferences are found, they will be listed, along with the location of those byte(s) relative to the start of the file. If the files match exactly, no messages are displayed.

The EF parameter causes any existing file on the output drive with the same name as a new output file to be erased before a given copy operation is done. Normally, a file is copied to a dummy file (x.$$$), and once the copy is done, the original file is erased and the dummy file is renamed with the actual name. This prevents losing a previous version of the file if the copy fails midway, etc. The drawback to this approach is that there must be room for both the original and the dummy, which is not always the case. This parameter in effect allows you to defeat this mechanism.

The RS option resets all drives to Read/Write status before any actual copying is done. It duplicates the function of the 'RESET' system command.

6.4 System Commands

The system commands (preceded by a slash) may be any one of the following CCP-like commands:

DIR [afnx]	list directory of files satisfying 'afnx'
DRIVE d:	select drive d: (A through P)
ERA afnx	erase all files satisfying 'afnx'
ERA afnx Q	erase all files satisfying 'afnx' w/ query
HELP	list available system commands
LIST ufn	list specified file on list device
REN new=old	rename a single ufn
RESET	make all drives R/W again
SET afnx $atr	give specified files attribute atr
SIZE afnx	display size of specified file(s)
SPACE [d:]	display space available on specified drive
TYPE ufn	list a single ufn to the console
USER n	switch to user number n (0-31)

Note that these commands may be used in interactive mode or in indirect command files. The slash (/) must come the first column, and the system command directly follow it (e.g. /dir *.asm).

Most commands accept an ambiguous filename expression, complete with user numbers, e.g. /DIR A7:*.ASM+*.COM would give a list of all filenames with type .ASM or .COM on drive A:, user 7.

With the SET command, any attribute may be set on all files satisfying a particular afnx. Legal attributes are:

DIR	make file visible in directory
SYS	make file invisible in directory
R/W	make file read/write
R/O	make file read only

as an example, to set all COM files to read only: /SET *.COM $R/O

The DIR command will not list 'invisible' files, and read-only files will be noted by a preceding '>' instead of the usual ':'. Also note that only files for the current user will be listed.

Even if a new drive is selected during MYCOPY, when control is returned to CP/M, the drive which was selected when COPY was initiated will still be the default drive. If the user number is changed, however, it will remain changed upon return to CP/M.

This version will default to a fast, sector by sector physical transfer of data (until physical end of information), instead of the slower character by character logical copy (until the first Ctrl-Z), unless any of the following conditions occur:

1) Input and/or output are devices
2) Concatenation is selected
3) The U,L,F or X options are selected

6.5. Examples Of Using MYCOPY

```
MYCOPY I=*.*,O=B:
```

Copy all files from current drive and user area to drive B:, current user area.

```
MYCOPY I=B:*.ASM+*.SRC-XYZ*.*,O=A2:
```

Copy all files of type ASM or SRC except those with names beginning with XYZ, from current user area of drive B: to user area 2 of drive A:.

```
MYCOPY I=*.*-C:*.*,O=C:
```

Copy all files from current drive and user area to current user area of drive C:, that are not already there.

```
MYCOPY I=*.PRN-FRED.*,LP
```

Copy all files of type PRN except for FRED.PRN to the list device, with a page length of 60 and a form feed at the end of each file.

```
MYCOPY I=*.ASM,O=C2:,Q,V
```

Copy all files of type ASM from current drive and user area to drive C:, user 2. On each file, query the operator for a Yes or No before doing the copy. Verify the output file as it is written.

```
MYCOPY /DIR A3:
```

Display the directory of all files on drive A:, user 3.

```
MYCOPY
```

Bring up MYCOPY in interactive mode, then enter any number of commands, one in response to each ? prompt. To exit to CP/M, enter a single Ctrl-Z, followed by a Carriage Return.

```
MYCOPY @FRED
```

Perform the MYCOPY commands found on the file 'FRED.IND', then return to CP/M.

7. Summary Of Formats

The following formats are supported by COMPAT and FORMAT as of the writing of this manual. More are being added all the time.

