RAPPORT

By John Whelan

Occasionally it's interesting to take a look at the fast-moving world of Electronic Mail. In the world of BBS Electronic Mail, we have been spoiled for many years with features like transparent data-compressed messages, menu-driven systems and efficient file transfer protocols such as Xmodem, Ymodem, Ymodem-Batch and Zmodem.

In the commercial world of mini-computers, newer features have progressed at a much slower rate. Telecom Canada's Envoy, for example, is entirely command-driven. If you forget a command, you've got problems. (i.e., you must remember to terminate messages with ""<Enter>"."). A help feature is available provided that you remember the command for it. Even then, it doesn't provide a list of all of the commands.

Envoy (sometimes referred to as iNet) is one of the largest E-mail systems in Canada. The federal government uses it as a basis for the Government Electronic Mail & Document Exchange System (GEMDES). DEC and Wang Canada's X.400 both interface with Envoy. Rapport now "front ends" Envoy.

Rapport was developed by Teal Communications to enable a group of accountants to use Envoy. It has existed for a number of years and has just recently undergone major changes as Envoy itself has incorporated new features such as XModem file transfers and 8-bit character sets.

The package is basically an off-line mail system that connects to a host system much like our points. It is entirely menu-driven and is available in English and French.

On entering the program, the user is presented with a "desk" covered with icons. This "desk" can be custom-programmed, if either a large enough fee is paid or a sufficient number of copies are ordered. It has been picked up by Telecom Canada as a supported product. Telecom Canada provides a 1-800 number support for Envoy and Rapport users. A context-sensitive help system is also included.

Messages are first composed off-line and held until the "Mail Send and Receive" command is actioned. User IDs and system passwords can be either embedded within the program or entered each time the package is accessed. If the PC is at home, I recommend embedding the password. In the office, however, I would recommend entering it each time Rapport is accessed.

The "Send and Receive" function can be set to run automatically at preset times. However, the password must be embedded in order to do this. When sending messages, it does not use the "Compose Batch" feature of Envoy. This feature can add to your transmission costs if you are using GPN (Government Packet Network X.25) and are paying for the transmission costs. However, this has little impact on regular users.

Rapport is well geared for commercial use. It allows both copies (cc's) and Blind copies (bcc's) to be added to messages. (When I am in the private message area of the PUB, I often think of how nice it would be able to send a message to one person and have it cc'd to another person.)

A copy of all messages sent is retained by the user along with the date and time of each message. Up to around 300 messages can be filed

NEXT MEETING

The next meeting of the OPCUG will be held on February 28, 1990. The Guest Speaker will be Carl Revine of Heath Zenith Computers. His topic will be the revolutionary Zenith 486 with parallel processors. This 6 ms. access-time machine, which is ideally suited for computer-intensive applications, can easily be used as a LAN (Local Area Network) server (See BYTE Magazine, February 1990.

The Guest Speakers for the March meeting will be a team from the Hewlett Packard Company of Canada. Their topic will be "An Answer to PC Users' dreams: HP's New Wave Environment".

* * *

FROM THE EDITOR

By Bonnie Carter

Hello fellow Members! This month's Newsletter introduces a series of articles on how to program in BASIC. Follow the series and learn how to program in BASIC.

Next month, the Newsletter will contain more pages for your reading enjoyment. Happy reading!

* * *

Contents

Meetings are held at 100 Sussex Drive, Ottawa
Last Wednesday of the month - 8:00 p.m. to 10:00 p.m.
No meetings in July and December

RAPPORT

in folders. Messages are kept in subdirectories as ASCII text. Although Rapport does not provide a message search, WordPerfect can be used to find a particular message.

Sending a file as an attached binary file is simple matter of using the menu to attach an Xmodem file and possibly moving around subdirectories, highlighting the file and pressing "<Enter>" (very similar to WordPerfect). Received files are placed in a data subdirectory along with the name of the file and its accompanying message.

Rapport also comes with a built-in editor. Messages may also be composed elsewhere and imported into the package. A pop-up list of users is included. Type "J" and a list of names that begin with a "J" appears. When the correct name is found, press "<Enter>" to select that name. Hit the space bar to select another name. Press the space bar again, followed by "<Enter>", to include both names in the "To:" field.

