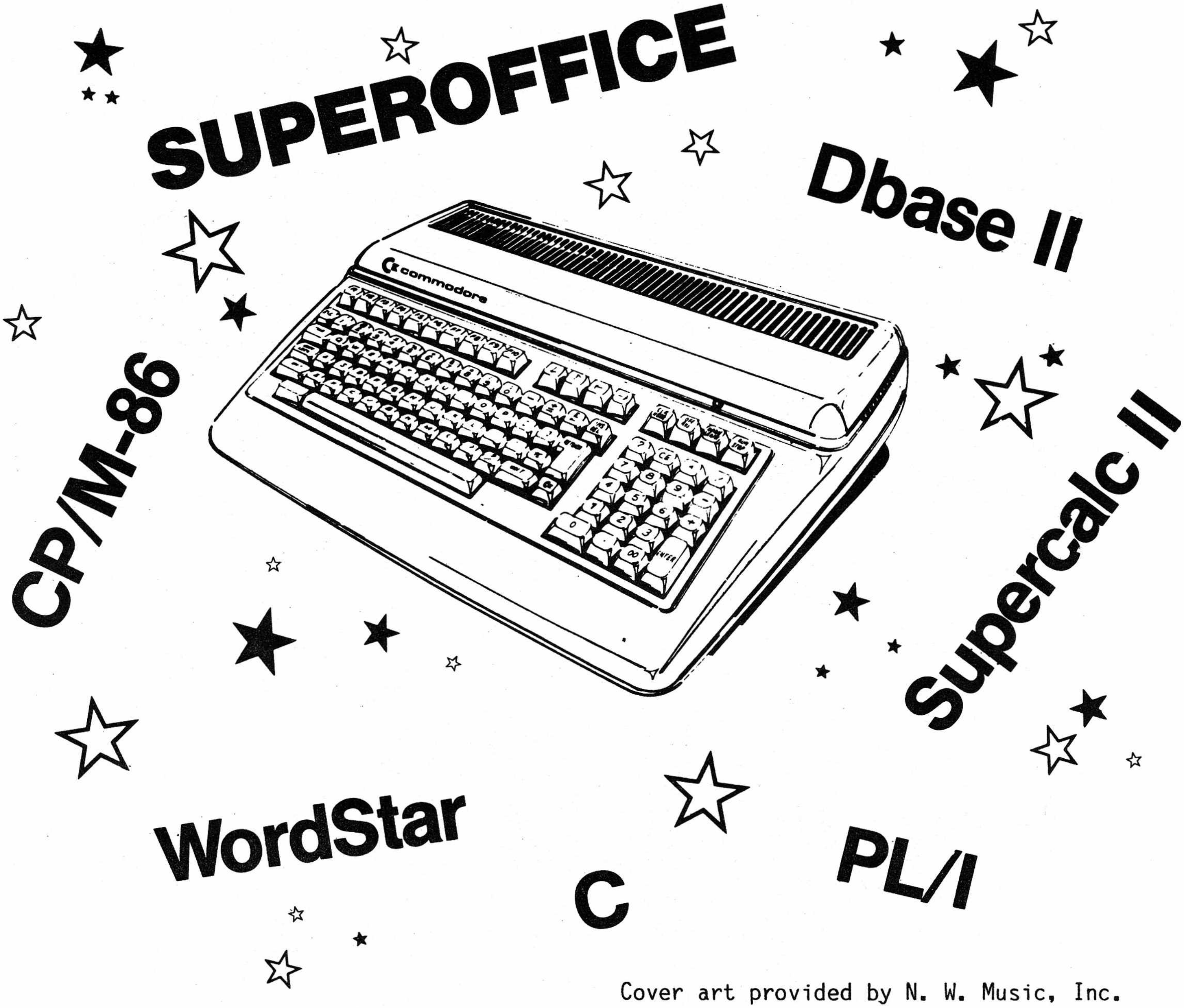
CHICAGO B128 USER'S GROUP -- INTERNATIONAL
4102 N. Odell. -- Norridge, Il. 60634 USA

**DATED MATERIAL
DO NOT DELAY!!**

- -

Return the ENTIRE Page -- including the mailing label. Make corrections as necessary
Please help us out. Print very legibly or type

- -



Cover art provided by N. W. Music, Inc.

USE PHOTOCOPIES FOR MULTIPLE ORDERS – ADDITIONAL ORDER FORMS WILL BE INCLUDED WITH SHIPMENT W/S 87

NAME ______________________________

ADDRES ______________________________

CITY ____________________ STATE ______ ZIP __________

THIS IS YOUR SHIPPING LABEL — PRINT OR TYPE NEATLY

ORDERS OVER 4 DISKS OR PHYSICAL EXAM SHIPPED BY BY UPS WITHIN CONTINENTAL U.S. YOU MUST USE STREET ADDRESSES FOR SUCH ORDERS — UPS CAN NOT DELIVER TO POST OFFICE BOXES!

Your phone # ______________________________

Remit to: CBUG, Inc., 4102 N. Odell, Norridge, Il. 60634 U.S.A. All payments must be in US funds.
IMPORTANT: If your shipping address is different than your membership address, please give your membership address!!

My Membership Address (if different) is: ______________ street ______________ city ______ state ______ zip

- -

R ck mo ca pr S

- -

Description	Quantity	Stock #	Price	Extension
SSDD Premium Opus Disks/Pkg of 10 w/ sleeves & labels		10017	$ 5.70	
COMMODORE IEEE to Centronics Adaptor Board (6400 type)		11221	35.00	
CBUG connector type changer for above (changes output connector to standard Centronics "blue ribbon" D shell type		11236	15.00	
< Data Cases Available To > TOTAL FROM DATA CASE ORDER FORM (back cover) --------> < N. America Addresses ONLY >				
MERCHANDISE SUB TOTAL				
Illinois residents add 7% Sales tax above items only				
CBUG #M54 CBUG MISC.		11930	9.00	
CBUG #M55 CBUG MISC - ML Programming Info		11944	9.00	
CBUG #56 Harrison's Assembler, revised, #5.5 v8 (If upgrading your prior copy, attach label and extend at $5.)		11959	35.00	
CBUG #57 Goceliak Strikes Again. INCLUDES tape and drive codes!		11963	9.00	
CBUG #58 Dittinger's Utilities +		11925	12.00	
CBUG #59 Winter/Spring 87 Print Files (This issue on disk)		11978	9.00	
CBUG #51 THE JCL WORSHOP & ASSEMBLER w/ wordprocessor All on one disk. ++ Superscript version of instructions on accompanying disk!)		11894	29.00	
CBUG #53 Fall 1986 Print Files		11911	9.00	
KNIGHT's 8050 (DOS 2.7) COPY UTILITY		11204	20.00	
PHYSICAL EXAM for the 8050 Disk Drive		11219	35.00	
PHYSICAL EXAM for the 8250 and SFD-1001 Disk Drives		12192	35.00	
BEELINE v2.1 Telecommunications Program		12280	40.00	
9060/9090 Service Manual - photocopy (allow extra week)		12295	25.00	
SFD-1001 Schematics by photocopy of stat reductions, 11" x 17"		12308	5.00	
Recopy fee remittance		12401	5.00	
BALANCE FORWARD FROM PAGE 2 RIGHT HAND SIDE OF FORM ------->				
BALANCE FWD FROM ARCHIVE ORDER FORM (last page of library section) ------->				
MERCHANDISE SHIPPING AND HANDLING CHARGE --- ALWAYS INCLUDED, NO EXCEPTIONS				$ 2 . 0 0
SUBTOTAL				$
Free Will Contribution to CBUG.		12416		
Extra Copy of this issue, Winter/Spring 1987 CBUG ESCAPE		12608	7.00	

BY CHECK ___ BY MONEY ORDER ___ TOTAL REMITTED $ __________

NOTE: We endeavor to ship against money order within 3 business days of receipt. Others delayed 14 to 18 days

THE CBUG ESCAPE

FIFTH ISSUE

Winter/Spring 1987

THE CBUG ESCAPE is published 4 times a year, more or less, by the Chicago B128 User's Group - International (CBUG, Inc.), an international membership organization in support of applications and usage of the Commodore B128 Computer.

CHECK YOUR ADDRESS LABEL FOR EXPIRATION DATE. The expiration date is to the right of the postal code in the form of YYMM, eg. 8712 indicates an expiration at the end of December 1987. CBUG will be unable to mail publications without renewal. PLEASE KEEP YOUR MAILING ADDRESS CURRENT.

CBUG is NOT affiliated or allied with any other organization, user's group, business or other entity of any kind, except in support of CBUG chapters.

Advertisements, articles and contents of disks are solely the responsibility of the individual authors. Their existence in a CBUG publication implies no endorsement by CBUG. However, please report to CBUG as to exceptional performance, either pro or con.

Publishing address: CBUG, Inc. c/o Norman Deltzke, 4102 N. Odell, Norridge, Il. 60634 USA

Cover price this issue: $7.00. 1987 subscription rate: $14.00 (bulk rate, U.S. & possessions ONLY); $20.00 (first class, U.S. & possessions, Canada & Mexico); $21.00 (surface <boat> first class, all others); $35.00 (small packet rate air mail, any country).

TABLE OF CONTENTS

CBUG SCRATCH PAD

By: Norman Deltzke

June 1, 1986

Every issue I've had to apologize for being late with the issue. So I guess we are just going to have to adopt the motto "Published more or less 4 times a year." More about that later though.

First though so no one misses this, I'll often put my two cents in the middle of an article. ALWAYS, the comment is enclosed between double angle brackets: <<comments as in this form>>.

For the good news, carefully read Rev. Schwarzbauer's, Mr. Faierson's and Mr. Matos' articles. For that matter, almost every article has something new and unexpected. Too good to be true! Another friend of CBUG, Dennis Jarvis, is about to publish a huge compendium of all Commodore 6502/6509 based computer memory maps, cross indexes, the whole thing, including our baby, the B128! - Soon the issue of modifying programs from VIC, C64, C128, Pet thru 8096 will be made far easier. He claims all data has be verified rather than merely reproduced from various incorrect CBM publishings or attempts to reverse engineer machines. The book will be published by Transactor Magazine and I expect CBUG will also participate in its distribution. Dennis and I wish to acknowledge the kind assistance of CBM in this project and specifically the permission given to CBUG to share the highly proprietary materials CBUG has accumulated so that Dennis could include them in his publication. Dennis has just bought a B128 and expects to become an active contributor in the near future. (Incidentally, Dennis is chief robotics engineer for Disney's Epcot Center.) He has already written code and designed a cassette port to serial drive adaptor (Gary Anderson will likely be manufacturing it). This will allow direct use of 1541, 1571,1581 and other similar drives on the B128.

Another interesting point of good news. While some of our members have suffered from equipment failure, my "gut instinct" statistics suggest that the B128 system is one of the heaviest duty and reliable personal computers ever brought to market. This is quite understandable as the system has its roots in the old PET products, several of which such as the 8032 computer have been mainstays in public education (computer & scientific), the workhorses of thousands of laboratories in schools, industry, and military. Commodore's commercial products (vs consumer products) generally have a well deserved reputation for being over built and reliable equipment. As CBUG matures we will see more competent service shops and parts suppliers come on line. The biggest worry is that they will find themselves in the position of the Maytag repairman, nothing to do because there are so few breakdowns. It is noteworthy that the mean time between repairs on several of the most popular personal computers (and not the cheapy imports either), has been reported at 4 months! No one in the CBUG group has suffered anything approaching that; most have never needed a repair shop!

HELP US -- ARTICLES WANTED

We are going to make a serious effort in getting the Summer issue out by early September. That's a whole summer away. BUT, if we do not receive most of the material to publish by very early August, it just won't happen. I pray that our authors (and many are very good about being early) try to move up their transmission date.

As Angel Matos points out in his article, too few members are carrying the authorship load. I talk to dozens of different members every week who have discovered or written things of terrific value to other members. Publish! PUBLISH!! -- if only for a basic starting point for some other member to use and modify. Let's get some more authors on line. Let's get some more library programs going!

P.S. the pay is truly fantastic. Not a penny!. Periodically we go back thru our current contributor lists and send out chits (ticket coupons) in modest quantity to the contributors. These are redeemable for library disks, blank disks, etc. At the moment we have not done this yet this year, so this will be our late June project (for our existing contributors).

PUBLICATION & MEMBERSHIP COMMENTS

It seems that when The ESCAPE gets on the street, everyone stops to read and reread it-- for weeks if not months. So the blood stops flowing. It restarts when people realize that the next due date is already past. In March we did not have enough articles to fill a handbill for an amateur stage show production! Then it picked up and soon we had twice what was needed -- but then the issue gets twice as big. So this issue again is a double issue -- 2 of the 4 for the year, but then, get more articles and library material in sooner and there just might be more than the planned issues. With only one exception, to the best of our indexing capability, everything on hand right thru the last minute, has been included in this issue. Our policy to date has been not to hold for the future!

As you will read in this issue, all sorts of wonderful things are beginning to happen. Shepherding them has become incredibly time consumptive; but certainly worth the delay in publishing.

One additional sink hole for time over the last six months is the problem with SEI Service Center discussed in the front insert.

In brief summary here per chance the insert was not included in this copy: In the two summer 1986 issues, SEI Service Center ran ads and articles regarding their upgrade and repair service for the 8050 drives. Unfortunately something went wrong and to date no such upgrades have been delivered, and generally speaking the company is refusing to communicate with customers, attorneys general, and several law enforcement agencies. DO NOT send equipment or money to SEI till this matter is resolved. If you are a victim and have not reported to CBUG, send CBUG a detailed diary of what was said and done, when, copies of all documents including front and back of checks, waybills, etc. This information will be forwarded to the several agencies currently investigating the affair.

Membership renewals this spring were amazing. Renewal notices were mailed December 27 via first class. About one third of the membership had renewed by mid February. Even today as I write, membership renewals are coming in. About three quarter of the Dec 86 membership roster has now renewed. Give some thought next December when you get your renewal notice as to what would happen had we published the Winter 87 ESCAPE in March as due? Those not renewing promptly would miss the issue but CBUG would feel obliged to send the issue, of necessity, hand addressed and by first class mail! Mechanically it would be a nightmare to process and also cost an extra buck fifty to mail. CBUG operates with annual subscriptions at this time. Late joiners in any year should automatically receive that year's past issues, thus joining in Sept 86 would have produced catchup issues in a few weeks, but renewal was still Jan. 1987. If anyone wishes to express their opinion on the renewal question, send me a post card (no disertations please) with your vote of the following options:

1.) Continue as is.
2.) Convert to subscriptions running one year from date of receipt of payment and no back issue catch up. (purchase from library as desired).
3.) Continue as is with respect to new members, HOWEVER NO catch up on tardy renewals; receiving only the balance of the year regardless of renewal date. Forfeited issues would be purchasable from library.

It is most interesting to note the continued flow of new membership applications. We receive over a dozen calls a month from prospective members -- generally people who've been using their systems successfully, but now find they don't know it all! Then too we receive a few membership applications from people who bought a system from a non-member, and a few from non B128 users just because of the wealth of unique information we are publishing. The most interesting catagory is the applications flowing from the first issue in November 1985 -- YES 18 months ago. Last month we even received an application which was distributed locally in June 1985!. Take a few minutes and think about how to balance all this information and remain sane (and not make any mistakes!).

As to mistakes, we've been making more than our share. We believe all shipments of library material EXCEPT the high boy schematics are completed. If you've been shorted, etc., please let us know. The family's been doing better than I, --- at least library shipments are current

ARE YOU MOVING? --- Hey you!, let us know!! CBUG spends several hundred dollars an issue to mail duplicate copies to people who have moved -- if and when the Post Office returns the notice of non-delivery/moved persuant to the request for same on each issue. When we sent out the "we are late" post cards in May, over 200 came back either from the post office or from the addressee indicating a move. Had that been an issue, $400 would have been wasted. Worse yet, had the forwarding order expired, we would never know where you moved.

ZIP CODES and Names. We receive many letters with only a signature. ALWAYS put your name and zip code on each page and your full address somewhere in LEGIBLE form. Also provide an evening phone number. On change of address notifications be sure and give us your old address and zip else it takes us a 20 minute database search to find you by name -- assuming your subscription was under the exact same name.

PUBLISHING NOTICE

The CBUG ESCAPE is published by CBUG, Inc., c/o Norman Deltzke, 4102 N. Odell, Norridge, Il. 60634. 312 456 8720 7pm to 10pm CST Wed thru Sat. All Rights Reserved. Permission of both authors and CBUG, Inc. required for reprinting or republication. The opinions, averments, statements and representations of authors and also those of advertisers are the sole responsibility of those authors and advertisers. CBUG assumes no obligation to apply editorial prerogative. Evaluation of materials appearing in the CBUG ESCAPE is left solely to the reading member.

A WORD OF THANKS

To my wife Barbara, who has allowed CBUG to overrun our home. The family room is wall to wall computers, drives, printers, library shipping area, filing cabinets, even the walls are totally covered with shelves full of CBUG! Worse yet the dining room remains the home of the disk duplicator, cases upon cases of printing past and current. Every few months the picnic table moves into the dining room for collating and mailing of The ESCAPE -- replete with about 60 post office mail bag. Even the kitchen isn't spared -- it gets the industrial power stapler and cartons of paper.

And to our children, Bill and Jennifer who along with a half dozen of their friends join the collating and mailing task. Even my mother-in-law chips in!

An unusual home for sure. But the thanks we so often hears says its an investment well made. And to you, CBUG members, I thank you for the opportunity to contribute what I'm able.

ANNOUNCEMENTS

PARTS & SERVICE SOURCES

N.W. Music (see center fold advertising) advises they have acquired quantities of spare parts for the B128 system including the special CBM only chips.

Jim Robbins, Midwest Computer, 2140 N. Westwood, Poplar Bluff, Mo. 63901. 314 686 2292. CBUG has received several favorable reports on Midwest and none adverse. Midwest claims to be experienced in repair of 8050, 8250, SFD and other equipment.

Second Source Engineering, Sacramento Ca. We have received a complaint regarding slow delivery. In review of the incident it appears that Second Source did not have the CBM service manual and diagnostics for the 8050 on hand, hence the only "sin" was taking in equipment without those items on hand. CBM at that time was notoriously slow (often never) about filing orders for those items. The repair was successfully completed.

Consult prior issues for the rest of the service stations listed with CBUG. Also check the advertising sections of The ESCAPE.

HINTS & TIPS

Superscript II's Search and Replace can also be used with control characters with this trick. For example, if you need to change the underline symbols to bold symbols the immediate problem is that those symbols can not be directly typed on the control line. Simply set up a line that reads ' *nb "x""y""z" ' at the top of the page. Where x,y,z can be any control symbol, reverse asterisk phrase such as *lm3 etc. To put the symbol in the control line field, put the cursor at the front of the symbol line, then use the Commodore key. Each time you hit the C= key the next symbol between quotes appears. Hit return when you get the correct one. You can also search and replace a series of linked files. Just take care to make sure the same sequence does not occur in normal text. Strings like Search *lm1 Replace *lm3: should be adequately unique.

I've not had a chance to try Goceliak's patch which allows the operation of a C2N cassette recorder with the B128. HOWEVER, from superscript, instead of selecting a disk drive device number at the opening menu, type ' t ' instead. The when you save or load you get the tape operation screens. Take note that the cassette is powered only while in Superscript tape mode so you must fast forward and rewind before exiting tape mode. The Directory commands however go straight to disk which is a bit confusing.

From R. VanNieuwenhuirzen: In order to assign your 4023

to device #5 you must go to U4D on the board and pull out **pin #11. This** will make the 8023 device 4 and the 4023 **device #5.** This will enable you to have both printers on **line and** direct the print to either one. <<The chip **number** for setting device number in the 8023 is U10, and **is also** a 6532. Opening pin 11 adds one to the device **number,** pin 12 will add 2, and pin 13 ads 4 (which would **put** the printer at 8 which is reserved for disk drives.) See the discussion on device number changes in the 8050 manual. The schematics all clearly show the above "pattern matching" pins.>>

From Dick Forman: On the 4023 when printing single sheet the paper out switch is a major irritant. An easy way to defeat the switch is <<I assume also on the 8023>> to insert a scrap of paper over the switch (located at about 45 degrees above the rear bottom of the platen roller. Secure it with Scotch tape or even easier, use a post it note.

From Bob Bushre: A bit more useful fix for paper out: Install two switches: a SPSP toggle switch in series with one of the wires going to the paper out switch. Opening the circuit disables the paper out. A SPSP NO push button across the two original paper out wires (before your new toggle switch) serves as a pause control. There is adequate space on the front left printer pannels to install the switches.

NEW B128 BULLETIN BOARD

David L. Evans II has opened a BBS running on his B128. The phone is 208 454 8421. 913 1/2 Dearborn, Caldwell Id. 83605. Voice 208 459 3279.

ERATA

On the CBM Diagnostics Disk CBUG #48, there is an error in "mpi test" program at line 1420. Remove the extra quote " sign.

MEMBER QUESTIONS

From Richard McNamara, 3CON. Wille St. Mt. Prospect, Il. 60056. re: CBUG #5 Financial Utilities, Mortgage program, printing an amortization table from.

"The program runs well until the last year is printed, then the year starts with the 13th month and not totals printed at end of last year. Please respond directly to Richard and also to CBUG with required fixes for future publication in print.

From David Farmer, Rt 1 Box 145, Grundy Va. 24614. Is there a program that will print sideways on the B with a 4023? Respond directly & to CBUG.

THE B-128 (CBM600/700) IS ALIVE AND WELL IN GERMANY!

By: Mark Schwarzbauer

If you recall my last article, I mentioned that there were B-128s stacked up in the warehouses in England. Well, it seems that most, if not all, of these made their way to Germany and the surrounding areas of Denmark and Austria. There were already previous shipments that had made their presence known in those countries.

When I was in England, Gail Wellington (now tranferred back over to Commordore America from Commordore International) gave me a copy of "Commodore Computing International". It contained the name of a gentleman in the classified section who was looking for information on the CBM 610. I passed it on to Norm and CBUG became known amongst the then organizing CBM 600/700 Selfhelp group in Germany.

Having a sister group has proved to be very helpful. They already have materials we do not yet have and of course there is much we have that they do not. They are a very hardware oriented group by nature and aside from a few of our resident guru's like King and Anderson, we are more software oriented.

As I write this article for the Escape, we are in the midst of exchanging information and materials, arranging legal matters of distribution overseas, and the royalty agreements. Though this issue of the Escape is late, the wheels at CBUG have been constantly turning. A few of the materials they have available are...

1. A CPM-80 emulator with no co-processor board needed. <<The co-processor board supports CPM86 for the 8088 processor.>>
2. A cassette port to 1541/1571 drive cartridge
3. An advanced 8432 emulator with instructions allowing multitasking
4. Macro-basic, a powerful basic extension
5. More is coming in.... stay tuned!

Most of the German users have SFD1001 drives or the 8280 (not a misprint). The 8280 is an 8" dual drive IEEE CPM/MS Dos drive unit designed for the B and never released here in America although some were built here. We know some were built here because Norm has one with the good ole MADE IN U.S.A. stamp on it! Why we never saw them no one seems to be able to tell us.

The B-600 machines sold for prices similar to those in the US. The drives were of course more. Because of the drive prices being higher they quickly made an adapter to run the 1541 on the B! Included in the adapter cartridge is TURBO-TAPE. These are offered for around $50.00 in Germany. However, there are versions that the German group is making that may be cheaper and I understand that Gary has been putting one together also.

We have so much available here that they do not have. For example, they don't yet have SuperScript/Superbase. The only word processor they have is hundreds of dollars and not very efficent. By the way, SUPEROFFICE has been updated. It now contains SUPERBASE version 2.08. This is even a higher level than SUPERBASE II. If you don't have SUPEROFFICE yet you are missing out on more than you can imagine. The integration of the two programs really gets the work done. Check the Northwest Music ads for more information.

The German group has been doing one meg upgrades along the same lines as our folks. Unfortunately, their 1 meg machines, just like ours, will not support the co-processors. We assume that there is a timing problem here but we are not certain. They have even pondered upgrading the 6509 to a 65816 and thereby speeding up the machine with a 6509 emulation. Much more awaits us as we merge our efforts to support the B.

We are getting materials translated at this time and we need more help. One of our members Jergen Bilhofer has been swamped by the materials and we need more German tranlators. Sprechen sie deutsch? Anruf mich auf 414-743-4151 und haben sie ein guten lesen. Danke.

We'll tell you more as it comes so hang close to your ESCAPE! AUF WEIDERSEHEN

Notes from the Underground

By Angel M. Matos
(c) by Angel M. Matos

**********Bulletin**********
**********Bulletin**********

As we are in the process of submitting this article to Chicago for publication, major changes are taking place on the Delphi Network. By the time you read this, Flagship

Commodore will be no more. A new Commodore specific SIG (special interest group) will be onltne. As of today's date, the name for this new SIG is "Commodore SIGs". To get into the SIG directly from the Main Menu on Delphi Network, you can type "GR COM", or "GR CBUG". If you go to the GROUPS Menu first, you can type either "COM" or "CBUG" to get in. You will then see a sub menu listing the four SIGs that currently service the Commodore user, CBUG OnLine will be the first item on this sub menu. Simply type "CB", and you will be in the CBUG OnLine area.

PLEASE NOTE: by the time you read this article, the name for the CBM sigs might have changed (it now under its third title), and the entry methodology might be slightly different. Not to worry, it should not be that difficult to find the Commodore SIG (regardless of name) in the Groups Menu, and "gr cbug" should still work from the Main Menu. <<shortcut direct entry "GR CB CB".>>

The new SIG will have four CBUG specific areas, with which we expect to serve you better, we expect to make it CBUG Central. Listed below, you will see the four topic areas that have been set up, as well as each areas sub-topics:

*CBUG OnLine--
--Announcements
--El Cheapo's
--For Sale
--Odds & Ends

*Software for the B-128
--SuperScript
--SuperBase
--CalcResult
--WordResult
--Info Design
--Library

*Hardware for the B-128
-B's
-DiskDrives
-4023/8023
-6400
-Other Printers
-MS DOS & CP/M

*Communications for the B-128
--Modems
--BeeLine
--TelrTerm80
--BTerm
--Null Modems

Keep in mind that things are still under development, so drop in and see all the great goings on.

In the last column I mentioned that we had a "free sign on port" for CBUG members. Follow the instruction laid out in the last column, but the new Username is "JOINCBUG", and the new Password is "NEWMEM". So dial in, join us, and check things out. CBUG OnLine = CBUG Central.

By the way, this column will be telecommunicated to Norm in Chicago, via Delphi's EMail (electronic mail) to make it as up-to-date as possible.

***********Bulletin**********
***********Bulletin**********

We now return you to our regularly scheduled ramblings.