7.1. 48 tpi Single Sided Formats

Format	Type	Size	System(s)
ACCESS	SS/DD	169k	Access Matrix, Actrix SS
CCSS0	SS/SD	70k	CCS SD 128 byte x 18 sector
CCSS1	SS/SD	78k	CCS SD 256 byte x 10 sector
CCSS2	SS/SD	78k	CCS SD 512 byte x 5 sector
CCSD1	SS/DD	142k	CCS DD 256 byte x 18 sector
CCSD2	SS/DD	158k	CCS DD 512 byte x 10 sector
CCSD3	SS/DD	158k	CCS DD 1024 byte x 5 sector
CROM1	SS/SD	81k	Cromemco SS/SD
CROM2	SS/DD	188k	Cromemco SS/DD
IBMPC1	SS/DD	153k	IBM/DRI CP/M 86 SS
KAYPRO	SS/DD	191k	Kaypro SS
LOBOSS3	SS/DD	142k	Lobo Max-80 SS 35 trk
LOBOSS4	SS/DD	164k	Lobo Max-80 SS 40 trk
MEMO0	SS/SD	71k	Memotech format 0 SS/SD
MEMO2	SS/DD	151k	Memotech format 2 SS/DD
MONT	SS/DD	166k	Montezuma Radio Shack IV CP/M
MORROW	SS/DD	186k	Morrow Micro Decision SS
MS8S	SS/DD	156k	MSDOS SS, 8 sector/trk
MS9S	SS/DD	175k	MSDOS SS, 9 sector/trk
NECPC	SS/DD	150k	NEC PC-8000
OMIK35	SS/SD	70k	Omikron Mapper 35 trk
OMIK40	SS/SD	81k	Omikron Mapper 40 trk
OSBORN1	SS/SD	90k	Osborne SD
OSBORN2	SS/DD	183k	Osborne DD
RSM4	SS/DD	154k	Radio Shack Mod IV CP/M+
SBRAIN1	SS/DD	162k	Superbrain/Compustar SS
SEEQUA	SS/DD	154k	Seequa Chameleon CP/M-80
SYSTEL1	SS/DD	167k	Systel/Olympia EX-100 SS
VT180	SS/DD	169k	DEC VT-180
XEROX1	SS/SD	81k	Xerox SS/SD
XEROX2	SS/DD	155k	Xerox SS/DD
Z90SD	SS/SD	90k	Zenith Z90 SD
Z90	SS/DD	148k	Zenith Z90 DD
Z100S	SS/DD	148k	Zenith Z100 SS

7.2. 48 tpi Double Sided Formats

Format	Type	Size	System(s)
AVATAR	DS/DD	380k	3R Avatar, Datamaxx Exxpert
BILLNGS	DS/DD	241k	Billings
CASIO	DS/DD	300k	Casio FP-1000
CDPPC	DS/DD	312k	Columbia CP/M-86 DS
CROM4	DS/DD	386k	Cromemco DS/DD
DIM68	DS/DD	386k	Dimension CP/M-68k
EPSON	DS/DD	300k	Epson QX-10 256 byte/sector
EPSON2	DS/DD	376k	Epson QX-10 512 byte/sector
HP	DS/DD	248k	HP 125
IBMPC2	DS/DD	314k	IBM/DRI CP/M-86 DS
KAYPRO4	DS/DD	390k	Kaypro IV, 10, 2x
LOBODS3	DS/DD	296k	Lobo Max-80 DS, 35 trk
LOBODS4	DS/DD	342k	Lobo Max-80 DS, 40 trk
MEMO1	DS/SD	151k	Memotech format 1, DS/SD
MEMO3	DS/DD	310k	Memotech format 3, DS/DD
MOLEC	DS/DD	356k	Molecular
MORROW2	DS/DD	384k	Morrow DS (MD3)
MS8D	DS/DD	315k	MSDOS DS, 8 sector/trk
MS9D	DS/DD	354k	MSDOS DS, 9 sector/trk
MSOURCE	DS/DD	376k	Microsource, Microstandard
NCRV	DS/DD	304k	NCR Decision Mate V
NEC88A	DS/DD	300k	NEC 8800 256 byte/sector
NEC88B	DS/DD	300k	NEC 8800 512 byte/sector
OTRONA	DS/DD	360k	Otrona Attache
PMC	DS/DD	386k	PMC Micromate CP/M+
SANYO1	DS/DD	310k	Sanyo MBC 1000, 1150
SBRAIN2	DS/DD	338k	Superbrain, Compustar DS
SYSTEL2	DS/DD	344k	Systel/Olympia DS
T100	DS/DD	254k	Toshiba T100
TVIDEO	DS/DD	340k	Televideo 802/803/806
UNISYS	DS/DD	386k	Hazeltine Unisystem
Z100D	DS/DD	304k	Heath/Zenith Z100 DS
ZORBAD	DS/DD	388k	Zorba DD
Z2000D	DS/DD	382k	Zorba 2000/4

7.3. 96 TPI Single Sided Formats

Format	Type	Size	Systems(s)
EAGLE2	SS/DD	384k	Eagle II
ITHACA1	SS/DD	346k	Ithaca Intersystems SS
LOBOSS8	SS/DD	342k	Lobo Max-80 SS 80 Trk
MONROE	SS/DD	306k	Monroe 8810/8820
RAINBOW	SS/DD	386k	DEC Rainbow
SANYO2	SS/DD	302k	Sanyo MBC 2000
TMI	SS/DD	346k	T.M.I.