The package has numerous other features. My intent has been to convey a flavour of it to you. If you use Envoy or GEMDES, give Rapport serious consideration. In the BBS world already some users are using points. I suspect that front-end software interfacing will arrive soon to TBBS boards, such as the PUB. It would be less costly than the addition of more telephone lines, and, as local time-measured service starts to creep in from phone companies, both sysops and users will favour this easy-to-use type of approach.

* * *

MEMORY: EXTENDED & EXPANDED

By Mike Schupan

When talking about the random access memory (RAM) on an IBM PC or a compatible, you may have heard it referred to as either *conventional*, *extended*, or *expanded* memory. You do not need to know what the differences between them are, and you may not even be curious, but if you want to work with a huge application with lots of information, knowing the differences may help you avoid a "Not enough memory!" problem.

When the IBM PC was first introduced, it's operating system (DOS) was designed to work with up to 640 kilobytes (Kb) of user memory (1 Kb = 1024 bytes). This was considered an incredible amount of memory, because the earlier CP/M microcomputers only had 64 Kb of RAM. As the PC's popularity grew, application programs, like Lotus 1-2-3, allowed people to manage/analyze more data at one time. People soon ran into memory limitations with DOS's 640 Kb of "*conventional*" RAM. They wanted more.

To resolve this problem, Lotus, Intel, and Microsoft developed the specifications for "*expanded*" memory, which allowed DOS based applications to work with up to 16 megabytes (Mb) of data (1Mb = 1024 Kb). This was accomplished by only working with small portions (pages) of the expanded memory at any one time. These pages would be "swapped" in & out of an area within DOS's workspace as needed. (The total DOS workspace is 1 Mb. This includes video and other hardware memory in addition to the user's 640 Kb of RAM.)

When IBM introduced the AT microcomputer with its 80286 central processing unit (CPU), the memory issue got more confusing. The 80286 is able to work with up to 4 terabytes (Tb) of RAM (1 Tb = 1,024 Mb) in its protected mode of operation. This memory is referred to as "*extended*" memory, because it extends continuously beyond the first 1 Mb of RAM. Figure 1 illustrates

the differences between the two types of memory. Unfortunately, because DOS was designed to work with the 8088/8086 CPU in Real mode, it cannot take advantage of the 80286's increased memory capabilities. ("8086 Real Mode vs 80286 Protected Mode" might be the topic for a future PC Tech Talk!)
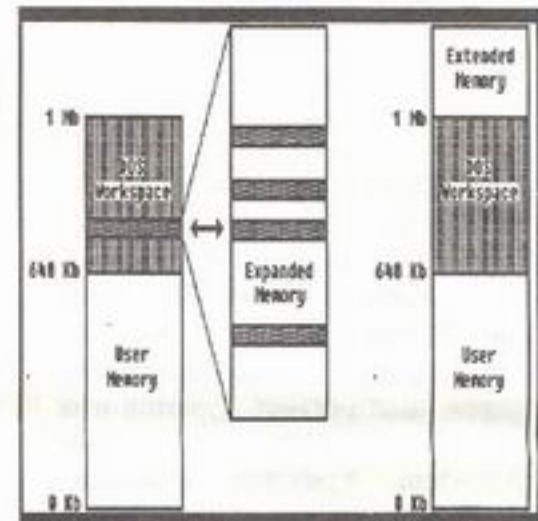

Figure 1: Expanded vs Extended Memory

In general, when using DOS, expanded memory is more useful than extended memory. Chances are, if you already have expanded memory, your applications are already using it to work with larger amounts of data. But what if you have extended memory? What can you do with it if DOS can't use it? DOS supplies a device driver called VDISK.SYS that allows you to use it as a virtual disk that works faster than your regular disks, but you can't save anything when you turn the power off. Or, you could switch to the OS/2 operating system which is designed for the 80286 CPU and larger memory configurations. The best alternative is to use an "expanded memory simulator" that makes extended memory look like expanded memory. (If you have a PC with more than 640 Kb of RAM, you can obtain the EMS40.SYS simulator from the CCA Help Desk.) With it, you can regain some "breathing room" for your larger spreadsheets and databases.