Many thanks to all of you that have contacted me since my last column, its good to get some direct feedback from the folk. I will try to address some of the issues you raised with me. Some of the points are follow-ups to issues addressed in the last CBUG "Escape", some are totally new issues. Also, please accept an apology for those rough sections in the last column, but unfortunately some words, as well as whole sections were chopped up (out) in the pre-publication editing and formatting process. Apologies from Norm and myself. Well nothing to do now but, chaaaaaargeeeee !!!

This Issue's Topics

<1> Catching up with the last column
<2> The other modems, plus
<3> A note on SuperScript II
<4> Keeping the power on
<5> Protecting your equipment
<6> Computers and the Blind
<7> Closing comments

<1> Catching up with the last column

Among the items that drew the most comments and inquiries were:

The availability and price of the CBM 6400 printer.
The availability and price of the Star PowerType printer.
The differences(s) between the Avatex 1200 and 1200HC modems.

1.1- Re: CBM 6400---

To understand the saga of the CBM printer, it is best if we start with a basic review of some of the facts in the computer hardware industry. Many products are manufactured by one company, and marketed by the manufacturer (sometimes) as well as another company (or various companies) under its (their) label(s). One of the best examples is Epson's famous MX-80, which also served as the original IBM printer, as well as a whole series of other printers.

In the automobile industry this is best typified by a "generic" GMC product which is marketed by their Buick, Pontiac, Cheverolet, etc., divisions with slight cosmetic differences. Sometimes there are production modification which make one division's product superior to another's, for example some folk still feel that the Pontiac power plants are superior.

The same conditions exist in our market, as well as the electronic market in general. Therefore, a generic product can also exist with or without cosmetic differences, major or minor modifications, under the name of its manufacturer, or that of one or more other companies. Now lets talk about the CBM 6400. The CBM 6400 comes from the generic model F-10 manufactured by TEC of Japan. Besides its generic and CBM incarnations, it also existed as the C.Itoh Starwriter F10-40, and as the Facit 4565, and who knows how many more. These magnificent machines have been on the liquidators block for the about the last 15 months. Depending on the model, the liquidator, and the timing, the prices have varied. Considering that the original list prices ran from a low of $1295, to a high of $1895, the liquidation prices have been delicious.

A few more notes on the phenomenon of "generic" and "label" products. Sometimes a product is designed by the company that manufacturers the product, and sometimes the product is designed by the company that will eventually market it. Two products that exemplify these concepts are the Tandy 100 Portable NoteBook Computer, and the IBM PCjr Computer. The generic manufacturer of the Tandy 100 is Kyocera of Japan, the generic product is the Kyotronic 85 portable (currently being liquidated for $149 by D. L. Liquidators, see above), the label products are the Tandy 100, the NEC PC-8201, and the Olivetti M-10. Each model has enough differences to make them incompatible and unique. The IBM PCjr was designed by IBM, but it was manufactured by Teledyne. Teledyne is a conglomerate that bring you Acoustic Research audio speakers, WaterPick dental hygiene products, and various "toys" for the defense industry.

By the way, from some bizarre reason the company that slaps a label on a product is called an OEM, or Original Equipment Manufacturer, which doesn't make much sense, but such is life. If anyone out there knows why the company that slaps a label on a product is called an OEM, instead of the company that manufactured the product, please drop me a note and I share it with everyone in the next column.

<<Once upon a time, life was much simpler and many companies make the perponderance of the product they sold from small puchased or inhouse manufactured parts. Now many firms see advantages in buying many manufacturing

operations in the form of completed sub-assemblies either from subsidary firms here and abroad, or often easier, just subcontracting the manufacturing to contract manufacturing firms. In the reverse, foreign manufacturings seeing the political destiny in making their products in the country of sale, simply set up "satelite" assembly plants with the parts all made in the firms home country or foreign plants -- often with the cost/profits filtered thru to the home office, tax haven, etc.>> While my days belong to a truly legitimate OEM manufacture as its chief engineer, it is doubtful that there will be but a fraction as many true OEM's in the US of any size in a few years.>>

1.2- Re: The Star PowerType---

Since our last column, the price on this item has dropped into the $189-219 bracket. For those that have, or are considering this product, a note. The PowerType implements a reverse line feed function in its Epson mode. Since the PowerType incorporated this feature before it was officially implemented by Epson, the codes do not match. This is not a major problem for B128 programs. Since the programs were written before anyone had implemented a reverse line feed feature, they do not include codes for RLF. However, if you also use this printer with another computer, who's software implements Epson's RLF codes, be careful. The best solution is to patch, if possible, the printer driver with the proper code. The PowerType can be had for $189 from Lyco Computers, 800 233 8760 or 717 494 1030. The tractor and sheet feeder are available from MidWest Micro-Peripherals ($49 and $169, respectively), 800 423 8215, in Ohio 800 321 7731.

1.3- Re: The Avatex 1200 and 1200HC---

The two basic differences between these two modems is the degree of Hayes code compatibility, and exclusion-inclusion of a "monitor speaker". The Avatex 1200 does NOT respond to all of the Hayes codes, and does NOT include a monitor speaker. On the other hand, the Avatex 1200HC will respond to ALL Hayes codes, and DOES include a monitor speaker. Total Hayes compatibility is not an important factor if you only use your modem with the B128, if the modem will also be used with an IBM compatible, then spend the few extra bucks for the 1200HC. Most IBM telecommunication software expects the modem to be totally Hayes code compatible, and issues these codes accordingly. This is not true with existing B128 telecommunications software.

1.4- Re: Printer DaisyWheels

The following information was mistakenly edited out of my last article. The best, most complete source for Printer DaisyWheels and Thimbles (also ribbons) I have found is Business Support Services, Inc., 705 Butternut Avenue, Royal Oak, MI 48073, (313) 585 4736. They specialize in these products, and provide the largest variety I've run across. <<And let's not forget member Jim White dba S&W Supply of Rapid City S.D. who has advertised in every CBUG issue to date with some of the best deals in CBM ribbons today -- and he gave those prices to CBUG members for the entire last 2 years!>>

<2> The other modems, plus

2.1- Re: Zipper 1200+

In the last column, I also mentioned the Zipper 1200 modem, from Priority One Electronics. There has been a change in this model, a change for the better. As I mentioned last time the Zipper 1200 was actually a Prometheus 1200G, a non-expandable unit. The unit currently being shipped, the Zipper 1200+, is a Prometheus 1200, a fully expandable unit which will accept Prometheus options, such as a 512K buffer, a parallel or series printer port, a 12 digit alpha-numeric LED display panel, and others. The price for the Zipper 1200+ is the same as the previous unit, $159. Yes, the Zipper 1200+ is Hayes compatible and incorporates a built in monitor speaker. Priority One Electronics can be reached at 800 423 5922.

2.2- Re: ADC 1200

Another reasonable priced 1200 baud modem that has appeared as of late, is the ADC 1200. This modem is totally Hayes compatible, and incorporates a built in speaker. There is only one national dealer carrying them DAK, at a very reasonable $129.95. The feedback on various SIG's, is that this is an excellent product. By the way ADC, once an American audio company, is now a subsidiary of the Japanese conglomerate BSR (once a British turntable manufacturer), as is dbx, another American audio manufacturer. And around and around we go. DAK can be reached at 800 325 0800.

2.3- Re: WorldPort 1200

One very, very interesting modem to come to market is the WorldPort 1200 from Touchbase Systems. This modem is primarily intended for use with portable computers. It weighs 6.5 ounces and is the size of a pack of cigarettes, easily fitting in a shirt pocket. Among the features offered by this unit are: it is totally Hayes compatible; it operates at both 300 and 1200 baud; it supports both American (Bell) and international (CCITT) telecommunications standards; operates on an internal 9 volt battery (8 to 10 hours) or an external AC adapter; and a very special, unique feature-- it can be fed by the Radio Shack acoustic coupler cups, and run at 1200 baud. This feature can be a blessing when you run across hotel telephones that are hard-wired into the wall, instead of terminating in a standard modular plug. Also, it is unique in that acoustic cups are usually unreliable beyond 600 baud. Of course, this unit is also perfect for those of you that have run out of desk space, considering that it is about one fourth the size of a regular desk modem. It can be purchased with either a male or female RS232 connector. One more note, this unit was originally released under the name of WorldLink 1200, which was recently changed to WorldPort 1200. The list on this unit is $199. The average discount price seems to be about $175. For more info or a local dealer call Touchbase Systems at 516 261 0423.

2.4- Modem (and printer) cables

One of the best, and usually cheapest, sources for modem and printer cables is Magitronic Technology Inc. Current, average prices run $6.00 for a RS232 male to male 6 foot cable, and $8.00 for a Centronics male to male 6 foot cable. Most other cable configurations are also available, as are 10 foot cables. By the way, if you are looking for RS232 or Centronics 2 to 1, or 4 to 1 switch boxes, Magitronics offer the best prices. $24 for the 2 to 1 boxes, and $33 for the 4 to 1 boxes. Magitronic can be reached at 800 227 5454 or 718 706 7670.

<3> A note on SuperScript II

One of our members, Don Trednick of Saint Marys, Georgia, has discovered a bug in SuperScript II. Most of you will never run into this bug, and to most of you, it will never be a problem. It seems that the "paging" feature runs into problems at page 255. First, in the preview mode, page 255 scrolls by, it does not stop/pause as it is suppose to. The bug also means that you cannot start a numbering sequence with a page number higher than 254. I assume this is a universal bug in Superscript II, since our member's copy had the bug, and so do my two copies of Superscript II. Obviously, this bug is only important to folk working on long documents, but it is a bug none the less. I would be appreciative if those of you that have SuperScript III would run a test to see if the bug carried over to SuperScript III, or if it was exterminated, please let me know the result. Many thanks to Don for bringing this bug to my/our attention.

<4> Keeping the power on

One of the issues that has attracted attention is the logic behind keeping your computer system on continuously. Well the first basic rule is, it is better to keep your system on instead of turning it on and off. As an example, if you are in your office for eight hours, but only use your system for limited chores, say a net of two hours out of the total eight, it is preferable to keep the system on for the total eight hours, instead of turning on and off as your task require it. In another example, you using your system in your home/business, your day spans 16-18 hours, with a net of 8 hours of actual computer use time. Again the basic rule holds, keep the system on for the 16-18 hours continuously. In a third example, you work late into the night, you decide to take a 6 hour sleep "break" and expect to start working soon after getting up. Should you turn the system off ? If you've gotten the drift by now, the answer is NO, leave the system on.

The reason that the continuous power on condition is preferable has to do with the nature of electronic components. The components that die in electronic equipment are sensitive to the electrical and thermal shock caused by powering the equipment on and off. This can be best exemplified by simple light bulb, if you leave it on continuously, it will provide many more hours of net service, than if you keep flicking it on and off. As the the electricity hits the wire in the bulb, the wire expands with the heat, when power is turned off, the wire contract as it cools. The repeated expansion and contraction causes more damage than does the continuous usage. In continuous usage, the wire stabilizes as the net heat stabilizes. In the worst case, turning the bulb on a couple of seconds after having turned it off, the thermal shock the wire receives when power is turned on just as the wire is in the process of cooling down, is the harshes and most destructive. Every component in your computer system goes through the same sequence as the lowly light bulb. This is the basic reason it is preferable to leave your system on, versus turning it on and off. Electrically this is also reflected in the fact that when an electrical device is initially turned on, it sucks up two to three times the net regular consumption. Considering that the average B system (B+monitor+8050) draws, at most, 100 watts, it means that for a few seconds, your system will consume 200 to 300 watts. This factor is also involved in the damage that can be done as you turn your equipment on and off.

A couple of factors to be aware of, when you leave the power on continuously:
<A> Be sure that all the venting ports are clear, sitting a manual on top of the B's, or the disk drives' venting holes will cause a significant, and possibly dangerous increase in the internal temperature. A quick three to four minutes of blocked ventalation can cause nasty results.
<B> The monitor, also requires a bit of caution. When you are going to stop working for any period of time over 10 minutes, turn down the Black Level (brightness). This prevents the CRT phosphor coating from being damaged.
<C> If your system includes a Hard Disk Drive and you keep your system on continuously for long stretches, remember to monitor the wear on the hard disk's bearings. A hard disk is always rotating once power is applied, so good preventative medicine dictates that this is the spot to watch.
<D> There are three different power supplies in the B's (that I know of). The original is on most machines with a serial number under 8000 (approximately). This power supply does not have a rear panel mounted fuse on units with a serial number under 2800 (approx.). This unit has provided an interesting fact for the three machine I've run, that used it. It seems that the power supply temperature continues to raise over the first three hours, however by the fourth hour the temperature has dropped, leveled, and stabilized. From this point on, it runs very cool. The power supply on machines with serial number above 8000, runs cool from power up to power down, on the two machines I've worked with. The third power supply is found on machines built in Germany, from the limited info I have, it is suppose to be a universal (110 and 220 volt) power supply and it runs cool. <<The cool running power supplies have rather substantial double wound transformers rather than the more common switcher type design. The transformer design is several pounds heavier, and is a far better choice if you can find them.>>
<<E>> Make sure that your power lines are filtered and protected, see the next section for a full discussion on this topic.

<5> Protecting your equipment

In the previous section we touched on the electrical and thermal shocks your equipment suffers on power up. As I said, on power up any electrical device draws two to three times it normal power consumption, this is known as a power surge. Power surges caused by other devices can cause electrical "spikes" to be transmitted through out your electrical system and into your computer system. <<This is the reason that you are always advised to power up your computer LAST, when powering up each item in you system individually.>> The worst offenders are electrical devices that include a motor, most common in a home setting: refrigerators, washing machines, vacuum cleaners, power tools, fans, etc. These are best exemplified by the static lines that run through your television screen when you turn on a vacuum cleaner. Some older refrigerators also cause this problem, as they go through each of their on and off cycles.

Another enemy to computer devices is RFI (radio frequency interference), again almost every electrical device generates radio waves, if you have a television anywhere near your computer system, you will probably notice diagonal, wavy lines (called a herringbone pattern), these are usually most notable on channels 3,4,9, and 11. The original three to four production runs of the Commodore 64 computers were know as "monster RFI generators". They caused so much interference that television sets in other rooms could be effected. Some of this RFI is transmitted over the electrical wiring, and some of it is transmitted over through the air waves. There's not much that can be done about the stuff that floats through the air, unless your going to enclose devices in grounded metal boxes.

However, there are a variety of problems that can be caused by electrical surges/spikes and by RFI. These problems can manifest themselves in a variety of ways. The worst possible situation is when a power surge/spike causes a reset, in other words you end up with a "ready" prompt on your screen. The simpler side effect, is that some or all of your data in memory may be scrambled. Sometimes the data on screen will also be scrambled, but most of the times it won't. Another invisible way in which power line garbage can get you, is during a save to disk. If a surge/spike strikes as you are saving a file, the data actually saved to disk maybe scrambled with out you realizing it. This also happens during a file load, but most of us would assume that the scrambled data was cause by one of components in the system, without realizing the real cause. Beside, simply clearing the system and re-loading the file will cure the problem, therefore most folk rarely give it a second thought, or even remember it at a later point.

Now that we have discussed the problems and their causes, let discuss the problem solvers. There are a multiplicity of devices on the market that will filter some, if not all of these power line problems. The simplest will provide minimal spikes/surges suppression, these basic devices run ten to twenty bucks. The next level of devices will provide significant surge/spike suppression and basic (minimal) RFI suppression, these devices run in the forty to seventy dollar bracket. The next level up, eighty bucks and up, provide heavy to heaviest spike/surge suppression, and moderate to super RFI suppression. These devices vary as to the other features included, some provide a master power switch, a line circuit breaker, individual power switches for four to six devices (besides a master power switch), indicator lights for the various switches, and some of the newer devices also include surge/spike and RFI suppression for your telephone line. More on this later.

These devices are usually constructed in three different ways: <1> a master filter that provides common protection to however many "filtered" sockets may be provided; <2> individually isolated filters for each socket (or bank of two); and <3> cascaded filters, each filter provides a degree of filtering between sockets, and higher isolation from the power line as you move up each cascaded filter; in other words if you have four filters, and each filter provides 30db of isolation, you would have 30db of isolation between each socket, but the isolation from line would vary at each socket. The first socket would provide 30db, the second 60db, the third 90db, and the fourth would provide 120db isolation from the line. Which device is preferable, depends on you susceptibility to power line garbage, and on your particular set up.

<<In non/industrial settings, the most common (and usually fatal) cause of data loss is running a vaccuum cleaner on the same line as a computer. Rule of thumb: if your appliance interferes with a cheap TV or AM radio, it will likely damage data. Make an effort to operate noisey appliances such as vaccuum, hair dryers, food mixers, etc on a separate circuit preferable fed in separate conduit or BX.>>

We quickly touched on telephone line protection, let explore this further. Modem are sensitive to all the garbage that can be transmitted via the power lines, but it is also sensitive to power surges/spikes that may come through the telephone lines. The most common, and highly destructive, source is lightning. No, it doesn't take a direct strike. Any lightning strike can induce enough electrical energy to cause a surge/spike of up to 3000 volts in magnitude. When this happens, it makes no difference if you equipment is on or off, it will fry your modem, and with a little dose of bad luck your computer. Remember they're interconnected via the RS232 cable. This phenomenon (called EMI, Electro-Magnetic Induction) has brought telephone line suppressors to market, these filters start at thirteen bucks. In some cases, they may be part of of a larger power line unit as alluded to above. NOTE: Most of the currently availabe units offer only surge/spike protection for EMI, the more expensive units also provide RFI filtering for telephone line.

Are these devices worth using, in my opinion YES, a must for the serious computerist. I live in an urban setting, an apartment building in New York City, and even though I have been advised that my power lines are isolated from the balance of the building, and that thee fused sub-lines in the apartment are isolated from each other, I have found this NOT to be true. I have suffered from power line problems that have not only affected my computer systems, but also my audio, video, and shortwave equipment. And last summer, during a particularly violent storm, I lost a modem to a lightning induced *telephone line* surge/spike. If you live in a rural area where power and telephone lines are above ground, then you are most susceptible to lightning caused damage. Currently all of my sensitive electronic equipment is fed through one of these filtering devices, and in the case of my computer systems, they are double filtered.

In my particular installation, the most harmful item is a large electrical fan. During the summer months I have to remember to turn on the fan BEFORE I turn on the computer system, and on the other end, I have to remember to turn the fan off AFTER I turn off the computer system. Otherwise the power line spike from the fan will totally corrupt the computer memory, which means that the data is corrupted, or worst --- the loaded program will be corrupted, which will lead to a system lock-up (or freeze-up). It should be noted that all my computer systems are on a filtered - protected sub-line (fused line), and each system has its own power line feed filtered and protected. Yet, the spike from this simple house fan travels through the electrical system and zaps my computer systems. Never underestimate the garbage that can come in through the electrical line.

The power line devices I currently use are manufactured by a variety of companies, and there are many more on the market. The devices I use are by Electronic Specialist, Inc. (800 225 4876 or 617 655 1532), Kensington Microware (800 535 4242 or 212 475 5200), TrippLite (312 329 1777), and Radio Shack (anywhere and everywhere). For telephone line protection, Radio Shack (anywhere), Priority One Electronics (see above), and Altex Electronics, Inc. (800 531 5369 or 512 828 0503).

P.S. I have only addressed filters, for really miserable conditions, there are devices known as power line stabilizers, conditioners, and UPS (uninterrupted power supplies). More about these in a future column.

<6> Computers and the Blind

In the last issue we touched on how the handicapped can use, and be assisted by computers. In the interim a new publication has come to market, specifically for the visually impaired. "ADD-ONS: The Ultimate Guide to Peripherals for the Blind Computer User" by the National Braille Press Inc (617 266 6160) is available in print ($19.95, +s/h), in braille ($16.95, free s/h), and on audio tape ($16.95, also free s/h).

<7> Closing comments

It is very interesting that even though CBUG's membership currently numbers over 2000 members, the active core numbers no more than 25 or so members. By core, I mean those members that regularly contribute to the groups activities, this situation cannot continue. If you (YES - I MEAN YOU) do not get involved and contribute to the effort, those of us that are active will burn out from having to carry the full load. We need everyone to contribute to this effort, otherwise we will disappear, and there will be no source of support for our systems. I don't think anyone wants this to happened. You don't have to contribute a mosterous document, or need to have the writing skills of Hemingway, BUT each little piece helps. Hey, I'm the King of poor spelling, lousy grammar, and the run on sentence. But I just try to communicate "the facts, just the facts, ma'am" (borrowing from Joe Friday of Dragnet). So please, take a little time and write for the Escape, do your mad scientist routine and develope some hardware or software for the orphan, or bring your experiences to our "***NEW and IMPROVED*** section on the Delphi Network. All contributions are welcomed.

Well folks, this brings this column to a close, it's shorter than my average, but with the added responsibility of Sysop duty on Delphi, there is but so much time. As always you can contact me via:

<1> Delphi ----- Username: AMATOS (BEST method)
<2> Compuserve-- ID# 75146,2224
<3> Ma Bell----- 212 231 6028 (Evenings 8 to 11pm;
11am to 6pm on weekends, eastern time
zone.
<4> US Mail----- Mr. Angel M. Matos
3176 Decatur Ave #6E
Bronx, New York 10467
Attn: CBUG

'Till next time, Keep On Truckin'. /30/

OVERSEAS DELPHI SERVICE AVAILABILTY

Delphi offers direct service via common carriers from 67 countries. Dozens of other countries without direct service can access Delphi thru neighboring facilties. The prices range from reasonable to outrageous. We can only afford to list the countries in The CBUG ESCAPE as the detail data on some countries runs to several pages. If you are interested in participating in the CBUG areas on Delphi, send us a post card or letter so indicating, and we will download the appropriate file(s) for you and mail them back via Air Mail.

67 Countries Available
1 ALASKA
2 ANTIGUA
3 ARGENTINA
4 AUSTRALIA
5 AUSTRIA
6 BAHAMAS
7 BAHRAIN
8 BARBADOS
9 BERMUDA
10 BRAZIL
11 CANADA
12 CAYMAN ISLANDS
13 CHILE
14 CHINA
15 COSTA RICA
16 DENMARK
17 DOMINICAN REPUBLIC
18 EGYPT
19 ENGLAND
20 FINLAND
21 FRANCE
22 FRENCH OVERSEAS TERRITORIES
23 GABON
24 GAMBIA
25 GERMANY
26 GREECE
27 GUAM
28 GUATEMALA
29 HONDURAS
30 HONG KONG
31 HUNGARY
32 ICELAND
33 INDONESIA
34 IRAQ
35 IRELAND
36 ISRAEL
37 ITALY
38 IVORY COAST
39 JAMAICA
40 JAPAN
41 KOREA
42 KUWAIT
43 LIECHTENSTEIN
44 LUXEMBOURG
45 MALAYSIA
46 MEXICO
47 NETHERLANDS
48 NEW ZEALAND
49 NORWAY
50 PANAMA
51 PERU
52 PHILIPPINES
53 PORTUGAL
54 PUERTO RICO
55 QATAR
56 SAUDI ARABIA
57 SINGAPORE
58 SOUTH AFRICA
59 SOUTH KOREA
60 SPAIN
61 SWEDEN
62 SWITZERLAND
63 TAIWAN
64 THAILAND
65 UAE
66 UNITED KINGDOM
67 UNITED STATES
-10h GMT) Last Update: January 31, 1987 (See U.S.)

There are numerous countries whose access is via their neighboring countries, most notably in the middle east.

BASIC 4.0+ Tutorial (Abridged)

by Warren D. Swan

<<Before I give Warren the forum for the next dozen pages of so, time has come to give plaudits galore to his efforts for CBUG members locally and nationally. He has tirelessly prepared courses and taught them for as long as 6 months at a time. Often traveling nearly 2 hours each way to teach B128 programming to local members at local meetings. He's set the courses down in writing (of which the following is abstracted) -- the whole series is unabridged on library disk. His work is always impecable, precisely and neatly laid out, error free. And all this with no more than token thanks. Give a word of thanks, some praise in your prayers for this man who always gives far more than he receives, and insists on perfection in everything he does for CBUG members.>>

<<Note: Due to the inability to reproduce the up arrow symbol ( ), it is shown as (¶)

2 BASIC DATA MANIPULATION

2.1 Variables; Assignment (LET) [Handling Data 1]

In high school algebra you learned to use variables to REPRESENT values. For example: "If Susan's age is twice Leonard's, and the sum of their ages is 18, how old are both Susan and Leonard?" We learned to solve this using a variable for Leonard's age; we'll use AGE. Then Susan's age, being twice Leonard's is (2 x AGE). The sum of their ages is 18, so that:

(2 x AGE) + AGE = 18

which we reduced to (3 x AGE) = 18, or AGE = 6 (Leonard's age) and (2 x AGE) = 12 (Susan's age).

When programming in BASIC, we will use variables to more than REPRESENT values; we can think of variables as STORING values. There is very little difference in purely mathematical terms, but computer scientists often forget to separate data from its method of manipulation in computers. Thus, variables STORE values, rather than just REPRESENT them.

Commodore BASIC allows us to use variable names that are several characters long. That is, in the example above we used AGE rather than A, as you might have originally done when you learned algebra. Variable names may contain letters or digits, but the first character must be a letter. So, A1 is a valid variable name, but not 1A.

There are just 3 things to remember when choosing a name for a variable:

(1) A variables name cannot contain any BASIC keyword. Here the term "keyword" means any of the BASIC instructions (remember print, run, list, end, stop, cont, goto, & rem?), plus any of the BASIC functions which we will be learning later (sin(x), pos(x), etc.). That is, "atoz" is not a valid BASIC variable name because it contains the word to in it. If we tried to use "atoz" as a variable, we would invariably get a syntax error.

(2) Regardless of the length of the variable name, BASIC only "cares about" the first 2 characters of the name. Thus, the variable "age" is the same as the variable "agree" as far as BASIC is concerned. If you change the value stored in "age", you are also changing "agree". The second character of a BASIC variable name may be a digit (0 to 9), so that "a0" is just as valid a name as "age". Of course, variables names can just be a single letter too.

(3) Some variables have predefined meanings and cannot be used as general purpose variables. These include: st, ds, ds$, and ti$ (the dollar signs will be explained in the next section).

A by product of rule 1 is that some harmless variables' names may cause syntax errors if you don't use proper spacing in your statements. For example, we want to check to see if the value in variable t when anded (AND is explained later) with the value 32 is true, we might write a statement such as:

```
ch = tand32
```

This harmless looking statement will cause a syntax error because the BASIC interpreter will "see" the word "tan" and think that you were referring to the tan(gent) function, rather than to "t and 32." Problems like this disappear with good programming practices and experience.