7.4. 96 TPI Double Sided Formats

Format	Type	Size	System(s)
ALTOS	DS/DD	712k	Altos
DATAVUE	DS/DD	776k	Datavue
FOX	DS/DD	624k	DMS Fox
INTEL	DS/DD	606k	Intel PDS
LANIER	DS/DD	616k	Lanier EZ-1 WP
LOBODS8	DS/DD	698k	Lobo Max-80 DS 80 Trk
PEOPLE	DS/DD	618k	Olympia People WP
PIPER	DS/DD	776k	Pied Piper
SANYO12	DS/DD	620k	Sanyo MBC 1250
SEIKO	DS/DD	628k	Seiko 8600
SYFA	DS/DD	758k	Micro SyFA
TIC	DS/DD	628k	Tech. Intn'l. Corp.
TVIDEOQ	DS/DD	692k	Televideo QD
XUPORT	DS/DD	776k	Execuport
ZORBAQ	DS/DD	784k	Zorba QD (Nomis)
Z2000Q	DS/DD	776k	Zorba 2000/8 QD
Z90Q	DS/DD	614k	Heath/Zenith Z90 QD

8. Notes

1. Additional CP/M formats can be added fairly easily in many cases. For the details, please contact us directly at Mycroft Labs Inc. (904) 385-1141.

2. Systems with single sided drives can access only single sided formats. Systems with double sided drives can work equally well with either single sided or double sided formats.

3. Systems with 48 tpi drives can access only 48 tpi formats. On systems with 96 tpi drives, reading 48 tpi disks works great. However, due to a weaker signal in the 96 tpi drives, 48 tpi disks written on these systems will not always be readable on real 48 tpi drives. To increase the probability of this working, follow these recommendations:

Use the highest quality 96 tpi media available
Format the disk AND write the data on the 96 tpi drives
Write several copies of the file(s) (at least one may work)
Transfer the files to real 48 tpi disks as soon as possible
As the outer tracks work best, do not fill the disk

4. Due to limitations of the hardware, the following versions are unable to work completely with certain formats:

A. Zenith Z100 is unable to read/write disks which are formatted with side number field of 0 on both sides.

Affected in COMPAT and FORMAT:

KAYPRO4, MSOURCE, ZORBAD

B. SANYO 1000,1150,1250 are unable to read/write in single density

Affected in COMPAT and FORMAT:

CCSS0, CCSS1, CCSS2, CROM1, MEMO0, MEMO1, OMIK35, OMIK40, OSBORN1, XEROX1, Z90SD

Affected in FORMAT (at least one track SD):

CCSD1, CCSD2, CCSD3, CROM2, CROM4, INTEL, LANIER, MEMO2, MEMO3, T100, XEROX2

9. Installation

The COMPAT.COM, FORMAT.COM and MS.COM programs supplied on the distribution disk are NOT installed, and will inform you of that fact if you try to run them before performing an installation for a given machine. In order to use this package, you need to do the following steps:

1. Copy all files on the distribution disk onto a working disk (which on a Kaypro 10 could be an area of the hard disk).

2. Put the master disk away as a final backup, and to send back in for later updates, etc. **DO NOT EVER WRITE ANYTHING TO THIS DISK.**

3. Log into the working disk you just made, and run INSTALL by typing its name:

```
A>INSTALL <CR>
```

4. The first menu will allow you to select which family of machines you wish to install COMPAT for (Kaypro, Zorba, etc.). Your distribution disk will only include the 'drivers' (device dependent modules) for the machines in that family. For example, if you have a Kaypro version of COMPAT, you can install it for any 8 bit Kaypro machine, but not for a Zorba.

5. Depending on the machine family selected, you will see a second menu of machines in that family. Select the specific machine for which you want to install COMPAT.

6. Wait while INSTALL informs you of its progress.

7. When the process is complete, all you really need from this working disk are the following files (which can be easily copied to other disks as needed:)

```
COMPAT.COM - if you want to set any drive(s) to other types
FORMAT.COM - if you want to initialize blank disks
MS.COM     - if you want to transfer files to/from MSDOS
MYCOPY.COM - for general file/device transfer anytime
```

none of the other files are needed for normal operation. We also recommend you rename MYCOPY.COM to MC.COM or C.COM, as you will find it to be one of your most commonly used commands. We typically have it on every working disk in our library.

8. Finally store the working disk you just created in a safe place as a second backup.