* * *

BACK TO BASICs

By Harry Gross

Welcome all you keen readers to an introductory course on programming in BASIC. The purpose of this series of articles is to provide you with an introduction to programming in BASIC on a PC. If I assume too little about your background knowledge of BASIC, we would start with "Dick and Jane do BASIC Things to Spot". If I assume too much, I waste everybody's time. So, at the risk of my peace and quiet, if you are stuck, give me a call at 733-7989.

First, some introductory words about BASIC are in order. In general, there are two main flavours of BASIC, "interpreted" and "compiled". A third in-between variation is known as "Incrementally Compiled", which we will ignore for the present.

The BASIC that comes with the machine is "interpreted" and has two variations, IBM and everything else. The IBM version is partially in ROM and is known as "Cassette BASIC". This is what you get when you have an IBM PC with no disc drives and a tape cassette deck for storage. Other BASIC commands relating to disc reside in the file BASIC.COM with additional graphic functions in BASICA.COM. The reason for this relates to machine space and copyright laws and can be ignored for all intents and purposes. Those with disc drives have a file on their disc named GWBASIC.COM. Apart from these differences, they all operate in the same manner.

I will start this series with "interpreted" BASIC, and later on, I will introduce the compiler. The Borland Turbo Basic will be used for purposes of illustration, but the principles involved will be quite general.

Let's start with a few words about the conventions I will use. When I say enter a line or something, type it in as shown. Press the key directly to the right of the alphabetical group of keys labelled "ENTER" or marked with a curved arrow. Words that are enclosed with angled brackets refer to a particular key. As an example, <Home> means press the key labelled "Home". <PgUp> means press the "Page Up" key, etc.

<Shift>, <Ctrl> or <Alt> beside another key, means press the two keys together. So <Shift><A> would give a capital letter "A", while <A> alone refers to the lower case letter "a". A "//" followed by text are comments rather than words to be entered at the keyboard.

If you have a fixed disc on your machine, set up a separate subdirectory called BASIC. Start up your machine, and at the DOS prompt, enter:

```
MD C:\BASIC
COPY C:\DOS\BASIC*.*  C:\BASIC\*.*
//for the IBM or Compaq or//
COPY C:\DOS\GWBASIC.*  C:\BASIC\*.*
//for the rest of the world//.
```

The reason for creating a separate subdirectory allows you to keep your BASIC programs from cluttering up everything else.

To start rolling, enter:

```
CD\BASIC
BASICA  //for the IBM or Compaq or//
GWBASIC  //for the rest of the world//.
```

This loads the BASIC interpreter.

At the top of the screen, you will see some legal mumbo-jumbo which is used to keep lawyers fat, happy, and employed. Below it, you will see the letters "OK" and the underscore. "OK" is used by the interpreter to indicate that it is waiting for input.

Our first exercise uses the machine in the direct, or the "big-expensive-hand-calculator" mode. Type 5+4-1 and press <Enter>. The screen will display the answer 8. Try a few other mathematical operations, such as multiplication, by typing 5*3, division, 5/3, and a power, 4^2. Simple and trivial so far, but play with it just for familiarity.

OK. Now lets try another other function, say PRINT. Enter PRINT 4+5*3 and see what happens. The result goes to the screen. Now turn on the printer and enter LPRINT 4+5*3. The output now goes to the printer.

Still trivial, but now enter the line:

```
10 LPRINT 4+5*3
```

What happened when you pressed <ENTER>? Nothing. Now try entering <RUN>. This time the answer went to the printer. The difference is in putting the line number 10 in front of the LPRINT statement. The interpreter considers this entry to be a part of a program, rather than something to be done as soon as <Enter> is pressed. To do something, you have to tell the interpreter to RUN the program.

The next article will focus on the BASIC editor.

* * * * * * * *

FROM THE PRESIDENT...

By David Terroux

Greetings to you all! You elected the Executive to serve you. Our efforts towards meeting your needs can only be met if we are aware of what they are. You can communicate these needs to us through the PUB, the Newsletter, at regular meetings or by telephone. Your active participation in the various Group activities will go a long way towards helping us serve you.

One of our current priorities is the revision of our Constitution which was found to be inadequate. Eric Clyde has already produced three drafts which have been discussed at length. The final draft will be presented to you for discussion when the more contentious points have been cleared up. We are also presently upgrading the Bulletin Board with the addition of a larger, faster hard drive and a fourth access port. A laser printer is being acquired for the Newsletter.