Of course, a variable is only good if it can be assigned a value; that is, if it can store a value. To do this we merely use a statement like:

```
age = 10
```

The variable that we wish to give a value is always placed on the left side of the equal sign, while the value, which may be an expression, is placed on the right side of the equal sign. That is:

```
40 age + 1 = age
```

is definitely WRONG. The "thing" on the left of the equal sign is not a mere variable name. We probably meant:

```
40 age = age + 1
```

Yikes! This brings up still another point. Any school child could tell you that something cannot equal itself plus one. This is a mathematical impossibility! The above statement really means: take the value stored in age and add one to it, then store the result back into age. Is that clearer? It will be with experience.

You will notice that the ASSIGNMENT STATEMENTS that we have been presenting here don't seem to fit into the pattern of other statements, which usually require an INSTRUCTION (such as **print** or list) followed by OPERANDS. The reason for this is that we have actually been using an abbreviation for the real assignment instruction, namely **let**. We could have actually written:

```
40 let age = age + 1
```

Actually, the use of the **let** here might actually clear up the ambiguity we talked about just 3 paragraphs ago. Somewhere along the way the designers of the BASIC language decided that the word **let** should be implied if a statement did not start with any other instruction word. From then on, we BASIC programmers have decidedly ignored the **let** instruction word, and merrily went on our way constructing assignment statements without it. The choice is up to you. The real speed minded programmer may take note that BASIC operates a smidgeon faster when you forget to use the **let**. So let's just all forget that it was ever mentioned, except that you'll know what it means if you ever see it in someone's program (surprise!).

2.2 Types: Real, Integer, String

[Handling Data 2]

Up until now we have been using variables to store numbers. Technically, we have been using them to store REAL numbers, as opposed to imaginary, complex, or any other such hairy things. Variables can store irrational numbers such as the square root of 2, or pi, or even numbers containing scientific notation. You remember things like 6.28 times 10 to the 18? Well, in BASIC, we would represent this as: 6.28e18. Notice that no space should be put around the letter "e". We can also represent very small numbers, such as 1 times 10 to the -25:

```
40 pl = 1e-25
```

This number is positive, even though there is a minus sign in it. The minus sign is on the exponent of 10, not the characteristic part of the number (the 1). Can you see how this differs from -1e-12 or -1e12? Think about it.

But BASIC doesn't stop at real numbers. BASIC also allows us to store integer-only numbers; that is, whole numbers without a decimal point and fractional part. To do so, we simply plop a percent sign after the variable name, like this:

```
40 age% = age% + 1
```

The tutorial disk includes a discussion of the relative merits of using integer variables at this point.

But BASIC still doesn't stop at integers. What if we want to store something other than numbers, such as someone's name? To do this, we simply use a variable name with a dollar sign plunked onto the end, such as **name$**, which is the same as na$, right? To assign such a variable a value we would type:

```
100 name$ = "Warren Swan"
```

This is called a STRING data item. Notice the quotes around the name. These are required, just as they were when we wanted to print a message with the print instruction. This is how BASIC "knows" that we want to use the actual characters W, a, r, etc. rather than some expression with two variables Warren and Swan. Also notice that the characters in the quoted data item here can be any upper or lower case letters or any graphic or special characters on the keyboard, with very few exceptions we will find out later. A string variable can store a string of up to 255 characters. The actual "size" of a string variable shrinks or expands when it is given a new value bigger or smaller than it had.

One little quirk about Commodore BASIC is that the trailing quote is not necessary here, unless the string is supposed to contain trailing spaces, or unless another statement is to follow this one on the same line (discussed later). So we could have written:

```
100 name$ = "Warren Swan
```

and left off the trailing quote.

What can we do with these new data types: integer and string? Well, we can do some of the same things we have been doing with reals, such as printing them:

```
1000 print "His name is "; name$;".
```

which will print: His name is Warren Swan.

Notice that the last string in this print statement also did not contain a trailing quote. Also notice that the quotes are not stored in the variable name$, nor printed from the print statement. Later we will see how we can tell BASIC to store quotes in strings if we want them.

One remaining point should be brought up here. What values does the computer give to variables when you run the program? When you type RUN, the computer clears out the area where variables will be stored. Variables are "created" the first time they are used or given a value. If the first thing we do with a variable is use it without having given it a value, BASIC will assume that the value in the numeric variable should be zero (0). With strings, it assumes that the string variable should contain nothing. Strings can be reset to this initial condition (cleared) by a statement such as:

```
10 message$ = ""
```

We could have left off the second quote in this case, remember?

2.3 Expressions: Operators & Hierarchy

[Handling Data 3]

Earlier we saw some simple expressions and how to print the result of evaluating an expression, and how to assign the result to a variable. Now we will learn some simple rules on how to write expressions so that the computer will know what you want. First, we must realize that you cannot assign a string value to a numeric (real or integer) variable, and vice versa. Later we will see some functions that allow us to convert between numeric data and string data, but such conversion is not automatic.

However, we can assign an integer value to a real variable, and the conversion will be done automatically:

```
1020 a% = -32
1040 age = a%
```

Likewise from real to integer:

```
1060 real = 3.5
1070 in% = real
```

In this last situation, the variable in% will be given the value 3, since it cannot store the fractional part (.5). The conversion from real to integer requires finding the GREATEST INTEGER NOT GREATER THAN the real value, and using that as the result. So, if we were to have used:

```
   1060 real = -3.5
   1070 in% = real
```

the result would be to assign -4 (not -3) to in%. Negative 4 is the greatest integer not greater than -3.5. The tutorial disk includes a short discussion pointing out the difference between BASIC and FORTRAN at this point.

The only remaining words of caution concerning conversion between types is that integer variables can only store values from -32768 to +32767. Thus, trying to assign in% = 1e10 would result in an illegal quantity error.

Expressions are generally evaluated left to right. However there are a set of rules that determine when parts of an expression (subexpressions) take precedence. These are called rules of hierarcny. Some of the operators listed here only apply to numeric (non-string) values, and will be identified as such.

1. Parenthesized subexpressions take precedence. These include calls to functions. Thus, 3*(4+1) will be 3*5 or 15. Also, 3*abs(1-4) will be 3*3 or 9. Likewise, "abc"+right$("defghi",3) will be "abc"+"ghi" or "abcghi".

2. Unary negation is done next. Thus -3+4 is negative 3 plus 4, or 1; NOT minus 3+4 or -7. Of course this only applies to numeric data.

3. Exponentiation takes the next precedence; exponentiation is done using the up-arrow operator (¶). For example, 3*2¶3 is 3*8 (2¶3 = 8), or 24. The multiplication is held off until after the exponentiation is done. If we had wanted 3*2 to the 3rd power, we would write: (3*2)¶3 to force the 3*2 first. This, too, only applies to numeric data.

4. Multiplications and divisions are done next with equal precedence. So, 3*2/1.5 is 6/1.5 or 4. Also, 3/2*1.5 is 1.5*1.5 or 2.25. Note that 2*3+4*5 is 6+20 or 26. This only applies to numeric data.

5. Additions and subtractions are performed next and are done with equal rank. So, 1+2-3 is 3-3 or 0. Also, 1-2+3 is -1+3 or 2. String data only allows the + operator, which is called "concatenation" not "addition". The use of the + operator on strings simply puts the two strings together. Thus,
   "com"+"et" becomes "comet", and (if name$="Fred")
   "Name: "+name$+" Flintstone" becomes:
   "Name: Fred Flintstone".

6. Comparisons are done next. These usually occur in if/then statements but are not limited to them. Comparisons include =, <, >, <=, >=, and <> which are equal, less than, greater than, less than or equal to, greater than or equal to, and not equal to, in order. The result of a comparison is either a true value (-1 in Commodore BASIC), or a false value (0). For example:

   ```
   50 test = 2+3 < 4+5
   ```

   would assign a value of -1 (true) to test. The reason for this will become apparent when we learn if/then.

   Comparisons may also be done on string data. A comparison of "less than", as in "abc" < "def" roughly means "Does the first string come before the second in dictionary ordering?" Two strings must be identical to cause an = test to result in true, or a <> test to result in false. If one string is shorter than another the test is made to the first character where they differ and the one with the character earlier in the ASCII code sequence is considered "less than" the other string. If both strings are the same except one has more characters, the shorter string is considered "less than" the longer string (ie. "abd" < "abcd").

7. The not operator is performed next. This operator "turns around" a truth value from false to true or from true to false. Thus,

   ```
   60 test = not 2+3 < 4+5
   ```

   would assign a value of 0 (false) to test. More about this later.

8. The and operator is performed next. This operator takes two truth values and returns true (-1) only if BOTH operands are true. If either operand is false, the result of an and is false (0). More about this later.

9. The or operator is performed next. This operator also takes two truth values and returns false (0) only if BOTH operands are false. If either operand is true, the result of the or is true (-1). More about this later.

As an example of the last two operators, we will simply look at an expression and simplify it. Remember that we are supplying the spacing for clarity. Spacing can be misleading to your eyes.

```
result = not 2+3 < 4+5 or "0">"/" and not 6<5
result = not  5  <  9  or "0">"/" and not 6<5
result = not    -1     or   -1    and not  0
result =      0        or   -1    and    -1
result =      0        or           -1
result =               -1
```

The tutorial disk describes the and, or and not operators as binary bitwise operators in this section, and in section 6.3. These discussions will not be included in these articles for brevity.

2.4 Cursor Controls In Strings

[Handling Data 4]

We have learned how to print out strings and how to do some basic operations with them. A later section on functions points out even more functions that can manipulate strings. One important aspect of Commodore BASIC is that many special characters may actually be used in strings. To do this, Commodore made their computers so that there is something called a "quote mode". This mode is entered whenever an odd number of quotes (") have been echoed back from being typed. When an even number of quotes have been typed, or the RETURN key has been pressed, the quotes mode is left and the computer behaves as usual.

While in this quotes mode, the computer no longer performs the cursor movements when you press any of the cursor control keys. Instead, a special character is entered into the quoted string which represents the character just typed. For example: we can enter:

```
  10 cd$ = "<cursor down>
  20 print cd$;"Message
```

(Don't type the 13 characters <cursor down>, but rather press the cursor down key with the down arrow.) As you could see, a funny reversed character appeared after the quote on line 10 when you pressed the cursor down key. The cursor down key was actually "programmed in" to the string, rather than the cursor moving down off of line 10 immediately. When the program is run, the print statement will start by printing a cursor down character, and then the Message. This is a good way of skipping lines. For example,

```
10 print"<cursor down>Message
```

is actually easier to read than

```
10 print
20 print"Message
```

since it is more compact and less verbose.

Any of the 4 cursor control keys can be "programmed in" to strings during quotes mode, as can HOME, CLR, RVS, OFF, NORM, GRAPH, TAB, SHIFT TAB, and many control characters (the ones that aren't the graphic characters). Of course, the letters and graphic characters and digits and operators, etc. can also be placed in a string. The few exceptions are ESC, RETURN (ENTER), INS, DEL, RUN and STOP or any of the function keys. A function introduced later will show how we can place some of these characters, plus even the quotes itself, into a string variable.

Many programs start out by clearing the screen and printing an identification message. With some (inferior) computers, this requires a separate instruction such as CLS or HOME, which are not needed in Commodore BASIC. Instead, your statement would look like:

```
100 print "<CLR>MYGAME
```

Again, the term <CLR> is used to represent pressing the SHIFT key and the CLR/HOME key, which we cannot represent directly in the word processor we used to type this tutorial. You get the idea.

2.5 INPUT, GET

[Basic Input 1]

So far we have been talking about manipulating data by using it in expressions, assigning it to variables, and printing it. But how does a program get the data in the first place?

Later we will be learning how to get data from disk files. But for now we'll learn how to get some data from the keyboard. As an example, we want to read in 3 numbers and print their average. We'll use the variables a, b, and c to hold the 3 numbers:

```
100 input a,b,c
120 print "Average of ";a;b;c;"is ";(a+b+c)/3
```

We then run this program. When the input statement is executed in line 10, the computer will print a question mark and a space, then turn on the cursor. It is waiting for you to type in the values for a, b, and c. To do so, we might enter (remember "? " came from the computer):

```
? 10,20,60
```

We separate the numbers by commas. The computer will then respond:

```
Average of 10  20  60 is  30
```

The tutorial disk discusses what would happen if we were to input something not expected by BASIC, such as too much or too little data.

The person who hasn't used our program before would definitely wonder why the computer spat out the question mark though. So we should add some prompt such as:

```
50 print "Please enter the 3 numbers to be averaged.
```

If we could change this prompt into a question and let the computer supply the question mark:

```
50 print "What are the 3 numbers to be averaged";
```

Notice that we didn't have it print any punctuation at the end of this message. The computer will supply the question mark immediately following the word "averaged":

```
What are the 3 numbers to be averaged? 10, 20, 50
```

With this example we can see that the numbers can be spaced out for readability, as long as we put the commas between them.

Commodore BASIC allows us to combine lines 50 and 100, the prompt print and the input statement:

```
100 input"What are the 3 numbers to be averaged"; a, b, c
```

Again we did not have to put the question mark in our prompt string because BASIC will automatically supply it and the space after it. The message must be separated from the first variable in the list by a semicolon. When the prompt message is put in the input statement, the question mark will always follow the prompt message immediately, rather than being on a separate line.

Can we use input to read integer and string values too? Yes, integer variables will behave just like the real ones did in these examples. String values have to be input with some extra rules in mind. Suppose we want to input a name and age. We might have:

```
100 input "Your name and age"; name$,age
```

When the program is run, we could respond like this:

```
Your name and age? Snidely Whiplash, 50
```

The result would be name$="Snidely Whiplash" and age=50. The extra rule to keep in mind is that quotes must be used around string values in certain instances:

(1) If we want the string value to contain leading or trailing spaces:

```
Your name and age? " Snidely Whiplash ",50
```

In this case, name$=" Snidely Whiplash ". Without the quotes, BASIC would think the leading and trailing spaces where only for readability. Imbedded spaces cause no trouble.

(2) If the string is to contain commas or colons:

```
Your name and age? "Johann Sebastian, Jr.", 15
```

This is the only way that BASIC can tell that the comma is part of the name, rather than the separator between the name and the age. Unfortunately, Commodore BASIC will ignore colons and everything after them when typed in for an input, unless the string had a quote at the beginning:

```
Your name and age? Funny name:,30
??
```

BASIC set name$="Funny name" and ignored the colon, comma, and 30. Then it noticed that it hadn't received the age, thus the double question mark prompt.

```
Your name and age? Tommy, 12: i'll be 13 in 1 month
though.
extra ignored
```

The tutorial disk includes a point of trivia concerning putting a quote in the middle of a string response.

As with strings in programs, string values given to an input statement do not have to have the ending quote if the string is the last thing input:

```
100 input "Your name is"; name$
run
Your name is? "George Jacob Hinkleheimer Schmidt
```

The quote before George was not really necessary here. It's just there for illustration.

The input statement differs from all other statements in one way. If you want to stop the program when the computer is waiting for input (the cursor is blinking), you must PRESS AND HOLD THE STOP KEY AND PRESS THE RETURN KEY WHILE PRESSING THE STOP KEY. BASIC does not "notice" that you are pressing the stop key until you complete the input statement by pressing RETURN.

Often we don't need to input a whole word to tell the program something, we only need to get one character. The get instruction can does this. The funny thing about the get instruction is that it doesn't wait for you to type something. If you haven't already typed the character when the get instruction occurs, it will simply set the variable in the get statement to nothing (cleared):

```
 10 get c$
 20 print c$;
 30 go to 10
```

This small program loops around getting one character at a time. If no character is typed, the print statement will print nothing (just waste time), and will leave the cursor right where it was (because of the semicolon, remember?). When you type something, the print statement will "echo" it back to the screen. You must press STOP to get out of this program.

The get statement differs from input in two other ways. (1) It doesn't automatically "echo" back what you typed. Your program may want to echo the character, or it may choose not to - a lot of games choose the latter so that the screen doesn't get messed up while you are typing characters to be used by a get. (2) get does not turn on the cursor like input does. You must just know when to type a character when a program uses get. The idea is that get allows your program to "suck in" a character that the user typed without it having to show it on the screen.

get may also be used with numeric variables, although this is seldom done in reality. The reason is that it only allows you to type 1 digit, and if you press anything else, like DEL, the program will stop with a ?syntax error; even though there was nothing wrong with the program! Also, get allows you to use a list of variables just as input does:

```
 200 get a$, b$, c$
```

But this is also seldom used. Remember that you must have already typed 3 characters before line 200 is executed. If you typed only 1, a$ would get that character and b$ and c$ would be cleared.

Commodore BASIC allows what is called "type ahead". You can type up to 9 characters before BASIC actually gets around to reading them. When it does execute the next get or input, it will just act like the characters you had typed where just now typed. In some cases this type ahead is undesirable because the characters may have been the rapid input to some game that finished before all the characters were read with a get (say). To "gobble up" these characters so that they aren't used in the next input or get, we could use:

```
 100 get c$,c$,c$,c$,c$,c$,c$,c$,c$
```

This would gobble up to 9 characters. The variable c$ was used repeatedly because we don't really care what get got. With each character that had been typed, c$ would be set to just that character, only to be reset to another character or nothing in just an instant. Later we'll see another way of doing the same thing.

Chapter 2 will continue next issue.

Any questions arising from this tutorial should be sent directly to the author, whose address is given below. Also, you may obtain disks containing the entire tutorial directly from the author. It comes in either a dot-matrix (4023, 4022, 2022, etc.) version, or a letter quality version (6400, 8023, etc.). Each version (1 disk) costs fifteen dollars, or you can obtain both versions for twenty dollars (both disks). Handling is included. Write to:

Warren D. Swan
1 N 114 Woods Avenue
Wheaton, IL 60188

THE CASE OF THE CONFUSED 4023 CHARACTERS

By: Warren D. Swan

This article will present a simplified approach to printing the characters normally thought to be unprintable using Superscript II. These are the left bracket, the backslash, the right bracket, the up arrow, and the left arrow. The Commodore dot matrix printers have always had a bug in their character generation ROMs that cause these characters to be switched with some graphic characters when they are printed in the upper/lower case mode. The characters were never lost, they were just confused.

A later section of this article explains the exact problem. Another section describes a utility for printing non-Superscript II files to a Commodore dot matrix printer so that the above characters get printed correctly. Before that I will describe a trick that allows us to get to these characters using Superscript II without a lot of work, or changing tables, or patching in machine language routines.

Please note that if you are writing a document that will normally be printed by both letter quality and dot matrix printers, you should probably avoid using any of the above characters but the left and right bracket. Why? Because the letter quality printers usually print characters that might not have been expected or desired.

The reason being that the character codes used by the Commodore computers and dot matrix printers are based on (a mutilated version of) the 1964 American Standard Code for Information Interchange (ASCII) character set, whereas all of the letter quality printers use the 1971 ASCII character set. The later character set replaced the up arrow with a "hat" (also called "circumflex" or "carat"), and the left arrow with an underscore. Also, each letter quality printer (or print wheel or thimble) maker seems to have a different idea as to what to replace the backslash with.

For that reason, in this article I have chosen to write out the words (like up arrow) instead of using the actual symbol, since I don't know what would be printed when my article gets published! Also, since the published article will not be able to use reversed characters, such as the reversed * that Superscript II displays when RVS is pressed, I will underline all characters that are to look reversed on your screen when using Superscript.

THE PROGRAMMING TRICK:

We will use a little known and lesser understood feature of Superscript known as the "format=" command. Yes, it does work on all versions of Superscript, but none of us knew how to use it for the longest time. The purpose of this command is to send special characters or codes to the printer that normally would not be accessible using Superscript. For example, if we want to print the character whose CBM ASCII code is 91, we would pick a format (represented by a digit from 0 to 9) to store the code. This is done with:

```
 *1=91<
```

Other commands can precede or follow this command by using a colon to separate them, as usual. Please note, that as explained above, the underline represents which characters will be in reverse when displayed in Superscript.

The above would be done somewhere near the top of the document. At the point where the character (code 91 in our example) is desired in the document, we would press ESCape 1, which will show as a reversed 1 (1) in Superscript. When printed, Superscript will output the 91 code when it encounters the 1. It will assume that the code caused the printer to print a character, that is, to use a column on the output.

Superscript uses the last encountered definition for a format (if there is more than one) when it finds the reversed digit in the text.

Armed with this knowledge, whenever you want to make a document intended for either Commodore dot matrix printers or other printers, put these lines at the top of your file:

```
*cm 1=lb, 2=bs, 3=rb, 4=ua, 5=la<
*1=219:2=220:3=221:4=222:5=223:cm CBM dot matrix<
*1=91:2=92:3=93:4=94:5=95:cm All other printers.<
```

The first line is just a reminder comment line. Replace the lb with the left bracket symbol, the bs with the British pound symbol (representing back slash), the rb with the right bracket, the ua with the up arrow, and the la with the left arrow.

Now, whenever you need a left bracket in your text, don't put a left bracket, put in an ESCape 1 (1)instead. Similarly, use an ESCape 3 (3) in place of a right bracket, and so on.

Now, when you print to a Commodore printer with the infamous bug, insert a cm just before the first 1 in the third line above, as in:

```
*cm1=91:2=92:3=93:4=94:5=95:cm All other printers.<
```

which will cause Superscript to ignore the entire line. So, Superscript will use the first set of definitions when it sees the 1, 2, etc. And, when you print to any other printer without the bug, just make sure you don't have the cm at the beginning of the third line. That will cause it to redefine each of the format codes and use the later set when it encounters a 1, 2, etc.

Please note that in the *1=... command the digits should not be in reverse. But when the codes are put in the text, they must be (eg. 1). Make sure you get this straight right away to save yourself some confusion dowr the road.

The format code command may be used to access other printer characters, such as the degree symbol, or whatever the print wheel has.

THE REAL PROBLEM:

You can ignore this section if you don't care why the above contortions are necessary.

The Commodore dot matrix printers work fine in the capitals/graphics mode, which is the mode that they start up in. There are two ways to enter the lower/upper case mode: (1) Print a chr$(17) (cursor down), which causes it to temporarily go into the lower/upper case mode, until either the end of the line is printed, or until a chr$(145) (cursor up) is printed. (2) Issue the following commands:

```
   open 7,4,7: print#7: close 7
```

which will cause it to stay in lower/upper case mode.

Once in the lower/upper case mode, there are four ways to go back into the capitals/graphics mode: (1) Print a chr$(17) (undone by chr$(145) or end of the line). (2) Issue the commands:

```
   open 8,4,8: print#8: close 8
```

(which is not available on the older Commodore dot matrix printers, such as the 2022 or 2023). (3) Reset the power to the printer. (4) Issue a command to the printer to reset using:

```
   open 10,4,10: print#10: close 10
```

Superscript assumes that you haven't issued commands to the printer to change it to lower/upper case mode. So it sends a cursor down prior to each line it prints, if printing to a Commodore dot matrix printer. (This is what causes the q at the beginning of each line of a Superscript output file read back in. It was actually a cursor down transformed to a q.)

Regardless of how it gets into lower/upper case mode, the printer confuses certain CBM ASCII codes as long as it is in this mode. In particular, it prints the character represented by a chr$(219) when given a chr$(91) (the left bracket) and a chr$(91) when given a chr$(219). It prints a chr$(220) when given a chr$(92) and a chr$(92) when given a chr$(220). And so on to a chr$(223) when given a chr$(95) and vice versa. That is, these 5 characters were inadvertantly switched in the lower/upper case mode section of the character generator ROM.

There are 2 ways to unconfuse these characters: (1) go back to capitals/graphics mode temporarily just when printing the confused characters, or (2) deliberately use the "wrong" character for each confused character and let the printer confuse it back to the right character. The programming trick we used above did the latter, but only when printing to a Commodore dot matrix printer. So does the utility program described below.

A USEFUL UTILITY:

On the Swan's Utility disk is a program named " copy" which copies files from any (readable) device to any (writable) device. For example, from keyboard to disk file, or from keyboard to printer, or from disk file to printer, or from disk file to screen, or RS232 to disk file, or disk file to RS232, and so on, and so on.

When sending data to the printer the program asks whether it is one of the Commodore printers in lower/upper case mode. If you answer positively, it sets a flag for the machine language portion of the program to deliberately confuse the 5 confused characters on the way to the printer, which re-confuses them back to what they should be. The data can be anything - Something you downloaded from a BBS, a Superscript output file - any CBM ASCII data. Happy printing!

4023 - 8023P PRINTER ROMS

by: David Ritterbusch

This article describes the changes/additions I have made to the 4023 & 8023P printer ROMs which make the printers compatible with the B-128 keyboard. For example when you type a left bracket the printer also prints the left bracket and not the graphics character. There are a total of 19 corrections to the 8023P ROM and 20 corrections to the 4023 ROM. The extra correction to the 4023 ROM is the small (k) which is changed to NOT look like a capital (K). These changes will also facilitate some graphics on Superscript III.

A good place to start this description is with reference to the Silver/Blue manual 'B SERIES - COMMODORE USERS GUIDE' that came with the B-128. Appendix D pages 116 & 117 is a list of the B-128 Chr$ codes and characters. The notes at the bottom of page 117 of the manual should read as follows:

```
GRAPHICS MODE:
    CODES   96 - 127  are the same as   32 - 63
    CODES  224 - 253  are the same as  160 -190
    CODE   255        is the same as 222
    CODE   254        is the programmable character.
NORMAL MODE:
    CODES   96 - 127  are the same as  160 - 190
    CODES  224 - 253  are the same as   32 - 61
    CODE   255        is the same as 222
    CODE   254        is the programmable character.
```

The following is a list of characters that have been changed to match the keyboard in both the NORMal and GRAPHics modes.

CHR$ CODE	MODE	ACTION
chr$ 91	- normal	changed to left bracket
chr$ 91	- graphics	added degree symbol
chr$ 92	- both	changed to UK pound sign
chr$ 93	- normal	changed to right bracket
chr$ 93	- graphics	added combined +/- sign
chr$ 94	- normal	changed to up arrow
chr$ 95	- normal	changed to back arrow
chr$ 167	- normal	changed to right vert. bar
chr$ 170	- both	changed right vertical bar
chr$ 180	- both	changed left vertical bar
chr$ 181	- both	changed left vertical bar
chr$ 182	- both	changed right vartical bar
chr$ 183	- graphics	changed upper horizontal bar
chr$ 184	- graphics	changed lower horizontal bar
chr$ 219	- normal	added degree symbol
chr$ 220	- normal	changed to UK pound sign
chr$ 221	- normal	added combined +/- sign
chr$ 222	- both	redesigned PI symbol
chr$ 223	- normal	changed graphic character

What does all of this mean? It means you do NOT have to use a special utility to be able to print brackets. Also, when you receive data from a network such as DELPHI or COMPUSERVE the information you see on the screen is what the printer will print.

The two added characters, the degree symbol and the combined +/- sign can be printed using a command PRINT CHR$(219) or PRINT CHR$(221) respectively with the printer in the UPPER/LOWER case mode. These can easily be printed from Superscript III using the FEATURE feature when the printer DEFAULT file is set up to use this capability.

Since these ROM changes and Superscript III are related I will describe the FEATURE set-up that will allow printing 10 special characters.