Speakers have been arranged for the next few meetings, although non have usggested they will be supplying marvellous munchies! Guests are always welcome. In fact, there is an incentive program in the works for those members who sign up new members.

* * *

BAT HINTS

By Morris Turpin

CLARIFICATION

I would like to clarify the CONFIG.SYS file as it was printed in the Newsletter last month. The SHELL line should begin with "shell=" and end with "/p" all on one line. There should be a space between "c:\dos" and "/e:1024". It is not obvious the way the line is wrapped in the published article.

ANSI.SYS ESCAPE SEQUENCES

ANSI.SYS is a console/video driver that provides a measure of user control greater than that allowed by DOS. Last month we took the first step to enable us to make use of this control by adding the line "device=[path]]ansi.sys" to our CONFIG.SYS file.

All ANSI.SYS commands begin with the escape character (ASCII code 27 decimal or 1B hexadecimal) followed by the left square bracket ([). I will discuss three methods for entering the escape character: 1) from the command line using the prompt command; 2) using the DOS line editor EDLIN; and 3) using word processors or text editors. Each method produces different looking characters. Let's define some terminology. I will use "ESC" for the escape character itself and "ESC[" for the beginning of the ANSI.SYS command.

The escape character may be entered using the PROMPT command. The PROMPT code for escape is the word prompt followed by $e. The beginning of the ANSI.SYS command then will be:

prompt $e[== ESC[

The prompt command can be used to enter the escape character since DOS does not provide any other way to directly enter this character. The PROMPT method may be used at the command line directly or in a batch file.

The escape character may also be entered using EDLIN. This method may be confusing, so if you do not use EDLIN you may wish to skip to the next section. To create the escape character while in EDLIN's insert mode, press CTRL-V then [(hold the Control key down while you press V, then press [). This CTRL-V[is equivalent to ESC. The beginning of the ANSI.SYS command will be:

CTRL-V[[== ESC[

To further complicate the issue, EDLIN will echo ESC as ^V[and will echo the beginning of the ANSI.SYS command as ^V[[, but when the page or list command is used in version 3.x of DOS, EDLIN will echo the beginning of the ANSI.SYS command as:

[^[== ESC[

Most word processing programs (i.e. WordPerfect and Microsoft Word) allow the user to enter the escape character using the ALT key and the numeric keypad. With WordPerfect, hold down the ALT key and press 2 then 7 on the numeric keypad (do not use the number keys located above your alphabet keys). When you lift your finger off the ALT key, a left arrow will appear. This is the escape character. When using a word processor for such entries, you must save your file as an ASCII text file (CTRL-F5 Text In/Out key in WordPerfect). The text that you entered will appear differently after you save the file and retrieve it back into your word processor. In this case, the escape character will appear as ^[and the beginning of the ANSI.SYS command will appear as:

^[[== ESC[

Text editors vary in how they deal with the escape character. Many require CTRL-P then the ESCape key for ESC. Check the documentation of your text editor to see what it requires. Most, however, will display ESC as ^[and the beginning of the ANSI.SYS command as:

^[[== ESC[

THE ANSI.SYS COMMANDS

ANSI.SYS commands are of the general form ESC[# followed by a letter. The number symbol (#) shown above represents a digit or series of digits and semicolons appropriate for that command. The commands are shown below in tabular form.

COMMAND	FUNCTION	DETAILS
ESC[#A	moves cursor up by # of rows	The column is not changed; # ranges from 1-24; if # is omitted, the cursor will be moved up one row.
ESC[#B	moves cursor down by # of rows	The column is not changed; # ranges from 1-24; if # is omitted, the cursor will be moved down one row.
ESC[#C	moves cursor right by # of columns	The row is not changed; # ranges from 1-79; if # is omitted, the cursor will be moved right one column.
ESC[#D	moves cursor left by # of columns	The row is not changed; # ranges from 1-79; if # is omitted, the cursor will be moved left one column.
ESC[#;#H	moves cursor to row # and column # (row# is the first # and column is the second #)	The first # specifies the row and ranges or from 1-24; the second specifies the ESC[#;#f column and ranges from 1-79; If row # is omitted, the cursor is moved to row 1; if column # is omitted, the cursor is moved to column 1; if both row # and column # are omitted, the cursor is moved to row 1 and column 1 (the home position); enter the semicolon when row # is omitted.