Special NOTE: If you do not want to have to press the space bar each time you see the message below displayed, change the first character in each line to a 'layout character' (CNTL-L). Superscript III will then bypass the message and the prompt and proceed directly to the edit mode.

```
*cm:sk5;Your printer has been defined as :
*cm:CBM Dot matrix
*cm:Press the space bar to continue
```

In the default file you will find a group of lines that start with a (:) followed by the word 'feature' plus a number. Below is a listing of the characters that I have assigned to the FEATURE selections. The number in front of the (:) is the CHR$ code of the character to be printed. The description in parenthisis is for your information as to what you have assigned to that feature selection.

```
219:   feature 1   (print degree symbol)
221:   feature 2   (+/- symbol)
163:   feature 3   (print underline character)
164:   feature 4   (draw line character)
165:   feature 5   (left verticle bar character)
167:   feature 6   (right verticle bar character)
:      feature 7   NOT ASSIGNED YET
:      feature 8   NOT ASSIGNED YET
186:   feature 9   (check mark symbol)
254:   feature 0   (user programmable character)
```

Once this has been entered into the defaults file then resave the file to disk and then execute the following commands to reestablish the default set-up.
CAUTION - ALL TEXT WILL BE LOST SO SAVE TO DISK FIRST
F1 P I P defaults (return). Type 'defaults' OR the name of your printer set-up file when prompted for "Enter Printer Type:"

To use the FEATURE features in your text execute the following sequence when you want to print one or more of the special characters.
F1 F F (n) where (n) is the number of the feature to be printed. A CTRL-R will repeat the command which makes it easy to draw a line 15 characters long.
The text on the screen will display a reverse (n) but the printer will print the appropriate character. The set-up described above matches the GRAPHics characters on keys 3 thru 6. For example an F1 F F 4 CTRL-R R R R R R R R R R R will draw a line 12 characters long for use as a signiture line if done just above your name in a letter.

The printer ROM changes allow the printer to print the right verticle bar character as shown on the 6 key. Without the ROM change it will print a duplicate of the character on the 3 key.

Special NOTE: A user designed character can be included in the defaults file and per the list above printed with the feature 0 command. In the defaults file below the features is a group of lines that look like this.

```
:                   set 8      Lines/Inch
:                   set 12     Lines/Inch
s5,4,12,20,60,68,132: User - Programmble character
35: #
36: $
```

The line starting with s5 is the command to send the character data to the printer and print it when the FEATURE 0 command is used. The 4023 printer uses 8 sets of data where the 8023P uses 6.

The printer ROM exchange is very simple for both of the printers. It requires no special tools. You need only remove the cover/lid, unplug the old ROM, plug in the new ROM and replace the cover/lid. The instructions supplied with the ROM explain the procedure in detail with photos of how to perform the printer ROM exchange.

The B-128 character ROM has a couple of errors which I have also corrected. This ROM is not so easy to replace since it must be unsoldered from the main board. The ROM changes reverse the GRAPHics characters on the C & V keys. I also changed the shape of the capital (O). See the ad for RITTERBUSCH BUSINESS VENTURES in this issue for ordering information.

Superscript III useful Hints & Tips

By: David Ritterbusch

These lines added to the 'ss en 30jun86' program of Superscript III will perform the following uses.

```
poke 55296,9 :poke 55297,9     rem adds space between text
poke 55296,11:poke 55297,8     rem changes size of cursor

open13,4,13:print#13:close13  rem Set NLQ mode of
open15,4,15:print#15:close15  rem 8023P printer
```

Superscript III text Tips

NOTE: A blank line is required before each group of commands when used in the text (edit mode) of SS III to send the 8023P secondary address commands.

8023P Hints & Tips

```
                   rem  mandatory blank line
  *sa13,13         rem  these two lines set 8023P
  *sa15,13         rem  to NLQ mode

                   rem  mandatory blank line
  *sa14,14         rem  these two lines reset 8023P
  *sa16,14         rem  to Draft mode

                   rem  mandatory blank line
  *sa6,12          rem  sets single line spacing
  *pl66            rem  sets page length to 66 lines

                   rem  mandatory blank line
  *sa6,18          rem  sets 1 1/2 line spacing
  *pl44            rem  sets page length to 44 lines

                   rem  mandatory blank line
  *sa6,24          rem  sets double line spacing
  *pl33            rem  sets page length to 33 lines
```

It seems that setting the NORMal and GRAPHics modes with SS III are backwards with the 8023P printer.

```
  *sa8,8           rem  set NORMal mode.

  *sa7,7           rem  set GRAPHics mode.
```

4023 Hints & Tips

NOTE: I don't know if a blank line is mandatory with the 4023 printer before each set of secondary address commands.

```
  *sa6,36          rem  sets single line spacing
  *pl66            rem  sets page length to 66 lines

  *sa6,54          rem  sets 1 1/2 line spacing.
  *pl44            rem  sets page length to 44 lines

  *sa6,72          rem  sets double line spacing
  *pl33            rem  sets page length to 33 lines
```

I hope somebody will find some usefulness out of these peices of information.

*** BYE ***

HEXPRIMER
A Primer

By: John A. Berezinski

This is a primer on hexidecimal numbers. It's foundation is based on conversations I have had with friends attempting to explain machine language and how computers work. It is meant to be an introduction to a primer on machine language.

Based on my conversations, I have found that it helps to give an explanation of the electro-mechanical side of the computer first. This description may not be truly accurate because of some interesting and unusual techniques used in electonic designs, but it should give a working understanding of what goes on inside your pet. See last paragraph.

The computer is essentially a machine made of switches. Some of the switches can be directly controlled by you (keyboard). Many control other switches. And finally some control other things (disks, screen, printer). The microprocesser is the primary driving machine in the computer. It is wired to connect (address) a variety of switches and act on the state of those switches.

Through some action of the microprocesser or you, these switches can be turned on or off. When a switch is turned on or off, it is called writing. When a switch is checked to see whether it is on or off, it is called reading. A switch that is turned on conducts a charge of usually 5 volts and is represented by a "1". When a switch is turned off, it will be grounded and is represented by "0".

Turning on your machine, activates a preset program. This program is actually a set of permanently set switches which the microprocesser is wired to read and act on. This begins the cold start routine of the computer and instructs it to begin reading the keyboard and memory and various other switches.

Now that you have a thorough understanding of what is happening, we will look at hexidecimal. The previous explanation is unnecessary to grasp except to understand that we are representing switches and their condition of on or off with numbers.

In our world, we use the decimal system. With the invention of the zero, one of man's truly great creations, we have a concise and flexible numbering system. The zero functions as a place holder. So the 0-9 can represent any quantity from none to nine. But when zero is used as a place holder, the numbers 1-9 represent a quantity of quantities. 99 means nine groups of ten plus nine ones and 435 would mean four groups of one hundred plus three groups of ten plus five ones. Note that the hundred column of a number can also be said to mean ten groups of ten. One thousand (1000) then means ten groups of one hundreds and ten thousand (10,000) means ten groups of thousands. This is known as a base 10 system because all numbers are grouped in bunches of ten.

An alternative base system is base 2. In this system, the highest number is one. That means that it would count like this 0 1 10 11 100 101 110 111. Each column represents two of the previous group. 0-1 represent zero through one. 10 would mean one group of two ones and 11 would mean one group of two ones plus one or in decimal 3. A number like 111 would mean one group of two twos plus one group of two ones plus one one or 2*2+2+1 or 7.

binary	decimal			
0	0			
1	1			
10	2	2*1		
11	3	2*1+1	2+1	
100	4	2*(2*1)	2*2	
101	5	2*(2*1)+1	2*2+1	
110	6	2*(2*1)+(2*1)	2*2+2	
111	7	2*(2*1)+(2*1)+1	2*2+2+1	
1000	8	2*(2*(2*1))	2*(2*2)	2*4

If this chart appears confusing, ignore it for now.

Notice that the binary system could lend itself as a notation system for representing the conditions of various switches in a computer. Many of the switches in our computer are grouped in sets of eight. In fact our system and the 65xx (reads sixty five hundred series) chips which control it are often called eight bit chips. Also our memory is arranged in this eight switch grouping. A set of eight switches is referred to as a byte and any of the individual switches in that set are called bits.

In the B128 computer there are three banks of switches. The operating bank, bank 15, contains memory switches and the various chips containing the switches which operate the computer and make it do things. Banks 1 and 2 contain 65535 sets of eight switch bytes which can be set or read by the microprocesser as instructions or data. The first two bytes in all banks of the computer tell it where it is supposed to be reading or writing.

A problem arises with using binary to communicate how a set of switches are set. And that is that for every byte represented, eight digits would be needed. 01000110 which has three switches on and five switches off could be represented in decimal as 70. When you use the basic POKE statement, POKE####,70, the computer looks at the byte of memory represented by #### and sets the eight switches (bits) to be 01000110.

An alternative number system, known as hexidecimal (10+6) is often used to represent binary numbers. Hex. uses a base of sixteen or each column of a number equals sixteen of the previous column. The advantage of hex. is that it is even more compact than decimal. The disadvantage is that we do not have enough numeric symbols in our world. So we will rewrite the number system and it will look like this; 0 1 2 3 4 5 6 7 8 9 a b c d e f 10 11 12 13 etc. Notice that when the count reached 10 instead of using the next column, we created a new number "a" and remained in the same column. When we reached 16 in our count then we moved to the next column, so the one in the next column means one set of sixteen. 13 would be one set of sixteen plus three ones or 19.

The advantage of hex. is that if you break up a binary 01000110 into two sets of numbers 0100 0110, hex. will represent it vary well. The first half is 0100 or (0*2*2*2)+(1*2*2)+(0*2)+(0*1) or 4. The second half is 0110 or (0*2*2*2)+(1*2*2)+(1*2)+(0*1) or 6. When we put them together we get %01000110 as $46. The "$" is the standard nomenclature for hexidecimal notation and "%" is used for binary. Using the hexidecimal base, $46 is four sets of sixteen and six ones or (4*16)+(6*1)=70. So %01000110 binary equals $46 hexidecimal equals 70 decimal.

Again looking at the mechanical side of the computer, there is one major anomaly from the eight switch byte scheme of our system. When a memory switch set or read in our memory, it must first be adressed or (you must go to the mail box before you put a letter in syndrome). This is accomplished by telling the microprocesser the adress or number of the byte to be used. But we have a problem in that the eight bit byte is only capable of representing numbers between 0 and 255 and we have 655536 bytes in a memory bank. Now it happens that while eight bits are not enough, sixteen bits work perfect. %11111111 equals 65535.

I must deviate even further from hex numbers and explain that for convenience the sets of switches are always numbered starting with zero. The first byte is considered to be byte zero and the last byte is not 65536 but 65535. There are still 65536 switch sets, 1 to 65535 plus 0. An example is the numerals 0 1 2 3 4 5 6 7 8 9, and although nine is the highest number represented there are actually ten digits shown and each could be used to label something.

Back to the mechanics of the computer. As it happened, the designers of the microprocesser realized that it would have to adress more then 256 bytes and included a sixteen bit adress line in it architecture. That means we can adress from %0000000000000000 to %1111111111111111 or 65536 bytes but we still have a problem writing out all those 0s and 1s. However if a binary number is seperated into four digit units, hexidecimal will symbolize very neatly. %1111 1111 1111 1111 is $ f f f f and our 65536 switches can be represented by $0000 to $ffff

hex.	dec.	hex.	dec.	hex.	dec.	hex.	dec.
0	0	0	0	0	0	0	0
1	4096	1	256	1	16	1	1
2	8192	2	512	2	32	2	2
3	12288	3	768	3	48	3	3
4	16384	4	1024	4	64	4	4
5	20480	5	1280	5	80	5	5
6	24576	6	1536	6	96	6	6
7	28672	7	1792	7	112	7	7
8	32768	8	2048	8	128	8	8
9	36864	9	2304	9	144	9	9
a	40960	a	2560	a	160	a	10
b	45056	b	2816	b	176	b	11
c	49152	c	3072	c	192	c	12
d	53248	d	3328	d	208	d	13
e	57344	e	3584	e	224	e	14
f	61440	f	3840	f	240	f	15

If you use the chart above, the number $f48e would be 61440 + 1024 + 128 + 14 = 62606. And 35248 would be 32768 + 2304 + 176 + 0 or $89b0. Also hexidecimal conforms to standard arithmatic rules. $0f + $01 = $10 Note that we had $f (15) ones and added 1 one giving $10 (16) one set of sixteen ones and the one carried to the next column.

I realize from experience that the hardest part of hexidecimal is thinking in units of sixteen. Once you achieve this the math and counting come fairly easy.

One of the deviations from the switch scheme is the dynamic random access memory chip (dram). It was developed to be much cheaper and simpler memory than a static ram. Where as a static ram is a set of switches, the dram is a set of capacitors. A capacitor is essentially a tiny rechargeable battery. So when a "1" is wanted, the cap. or battery is charged and it shows +5 volts and when a "0" is wanted, the cap. is drained and acts as a ground. Again the advantage of dram is the simplicity of the circuits on the chip, but the disadvantage is that these capacitors or batteries have to constantly be charged or drained in order to maintain the state in which they are set. This is known as a refresh and is handled by other chips in the computer.

John A. Berezinski
512 Apache Ave
Carpentersville, Il. 60110
312-426-2221 6pm to 10 pm ONLY

NMW Inventory Program Review #2
NWM Updated Inventory Program - or - The Good Just Got Better!

By: Bob Leoffler

Having used NWM's Inventory Program for over half a year now, I find I am amazed at how "clean" this program is. As those of you who have used the CABS software know, you can put a lot of work into something, only to have the program self-destruct hours of input - as in the case of inputting Accounts Receivable invoices and not updating them before you use another function.

Just to give you an idea of what we are going through to get our inventory on disk - as I mentioned, I've been using this software for over six months and the end is still not in sight. Of course, I don't work at it eight hours a day, but I try to work at it part of every day. As anxious as I was to get a workable Inventory Program, I must admit to some apprehension to committing this much time and effort to an unproven piece of software (especially in view of the experiences I had with some of my former company's new-release software).

But it has turned out to be a "not to worry" situation. Bruce Faierson had his P's and Q's together when he wrote this program. It is completely docile - no unpleasant surprises and no bad habits.

However, no matter how much thought you put into something, there's always going to be somebody around that wants things a little slicker and I'm no exception. So just after the first of the year I fired a wish list off to Bruce. That led to a series of phone calls between us and sure enough, it wasn't long before a new program disk arrived with almost everything I had suggested and in some cases, slicker ideas than I had suggested and some I had not suggested, but welcomed with open arms. When you take an excellent program and begin to add "magic" to it, that's when it REALLY becomes fun to work with it - that's just what Bruce has done with this program.

The following is the "wish" list I sent to Bruce, along with the resultant enhancements he implemented.

Merchandise Receipts/Issues

Request:
Would like to be able to use the "RET" (or whatever) key to "step" through the request for cost and/or retail prices when there is no change.

Solution:
You now can "step through" the prices by using "0 RET" keystrokes - this is a tremendous keystroke/time-saver.

Merchandise Issued

Request:
Stock <=0, etc., message takes too long - should be speeded up or limited to two flashes and/or have a beep tone with it or have the option of turning it off entirely. I personally have no need for it as I'm always looking at the stock on hand when I'm entering the number issued.

Solution:
It was not practical to employ the "beep" tone scheme, however it is now reduced to two flashes. I have no problem with this now because with the eight keystroke type-ahead feature, the timing between the flashes and the keystrokes is just right.

Delete Record

Request:
Would like to have the message that it can't locate an item remain on screen until you tell it to go away - that way, if you happen to be looking away (as I often am), you won't be fooled into thinking the item has actually been deleted.

Solution:
The message now remains on the screen until you hit the "RET" key.

Menus

Request:
Use number 0 or 8 to return to the main menus. Even having used the program this long, my mind refuses to memorise which menu call uses what number.

Solution:
You now can either type the function number as before or hit the "Home" key to get back to the main menu - - - (Love it!).

Merchandise Issued, etc.

Request:
When you enter an item incorrectly and it says it can't locate the item, it would be nice to have the characters you typed stay on the screen so you can see what you've typed wrong, until you tell it to go away.

Solution:
Here's some of the "magic" I told you about. Bruce went one better on this request. If you enter an item incorrectly, the system will still say it can't locate the item but will now ask if you want to search for the item. The "RET" will initiate the search, will find the item that resembles the item closest to what you typed and will display it on the screen and gives you the option of using that item or not. The closer the item description you typed is to the real thing, the better your odds of landing on the correct one is. If it is too far away, it will display the closest thing it can find, which is usually pretty close to the correct item. You can then scroll up or down with the Cursor Left and Right keys until you land on it.

Prior to this feature, I had to have an inventory report next to the keyboard, at all times, to make sure I typed in each item 100% correct, or the system would not be able to locate it. Now, all I have to do is be reasonably close. Once in a blue moon it will land on something a long way away, but I've usually typed the description a long ways away from the real thing.

Enter Record

Request:
Would suggest three places in "Cat/Loc" to accommodate 999 storage locations or a scheme such as 21A, 21B, etc.

Solution:
Bruce advised that implementing this would use up considerable space, so did not. This did not disturb me as it was not very high on the priority list.

Merchandise Issues/Receipts, etc.

Request:
If you select these functions and then change your mind, there is no way to get out except for plugging in numbers until you get to the "Store y/n" function.

Solution:
You can now exit the function with "? RET" keystrokes, provided you have not entered any numbers and hit "RET", then you are committed to go to the "Store" function as before.

Bruce also made the following changes:

Under normal circumstances, each time you "stored" something you were asked "Shall I continue?" and you were forced to hit the "RET" or "y" keys. It will now automatically continue to allow you to input another item until you enter a "?" to exit the function - another real time-saver.

Experience has proven that most people more often than not, do not need to use the calculator. As it is now, you are asked if you want to use the calculator on each item and must acknowledge same.

Bruce has changed the program so you can turn off the "Use Calculator" inquiry, which eliminates the necessity of an answer, however, it will be available upon YOUR request when YOU want to use it - it's proving to be quite a timesaver also .... as they say in the computer typesetting industry, "keystrokes are time and time is money - the fewer there are, the less it costs you".

Reports

Request:
I've noticed that some print functions will form feed to the end of the page, while others don't - lost my list so can't tell you just which ones do and don't, but would rather have them all do one or the other.

Solution:
This has been implemented - all reports now form feed to the end of the page

Summary:

All in all, if you are still using the CABS inventory program and don't like it, I guarantee you'll like this program. And if you have an earlier version of this program, you will definitely want to up-grade to this version - - it's a real winner!!

Bob Loeffler
Green Lake Hardware
511 Mill St. Box 218
Green Lake, Wi. 54941
414/294-6412

SUPEROFFICE NOTES

By: Bruce Faierson

After a year of negotiations, Precision Software has released Superoffice. This has come about through the hard work of numerous people including Mark Scharzbauer, Norm Deltzke and Northwest Music Center, Inc. In late December, Northwest Music Center Inc. purchased the North American rights to distribute SuperOffice from Precision

Software for a sizeable sum. Due to the amount of work put in by CBUG members to acquire this program, we have elected to release this program for only $49.95. The original selling price was around $1,000.

What is SUPEROFFICE ?
What is the significance for B-128 users ?
Why would I want SUPEROFFICE ?
What are the hardware requirements to run SUPEROFFICE ?

I will try to answer all of these questions in the following article. SuperOffice was developed by Precision Software several years ago and was a program ahead of its time. Most microcomputer programs at the time SuperOffice was developed were stand alone programs. The user had to exit one program, load another program and then import and reformat data to transfer information from one program to another. Needless to say this was annoying at best and a waste of time. SuperOffice would have filled this need but never made it to the market in the United States.

SuperOffice is an integrated version of Superscript II and Superbase II (or Superbase I, we have both versions. Note Superbase II in Superoffice is version 2.08, which is later than any published version of which we are aware. It has all the features of each program with its own command structure. This facilitates data transfer and switching between the two programs. An integrated program allows you to load more than one program into memory and pass control and data between program modules.

The most simplistic example of SuperOffice use is to type a document in Superscript, exit Superscript to Superbase, look up an address in your address file and return to Superscript. Note that neither the document or database file is erased when you switch between the program modules. To accomplish the same simple task before you either had to have two complete systems or exit and reload programs twice.

The significance of SuperOffice is that it finally brings the B user into the future. Most of the major computers have integrated programs available for them. None of them are quite as easy to use as SuperOffice. At the most basic level the user can save time and effort switching between the database and word processor. On the most advanced level the user can not only have automated databases but also automate the word processing functions. The SuperOffice applications language can control both Superscript and Superbase operations.

A simple example of programming the word processor is to create a small file format for document information. The user can then write a simple program to prompt for a text file name. If the file exists it would be loaded for editing in one of the text banks. If the file does not exist the program could ask whether to create it! Mail merging is also a snap with SuperOffice. The users imagination is the only limit to what can be done with SuperOffice.

The simple example above can save the user many annoying searches looking for text files. Anyone who has saved over fifty documents to disk knows what a delight it is to watch the screen scroll by when all those neat files are displayed. Of course you can page the directory up and down by screen but if the files are similar in content it is difficult to remember what is in them. With SuperOffice you can do a search by keyname, sliding match or a field specific match. With this capability think how much easier text file storage and retrieval would be.

With SuperOffice the user can have multiple drives and printers on line. With the programming capability the user can have Superscript files on drive 8 and Superbase files on drive 9 and switch between them. The user could have an 8050 as drive 8 with spellcheck and use a SFD 1001 as a drive 9. The 1 megabyte capability of the SFD is perfect for large database applications. Printer numbers can also be changed easily through SuperOffice programming.

SuperOffice can also utilize 576k of memory. A rounder figure would be 512k of ram or one half megabyte. This allows for seven text banks and three database files open. Amazing stuff!

To run SuperOffice you only need a 256k machine. This does not have to be a stock CBM 256-80 as any upgrade machine will run with it. The basic roms in the machine do not interfere with the functions of SuperOffice. Since most users have not upgraded to 256k, probably for fear of being without their machines for a while. We have contracted to have a user installable 128k upgrade designed. This a plug in board that will retail for $125. The installation is completely user friendly, reversible and non-destructive. The upgrade will also allow you to use over 12,000 cells of Calc Result, 3 banks of Superscript, Word Result-Calc Result integrated and 4 banks of 8432 emulator.

Why should I want SuperOffice ? The answer to this question is as simple as, why did you buy a computer. To make your business more cost effective and to save time! SuperOffice is certainly the lowest priced integrated system of quality for an microcomputer on the market today! The time saving conveniences are worth far more than the $49.95 introductory price.

Besides helping yourselves you will also be supporting CBUG as we will be making a $5 contribution for every copy of SuperOffice sold. Everyone has to admire the long and dedicated hours that Norman Deltzke has put in to support the B-128 users. Without "STORMIN NORMAN" it would be safe to say that the B users would have almost none of the information or programs that it now has available. We vow to do our share in supporting the B128 cause!

We will also support a section in our ads for SuperOffice programming tips as the use of SuperOffice increases. We welcome any user tips and suggestions. We are available to answer questions at 312-299-2270 between the hours of 3:30 and 8:30 Monday thru Thursday and Saturday 9-4 p.m.

NEWS FROM THE CP/M-86* WORLD

By: Bruce Faierson

So you didn't think it could be done! You thought that Commodore was just blowing wind when they stated they had developed 8088 co-processors that could run CP/M-86 and other related operating systems. We have not only got the operating system up and running but are operating many of the major programs and language software. Since late October, NWM Inc. has spent a fortune buying all the CP/M-86 software it could find. We also have spent hundreds of man hours scrounging information, porting software and rewriting a custom bios to get a quality terminal program up and running.

First, a little history on the CP/M family of operating systems. CP/M and all of its descendants are registered trademarks of Digital Research Inc. At the time that the B-128 was in development, CP/M was still the dominant operating system for most microcomputers. CP/M itself meant Control Program/Monitor though today it is usually referred to as Control Program for Microcomputers. The creator of CP/M was Gary Kildall.

In the early 1970's, Mr. Kildall was a computer science instructor and a software consultant for none other than Intel, designer of the 8088. Due to his interest in the new microcomputer industry he accumulated spare components for a home computer system. At this time there were no operating systems to make it useful, so he had to write his own operating system from scratch. He used the IBM mainframe language PL/M to write CP/M. He tried to sell Intel on his new operating system but they were not interested because microcomputers were a rarity at this time. CP/M went on to become the most popular and foundation of modern microcomputer operating systems.

When IBM marketed their 16 bit personal computer in 1981, a new wave of operating systems came to the market. Seattle Computer products created QDOS or 86-DOS which Microsoft bought a license to market. This eventually became known as MS-DOS or PC-DOS which is a variant of it. Digital Research was beat to the punch in terms of getting an 8086 type operating system to the market. Eventually they produced what is referred to today as CP/M-86 and a number of variants ranging from a multi-processing Concurrent CP/M-86 to Concurrent PC Dos.

At present Digital Research has CP/M, CP/M-86, CCP/M-86, Concurrent PC Dos, CP/M-68K and CP/M Z8K on the market. The latter two support the 68000 series and the Z-8000 series chips. The Commodore B series supports use of the CP/M-86 operating system. This is of course providing it is a high profile machine with the correct PLA and an 8088 co-processor board built in.

The most important feature of the CP/M type of operating systems for B users is that programs written correctly to the operating system are transportable to most CP/M type systems. It does not matter what the physical or logical format of the disks are because that will not affect the actual porting. All you would need is a good terminal program like Move-it by Woolf Software to transfer the data. By the way, if any of you have wondered how Commodore decided on the 77 track format for the 8050, and still have not figured it out, this may be the answer. The IBM 3740 format is 77 tracks and is considered the only legitimate CP/M standard format.

Each manufacturer or user of a CP/M-86 compatible computer must modify the bios for the particular physical hardware of their system. The bios is the basic input/output system. The bios tells the operating system how to access various devices related to the particular system. The bios is the only user or manufacturer configurable part of the system and is not meant for a novice to configure. This is why correctly written programs can run on different brands of computers even though they have different hardware. CP/M treats each device as a logical device, not a physical device. The advantage appears to be that the manufacturer writes the bios for connecting physical devices while the program will access the logical device.

I will try to explain how CP/M-86 is actually implemented on the B series though I am not truly qualified. It must first be realized that most CP/M type systems have only one central processing unit, direct memory access and more similar drive formats. The CBMX-128 and 256 have two CPU's, IEEE access and of course unique GCR coding on their disks. I will not even try to get into the type of coding that was used on their CPM/86 formats other than they have 64 directory entries and 410k of usable disk storage on an 8050 disk. More on this in future issues of CBUG. You can see that there are a few problems to be worked out here.

The two CPU's, the 8088 and the 6509, must both be on continuously and passing control back and forth between each other when necessary. This concept dates back to Commodore's fling with the Superpet. This machine had a 6502 and 6809 co-processor arrangement. The logic here was that the 6509 side would handle all the i/o. This meant it would talk to the screen, handle keyboard input, drives, printers and the rs-232. The 8088 side would take care of data processing and calculations. The catch word here is concurrent processing.

Concurrent processing usually means that multiple tasks are seemingly occuring at the same time. This is not quite true, as in most cases the processor is just sharing time with multiple tasks when it is not busy. In terms of the B series it just means that both processors are on and relinquishing control back and for between each other. The problem here is that the 8088 side has to pass everything back to the 6509 side for screen output, disk access etc. Some operations are therefore slowed down.

The method upon which we load CP/M-86 reflects upon the above information. A boot program is used to load a cold start loader which in turn loads the CP/M-86 operating system. After the operating system is loaded into memory it passes control to CP/M-86 which then performs initialization operations and displays the opening "A>" prompt. After using Commodore computers for so long, it is hard to imagine coming up in anything other than Basic. Yes there are other operating systems different than ours. It will also be noted that we start in the Commodore operating system and transfer to the CP/M-86 system.

We will get into more of the specifics and operation of CP/M-86 in the next issue. I hope this opens up another area of discussion and interest with the B series computer. It is now time to get into some practical information about the use and value of CP/M-86.

If CBM had backed the B series the way it had intended, we would have had a lot more software in the native 6509 mode than we have now. If Digital Research had been quicker to get CP/M-86 to the market and had priced it without the assembler, we might have a different dominant operating system for the IBM then is now available. There would also be a lot more CP/M-86 programs available. What we have is an interesting cross situation. With the B series we have a limited selection of a few good programs available. With CP/M-86 we have a complete development system, major languages and major programs available. Together we have a good, practical system that can be operated and developed in many directions. With CP/M-86 we have now opened the doors to not only software developers for languages but the opportunity to use the same type of programs that our friends on the IBM and compatibles use.

Now before we all jump in and say "WOW THIS IS IT", it is time to get into the realities of what programs are available for CP/M-86. The first thing to understand is that the programs available are all highly respected professional software that cost real money. We are not talking about $9.95 either. Unless you can find unbelievable bargains on the languages, they are several hundreds of dollars. The mainstream programs like Wordstar, Dbase II and Supercalc are also in the hundreds. Net result, you pay for what you get!

Most these companies support their products very heavily and you will pay for that. These programs will also allow you to transport work between different computers using those programs and in the IBM world almost everyone uses some variation of these programs.

A partial listing of known programs that will operate on the B series appears below with annotations as to the developer or distributor and the particular usage.

LANGUAGES & OPERATING SYSTEM

1.) CIS Cobol DRI Inc. Level 1 Business language-Government certified.

2.) Cobol Level 2-DRI Inc. Only mainframe certified Cobol for micros.

3.) C DRI Inc. most popular development language today.

4.) CBasic 86	DRI Inc. most popular structured basic for business.
5.) CB86 Compiler-	DRI Inc. Compiler version of above.
6.) PL/I	DRI Inc. popular variation of powerful mainframe lang.
7.) Pascal MT+86	DRI Inc. popular version of Pascal language.
8.) SPP-86	DRI Inc. Speed programming package for above.
9.) CP/M-86	DRI Inc. makes all of the above work.

CP/M-86 PROGRAMS

1.) Wordstar	Micropro	most popular word processor ever. WYSIWYG type.
2.) Calcstar	Micropro	popular spreadsheet package.
3.) Spellstar	Micropro	integrated spell checker for Wordstar.
4.) Supercalc2	Sorcim	Very popular spreadsheet.
5.) Dbase II	Asht-Tate-	Most popular programmable, relational database.
6.) Move-it	Woolf	Ex. terminal program w/proprietary protocol.
7.) Pers. Pearl	Pearlsoft	non-programmable database manager.
8.) Modem86	Hersey	Well known Xmodem type terminal program.

This does not in any way represent all the commercial or public domain software available for CP/M-86. It only represents that which NWM, Inc. owns legitimate copies of and has tested to verify proper operation. These also represent generic versions of these products as machine specific versions do not usually have terminal installation files. It should also be stated that when buying the generic versions of these programs they will also run on the CP/M-86 side of the IBM. Therefore they are usuable if you change systems.

This also brings up an interesting feature of the CP/M-86 operating system. Our own B-128 emulates only a B-128 dumb terminal. It is therefore difficult to use our current terminal programs with any system that requires a particular emulation such as the standard VT-100 terminal. Sorry, we do not emulate the VT-100 terminal but we do emulate the ADM-3A terminal. This is what allows us to configure the above wonderful programs without mess and fuss. Thus we can use a terminal program under CP/M-86 and communicate intelligently with a system that is running the ADM-3A terminal configuration.

As a simple example, the ADM-3A terminal will allow us to access a program called Remote, by Microstuf, on the IBM PC. Remote will not only emulate many terminals but allows the user to remotely access and run programs. That's right you can run Dbase III or other fun programs from a remote location with your CBMX computer. Forget anything that relies to heavily on graphics though. You can also change and view directories and leave mail like on a bulletin board. With an Xmodem program you can also upload and download files. It's getting more fun isn't it?

We have spent an awful lot of time and money pursuing this matter. As a matter of fact much more of both than will ever be recovered by sales. The primary reason we researched this was to give more options for use of the B series computer. At this time it appears that we have achieved some worthwhile results. THERE IS A LOT MORE WORK TO DO AND WE WOULD APPRECIATE ANY HELP WE CAN GET !!! THE MORE YOU EXPERT PROGRAMMERS, HARDWARE GENIUSES AND USERS CAN HELP US THE MORE ALL OF US WILL BENEFIT !!! YOUR USER GROUP IS ONLY AS STRONG AS THOSE WHO CONTRIBUTE AND WE HOPE ALL OF YOU WILL !!!

The help we are referring to could be any or all of the following:

1.) Writing articles.
2.) Tracking down public domain and commercial programming for CP/M-86.
3.) Finding programming that could be ported to our machine from other languages like C, PL/I or Cobol etc.
4.) Writing programs for the CP/M-86 side let alone the 6509 side.
5.) Looking into writing a new bios for improvements.
6.) Looking into the development of Concurrent CP/M-86.
7.) Writing utilities for CP/M-86.
8.) Tracking down CP/M-86 bulletin boards.
9.) Development of DMA for CP/M-86.
0.) Tracking down information on the concurrent graphics controller.

With your support we hope to continue this section in the next issue of The CBUG ESCAPE. For more information regarding developments and offers of assistance in this project please call NWM Inc. at 312-299-2270 between the hours of 3 p.m. and 8 p.m. Monday thru Thursday and Saturday 9 to 4 p.m.

CP/M86 is a registered trademark of DRI Inc.

* *

CORRECTION TO ADVERTISING PAGE 8
REVISED CO-PROCESSOR INFORMATION

The 128K RAM expansion board as advertised in this issue by NWM will not operate with the 8088 co-processor.

* *

PAS-BAS: WHAT IS IT?

by: Mauricio J. de la Torre

PAS-BAS is a tutorial system and a translator. It teaches you to program in Pascal weather you know BASIC or not. The menu driven tutorial has been designed to help beginners with a special section. For those with previous experience in other computer languages there is an advanced placement test. If you want to keep track of your progress, the tutorial keeps a record of who you are and what you did during the last session.

The translator changes a program written in Pascal to a BASIC program so you can run it and see the results. It is also a syntax analyzer. If it finds a syntax error, it stops translating and shows you where the error is. As it translates, it displays what is translating and denotes the functional parts of your source program (ex: variable type, expression, function name, etc.).

Why Pascal?

Pascal is the most widely used language in universities around the world today. It has been chosen by many schools and professional programmers for several reasons. Tony Roberts in the June 1987 issue of COMPUTE! said: "Pascal [was] developed by Niklaus Wirth in the 1970s as a teaching language....Wirth reasoned that...[p]rogrammers who learned Pascal would have a solid understanding of programming and could easily pick up other languages as needed. As it turned out, Pascal grew beyond the classroom and has become a development tool for many programmers who needed a language that's easier to work with...."

Pascal is a structured programming language. This means that you do not have to keep track of line numbers for subroutines or even have a need for a 'goto'. In Pascal you call a function or subroutine by name and have a wider range of data (variables) types. This and other advantages in Pascal allows the programmer to write and debug his programs with ease. However, for anyone trying to learn Pascal as his first computer language, or for anyone with the desire to step up from BASIC, one has to have a way to learn it and a way to use it.

Tutorial systems are practical and flexible learning tools. They are available in some public domain libraries, but if you are serious about learning a computer language you also want to test your skills, practice what you have learned, and get some positive feedback.

PAS-BAS was developed with the above considerations and with program adaptability in mind.

Why a Pascal to BASIC translator?.

Most personal computers include the BASIC language as standard. The majority of the people that own personal computers are familiar with the BASIC language. A translator is an excellent tool to get the necessary feedback when writing your own Pascal programs. It allows to compare equivalent programs in different languages.

Adaptability is another reason to choose a translator over a compiler. If you want to have a program running in different computers you may use a compiler for each or a translator for all. This is specially convenient for programs you want to transport to computers of different brand. However this flexibility can only be done with some tradeoff. A translation is not directly related with the type of processor a computer has, therefore it presents flexibility in one way and some limitations in other. Flexibility because you can alter easily any part of the translated program, and limitations because you cannot use graphics or special features of the particular computer from the source (Pascal) program. Nevertheless if learning to program is what you want, a translator is more advantageous.

The PAS-BAS translator recognizes a wide subset of UCSD Pascal including nested and/or recursive functions and procedures, although it is limited by the possibilities of BASIC (i.e. PAS-BAS does not emulate some Pascal data types such as pointers and records).

The Pascal version supported by the translator includes the following: AND, BEGIN, CASE, CONST, DIV, DO, DOWNTO, ELSE, END, FILE, FOR, IF, MOD, NOT, OR, READ, READLN, REPEAT, THEN, TO, UNTIL, WHILE, WRITE, WRITELN, VAR, TRUE, FALSE, INTEGER, ARRAY, BOOLEAN, REAL, CHAR, STRING, TEXT, PROGRAM, FUNCTION, PROCEDURE, ABS, ATN, CHR, COS, EXP, LN, LOG, ODD, ORD, PRED, ROUND, SIN, SQR, SQRT, SUCC, TAN, TRUNC.

The PAS-BAS system also includes an editor and other convenient utilities. For more information call ISIT (713) 932-1691.

ALTERNATE OPERATING SYSTEM EXPANSION - Part 2

By: Gary Anderson

As stated in the last issue of the Escape, in this issue I will go into more detail on the Alternate Operating System Expansion (Alt OS Exp) board project presently nearing completion.

THE HARDWARE: The circuit board will be approximately five inches square with double sided plated through holes and solder mask and will plug onto an expansion pin field, P7, on the computer's main circuit board, or on the expansion pin field on the B-1024 1 Megabyte Memory Expansion Board on onto the possible future co-processor. This expansion pin field provides connections to +5V power, ground, CS1, phase2, R/W, hardware reset, data bus and address bus.

The three ROM (read only memory) ICs on the computer's main board get moved over and installed on the Alt OS Exp board. A ribbon cable connects the computer's main board ROM sockets to the Alt OS Exp board to pick up the chip selects for the three ROMs. Due to the physical differences between the low boy and high boy main circuit board ROM layouts a unique ribbon cable is needed for the particular machine being used.

The original hand wired prototype design has been modified resulting in improved capability. Besides the locations for the three original Commodore operating system ROMs there are locations for two other alternate operating systems of three memory ICs each. These two alternate areas are layed out to use the industry standard 2764 UVPROM (ultra violet eraseable programable read only memory), 2764 SRAM (static random accèss memory), and 2864 EEPROM (electrically eraseable programable read only memory) pin outs. Moveable jumpers select which operating system of the three boots from power up. Don't try powering up from 6264 SRAM as static ram looses its contents at each power down, SRAM can only be loaded and executed from after a boot from ROM, UVPROM, or EEPROM.

There are a small number of "S" and "LS" TTL support ICs that implement the address decoding, control register, control logic, power up initialization, and the read enable buffer. The amount of +5V current consumed for the whole board is about 200ma.

SOFTWARE CONTROL: The final version of the board has been decoded to use the bottom eight address, $F:D900 to $F:D907, of the page (256 bytes) called CS1 at $F:D900 to $F:D9FF for accessing the control register. All of the control bits are bidirectional, not only do you write to them but you can also read the state they are in relieving the user of placing and maintaining a copy of the control register in software.

The bits in the control register perform the following functions:

<u>BIT0</u>: A logic "0" forces execution from your original Commodore ROMs. A logic "1" causes execution from one of the two alternate operating systems selected by BIT1. The state of BIT0 at power up is initialized by a moveable jumper giving you the option of booting from an alternate operating system.

<u>BIT1</u>: A logic "0" results in execution from Alternate Operating System #1 and a logic "1" enables execution from Alternate Operating System #2 when BIT0 is a logic "1". The state of BIT1 at power up is also selectable by a moveable jumper giving you the option of selecting which Alternate Operating System boots provided that the jumper for BIT0 is placed accordingly.

<u>BIT2</u>: This is the Alt OS #1/Alt OS #2 write selection bit allowing selection of the alternate operating system which is to be written to for either initialization or modification. A logic "0" selects Alt OS #1 and a logic "1" selects Alt OS #2. From power up this bit is a logic "0" selecting Alt OS #1.

<u>BIT3</u>: This is the write enable bit. A logic "0" disables write capability to either alternate operating system. A logic "1" enables the writing of data to the alternate operating system selected by BIT2. From power up BIT3 is a logic "0" disabling write capability to both alternate operating systems. Only SRAM and EEPROM can be written to, UVPROM must be programmed in a special programmer.

<u>BIT4</u>: This is the hardware reset bit. Setting this bit to a logic "1" will reset the machine and bits 2 through 5 of the control register. BIT0 and BIT1 of the control register are left intact during a hardware reset as they select the operating system that is to be executed when the machine reboots. If the warm start vector is left intact before setting BIT4 a warm start occurs. If the warm start vector bytes are something other than what the operating system wants to see then a cold start occurs.

<u>BIT5</u>: Implemented in hardware and can be read but does not control anything.

<u>BIT6</u>: Not used

<u>BIT7</u>: Not used

OPERATION: Presently I have a diskette with a basic loader program and data files representing the B-128 and CBM-256 operating systems. I just insert the diskette into drive 0 and shift run to display a selection menu. I then select the operating system I want and which alternate area it is to be loaded in and it is all automatic from there. The 8K byte data blocks are loaded in and then a cold start occurs. Do not forget, to run the CBM-256 operating system in a 128K byte DRAM machine you must increase the DRAM memory to at least 256K bytes by adding banks 3 and 4. This can be done by having someone install 64K by 1 DRAM memory chips on your main board or by installing the B-1024 1 Megabyte Memory expansion board in your machine.

The flexible design of this Alt OS Exp board yields the option of loading an operating system into SRAM or EEPROM, transfering execution with a cold start to the newly loaded operating system, enabling the write line and then modifying the new operating system while it is being executed. Unless you know exactly what you are doing do not be surprised if you experience many a system crash.

The advantage of running the CBM-256 operating system is in the way variables are split up. Instead of all of the variables residing in bank 2 only as with the B-128 operating system the CBM-256 operating system splits up the variables over banks 2, 3, and 4 giving much more breathing room for variables, strings and arrays.

Look for my advertisements in the next issues announcing the Alternate Operating System Expansion board for the B-128/CBM-256.

Gary L. Anderson
2560 Glass Road N.E.
Cedar Rapids, Iowa; 52402

B-128 SWITCHING POWER SUPPLY UPDATE

By: Gary Anderson

I have come across a replacement transistor that can be used in the switching power supplies of our low profile B-128s that should be more reliable than the original transistor. The schematic diagram that I have seen on this 35 watt supply calls out a Motorola MJE-8501 as the switching transistor in the oscillator. This is the one that radiates the vast majority of the heat you feel rising from the top. I feel that the Motorola MJE-8503 is a better choice if the original ever needs to be replaced. Consider the following specs:

Motorola Type #	Continuous Collector Current Amps Max	Power Dissipation (Case) Watts @25C
MJE-8501	2.5	65
MJE-8503	5.0	80

As can be seen, the MJE-8503 can handle more collector current and dissipate more power than the MJE-8501. The MJE-8503 has the same case style, TO-220AB, as the MJE-8501 resulting in no mounting problems. As far as procurement you will have to call a Motorola distributor. One last comment, my two articles in this issue were written with a MJE-8503 installed in my power supply.

Gary L. Anderson
2560 Glass Road N.E.
Cedar Rapids, Iowa; 52402

A BUG FOUND IN ASSEMBLER 5.5

By: Howard Harrison

Have you run into a problem with Asm5.5 while assembling "bad source?" This has been brought to my attention that there is a bug in the assembler. The problem is when you assemble a line that has an error in a label.

```
        opt 1
        org $0400
ptr     equ $fb
label   equ $fd
        ldy #0
bad     lda label(ptr),y     ;oops i didn't realize that one
        rts
```

The assembler will lock up while trying to assemble the "bad" line. This is also true for "byt #" (byte immediate) and other keying mistakes. A fix:

Either you may purchase the "assembler 5.5 v8" upgrade or you may key in this source:

```
   (the comments are there for your information regarding
       the bug)
   (they are not required in the fix)
   (the change occurs during assembly, but asm 5.5 doesn't
       care)
   (remember to save this source before assembly just in
       case you make a boo boo!!!)
       opt 1
       bnk 15

error  equ $0044
asci   equ $52fb
       org $52dd
nolbl1 lda #1 ;bug: index not incremented after error found
  sta error   ;     there is an "inc index" and "jmp morop"
  jmp asci-2  ;     2 bytes before asci, so go there
  org $4000   ;save the asmblr. note comments can't be used
              ;after a filename!!
   sav assembler5.5
   res $2000
```

Please use the "bad" source or simulate the error and test the assembler, after installing the fix. You should encounter a "bad label error" in that line. This fix will cause the index to skip to the next character in the line instead of continuing to process that same error byte! I have been using my assembler for over 2 years before the release, and I haven't found this bug. and it existed from day1.

THE NEW JERSEY GOLD MINE

By: Mr. Anthony J. Goceliak

Sneaky move Norm! A little device like announcing that I've got my own column and now I'll bet you expect a regular supply of articles from me right?

OK this time, but if I don't have anything to say, you'll be the last to know. First thing I'd like to tell the membership is to hold on to any datasette cassette recorders they may have. When I first bought my b from Protecto, I looked carefully at the rear and saw the infamous cassette connector. The failure of Commodore to have pointed it out on page 17 of the silver user's guide didn't bother me in the least, because they had written half a chapter on the use of the Datasette (pages 45 to 50) in the equally infamous blue user's guide! By gum, I had a B128-80, it said so on the back of my computer, the blue user's guide said B128-80 right there on the cover, and so I ran out and got a Datasette. Anybody else who ever tried to use one with their b knows Exactly what I got when I followed those instructions, and intended results isn't included by a long shot.

I'm an obstinate cuss, and if I went out and spent good money on a dagnabbed cassette recorder [$22.35!], I want it to do something! This one doesn't even have it's own speaker, so audio tapes are out, crushing walnuts is no good either, since the plastic case doesn't look THAT sturdy, and that leaves only one option -=- make it do what it should have done in the first place!

Well I did, and that's what the business about saving your datasettes was all about. Why? First because I'm stubborn (have I ever mentioned that?), and second because it is a good, no let me say GREAT idea!

You all know about pets and vics and c-64's [p/v/c], and the tons of programs in the public domain available for each of them, but how do we transport these programs to the b? Modems and wires dangling accross the room? OK, but not every commodore computer owner has a modem, and most don't have two. Maybe just exchange disks? Also OK, but again, 2031 drive to read 1541 disks, oops, perhaps there are one or two of us who don't have a 2031, and one or two p/v/c owners who don't own disk drives at all! Take a list from the p/v/c and enter it by typing the whole thing on the b? Works fine, but oh my Gching fingers! Yeah, TAPES! Now that's really the common medium! Do you know that every computer that commodore ever built up to ours can read a tape written by any other commodore model? True! -=- and that now includes us b owners! Of course loading and running are not synonymous: the program must be suitable. If you list a pet program and the whole thing looks like this:

```
10 sys 1039
```

then the b will load it, but don't expect much if you try to run without modifying it.

What this listing indicates is a program which was written exclusively in machine language, with a single basic instruction to tell the computer where to begin executing the machine code. M/l object code tends to be written for one computer model alone, whereas basic does all those nasty housekeeping chores for you [for instance deciding where screen memory is on your computer and exactly what symbols to store in 'memory' when line 100 says print "hello"].

Mostly or purely basic language programming from the p/v/c's may need a tweak or two, or may run fine just as loaded.

Peeks, pokes, sys's, references to colors in the print statements, and all the usual stuff has to be reworked, but at least the program is now loadable into the b, and that is really about 91% of the battle!

By the way, you can also transport your marvel of basic [or m/l] programming to a p/v/c, or whatever almost as easily, just change the b's start-of-basic pointer, [as in Ms. Deal's configure pgm - on cbug disk #2], or somewhat harder, because I can't tell you EXACTLY what to do, tell the p/v/c owner to disable the automatic relocation feature of his machine during the load, and rechain the lines. On p/v/c's I'm NO expert, but it involves the mysterious 'load,1,1' versus load. I have even read of someone claiming to be able to load commodore tape programs into another manufacturer's computer, so you may be able to export programs there too. Importing programs from non-commodore computers is anyone's guess, but I'd say don't expect to do it.

Essentially what I have done is to disassemble, analyze and then 'borrow' a few ideas from the kernal of a b128 model 500 user (the b version with a working tape port) who was good enough to send the system dumps to Ms. Deal about a year ago, edit, modify and/or rewrite to make it compatible with our kernal, and presto, a tape operating system! Sounds easy here, I wonder why it wasn't when I actually did it?

Right now is a good time to publically thank Ms. Deal who was very kind to help me in researching tape parameters and testing the compatibility of the tape code with her PET. We can read and write to PETs, and C-64's, [I haven't found anyone yet willing to 'fess up to owning a VIC who also had time to test, but I'm confident we're compatible.].

From basic you can load, save or verify program files, or open for writing or reading the tape's equivalent to a seq data file [no commodore tape has rel or usr files], and generate end of tape markers using secondary address #2. [these are really neat - automatically shuts the cassette off and tells you you've reached the end of recordings on the tape instead of searching blank tape for a program or file that isn't there, especially if you use c90 tapes!] [i.e. 'save"my program",1,2' or 'open"ascii data file",1,1,2']

Commodore's standard tape format includes two copies of each block, and so we get two shots at each byte as it flies by. Should the first copy test bad, tape code logs it for re-trying when the second copy goes by. If that one reads ok, you're home free, and I won't even bother to tell you, but if you're really curious to know how good your load was, check for the number of first-try errors by typing [from basic] 'bank 15:?peek(174)', otherwise, just wait for either the 'ready' prompt [load or verify OK] or an error message.

You should also check the i/o status word 'st' [you DO check this following every disk and printer operation already, don't you?], for an indication of exactly what type of error you encountered if any. From basic type '?st' [and return]. I set the 'reserved system variable st' exactly as spelled out on page 208 of the Protecto PRG, [by the way, you may want to correct the footnote, because our '700' now has cassette i/o!]. If you don't understand the meanings of various st values, 0 is an OK, and 64 means you've read through to the end of the file without encountering any 'real' error. You'll get the end of tape indication only if you put it there, so if you didn't, consider any other value to be an error indication. I let tape code build a table of first-try error addresses, but that won't help you much. [i.e. if address $1111 is = #33 if the second try couldn't fix it, it doesn't do you much good.]

Just for the sake of completeness, do not perform a basic 'load' and then a 'sys5' to check for first-try errors, because the byte at $ae will be overwritten AS YOU ENTER THE MONITOR. From the monitor, performing a 'l"super m/l",01,02abcd and then checking $ae is valid.

I have tried to keep my use of b-system locations to those listed in the Protecto Programmer's Reference Guide, but some symbols are pretty obscure. [SHCNH?]? Chances are, if it has not been used already by the b's operating system, but it was listed as used in the PRG, it is used now for tape. I also re-use some system locations which should be non-conflicting with tape operations. At any rate, I have not found tape code to clobber any needed locations. Anyone finding such is encouraged to write to me. No 'unused, unlabelled' locations in the system pages are used, so if you've borrowed $fa and $fb for a m/l operation of your own, I won't guarantee compatibility, but I won't overwrite your code.

Tape hookup to the kernal is completely automated, basic loads are automatically re-located to the proper place, lines re-chained, and ready to be edited, even if the program came from a p/v/c. In other words, the thing works just like it should have, instead of mindlessly blabbing '?device not present' in response to every command referring to cassette.

From the monitor, you can 'l' or 's' using device #01 [remember to use 01 not 1], from-to any long address [bank number and address, with the exception of $0000 and $0001. A bit of thought shows that loading either of those locations during 'l' can take you to lunch city. Don't ask how much thought.

Experience may be the best teacher, but she's often late for school.

My basic loader bloads the m/l, and automatically connects same, and you have tape functions in your b, in as user-friendly a manner as commodore ever intended. Just follow the instructions in the user's guide that they sent us and you won't go wrong, [at least not any more!]

The m/l hooks up only with the kernal (that is, the basic command to load does some of it's own preparatory work and then turns to the kernal to do the 'guts' of the work), so although I HAVE NOT TESTED TAPE CODE WITH A B/256, I believe it should work. Both machines share the same kernal rom. However, since I am not receiving royalties for this, do not fairly expect me to make it work on a machine which I don't have, should there be a bug in regards to the b/256.

One minor point which deserves your attention before you rush off, pen in hand to order this language extension from cbug, you WILL need a ram expansion cartridge ala Anderson or Deal/Calc result. For my convienience, the code resides in bank 15, $6000 - $6aff. If the numbers don't mean anything to you, don't worry, you don't need to know them, just shift/run, answer the prompt question, and you are connected.

As a further point of interest to advanced m/l fanatics out there, I promise that this is the last m/l project that I will write using my own mnemonic system. I am now sufficiently impressed with the 650X chip's possibilities to learn the assembler mnemonics that everyone else in the whole world uses, and to those of you who bought my 'goceliak's patch' disk, please scratch the file entitled 'my dis no zpage'. I will no longer be using that mnemonic system, and that should leave a net number of 0 of us.