COMMAND	FUNCTION	DETAILS
ESC[s	saves cursor position	stores current cursor position; stored position can be implemented with restore.
ESC[u	restores cursor position	restores cursor to position saved with save command.
ESC[2J	erases display	erases screen; similar to CLS DOS command.
ESC[K	erases to end of line	erases display from cursor position to the end of that line.
ESC[=#h	sets display mode	sets display to monochrome or colour and sets graphics resolution; # ranges from 0-7; # codes are shown below: 0 25 rows x 40 cols, monochrome 1 25 rows x 40 cols, colour 2 25 rows x 80 cols, monochrome 3 25 rows x 80 cols, colour 4 200 x 320 graphics, colour 5 200 x 320 graphics, monochrome 6 200 x 640 graphics, monochrome 7 turn on line wrap, lines longer than 80 characters wrapped to next line
ESC[=7l	turns off line wrap	turns off line wrapping; lines are truncated at column 80
ESC[#m	sets display attribute(s)	omitting # or using 0 as the # turns off the previous attribute; multiple attributes may be entered by separating each # for that attribute with a semicolon and the # codes are shown below: 0 no attribute; turns off attributes 1 high intensity; bold 4 underline 5 blink 7 reverse video 8 invisible

Foreground #   Colour   Background #

Foreground #	Colour	Background #
30	black	40
31	red	41
32	green	42
33	yellow	43
34	blue	44
35	magenta	45
36	cyan	46
37	white	47

COMMAND	FUNCTION	DETAILS
ESC[#;#p	defines keys	defines keyboard keys as single or multiple characters; the first # is the ASCII code of the key to be defined; the second # is the new ASCII code of the character to be used for the key defined by the first #; the second # may be a character string enclosed by quotation marks, including executable filenames or words; # may be a combination of numbers specifying extended key codes for function keys or keys pressed in combination with shift, ctrl or alt (in which case the extended code has a first number of zero and is separated from the next number by a semicolon); any combination of key codes and character strings may be used provided that all are separated by semicolons; to restore a key to its original value, the first # must be identical to the second #.

We now have the ability to place the cursor where we wish on the screen, set colours or attributes, set display mode and re-define keys. I've run out of space, so hold on to this issue. We'll continue next month with some some examples.

* * *

DOS SYSTEM FILES

By David Smith

"Why doesn't this program work when I place the disk in Drive "A" and boot my computer?"

Generally, the first questions, I ask the person is whether or not their disk is bootable. More often than not, their reply comes with a frown and a voice saying, "Bootable? What does that mean ?".

One of the first things to comprehend is that a DOS-based computer requires only three files to operate. One of them (COMMAND.COM) can be seen on the screen while the other two (IBMBIO.COM, IBMDOS.COM) are hidden.

COMMAND.COM contains "internal files" such as Copy, Erase, Prompt, Dir and so on. If these three files are installed correctly on a floppy (or hard disk) <u>and are located in the root directory</u>, the computer will load them into a section of the computer's RAM where they become "memory resident" until the power is turned off.

The second area of confusion is how to "install" these three files in the Root directory. (Note that the word "Copy" has been avoided since it is impossible to use the Dos "copy" command to transfer the two hidden system files.) There are in fact, only two ways to transfer these two hidden files to a particular disk.

Let's suppose you wish to make a duplicate of your DOS disk(s). This may be done using "Diskcopy", the simplest method. "Diskcopy" creates an exact duplicate of the original disk AND INCLUDES THE SYSTEM FILES or by formatting a new disk using "Format /S". and then, using "Copy *.*" to copy the rest of the files onto the new (and now bootable) disk.

If your PC does not have a hard disk, having a bootable program disk eliminates the need to first load a DOS disk and then loading the program disks. For those who have hard disks, keep in mind that neither "Copy" or "Xcopy" will transfer hidden system files.

Should you wish to upgrade your Hard Disk DOS from an earlier version to a more current release, you must boot the computer with the version you wish to upgrade to. Then type SYS C: ("C" meaning the Hard Drive). This will overwrite the existing system files.

Now, delete the original DOS files from the DOS directory and copy the newer version to this directory.

The 'COPY' command will also copy COMMAND.COM to the DOS Directory. but as this system file will already be in your disk's root directory, erase this file from the DOS directory.

The bottom line is that should you wish to make blank "Bootable" floppies, use "Format /S". To make duplicates of any program disks which are bootable, use "Diskcopy" to transfer those slippery DOS System files.

* * *

CALL WAITING DEFEATED!

By Gordon Phillips

Bell Canada offers it's phone customers a "call waiting" feature. If someone tries to call you when you are using the phone, "call waiting" will "beep" to inform you that someone else is calling you. You can press the switch hook to put the current conversation on "hold" and talk to the other caller.

Unfortunately, this new feature causes problems for computer-users with modems. A "beep" in the middle of your download can cause garbage, a failed download, or, even a dropped line.

There are two simple solutions. You can have call waiting removed completely. Or, if you like the call waiting feature, you can dial *70 for touch tone, or 1170 for pulse dialling, which will disable call waiting and return to dial tone. You can then place a call without fear of being interrupted by an outside call. Call waiting will be restored when you hang up.

Most modems sold today are compatible with the "Hayes" command set. The "w" command, tells the modem to "wait for a dial tone" before continuing. Placing this "w" in the your communication software's dialling prefix after the code to cancel call waiting will prevent the modem from dialling until it gets the second dial tone. Adding "w" right after the "atdp" (or "atdt" for tone dialling) will also prevent your modem from dialling if an extension phone should happen to be in use.

All of this can be combined into one very useful dialling prefix: "atdpw1170w" for pulse dialling or "atdtw*70w" for touch-tone dialling. The first "w" says "wait for the first dial tone". The "1170" (or "*70") says "cancel call waiting". The second "w" says "wait for the second dial tone". If you drop the call-waiting feature, you must remove the "*70w" or the "1170w" from your dialling prefix.

* * *

# IT'S RENEWAL TIME!

By Carl-Henri Gomez
Membership Chairman

Renew early to avoid uncomfortable lineups! Membership privileges are numerous and the value of your membership fee is rapidly increasing.

Heath-Zenith Computers has agreed to give discounts of up to 25% on computers and accessories to card-carrying members of the Ottawa PC Users' Group. A discount schedule is forthcoming.

* * *
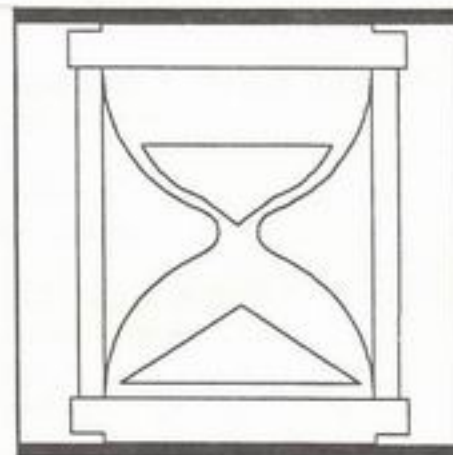
WORDPERFECT SOLUTIONS

By Bonnie Carter

The common method to move text is with the Move F-Key <CTRL F4>. However, it involves several key strokes and can be both time-consuming and inefficient. Most WordPerfect users know that the <F1> key's primary function is to cancel commands. You might be wondering why I am talking about two function keys that are unrelated. There is a good reason for this.

To move text from one area of a document to another, the first thing you always do is block the text you wish to move using the <ALT F4> key. Then you press <CTRL F4>, right? Well, maybe.

Did you know that you can use the <F1> key to move text in a way that is faster and more efficient than the "Move" key? You can use it to move sections of text from one line to another, one page to another, and even from one document to another. You can move a word, line, sentence, paragraph, a page or even more.

You start in the normal fashion by pressing <ALT F4> to block the text you wish to move.

Now, press the "delete" key. The blocked text will disappear from the screen. Don't panic! You haven't lost it. Move the cursor to wherever you want to place the "deleted" text.

Here is where you digress from the normal method of moving text. Press the <F1> key. The text that you deleted reappears exactly where you have positioned your cursor, and it is still highlighted. You will see two options, "(1) Restore" and "(2) Previous Deletion". Choose "Restore" and the highlighting will disappear, but not the text.

Using the same method, moving text from one document to another is simple and fast. Divide the screen into two parts by pressing <CTRL F3> (Screen). Choose (1) Window. You will be asked "How many lines in this window?". Type the number 12 to equally divide a 24-line screen in half.