As information to those who need it, the tape buffer is placed by the b's own kernal allocation routine at high memory, and 'normally' lives in bank #2. I recently had occasion to open up my b to fix a faulty memory chip that had annoyed me in bank 2, confusing bit #4 of only two locations in the whole bank, but with the diagnostic program for the b, it said chip u67, so while replacing that, I added banks 3 and 4. Kernal smart, basic dumb, my tape buffer now automatically moved to bank 4! All internal pointers(are to the best of my knowledge, properly done, so that you can generate and garbage out strings, re-define function keys, etc, etc, and tape code isn't even fazed

Now down to brass tacks. If you want to use this code you need:

1. b series computer. [you've got one]

2. plug-in memory in bank 15 covering at least $6000 - $6aff [Mr. Anderson's plug in cartridge or the Calc Result cartridge modification described by Ms. Deal on cbug disk #2 work equally well.]

3. a 'datasette' recorder, exactly the same as was sold by commodore and many other manufacturers for the [p/v/c] computers. Mine is a commodore model C2N 1530 datasette unit, sold as surplus about a year ago. This model is great, but definitely not the only one that will work.

4. patience. Tape code is slow, but reliable.

Enjoy the new-found compatibility with other commodore computers, and the use for that #$%²&*()! cassette connector on your b!

Anthony J. Goceliak
32 Cottage Street
Jersey City N. J. 07306

TAPE DRIVE CODE FOR THE B128

By: Mr. Anthony Goceliak

<<One more item of general interest which is properly library material>>

The tape code presented here has been 'liberally inspired' by the 500 series b-computer. As such, it will be distributed to CBUG members only under the terms of the CBM / CBUG lisence agreement. Since the code is specific to the b-128 series computer, and attempting to adopt it for use on other computers is either needless redundance or impossible due to hardware differences, we should have no difficulty in 'keeping to the letter of our agreement'.

Do not distribute this code to any non-CBUG member, or it may become actionable

Read the BLUE user's guide that came with your B-128 for an excellent description of how to load and save basic programs. My prompt messages differ slightly from those in the book, but mean the same thirg (it IS my code!)

Things to remember / known variations from commodore tape standards:

1. You may use pattern matching file names with tape. [i.e. load'???xyz*']

1a. Only 16 character filenames please! otherwise tape would need a separate filename buffer instead of using the b's own buffer. No error, but I won't warn you again.

2. You don't have to use a name at all with tape. [i.e. save]

3. tape equivalent of a disk seq file is Generated by open1,1,1, filename'

4. if 'filename' is to be the LAST THING ON THE TAPE, open1,1,2,'filename' this creates an 'end of tape'. In load, verify, or read, tape stops here

5. Read tape data files by using open1,1,0,'filename' Zero is mandatory!

6. all keys EXCEPT stop are de-activated during tape operations.

7. kernal tape operations 'l' and 's' using dev #01 and LONG ADDRESS work.

8. when loading/verifying actually begins, the 'bell' will sound.

9. tape is very slow. expect about 5 seconds per disk sized block to load.

A. if you change load vector to ram, do it after sys'ing this code, or crash!

This tape code has been brought to you by A. J. Goceliak, who is NOT paid by CBUG [or anyone else for that matter] for his efforts. If you use this code, and find a really neat program that you can now load to your b-128, send him and CBUG a copy.

COMMENTS ON CBUG CONTESTS

By: Mr. Anthony J. Goceliak

OK Norm, I've got a bone to pick with you. That contest of 'spot-a-name' sort of stuff. Now I'm not gonna enter, and I'll bet a whole bunch of other CBUG members aren't either.

Want to know why? I've been getting letters from people asking about things in the fall '86 Escape for over a week now, and judging from the earliest postmarks on the letters, your estimation of delivery dates wasn't so far off. Except of course for mine, and only the BIG G and the post office know how many others. A contest in which everybody else gets a two week headstart over me and only the first umpty ump correct replies win a 'top' prize doesn't seem to gibe with my idea of fairness.

You wanted to know when we got our escapes? I got mine on 7 JAN 87, and there were three letters postmarked last year from people referring to this issue. Admittedly we are in a privileged position calendar wise, but let's get organized on this stuff, ok?

Want to run a contest and increase orders? ok, no quarrels with that, but either give everybody who comes up with the right answer the same discount/freebie or whatever, or conduct a random drawing supervised by your wonderfully patient wife or offspring from all correct replies to determine the winner.

Or, let everybody get the same start. This section will need revision or your spouse will seek revenge, but how about withholding the particulars of the proposed contest until some reasonable date which allows all members to have their escape in hand, and then let everyone call a particular telephone answering machine which will give the specifics of the contest. (For example, add the first numbers appearing on each page and submit the total).

Or thirdly, how about offering the prize to who(m?)ever can solve the 'no-key-crypt' text problem on my goceliak's patch disk [the indefeasible title]. I'll enter that one!

Seriously, it was a good try, but you should really set the rules somewhat more equally for those of us who live in the areas served by illiterate, immobile or otherwise ineffectual mail carriers. [lmn!]

A SECRET B-128 KEY!

By: Mr. Anthony J. Goceliak

Have you ever wondered why Commodore installed a 'tab' key on the b-128? I sure have, and doggone if I haven't finally, after all this time figured out why!

CBM installed, hooked up and coded an extra-big key on your keyboard, and even assigned it a function that you can use without resorting to Superscript or some of my programs where I have used 'tab' as a 'specially defined by me key'. That function is, can you stand it, a tab function! Well no wonder the cursor flies to the right hand edge of the screen if you press it while in basic direct mode, it's looking for (but not finding) a tab to stop at.

No, I'm not talking about a command like print tab(30);"hello"

Try this on for size:

1. In basic, no program running, ready prompt and flashing cursor on screen.
2. Move cursor somewhere across the screen, say to the 20th position.
3. Press and hold 'shift' while pressing 'tab'. Release both.
4. Move cursor leftward of the tab you just set, whether by pressing return, cursor left, or whatever you desire.
5. Press 'tab' <u>without</u> pressing shift.
6. Wow!

In summary, shift/tab sets or clears a tab by toggling on or off <u>at the position of the cursor</u>, and tab alone is the 'search for previously set tabs' key.

Commodore did it again. Perhaps many of you already knew what the tab key did, but I didn't, and I just re-read the user's guide, and it didn't have much to say about this key. Just like the disk drive, good ol' commodore has equipped our machine with more functions than they cared to talk about.

The books never were much on accuracy anyway, consider how many of the 'ctrl' codes listed do what is claimed for them, but I always enjoy finding something new in my b's repertoire, and I am not exactly a novice, having used the same machine for well over a year.

What do you do with a key like this? Ever want to write neat & tidy data statements while writing a basic program? Or much much better, by a few simple pokes to bank 15 memory, you can 'pre-install' tabs and then when your program has a line like:
100 print chr$(9) [that's the tab code], guess what happens? Right! you tab over to the next set tab!

While there is probably a more elegant method of doing this, here is a rough and ready way to make the b duplicate a tab set-up.

1. In direct mode, set however many tabs you wish at whatever positions you desire.
2. Next type the following:

```
bank15
?peek(929)
?peek(930)
    ''
    ''
    ''
?peek(938)
```

Somewhere near the beginning of your program, say line 10:

```
10 bank15
20 poke929,[put the number that peek 929 was here]
30 poke930,[ditto for peek 930]
```

and so on.

You can automate the process by a loop, reading and poking from a list of data statements, or do whatever suits you.

For instance, if you have already set your tabs, the following simple basic program will pick them up and display them for you almost ready for inclusion in a data statement for your real program.

```
10 bank15: print "1000 data";
20 for x=0 to 9
30 ?peek(929+x);:if x<9 then print " , ";
40 next x
```

This will generate a display of your set tabs which you can make into a real line by positioning the cursor over it and pressing return.

Of course you will want to change the line number from 1000 to whatever you require, and also delete lines 10 - 40!

Again, in the manner of not pretty, but it works, you can type new after running the program above, and then dloading the program you wish to use the tab set-up in, and only then positioning the cursor on the data statement line and re-saving the program.

Next time I'll tell you about printing a few curaor up's, and poking a 13 into the keyboard buffer and a 1 into the number of keystrokes waiting location, which completes the automation of saving a tab set-up.

I have a good idea that there is a whole lot more 'unpublished functions' which the b can do, and if any of you out there have found one (or more), why not write 'em up and send 'em in?

A TRULY SECRET KEY!

By: Mr. Anthony J. Goceliak

During a casual telephone conversation with one of CBUG's pre-eminent programmer's, I mentioned that there was a real use for the escape z [alternate character set] function which had escaped(!) most people's notice.

If you wish to have a user enter a pass-word, or some short but sensitive data, which should be concealed from a casual onlooker, just preceed the input statement with a line such as:

```
250 print chr$(27)+"z"
```

chr$(27) is if course basic long-hand for the escape key. You will get a little procession of blank blocks marching across the screen as anyone types and the program echoes to screen. Just don't forget two things!

1. clear the screen before returning to the normal character set.
2. return to the normal character set!

```
270 print chr$(27)+"y"
```

Also keep in mind that a less-than casual observer can watch your fingers instead of the screen and still find out what you typed, so keep alert.

B-128 MONITOR and THE STACK

By: Mr. Anthony J. Goceliak

While I was de-bugging my tape code, I had frequent occasion to go from basic to the monitor and back. Are

THE CBUG LIBRARY

This issue of CBUG's LIBRARY is shorter than usual. But it offers many excellent new programs of outstanding quality and usefulness. Several contributors have done remarkable jobs of explaining their offerings, also. To that end, I'd like to make a few requests of our contributors.

When making contributions to Library, directory and anotated directories should be in a format we can quickly transfer into our printing files. Our new standard is 58 columns wide, which means the carriage return symbol can be in column 59 and none higher. With this we can then move your text into two columns as appropriate.

The preliminary comments about a disk or submission should be without carriage returns at the end of each line as we will reformat to 120 column wide layout for those components. CR's go only at the end of paragraphs in the general commentary area.

Always label the face of the disk with your NAME, ADDRESS, ZIP, DATE, and PHONE NUMBER. Insert in the sleeve any information you think CBUG should initially see. Name address, date, and phone should be affixed to the disk, not on a loose sheet.

Several well meaning members have submitted disks with other members' materials. This is fine and helpful IF they are public domain. Programs bearing copyright notices, Freeware notices, or were listed in The CBUG LIBRARY at prices greater than $9.00 can not be re-submitted, even if repaired, etc. without a release by the author. While you can not publish copyrighted code, you can print corrections to be made by specifying line number and your new code for that line or portion of a line. Submission to author is another approach. Several members have submitted disks with their favorite programs compiled on the disk, however, we can not publish without each copyright owner's permission. Please do not challenge CBUG to find those which have been improperly included. If anyone sees a copyrighted item being republished, please send us a note so we can expunge the offending item.

When submitting articles, kindly use SSII. If you must use SS3, do not use the formatting capabilities of SS3 as most will not work with SS2.

CBUG has the capability of reading and transffering to and from the common CBM serial format disk drives such as 1541 and 4040. Let us know if your input materials are of these other formats. We can not offer library materials in these other formats for obvious reasons.

Please observe that much of our library material is offered either on a royalty basis (fees paid) or Freeware Basis (where the consideration is owed by the consumer upon acceptance of a product). Please observe and remit accordingly so that we can continue to bring you these author's continued work.

Do observe copyrighted programs and never give or sell copies to anyone. This is all on the Honor System and those who have acquired copies without proper payment are merely serving to destroy the services CBUG can provide.

CBUG MISC. M54 NEW RELEASE 11930 $9.00

This is a general use disk of materials from 5 contributors.

* *

The first author is John Wright.

<NOTE: the program ' LIBRARY ' is a Superbase application with a modest database built in. Load Superbase then se this disk as the data disk, or as with all databases, you would be better off copying the database and its related iles to another disc, as this one is almost totally full!>>

Most of the programs on this disk were modified from previous CBUB releases. The major modification was to allow use of drive 1 on the 8050 disk drive. Liz Deal's Key Trix has been extensively used to renumber program lines, so most of the changed/added lines may not be the same as in the original programs. This file is written in Superscript II. The only "Formating" commands used are *fp0!

brief: Categorization of items on this disk.

by: John A. Wright
818 Juniper Dr.
Papillion Ne.,68046
Phone(s): [talk] 402-339-5729, [modem] 402-339-8918

The contents of this disk are of two general types:

1. A group of four utility programs that have been modified
to allow either drive 0 or drive 1 selection.

2. A Superbase file that I use to "catalog" magazine articles

Files that I have modified will include the original author.

brief: Original program name:"Boot v2"
Modified program name: "Autoboot"
Written by J.A. Wright to auto load the unpotected versions of Superbase and Superscript. Superbase auto load has problems due to problems with Casey's Scrubber (CBUG #28) Simple little program to auto load these two programs. I use it on all my disks where I am creating a lot of Superscript files

brief: Original program name: "f-key template"
Modified to allow use of either disk drive
Menu driven

brief: Original Program name:unknown (I may have written this one and forgotten it)
Modified Program Name."Copy File"
From CBUG #
Program written by:

Simple little menu driven program that copies files from one drive to another or to the same drive.

brief Original Program name:Seq Read/Print
Modified Program Name: (same)
From CBUG #4
Program written by:Roger B. Fry
Modified by: Kernighan
Modified by: J. Wright

Modified to allow drive 1 or 0

brief: Library Pgm section is a package to allow the user to keep a file of magazine articles or books. Written mainly for use as a cross reference for magazines.

by: J.A. Wright
818 Juniper Dr.
Papillion,Ne., 68046
Phone (talk) 402-339-5728 (modem) 402-339-8918

Modified program name: (same)
From CBUG pre release #4
Program written by Dick Wilkinson

Modified to: Print in capital letters instead of "Graphic" characters. Allow user to select either drive 1 or 0. Allow user to go to next key by simply pressing "n". Thus removing the requirement to use "shifted spaces" for every key. And to allow the user to quit the "input" stage by typing "finish" to exit

Instructions:
Follow the instruction from Pre-release #4. When you desire to proceed to the next key without entering a value for the present key simply press the "n" key. When complete type "finish" to get back to the menu.

brief: original program name:"Unscratch"
Modified Program name: "File unscratch"
From CBUG #4
Program written by:Luke Mester
Modified by: Kernaghan
Further modified by: J. Wright

Completely menu driven. Programs will direct input of new records or a search for articles by subject and will return the magazine issue/date and page number of the article. Look at the program "search.p" and the "Format" to gain a better understanding.

```
1    "-----jw-----"     seq  12   "seq read/print"   prg
1    ".."               seq  1    "fill 16"          prg
2    "ss.autoboot"      seq  1    "fill 17"          prg
1    "."                seq  1    "fill 18"          prg
22   "jw comments"      seq  1    "fill 19"          prg
2    "autoboot.p"       prg  1    "--LIBRARY--"      prg
1    "fill 7"           prg  7    "start.p"          seq
1    "fill 8"           prg  3    "magazine"         seq
1    "fill 9"           prg  40   "superspell.tm"    seq
1    "fill 10"          prg  27   "!!--NOTICE--!!"   seq
1    "---UTILITIES---"  prg  1    "LIBRARY"files     seq
31   "f-key template"   prg  3    "magazines"        seq
6    "file unscratch"   prg  11   "search.p"         seq
4    "copy file"        prg  1799 blocks free.
```

* *

by: Matnew Goldstein

' Mastermenu v1.3' is a full-featured <shift-run> automatically configured 1, 2 or 3 column disk directory that loads and runs (for prg files) or displays (for seq files) files selected by assigned menu number. It will run on both disk drives, includes print/copy/delete/rename utilities and belongs on every disk that contains more than a few different files. checkbook v2.3 is a complete, flexible, carefully-crafted, user-friendly and bug-free program that minimizes data-input drudgery and keeps a neat easily-accessible record of your checking account. The numerical methods programs are based on a textbook used in an undergraduate computer course that Mathew Goldstein completed this past fall semester. The non-executing program identifies the lines of code in BASIC programs that are not utilized by the computer. The program was downloaded from a C64 database on Delphi. The typist v2 program is an improvement on the program on the CBUG #32 disk.

```
36   " Mastermenu v1.3" prg <shift-run> for every disk w/     11   "multipleroots.mr" prg
20   " Mastermenu/c1.3" prg  more than 1 file. v1.3 shows     8    "newtonraphson.mr" prg
8    "Masterdirectory"  rel   filename of selected file.      9    "secant.mr"        prg
1    "--------------u+" seq                                   12   "gauss.ms"        prg .ms solves systems of equations
1    "---utilities----" seq                                   11   "gauss-jordan.ms"  prg
1    "--------------u-" seq                                   9    "gauss-seidel.ms"  prg
8    "non-executing"    prg identifies non-exec. BASIC        23   "cubicspline.mc"   prg .mc fits curves to data
1    "--------------a+" seq          lines                    8    "lagrangeintrp.mc" prg
1    "--applications--" seq                                   16   "multiregress.mc"  prg
1    "--------------a-" seq                                   8    "newtonintrp.mc"   prg
6    "checkbook v2.3"   prg Complete checking program w/      20   "polyregress.mc"   prg
134  "%checkbook v2.3"  prg bank reconciliations. Delete/     6    "gaussquadture.mi" prg .mi integrates
2    "+bank.ram"        prg insert/move/sort/edit entries     9    "romberg.mi"       prg
2    "check.sample"     seq  Fool-proof data input, user-     15   "simpson's.mi"     prg
3    "checkcatagory1"   seq    friendly.  Guaranteed no       7    "trapezoidal.mi"   prg
2    "checkcatagory2"   seq    bad bugs.  Create your own     12   "runga-kutta.md"   prg .md solves differential eq.
1    "checkcatagory4"   seq    checkcatagory files.           1    "--------------i+" seq
1    "checkcatagory3"   seq    Search/print by catagory       1    "--instructions--" seq
13   "typist v2"        prg faster w/moving col. indicator    1    "--------------i-" seq
1    "--------------m+" seq                                   43   " Master1.3.instr" seq
1    "--mathematics---" seq accurate numerical methods        8    "annotation.mg5"   seq
1    "--------------m-" seq                                   18   "checkbk2.3.instr" seq
8    "bisection.mr"     prg .mr finds roots of equations      15   "numerical.instr"  seq
10   "falseposition.mr" prg                                   4    "blurb.mg"         seq
7    "iteration.mr"     prg                                   2    "FROM.MG"          seq
```

* *

by: Neil Cumfer

"Graphics Finale" winds up the instructional series showing you how to teach SuperScript II to print graphics on the 4023. The programs it explains are already typed in for you in "finalgrafxtables" and "ss2 tables"final. "4023graphicsdemo" shows you how easy it is to dazzle your friends and impress your boss with horizontal and vertical graphs. Speed readers will love "veryfastreader", a sequential file reader that can display a 64K file to the screen in just 6 seconds, thanks to an imbedded machine language routine. Also useful for slowpokes; read files that you aren't going to change, without having to wait for SuperScript to load. Scroll forward or backward by the screenful (similar to SS's ESC SPACE and ESC SHIFT-SPACE).

```
12 "veryfastreader"  prg Basic: sequential file reader         2  "ss2 tables"final prg binary: output of above program
                         see 64K in 6 sec                      14 "4023graphicsdemo"seq SS: print out this file from SS
51 "graphics finale" seq SS: part 3, 4023 graphics with                                 to see graphics
                         SuperScript II                        4  "blurb.nc2"       seq SS: wild claims for these files
29 "finalgrafxtables"prg Basic: revision of SSII's             3  "annotations.nc2" seq SS: this file
                         "ss2 maketables"
```

* *

By: David Greene

The following is a short explanation of some of the programs on the disk. I have written most of these programs myself; the exceptions are the six text adventures and the renumbering routine. The programs listed are something of a miscellaneous collection; there are a quite a number of games and several utilities as well. They are separated on the disk directory by null programs.

"run me.dg" is a program to read Superscript files like this one without accessing Superscript.

"adventures" is a loader program for the eight adventure games.

"collab" is a three-dimensional maze game. It has two floors, each one being a 25x25 maze grid. There are a number of pits, elevators, teleport squares, objects, monsters, and other surprises hidden in various places. Making a map is advisable, but it will still take time to figure out how to escape.

"abrlab" is an abridged version of the above; everything in the larger maze is carried over to this version, but the maze is significantly smaller (only a 15x15 grid). This makes it less of a challenge but easier to explore and map.

"3d tictactoe" is a game of 4x4x4 tictactoe; you can play against the computer or another human player.

"roadrace.dg" is a simple roadrace game; you control a car speeding down a road full of potholes. There is a finish line; its distance from START is determined by the skill level you choose.

"snakes.dg" is a game where you control a snake and try to avoid obstacles.

I got the following adventure programs from a games disk put out by BUG; I improved them considerably, adding upper/lower case and correcting errors.

"haunted.adv" is an adventure game in which you must explore a haunted house and bring out valuables and treasures.

"pirate.adv" is a terribly slow adventure where you labor to build a pirate ship and sail the seas to Treasure Island.

"cavern.adv" is a rather uninspired offshoot of the original Adventure.

"sorcerer.adv" is also based on the original Adventure somewhat; however, it is original to a large extent and contains a few humorous touches.

"dogstar.adv" is a Star Wars adventure in which you must rescue Princess Leia.

"wisp.adv" is my personal favorite; it is fast and clever, though actually more of a spoof on adventures than a real adventure game.

"bigmaze.dg" is a simple maze game on a large grid (99x99). This program is mostly an demonstration of what it is possible to do with strings; it does not make a very interesting game by itself.

"mouse.dg" is a program which allows you to create mazes and run simulated "mice" through them. After the first run-through, you can have the mouse take the fastest path to the exit.

"taxman.dg" is an "educational game" involving factors.

"mazegen.redraw" is a maze generator that quickly and repeatedly redraws mazes. For all the "mazegen" programs, it is possible to control the direction the generator takes. Use (A) for up, (Z) for down, and < and > for left + right.

"mazegen2" draws mazes on a more condensed grid.

"mazegen3" is the same as "mazegen2", but with a different type of graphics.

"chase1" a rather abstract game program. The player is represented by a solid graphics block. He must maneuver (using (A), (Z), <, and >) and try to avoid monsters. There are obstacles -- land mines -- scattered around the field, and if either a monster or the player runs into one of these, it explodes. The object is to get rid of all the monsters by making them run into mines. "chase2" is a version of "chase" where the obstacles do not explode; the object is to grab a number of '$$'s which are scattered around. If two monsters run into each other, they will explode. Again, the player must avoid the monsters. push the obstacles around. The monsters can be killed by pushing objects into them, but every time this is done, the player must grab a '$$'. The object of this game is to kill all the monsters and also grab all the '$$'s.

"renumber.dg" is a utility which will renumber BASIC programs.

"hres editor.dg" allows you to draw pictures or designs and then print them out to a CBM 4023 printer. An entire screen comes out about two inches square.

"screen editor.dg" lets you design screen layouts for use in other programs.

"drawprint.dg" lets you draw on the screen and print the results on a printer in high resolution. This is a simpler version of the highres editor; it has fewer features, but is easier to use.

"diagprint.dg" prints out on the printer an N-sided polygon with all diagonals connected. This is a demonstration of my high-res SET/RESET/TEST/LINE drawing routines; you can remove the demo program and use the subroutines.

"graphics.dg" is a set of subroutines for onscreen, 80x50 SET, RESET, AND TEST.

"graphics2.dg" is the same as "graphics.dg", but with a matrix of 160x50.

Any programs that require controlling something should accept the following controls:

(A) key moves up,
(Z) key moves down,
(<) [comma] key moves left, and
(>) [period] key moves right.

If (Heaven forbid) you run into any problems with the programs and can not figure out how to fix them, you could write me:

David Greene
RR 1 Box 63 River Road
North Creek, NY 12853

I hope you can find something useful or entertaining in this collection.

2	"run me.dg"	prg	if you don't have Superscript
3	"adventures"	prg	loads Adventure programs
1	"-----games-----"	prg	(marker only)
158	"collab"	prg	Colossal Labyrinth
144	"abrlab"	prg	Abridged version of Colossal
36	"3d tictactoe"	prg	4x4x4 tictactoe player
13	"roadrace.dg"	prg	Avoid the potholes.
16	"snakes.dg"	prg	Dodge obstacles.
85	"haunted.adv"	prg	Haunted House adventure
72	"pirate.adv"	prg	Pirates' Cove adventure
67	"cavern.adv"	prg	Cavern of Riches adventure
60	"sorcerer.adv"	prg	Sorcerer's Castle adventure
64	"dogstar.adv"	prg	Dogstar (Star Wars) adventure
119	"wisp.adv"	prg	Will O' the Wisp adventure
6	"bigmaze.dg"	prg	demo of string uses
20	"mouse.dg"	prg	maze and mouse program
11	"taxman.dg"	prg	number game
4	"mazegen.redraw"	prg	maze generating program 79x49
4	"mazegen2"	prg	maze generating 159x49
4	"mazegen3"	prg	maze generating 159x49
6	"chase1"	prg	simple dodge-the-monsters game
7	"chase2"	prg	dodge monsters and grab $$
7	"chase3"	prg	push blocks and grab $$
1	"---utilities---"	prg	(marker only)
6	"renumber.dg"	prg	renumbering utility
18	"hres editor.dg"	prg	high-res graphics editor
5	"screen editor.dg"	pr	save screens for use elsewhere
7	"drawprint.dg"	pr	draw things and print high-res
4	"diagprint.dg"	prg	n-sided polygons high-res
3	"graphics.dg"	prg	set/reset/test routines 80x50
3	"graphics2.dg"	prg	set/reset/test routines 160x50
1	"from.dg"	seq	Superscript: name + address

"chase3" is yet another version; here, the player can

9	"adirectory.dg"	seq	what you are reading now
22	"readme.ss"	seq	blurb file on these programs

* *

by: Joe Rotello

See Joe's articles in this issue for more information on these programs.

12	"data entry.notes"	seq	4	"data entry a 1.2"	prg	3	"light.menu"	prg	3	"box.menu.notes"	seq
5	"ss2 brack.notes"	seq	3	"b128.dir.notes"	seq	10	"light.menu.notes"	seq	3	"master.box"	prg
4	"data entry c 1.2"	prg	7	"b128.dir"	prg	4	"box.menu"	prg	8	"master.box.notes"	seq

* *

118 blocks free

CBUG MISC. M55 NEW RELEASE 11944 $9.00

The first two items deal in training in ML, and some very interesting bank switching discoveries

* *

The first author is Dr. Gary Swindler

list and study these basic programs for your instruction. They show what needs to be done to get the bank ready for this or any ML program that needs to coexist with basic. If the ML code is used from direct mode none of this is necessary.

7	"rl1"	prg	43	"instruction.file"	prg	4	"qlist.3000"	prg	10	"don'tlikekeys"	prg
7	"rl2"	prg	3	"xboth.fe00"	prg	7	"hotel cal.file"	prg	4	"qlist.8000"	prg
									4	"qlist.0800"	prg

* *

By: John Berezinski

Please read the two articles in the main body of this issue for the details on these contributions.