Now, press SWITCH <SHIFT F3> to move the cursor to the lower half of the split screen. Note that the Tab markers now point downwards. To retrieve Document two, press <SHIFT F10>. The second document will appear under the bar.

To move text from, Document one to Document two, block the text as described above, delete it, and, using switch, move the cursor to document two. Position the cursor exactly where you want to add the text from Document one and press <F1>. Choose restore, and your blocked text will be retrieved into document two.

I find these methods of moving text very handy. In fact, I never use the "Move" key.

* * *

Due to a mechanical problem, the table in the Treasurer's Report was incorrect. The corrected table is below.

1989 TREASURER'S REPORT - SUMMARY

		1987	1988	1989	Projected 1990
BANK					
	Opening Balance		$12,436	$7,682	$10,689
	Revenues		$17,932	$15,986	$22,000
	Expenses		($22,686)	($12,979)	($26,000)
	Closing Balance	$12,436	$7,682	$10,689	$6,689
INCOME					
	Membership		$11,340	$9,660	$14,500
	Disk Sales		$6,304	$5,887	$7,000
	Bank Profits		$288	$439	$500
EXPENSES					
	Software Library		($5,453)	($4,480)	($5,000)
	Newsletter		($5,655)	($3,351)	($6,000)
	Bulletin Board		($10,780)	($4,134)	($6,500)
	Miscellaneous		($798)	($1,014)	($1,000)
	Incorporation				($1,500)
	Hardware				($6,000)
ASSETS					
	Coffeemaker			Membership software (Reflex)	
	Newsletter software (PageMaker)			Library (diskettes/boxes)	
	Corel Draw software			Public address system	
	BBS equipment/software			Telephone answering machine	

The 1989 Annual Balance Sheet was audited by Robert P. Dulude and Paul Baudette.

OTTAWA PC USERS' GROUP

EXECUTIVE

President	David Terroux	238-4895
Past President	Stu Moxley	592-4933
Treasurer	Tony Frith	231-7250 Weekdays
		671-0401 Weekends
Secretary	Claude Jarry	521-3366
Facilities Chairman	Douglas Poulter	745-8768
Membership Chairman	Carl-Henri Gomez	731-1462
Meeting Facilities	Stu Moxley	592-4933
Publicity	André Cyr	561-5207
Software Librarian	Chris Taylor	723-1329
Newsletter Editor	Bonnie Carter	236-1015
BBS System Operator	Mike Schupan	820-0293
Bulk Purchasing	Terry Mahoney	225-2630
		226-2615 FAX

ASSISTANTS

Software Assistant	John Ings	235-8132
Newsletter Assistants	Marc Riou	733-2092
	Ted Harvot	733-7210
Contributing Newsletter Editors	Chris Taylor	723-1329
	Marty Sells	829-5606
	David Smith	837-9291

DISK OF THE MONTH

One-year subscription (10 DOMs) 5 1/4"		$25.00
One-year subscription (10 DOMs) 3 1/2"		$45.00
Individual 3 1/2" surcharge		$2.00

MEMBERSHIP FEES SCHEDULE

From January 1, 1990 to March 31, 1990	$15.00
From April 1, 1990 to March 31, 1991	$25.00

THE OTTAWA PC USERS' GROUP MEMBERSHIP APPLICATION - Please Print

Last Name: _____ First Name: _____

Mailing Address: _____

Postal Code: _____ Telephone - Home: _____ Office: _____

Profession: _____ Date of Birth: _____ Sex: M _____ F _____

Membership Period: Present until March 31, 1990: _____ April 1, 1990 to March 30, 1991: _____

Disk of the Month: YES _____ NO _____ Size: 5 1/4" _____ 3 1/2" _____ Amount Enclosed $ _____

Are you: A new member? _____ Renewing your membership? _____

How did you find out about the group? _____

What in particular interests you in the Group? _____

Can you help in Group activities? Check off the activities that apply: Programming language Instruction _____

Newsletter Input _____ Memberships _____ Software Library _____ Promotion/Publicity _____

Hardware Techniques _____ Meeting Locations _____ Agendas & Speakers _____ Advertising _____

Bulletin Board_____ Other_____

What hardware/software do you own and/or use? _____

Comments and suggestions: _____