4	"bankall4096"	prg	4	"bankall16384"	prg	41	"hexprimer.s"	seq	2	"rpn8192"	prg
4	"bankall8192"	prg	4	"bankall24576"	prg	3	"rpn1024"	prg	7	"blackbox"	prg
						2	"rpn2048"	prg	27	"info"	seq

* *

By: Rev. Doug Gorsuch

This series was appended to this disk just for a place to put it as the general topic disk is full!
An additional copy of this same material will be appended to Mr. Goceliak's disk being released this issue.

Please read the article in the main body of this issue for the details on the contribution.

1	"SS2 CALENDAR"	seq	17	"30/mon"	seq	17	"31/wed"	seq	17	"31/fri"	seq
11	"Instructions"	seq	17	"29/mon"	seq	17	"30/wed"	seq	17	"30/fri"	seq
17	"31/sun"	seq	17	"28/mon"	seq	17	"29/wed"	seq	17	"29/fri"	seq
17	"30/sun"	seq	17	"31/tue"	seq	17	"28/wed"	seq	17	"28/fri"	seq
17	"29/sun"	seq	17	"30/tue"	seq	17	"31/thu"	seq	17	"31/sat"	seq
17	"28/sun"	seq	17	"29/tue"	seq	17	"30/thu"	seq	17	"30/sat"	seq
17	"31/mon"	seq	17	"28/tue"	seq	17	"29/thu"	seq	17	"29/sat"	seq
						17	"28/thu"	seq	17	"28/sat"	seq

HARRISON'S ASSEMBER 5.5 v8 CBUG 56 11959 $35.00

This is the latest version of Harrison's Assembler 5.5, upgraded to v8. If you wish to upgrade, you may attach your existing disk label to the order form and remit only $5.00 for the upgrade version.

5	"b128load"	prg	2	"reload.rel"	seq	3	"example.src"	seq	12	"birqdoc.src"	prg
33	"assembler5.5"	prg	2	"kernal.lib"	prg	6	"backup"	prg	2	"printdoc.ml"	prg
7	"assembler5.5.rel"	seq	10	"config.src"	prg	5	"birq"	prg	2	"printdoc.src"	seq
5	"reload"	prg	134	"asmdoc.src"	seq	2	"birq.rel"	seq	1822	blocks free	

GOCELIAK STRIKES AGAIN CBUG 57 NEW RELEASE 11963 $9.00

```
PPPPPP     AAA     SSSSSS                BBBBBB     AAA     SSSSSS
PP    PP   AA AA   SSS                   BB    BB   AA AA   SSS
PP   PPP  AA    AA  SSS                  BB   BBB  AA    AA  SSS
PP PPP    AAaaaAA    SSSS                BB BBB    AAaaaAA    SSSS
PP        AA    AA      SSS              BB   BBB  AA    AA      SSS
PP        AA    AA        SSS  _______   BB    BB  AA    AA        SSS
PP        AA    AA  SSSSSS     _______   BBBBBB    AA    AA  SSSSSS

█████████   █████████████   █████████   ██████████████
    █       █                   █              █
    █       █████████████       █              █
    █                   █       █              █
█████████   ████████████    █████████          █
```

PAS_BAS 1.0 tutor/translator is all you need to learn and use the powerful PASCAL programing language.

An interactive on-line tutorial gives you a self paced instruction on the standard UCSD PASCAL. It covers everything, from first principles to advanced techniques, from using variables to creating nested and recursive functions. Periodic quizzes test your expanding knowledge, and an efficient translator allows you to balance theory and practice.

The PASCAL to BASIC translator is a first of it's kind software writing tool. It makes a dynamic analysis of a PASCAL program and generates a compact and efficient equivalent program in BASIC. The translator covers a wide subset of UCSD PASCAL (Including recursion!). The BASIC code generated by PAS_BAS is 100% ready to run in any personal computer with a BASIC interpreter (Most computers), and is fully labeled so it can be compared with it's PASCAL source code.

You can write PASCAL programs in your Superscript II, or with the PAS_BAS easy to use integrated editor.

PAS_BAS was born and developed in a B-128 (THE BEST!) and will soon be available for other computers. Future versions will be announced in THE CBUG's ESCAPE.

ORDER YOURS NOW!   PAS_BAS ISIT!

International Software for Information and Translation

```
+-----------------------------------------------+
!Check_ Money Order_ Quantity_ Total$________   !
!                                               !
!NAME___________________________________________!
!ADDRESS________________________________________!
!CITY______________________STATE________ZIP_____!
+-----------------------------------------------+
```

PAS_BAS is copyrighted 1987 by Witte & de la Torre. To order fill your form and send $35.00+3.00 shipping to

I S I T
P.O. Box 773406
Houston , Texas 77215-3406

COMPUTER SPECIALTIES, INC.

77Ø5 TECHNOLOGY DRIVE
MELBOURNE, FLORIDA 329Ø4
3Ø5-725-6574

CSI's New Building

This is a picture of CSI's 14,000 square feet facilities into which we moved in March 1986.

This building houses:
- Sales Showroom
- Engineering Lab
- Service and Technical Support
- Manufacturing
- Training Center Class Rooms
- Corporate Offices
- Warehousing

CSI services NEC, IBM, Clones and other products in addition to Commodore products.

WHEN THE GOING (and Servicing) GETS TOUGH . . . CSI will assist you. That's right, when all else fails . . . Call CSI, and either speak to Roman for parts or ship it to us for repair.

We're SOOO easy to reach.

Direct Line to Offices & Service/Parts (305) 725-6574

Direct Sales Lines:

National	800 523-8275
Florida	800 341-4368
Local	(305) 725-6574
Minimum orders	$25.00
Minimum ship/handling	$5.00

SPECIAL NOTE re: Commodore

We stock a huge inventory of Commodore Repair Parts, and all are at low prices. Our Normal shipping is SAME DAY via UPS.

To place orders . . telephone our toll-free numbers anytime between 7a.m. and 7p.m. Eastern Time. Our account managers are eagerly awaiting to hear from you. Remember, we provide "same day shipping" on all orders received by 1:00 p.m.

WE HAVE THE LAST OF THE 64ØØ LETTER QUALITY PRINTERS AT $519.ØØ EACH. ONLY NINE LEFT AT PRESSTIME

ALSO AVAILABLE ARE P-I AND I-I CABLES AT $27.95 EACH.

CSI'S IEEE TO CENTRONIC INTERFACE IS ONLY $61.98

CSI BUYS AND SELLS ALL COMMODORE COMPUTERS AND PERIPHERALS

◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆

SERVICE CENTER:

TYCOM INC
503 East Street
Pittsfield, MA 01201
(413) 442-9771

Authorized Commodore service center, we repair B128, 8032, 4023, 8023, 6400, 4040, 8050, 8250, 2031, C-64, C-128, etc. Rates are $50/Hr plus parts. We will diagnose & quote repair cost for $25. Normal turn-around time is 5 business days. We ship via UPS. Limited supply 8032's & 4023's (new) at $189 each.

◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆

B-1024 1 MEGABYTE MEMORY EXPANSION CIRCUIT BOARD for the Low Profile B-128! Experience the full memory capability of your low profile "B" by installing Banks 0-14, Socketed 256K DRAM, Pin Fields for future I/O, Static RAM for $0800-$1FFF in BANK 15, Mounts Internally in 5 Min, Assembled and Tested, ML Memory Test Utility on 8050 Diskette, Documentation, Schematic Diagram

B-1024 1 MEGABYTE MEMORY EXPANSION BOARD	$349.00
US Funds, Iowa add 4%, Shipping/Handling	$ 10.00

ANDERSON
COMMUNICATIONS
ENGINEERING
2560 Glass Rd. NE.
Cedar Rapids, Iowa
U.S.A. 52402

24K RAM/ROM CARTRIDGE with Case for the B-Series:

Assembled+Tested (Socketed RAM Memory ICs)	$39.95
Bare Circuit Board and Case	$14.95
US Funds, Iowa add 4%, Shipping/Handling	$ 3.00

A NEW DATABASE FOR THE B-128

Mike Konshak's dfile DBMS for C-128 is now available for the B-128

Readers of RUN magazine will know Mike as a regular contributor, especially for the "Datafile" series of programs published in several installments beginning in November, 1984. In the nearly two years since then, it has evolved from a comparatively simple program to a full featured database for the C-128 computer. Now, with appropriate changes, it is released for the B-128 with a licensing agreement to Chicago B128 User's Group - International. Distribution initially is limited to the U.S. only.

The newest version, now sold commercially, is "dfile128". This program is the result of input from thousands of users of Mike's earlier programs. It is highly recommended in the Midnite Software Gazette.

Being memory-based, rather than keeping all the files on disk for processing, operation of this database is much faster than Superbase and similar databases. All processing takes place in memory, without waiting for disk access for each record. When entries or changes are completed, data is stored on disk in sequential files. This speed does have a trade-off: the amount of data that can be in memory is limited by memory size. However, most folks don't have an enormous number of records, and do desire speed and ease of use in preference to capacity and complexity. dfile, as an example, will allow an inventory with 8 columns to have 740 records. If that isn't enough, a program awaiting conversion for later issue can combine up to 30 compatible files into one large record file for printing reports, labels, etc.

dfile DBMS main menu options:

Create new file
Print records
View file on screen
Sort records by field
Disk commands
Print records

Add record to current file
Modify record in current file
Delete record in current file
Read old file from disk
Write new file to disk

dfile for B-128 includes provision for IEEE interfaced printers as well as RS-232-C printers at both 300 and 1200 bps. Escape codes can be sent to printers to control modes and styles. Output can be unformatted, report format (set up as desired), mailing labels, or calculated reports that offer column totalling, averaging, and calculated fields such as in a spreadsheet.

The complete instruction manual is included on the disk. A commercially published booklet (for the C-128 version) is available from Mike Konshak for $5. Printer options are the most noticeable difference in manuals,

A companion disk of utility programs is expected to be offered. These programs allow manipulation and modification of your record files. They were created because they were requested by users of dfile128.

One program will merge files for those requiring files larger than can be stored in memory. It can merge up to 30 compatible record files into one huge record file for printing with a print file utility.

Clone file will clone or copy the data from one record file into a new record file with a totally different structure. You can even combine the data from several old fields into one of the new. The ability to change the structure as the need arises, without re-entering all the data, makes expansion or modification much more feasible.

Xport file will strip all non-essential structure data found in a dfile128 record file and leave just the records so that the files can be transferred to other databases or wordprocessors for form letters.

Conversion effort applied to these already existing utilities will be proportional to CBUG member reception of dfile.

ORDER FROM THE CBUG LIBRARY, CBUG #47, stock #11856

A HIGH-RESOLUTION BOARD is available FOR THE B128

The board allows bit-mapped graphics. The HR pictures can coexist with the normal text screen. The resolution is 1024 across by 512 down; to that area the B's screen is a window. Any size window for HR graphics can be created, and it can be moved independently of the text screen. The board includes 64K of RAM to support the bit graphics. Installation is simple: a six by five inch board goes inside the B, and one line needs to be cut. The board was designed in cooperation with the people who are building 1meg expansion, in order to be compatible with that project.

Software is in Basic at the present time. Machine code routines will be developed as an independent project, as soon as possible.

HI RES TECHNOLOGIES
16 English Ivy Way
Toronto Ontario M2H 3M4
phone: (416) 497-6493
The price is $199 U.S.

BLANK DISK SALE

Continuing the special offer from CBUG, Inc., and OPUS

Just correct or insert these items on the order form

5.25"	Pkg/10	item #
SSDD	$5.70	10017
DSDD	6.00	10021
DS4D	16.00	10197
3.5"		
DSDD	19.00	10088

OPUS QUALITY DISKS!

OPUS disks are without doubt the finest disks made. Mil spec forumulas and exotic materials, promise a wear life over 5,000,000 revolutions per single track with less than a 25% loss in signal output while delivering the highest initial output of any disk! Heavyweight jacket and lubricated antistatic liner insure the ultimate in stability. Of course there is a 100% factory warranty.

We are proud of our product. Each disk bears the OPUS corner label; you can be confident you are buying the real article -- no misrepresented no-name seconds. Each disk is factory inserted in its sleeve. A supply of labels + write protect tabs is encluded in each factory sealed package.

SSDD and DSDD 5.25" product come 10 in a poly bag; Quad Density are 10 in a display type vinyl library case. 3.5" micro disks are packaged 10 in a Flip n' File box.

BUY THE BEST -- OPUS!
THRU CBUG, PAY THE LEAST!

THE BEST DEAL

The Best Quality!

The Best Prices Too!

WHAT A DEAL! No kidding, we know from you, our members, and members of many other user's groups and commercial customers that the OPUS quality disks cannot be surpassed by any other manufacturer. In the course of our duplication on the 8Ø5Ø drives, which write 5Ø2K per side rather than the usual 17ØK to 34ØK for conventional Double Density usage, the Opus disk reject rate has been less than .1% -- one part per thousand. Phenominal! Most of the major brands I've tried have exhibited reject rates in the order of 3% to 5%!

- * 1ØØ% Certified
- * 1ØØ% Manufacturer and CBUG Warrantee
- * Proprietary high output oxide formula
- * Highly polished surface
- * Highest saturation figures
- * Special Heavy duty binder
- * Heavyweight jackets
- * 5,ØØØ,ØØØ rev/track life ++
- * Antistatic liner
- * Lubricated liner
- * Hub rings.

++"media life shall exceed 5.Ø million revolutions (278 hrs) with the head(s) loaded normally on a single track. During this period, the average signal amplitude shall not be less than 75% of the original value." The high output oxide insures that even at the 75% or original level, the Opus disk will exceed the output of most other brands of disks when new.

Why is OPUS so good? Opus is one of the very few "manufacturers" of disks which makes the entire product. Infact, OPUS is primarily a manufacturer of computer and military critical use instrumentation tape; they've taken the technology used in making those far more critical products and applied it to produce the finest disk in the business. But they are not just satisfied to make it right, they then check every single piece they manufacture -- a necessity in the world of critical data applications. Opus is one of the largest disk manufacturers in the US, but generaly caters to the ultra critical duplicator market, hence few consumers know of the brand.

We stock the complete line of OPUS disks: QUAD DENSITY, SS and DS, IBM AT, 3.5" and 8"; retail packaged thru duplicator no label bulk packaged. Inquire for volume pricing.

CBUG
THE CHICAGO B128
Users Group
4102 N. Odell
Norridge, Il. 60634 USA

◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆

KNIGHT'S COPY UTILITY

With the advent of programs to upgrade Superscript and Superbase, if you have not bought this most important utility, now is the time. Never ever do surgery on an only copy. Knight's copy utility will copy Superscript and Superbase easy as can be. All you need is an 8Ø5Ø drive with DOS 2.7. To find out if your drive is DOS 2.5 or 2.7, type as follows upon first turning on your computer and drive: ' ?ds$ RETURN '. The screen will then show drive and DOS type.

Running Knight's Copy Utility is very simple. Just load the program in the usual manner, remove the program disk and insert the original to be copied (must have a write protect tab on it), then insert a blank disk in drive 1. Ten minutes later you have a complete copy. The indicators on the drive flash at the end of the cycle if the copy is good. You can continue inserting blank disks as long as you want for additional copies!

Don't risk ruining your Superscript or Superbase disks. Order from CBUG, #122Ø4 only $2Ø.ØØ ea.

PAID ADVERTISEMENT

-- BeeLine V2.1 -- BeeLine V2.1 --

>> Be sure to read the latest BeeLine review by Angel Matos <<

LOAD/SAVE in ASCII or CBM Format
Skip beginning of large files

VIEW memory Buffers using forward
and backward controls
Set Blocks for Saving & Printing
Transfer Data between Buffers
Find strings within Buffer Data

PRINT to CBM or ASCII Printers
Use various device numbers

RECEIVE directly to Disk files
Up to 500K using CBM 8050 Drive
Use with/without Memory Buffers

MESSAGE Editing within BeeLine
Use 9 seperate Messages with
screen editing & ESC codes
Transmit & Edit using Fkeys
Prepare Message from Buffer,
previous or blank Message

XMODEM Upload and Download
Exchange any Disk file
Visually monitor transfers

TRANSMIT Data from Messages,
any data file or Buffer
Delay after char. or line
Use XON/XOFF flow control

AVAILABLE IMMEDIATELY FOR ONLY $40.00 - ORDER FROM CBUG TODAY!

BeeLine V2.1 is Copyrighted 1986 by J & K Lemkelde
LemData Software, P.O. Box 175, Dover, PA 17315

RIBBONS - MEMORY KITS - FANS - POWER SUPPLIES

				12+	1-11
CBM	4023	Multistrike	ea.	$5.50	$6.00
CBM	8023	Nylon Fabric	ea.	$5.00	$5.50
CBM	1525	Nylon Fabric	ea.	$5.75	$6.25
CBM	1526	Multistrike (Carbon)	ea.	$5.50	$6.00
CBM	MP801	Nylon Fabric	ea.	$6.50	$7.15
CBM	PET	Nylon Spool	ea.	$2.90	$3.20
CBM	PET	Multistrike	ea.	$4.05	$4.40
CBM	6400	Nylon Fabric	ea.	$4.95	$5.45
CBM	6400	Multistrike	ea.	$4.95	$5.45

Minimum Postage: $2.00 plus $0.25 each additional ribbon
Please give Street Address for UPS shipments -- Foreign Add 25%

"IF IT PRINTS, I HAVE IT."

Memory Expansion Kits (each)	($3.00 UPS)	$75.00
Memory Upgrade (In Shop)		$100.00
Fan Kits	($5.00 UPS)	$25.00
B128-80 Power Supplies	($5.00 UPS)	$50.00

Will accept COD's

S & W Supply Company
Jim L. White
5308 Timberline Trail
Rapid City, South Dakota 57702
(605) 348-3696

[Prices Subject to change without notice.]

C-64™ • VIC™ • SX-64™ • C-128™ • Plus 4™ • C-16™ • B-128™ • PET™ • CBM™ • LCD™

One disk. 25 business programs, $19.95

The Intelligent Software Package is the one product for your Commodore that can take care of all your data processing needs.

Customers write: "*. . . accolades for the authors. This is as slick a deal as I have seen and more than adequate for all except fancy presentations. The best thing is the ease of use . .* "

"*I have come to consider these programs among the most valuable pieces of software I own.*"

There are no hidden fees for shipping or documentation, and no clubs to join. The package is not public domain software, and is sold only direct to customers by mail: it supports all available printers, and will run on any Commodore computer [except Amiga] with a minimum of 10k RAM, including the C-128 in C-128 mode.

What you get when you order the Package:

Database—A completê database manager. All fields completely user-definable. Can be used for any number of tasks, including accounting, checkbook and tax records, mailing lists, inventory control, catalog maintenance, or as an electronic rolodex. A customer writes: "*I am especially impressed with Database, and have used it to replace a half-dozen other 'database'-type programs I had been using.*"

Word Processor—A full-featured menu-driven word processor. Allows full control over margins, spacing, paging, indentation, and justification. "*Highly recommended.*" —Midnite Software Gazette. "*Provides good basic features.*" — Compute's Gazette.

Copycalc—An electronic spreadsheet. "*Excellent program for budgeting, estimating, or any math-oriented use . . . well worth the money. Highly recommended.*" — Midnite Software Gazette.

ReportGen—creates form letters, mailing lables, etc.

ReportMerge—creates statements invoices.

Basball Stats—compiles team batting statistics.

Index—indexes W/P's text files.

Wordcount—counts words in a text file.

WPConvert—converts files to other WP formats.

DBMerge—facilitates relational D/B applications.

DBStat, DBStat2—analyze D/B files.

ASCII—converts text files into program files.

Checkbook—reconciles checkbook.

Inventory—Maintains inventory records.

Paper Route—A/R for paper route.

Loan Analysis—computes finance terms, prints schedules.

Breakeven—computes breakeven analysis.

Depreciation—creates depreciation schedules.

Labeler—creates labels.

File Copier—copies sequential, program files.

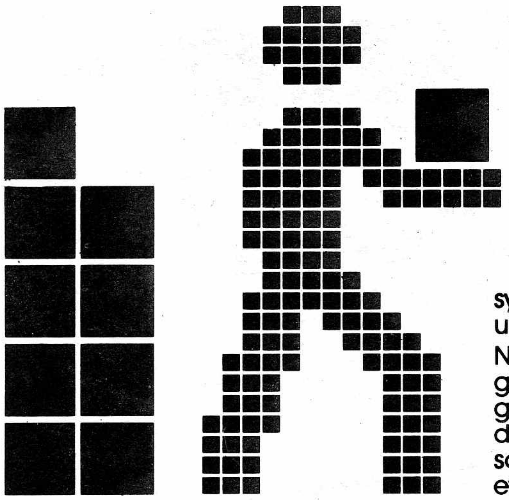
Correlation—calculates statistical correlation.

Also other Database and Word Processor utilities.

To order, send name, address, and $19.95 to address below. Please specifiy regular [1541/1571/2040/4040/2031] disk, 8050 disk, or cassette [cassette not available for Plus 4 or C-16]. Add $3 for credit card or COD orders; Calif. residents add 6%. No personal checks from outside USA. A sampling of program output is available for $1. **Tear this ad out and keep it handy!**

Intelligent Software
Quality Software since 1982

Box A Dept. M-8
San Anselmo, CA 94960
[415] 457-6153



NWM's INVENTORY CONTROL SYSTEM*

This system was developed by a user for users. It seems that most systems are developed by computer engineers that never see or use the product when finished.

NWM, inc. chose the toughest software analyst we could find to give a complete and unbiased review. Bob Loefler is the gentleman that tore apart the CABS Accounting system and documented all the bugs in most of the modules. To have someone of this caliber make suggestions means better, more efficient software available for CBUG members.

- loads program modules in less than **8 seconds** (superbase 2) to main menus in **3 seconds** or less
- on screen **pop-up calculator** in transaction modules
- most data centered functions use the **calculator keypad**
- versatile report features allow for **3 ways** to print the same report. User selects the fastest method
- built in sophisticated export program allows for **complete packing** of the database
- type ahead feature allowed
- you can display reports on screen
- access to superbase menu for user developed applications

Now it's even better!

Listen to how NWM has improved this already great Inventory Control System...

- **Partial-match key search now allowed in transaction modules.**
- **User can search forward and backward through file while in transaction modules.**
- **Elimination of most tiresome prompts for faster movements between subfunctions and menus.**
- **Simplification of entering prices and costs that have not changed when entering transactions.**
- **User programmable calculator allows you to turn the calculator function on or off within the transaction modules. If the user does not need the calculator, it saves time by eliminating the prompt and bypasses the calculator function.**

Only a $10 upgrade charge for owners of previous versions of NWM Inventory.

WE SUPPORT OUR CUSTOMERS!

And listen to our prices!

B Version 1 8050	**$49.95**	C-128 Version 1 1571	**$49.95**
B Version 2 8050	**$49.95**	B-128 Version 1&2 8050	**$54.95**

NWM, inc. • 1645 Rand Rd. • Des Plaines, IL 60016
(312) 299-2270

*Requires use of Superbase®

Superbase is a registered trademark of Precidion Software.
B128 and C128 are registered trademarks of Commodore Business Machines Limited.
©1986, NWM, Inc.

Vacation Notice
N.W. Music will
be closed from
Fri June 26 thru
July 3 inclusive

CBM 256-80
w/8088 Co-processor

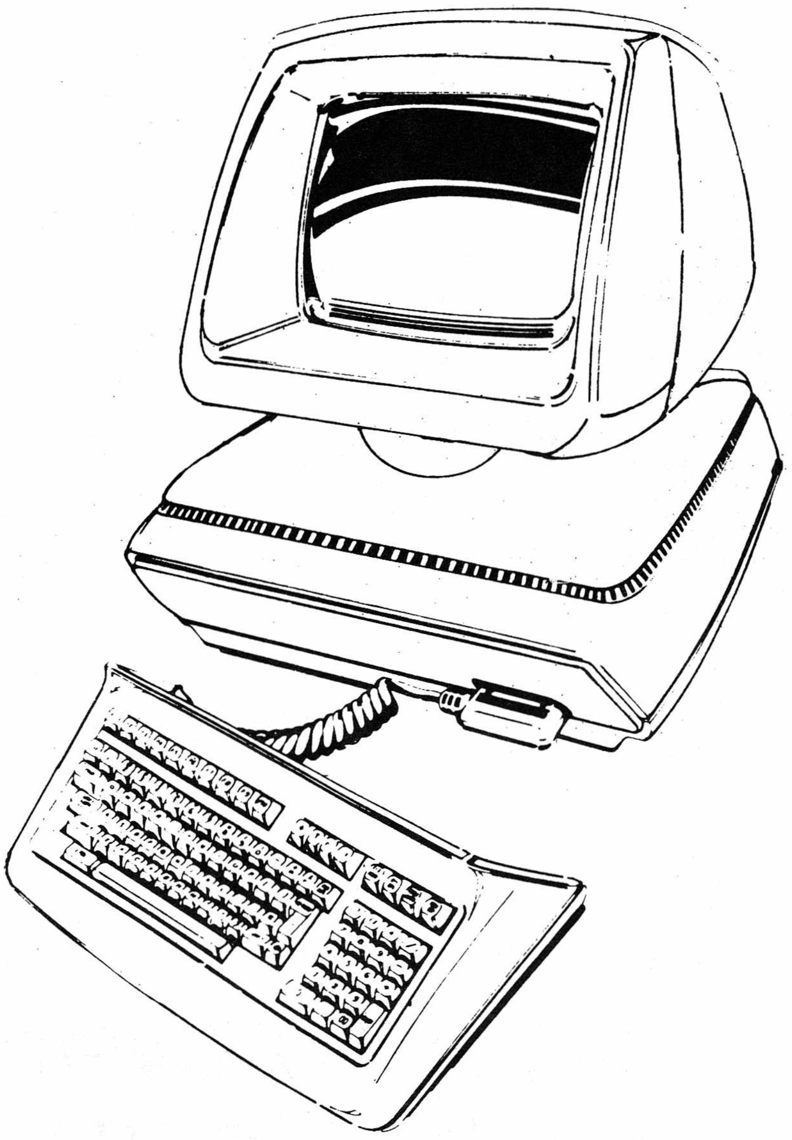
Call for Details
Very Limited Quantities

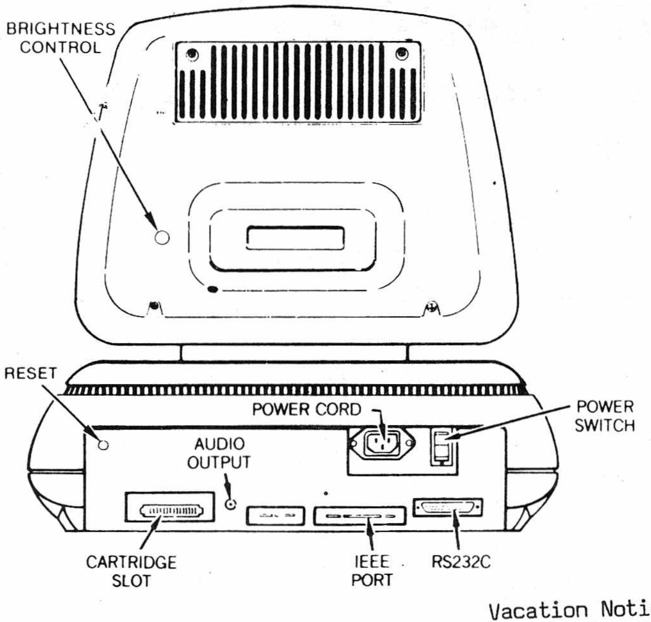
We are offering a limited number of used CBMX 256-80 computers for only $400. These come installed and tested with the original 8088 co-processor boards that until recently were hidden in the Commodore research labs.

The CP/M 86* operating system was implemented on the B system and has been tested and found to be very reliable. It had previously been stated that the co-processor would only work with the CBM 256-80. This is not true! The PLA that is installed on the hi-profile motherboard determines whether the operating system will run on that particular machine. We will also be offering the 8088 co-processor board for $150 providing the purchaser sends in their working hi-profile motherboard with the correct PLA. Call for details!

All the generic CP/M 86 software that we have tested will operate on the B series machine with an 8050 drive. The SFD and the 8250 can also be used with some restrictions. These same programs can also be run under the Digital Research CP/M 86 and Concurrent PC Dos operating system on an IBM. Therefore an investment in software is not wasted as it can be ported to other compatible computers.

*CP/M 86 is a trademark of DRI Inc.





CBM 128 & 256 features:

- SWIVEL MONITOR
- ADJUSTS HOR. and VERT.
- DETACHABLE KEYBOARD
- 9 x 14 PIXEL DISPLAY
- INCREDIBLE RESOLUTION
- DESIGNED FOR 2 INTERNAL DRIVES

Priced from $225 to $550 US
SHIPPING CHARGES EXTRA!

You really have to see the green phosphor display to believe it. This model has at least as good a display as the other company with those three big letters.

Vacation Notice
N.W. Music will
be closed from
Fri June 26 thru
July 3 inclusive

NORTH WEST MUSIC CENTER INC.

1645 Rand Road — Des Plaines, IL 60016
Hours — Phone (312) 299-2270
Mon.-Thurs. 12:30-8:30, Sat. 9:00-4:00

Super Office
List
$~~149~~00

SUPER OFFICE

SALE PRICE
$49.95*
U.S.

WITH SUPERBASE 2

ALL THE GREAT FEATURES OF SUPERSCRIPT II AND SUPERBASE INTEGRATED IN ONE FANTASTIC PACKAGE

Store important data then print out personalized custom reports inserting totals and names in the proper places. Do invoices, past due notices and many other things with Super Office. With Super Office you can do anything a small business needs.

Super Office is the ultimate in integrated programs for the "B" series market. Some features include left and right margin settings, tabs, decimal alignment, right justification, underlining, bold print, page numbering and a whole lot more. Super Office is easy to use with simple function key commands and an easy to use manual. There is even a spelling checker to keep you from making embarrassing spelling mistakes. We challenge you to find an easier integrated program or one with more features than Super Office. Includes the ultimate data base at an affordable price. You can do anything that requires data storage with Super Office. Simply set up your screens for easy entry of information (up to 4 screens). Then set up the way you'd like the information to be printed. Super Office does the rest! This program has full calculation capabilities and full data storage manipulation. You do not have to reload either program to access the other. Automated merging through the Super Office application generator.

(must have 256K-1 Meg. upgrade to operate)

* Purchase subject to signed disclaimer.

TAKE A LOOK AT THESE GREAT SUPER OFFICE FEATURES

- Help Screens
- Bibliographies
- Document Chaining
- Sales record • Home Budget
- Built-in programming language
- Highlighting • Inventory
- Up to 127 Fields per Record
- Calculation capabilities • Formatting
- Up to 4 Data Entry Screens per File
- Form Letter Printout
- Better than dbase II. • Mailing list
- User definable screen formats
- Applications Generator
- Up to 15 files per Database
- Keyed Access
- Report Generator
- Macro Capability

Shipping and handling charges $3.95

NEW 128K USER INSTALLABLE MEMORY EXPANSION! INTRODUCTORY PRICE OF ONLY $125.

NWM Inc. has contracted to have a user installable 128k memory expansion board designed. This non-destructive upgrade will turn a B-128 puppy into a full grown 256k monster machine! The board will install on the 40 pin co-processor and 60 pin expansion bus.

This will give you the capability to run the following programs.

- SuperOffice
- Superscript w/3 banks.
- Calc Result w/12,047 cells
- 8432 emulator w/4 banks.
- CP/M 86 extended programs*
- Word & Calc Result integ.

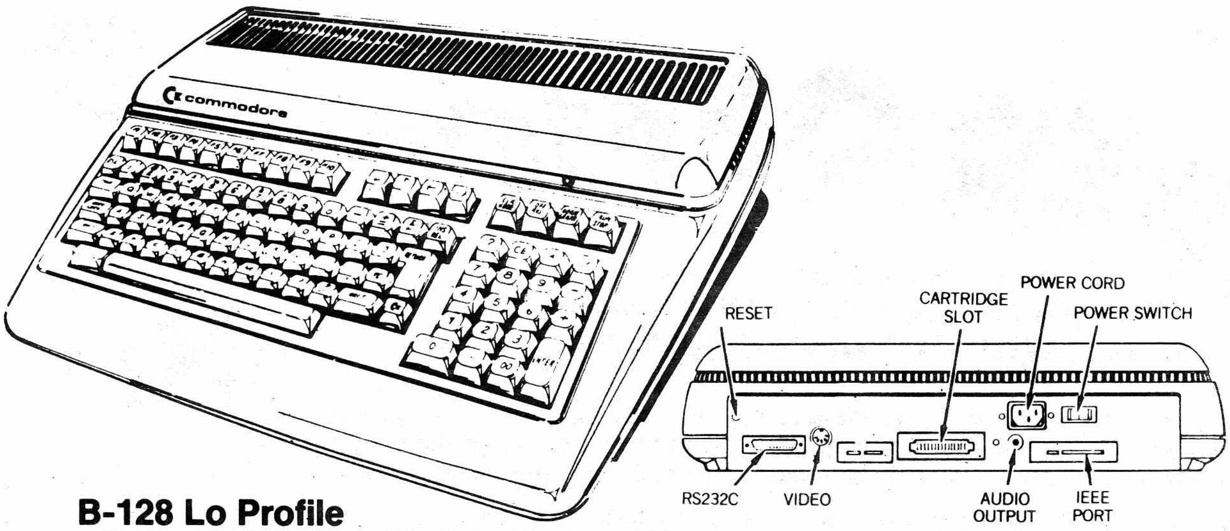
If you were afraid to send your machine away or did not have the ability to self install the upgrade, then this is for you. This is your opportunity to enjoy expanded memory and what other B-128 users have found it can do for you!

*W/8088 co-processor

Shipping and handling charges $3.95

NORTH WEST MUSIC CENTER INC.

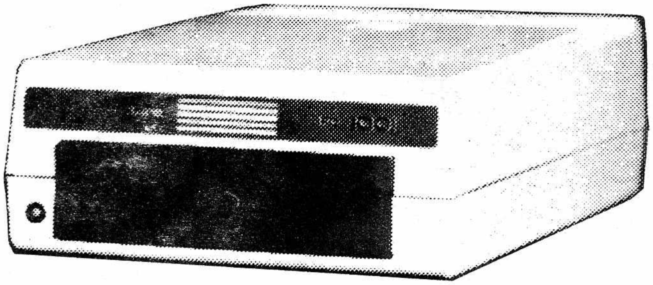
1645 Rand Road — Des Plaines, IL 60016
Hours — Phone (312) 299-2270
Mon.-Thurs. 12:30-8:30, Sat. 9:00-4:00



B-128 Lo Profile

We have just purchased the last of the lo and hi profile B-128 computers from Commodore. We have both new and used models of both of these computers. If you would like to have a backup computer for your system then this is it! When these are gone there are no more! At this time it is certainly very prudent to have a backup computer. If your computer goes down and has to be repaired, you will certainly feel more secure knowing you have a replacement if it takes time to find parts for the repair!

PRICED AT **$145** US
ADD $10.95 SHIPPING US



SFD 1001 1 Megabyte Drive double sided 8250 format IEEE interface

PRICED AT **$149.95** US
ADD $9.95 SHIPPING US

COMMODORE 8000-9000 SOFTWARE & MISC.

9000 Superpet	**$350**
64K exp for 8032	**$150**
Pet Switch	**$199**
Pet Daughters	**$105**
BPI General Ledger	**$25**
BPI Accts Payable	**$25**
BPI Job Cost	**$25**
BPI Accts Payable	**$25**
BPI Inventory	**$25**
Superscript 8032	**$79**
Superbase 8096	**$79**
OZZ Database	**$25**
Legal Time Acc.	**$25**
Dow Jones Program	**$25**
Info Designs 8032 Accounting System	**$50**
Superoffice 8096	**$149**
Calc Result 8032	**$89**

Vacation Notice
N.W. Music will
be closed from
Fri June 26 thru
July 3 inclusive

International Orders Call For Shipping Charges

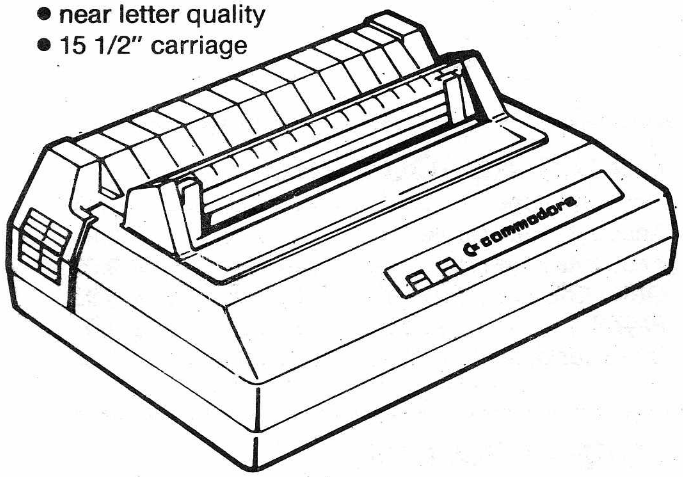
NORTH WEST MUSIC CENTER INC.

1645 Rand Road — Des Plaines, IL 60016
Hours — Phone (312) 299-2270
Mon.-Thurs. 12:30-8:30, Sat. 9:00-4:00

PRINTER CORNER

8023P 160 CPS Printer

- 100,000,000 characters
- near letter quality
- 15 1/2" carriage



Vacation Notice
N.W. Music will
be closed from
Fri June 26 thru
July 3 inclusive

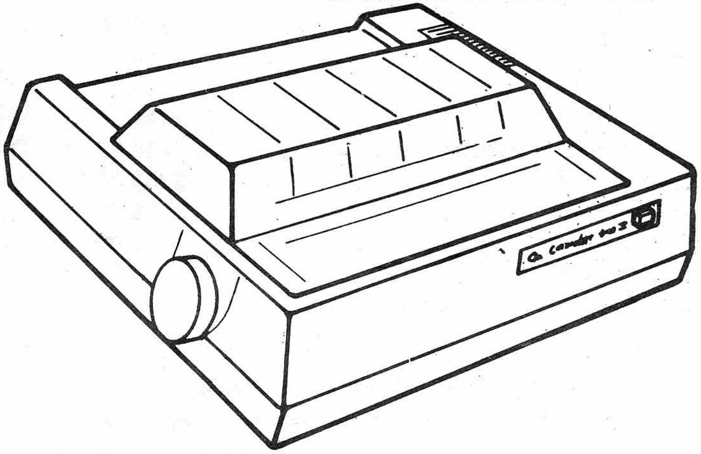
PRICED AT **$179** US
ADD $16.95 SHIPPING US

The 8023p supply is in an unpredictable position at the moment. Before we could place a large order with Commodore for these printers, they sold all 1,600 of them to a large import-export firm. At this time we have not been able to locate where they were shipped. The available supply of these may also be very limited.

CBM 4023 Printer

Unfortunately, all the remaining 4023p printers were also sold to a Canadian firm two months ago. We have located a small number of new and rehabilitated 4023p printers. The supply on these is larger than the 6400 printers but still is in a critical state. Note that a new printhead alone is almost $60 without installation. The available supply may last through early to mid-summer.

PRICED AT **$99** US
ADD $9.95 SHIPPING US



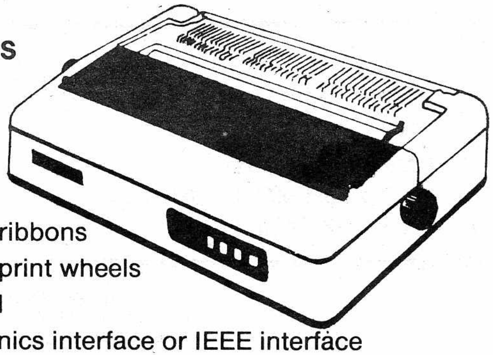
Commodore 6400

CBM 6400 Daisy Wheel Printer

Commodore has sold all their remaining 6400 printers to a company in Canada. We bought all the 6400 printers that we could find remaining in the U.S. If you still desire one of these printers we only have a few available. The supply is expected to be depleted by the end of April or early May. Don't wait on these because we do not anticipate locating any more.

CBM 6400 SPECS

- Bi-directional
- two copies
- 40 cps
- 136 column
- uses standard ribbons
- uses standard print wheels
- 40 ms line feed
- Parallel centronics interface or IEEE interface



w/Parallel centronics	**$450**
w/IEEE	**$450**

NORTHWEST MUSIC CENTER INC.

1645 Rand Road — Des Plaines, IL 60016
Hours — Phone (312) 299-2270
Mon.-Thurs. 12:30-8:30, Sat. 9:00-4:00

Hacker's Corner

One of a Kind • Surplus • Monthly Special • Closeouts

Limited quantities to stock on hand

PRECISION SOFTWARE

Superbase	$ 9.95
Superscript	$ 9.95

HANDIC SOFTWARE

Calc Result	$89.95
Word Result	$59.95

C.A.B.S. ACCOUNTING

ieneral Ledger	$ 9.95
Accounts Receivable	$ 9.95
Accounts Payable	$ 9.95
Order Entry	$ 9.95
Payroll	$ 9.95
Buy 5 for only	$24.95

B-128 GOODIES

B-128 Programmable Reference Guide was $29.95	$ 9.95
P-I	$14.95
I-I	$29.95
SFD 1001 1 meg drive	$149.95
Superbase The Book	$14.95
Superpet 9000	$350.00
Pet Switch	$199.00
Pet daughters	$105.00
8050 rehabs from	$250-$400

CALL FOR INFORMATION ON 8000 & 9000 SERIES SOFTWARE AND MISC.

SUPER DISK DOC

ONLY $17.95 WHILE SUPPLIES LAST!

CBM PROFESSIONAL USERS GUIDE

This is a reference manual on most of CBM'S Business Machines.

An excellent source for looking up useful information not found in most of the standard doc we have seen with most computers! ONLY $17.95 WHILE SUPPLIES LAST!

ONE OF A KIND ITEMS

WE HAVE A LIMITED NUMBER OF 9060 & 9090 HARD DRIVES CALL FOR DETAILS!

DENT AND SCRATCHED MONITORS	$99.95	SERIAL to IEEE	$49.95
SMITH CORONA		IEEE to PARALLEL	$89.95
DOT MATRIX DM200	$149.95	MONOCHROME MONITOR	$79.95
MJ-10 COLOR MONITOR	$125.00	C-WORD WRITER	$29.95
AVATEX 1200 MODEM	$94.95	C64-MULTIPLAN	$50.00

CALL FOR SHIPPING CHARGES AS ITEMS VARY—SHIPPING NOT INCLUDED
NWM Inc. reserves the right to limit quantities to stock on hand.

Vacation Notice
N.W. Music will
be closed from
Fri June 26 thru
July 3 inclusive

NORTH WEST MUSIC CENTER INC.

1645 Rand Road — Des Plaines, IL 60016
Hours — Phone (312) 299-2270
Mon.-Thurs. 12:30-8:30, Sat. 9:00-4:00

PHYSICAL EXAM — THE DISK ALIGNMENT PROGRAM

By now you've seen several articles and ads dealing with the speed instability problems of the 8Ø5Ø drives. This problem also applies to the 825Ø and most other CBM drives. Physical Exam is a superior in-computer measuring tool for seeing the speed of a drive on you monitor with great accuracy. The drive must be between 299 and 3Ø1 rpm to work properly. P.E. gives about 1 reading per second, leaving about 15 on the screen as they scroll. The accuracy is to two decimal places, 299.35 for example. You then simply turn the control on the inside to the drive is on speed. If you are going to add the new reference diodes you must have a way to reset the speed adjustments. You can easily and quickly check your drives as often as you like to insure that you will not have read or write problems, or worse yet trash a disk or write bad files within a large data base. A second section of the disk displays the mechanical alignment of the head tracking so this also can be adjusted when the occasion demands.

CBUG is now stocking P.E. for the 8Ø5Ø, 4Ø4Ø, 1541 and 1571. Though the 8Ø5Ø version will not "officially" operate on a SFD 1ØØ1 or an 825Ø, you can load the program from an 8Ø5Ø then switch drives without turning the computer off. You switch the drives after you have passed the first menu. By doing this you avoid the section of program which asked the drive what it is.

* *

BETTER ZENERS and REFERENCE DIODES

Finding parts is not always easy, so CBUG has purchased a supply of both replacement close tolerance zeners of the type recommened by Mr. Kernaghan, and a far superior reference diode. We are offering these, albeit at a modest markup, along with copies of the circuit board component location charts, with schematics and brief instructions. KNOW WHAT YOU'RE DOING OR GET SOMEONE WHO DOES TO HELP!

Zener Diode: ECG 5Ø14A is also known as a SK6A8 under the RCA numbering system. In reality it is a relabeled 1N5235B. These diodes can simply be swaped for the two located on the analogue board, one for each drive (Ø and 1) They are about 4 times better than the original part ratings. These parts are rated at .Ø5% per degree C drift. Two are provided in a package. CBUG order #1133Ø, $6.ØØ.

Precision Reference: LM329 is a tight tolerance near zero drift reference diode from National Semiconductor. It sports a .ØØ15% per degree C drift -- 33 times better than the zener diode. However, a bias resistor must also be changed, so 4 parts need to be substituted. Make twice as sure you are capable of this before attempting. Two LM329's and 2 2.7K resistors in a package. CBUG order #11344, $8.5Ø.

* *

INTERNAL IEEE CONVERTERS FOR THE 64ØØ PRINTER

If you are about to or have purchased a CBM 64ØØ printer from most any source, it likely will come with a Centronics Parallel (industry standard) connector. To use this interface system you must provide an IEEE to Centronics converter. CBUG has brought in a quantity of the original internally installed converters for the 64ØØ printers. It takes but a few minutes and only one small phillips screw driver to install. This is a Commodore interface and performs far better than the outboard devices with respect to many of the instruction codes. It is equiped with adjustable device number jumpers. We provide them with complete instructions and schematics. CBUG order #11221 #35.ØØ. Incidentally, there are two sets of dip switches on the back side of the control panel of the 64ØØ. Their proper settings for B128 use is: ØØØ1ØØØØ Ø11Ø1ØØ1ØØ.
When using Superscript with the 64ØØ printer, at the opening menu select option 3, Diablo!

JCL SOFTWARE 700 WORKSHOP

The special CBUG version of the Workshop is supplied on disk. It requires a 24K RAM pack mapped into the system bank cartridge space.

The disk includes.......

* EXTENDED BASIC which links over 39 new keywords into the BASIC interpreter. Your programs become faster, more reliable, smaller and generally better structured and readable. Many useful routines and small programs are included to demonstrate the flexibility and power of the system.

* SOFT LOADED EXTENSIONS for further BASIC extensions with your own machine language routines called by your choice of keyword. Many essential parsing and utility routines may be accessed from a convenient jump table.

* PROGRAMMING AIDS include AUTO line number generation, re-NUMBER, FIND, program JOIN and a true ASCII file MERGE.

* ASCII FILE EDITOR for assembler source file preparation. Includes screen scroller and special keywords for file loading, saving, viewing and printing.

* FULL SYMBOLIC ASSEMBLER that can assemble source files from disk or memory. The assembler and editor may be in memory together, reducing the need for repeated loading.

* DOS SUPPORT the classic disk handling utility.

* WORD PROCESSOR an example of the type of utility JCL created using the editor and assembler!

* DOCUMENTATION in file form which may be printed with the wordprocessor.

* 8050 DISK copyable, suitable for 128K RAM machines only.

700 ASSEMBLER

★ Editor with screen scroller, FIND, AUTO, etc.
★ RENUMBER for BASIC programs as well as for source file lines.
★ TYPE command provides adequate performance for simple word processor jobs.
★ 14 assembler directives.
★ 'B SYS' allows machine code to call kernal from any bank.

700 WORKSHOP

Your programs will become faster, more reliable, smaller and much easier to write!

★ 10 TOOLKIT-type programming aids
★ Enhanced BASIC with 39 new KEYWORDS.
★ Comprehensive data input systems with efficient File Interfacing.
★ DOS-SUPPORT available all the time.
★ ASCII-editor for text and assembler files with screen scroller.
★ Elementary word processing facilities.
★ BANK-SYS. Kernal in every bank.
★ Design and add your own BASIC extensions.
★ Extra RAM in system bank.
★ Comprehensive manual.

ORDER FROM THE CBUG LIBRARY, CBUG #51, stock #11894

◆◆◆

PHYSICAL EXAM

NOW FOR THE 8250 & SFD 1001

$35.00 ea, from CBUG

8250 or SFD 1001 order #12192; 8050 order #12219; 4040 order #12238
1541 order 12223; 1571 order #12242

FUNCTION LIST - JCL SOFTWARE 700 WORKSHOP

1. DOS Support . . . Enhanced version of the standard system.

2. Thirty nine NEW keywords:-
STOP KEY OFF and STOP KEY ON . . . mutes the STOP key for safety.
BELL . . . sound the internal bleeper.
DELAY . . . programmable wait period. 1mSec to 64 Seconds.
TRANSFER . . . move RAM contents from any source to any destination.
WINDOW . . . set a window on the screen to constrain cursor.
FLIP . . . reverse field the current window.
FLASH . . . flash the window to get attention.
FRAME . . . draw frame around window or at any other place on screen.
EVALUATE . . . calculates the value of an expression held in a string.
ROMOFF . . . Guess!!!
UPPER CASE and LOWER CASE . . . select screen print mode.
GETKEY . . . single key response of program defined letters.
RESPOND . . . safe and filtered replacement for INPUT with length limit.
READSCREEN . . . pick data off the screen and asign to string variable.
PLIST . . . print a listing of the current program, upper,lower and graphics.
DUMPLINE . . . print a specified screen line to the printer.
SAVE/GET/EXCHANGE SCREEN . . . move screen contents to and from memory.
SKETCH . . . Free drawing over the current window, used to design forms.
PRINT AT . . . co-ordinate specific printing system.
OUTPUT . . . alternative to PRINT# that sends ASCII or CBM mode.
ASCII/CBM . . . Switches OUTPUT to standard ASCII or CBM printer convention.
DISK . . .directory listing to screen window. Multi column, screen full pause.
LAST . . . specify alphabet limit for filters on RESPOND and SCRINP.

Screen Input (SCRINP) and related commands, forms generator/servicing.
DEFINE . . . set up characters to define screen limits.
SCRINP . . . let user enter data on to the SCREEN INPUT SYSTEM.
INPUT FIELD . . . read specific field into a specific string variable.
CLEAR FIELD . . . clear all or a specified field.
DEBRACKET . . . remove screen field characters prior to printing screen.
CHECK FIELDS . . . validate a screen field layout.
SEND . . . transmit screen field contents to a relative file record.
RECEIVE . . . fill screen fields from a relative file.
MATCH . . . compare screen fields with a file record.

3. EBR Machine Language Environment.
BANK SYS . . run code in any bank, any address. Kernal available in ALL banks.
FIXMEM . . reserve memory in bank 1 for integrated machine code/BASIC.

4. Programming Aids.
AUTO . . . automatic line numbers of specified increment.
NUMBER . . . three parameter re-number system.
JOIN . . . adds a BASIC program to the end of a current program.
MERGE . . . reads a sequential file and executes contents, in command level.
EDIT . . . turn off BASIC, edit ASCII files such as Assembler source code.
PUT . . . make a sequential file of ASCII text whilst in EDIT mode.
FETCH . . . read a sequential file into the ASCII editor.
TYPE . . . display/print an ASCII file with page breaks, right justify etc.
GOCOM . . . convert Assembler object files to CBM program files.

5. Soft loaded BASIC Extensions . . . Simple system to add your own keywords.
6. Blended Machine Code and BASIC . . . Simple system to produce hybrids.

CBM 700 (B Series) ASSEMBLER PACKAGE

* Compatible with CBM assemblers for 4000 and 8000 series machines.
Assembled CBM's own source files for the 700 Kernal as an independent test.

* Editor commands

EDIT - Activate editor mode, flagged by switching cursor shape.
PUT - Save all or part of a source file to disk.
FETCH - Load or join a source file back into memory.
DRIVE - Specify default disk drive for PUT/FETCH.
FIND - Look for a string in memory. (*)
AUTO - Generates line numbers for text entry.
NUMBER - Renumbers all or part of text. (*)
TYPE - Displays or prints a source or text file.
ASCII - Switch output from TYPE and PLIST to standard ASCII.
CBM - Reset to normal CBM output for TYPE and PLIST
SCROLL - ESC 'pi' followed by :-
CRSR up/down scrolls source up/down screen.
T/B/C re-writes screen from top/bottom/cursor.
GOCOM - Convert standard CBM Assembler hex file to program file.
KILL - Return to BASIC environment.

* General purpose commands

PLIST - Like LIST but outputs to printer without OPEN4:CMD4 etc... (*)
MERGE - Execute a sequential file's contents as if typed from keyboard. (*)
JOIN - Links a program file to the end of a program in memory. (*)

* Machine language execution

BSYS - Executes a program in any bank and automatically makes the Kernal available in the bank called.

* Assembler

ASM - Run two pass assembler with standard .BYT .WORD .PAGE .SKIP .OPT .END .FILE .LIB .LIST .NOLIST .NOSYM directives. plus .IFE and .IFN conditional assembly directives and .NARROW for printers with 80 character buffer limit.

- Source may be assembled from disk files or directly from core to produce screen or printed listings and hex or binary output files.

Commands marked with (*) are able to be used with BASIC as well. ASM and BSYS may be used in a BASIC program so a command file can automate the assembly cycle. TYPE includes optional automatic justification and paging so can be used to produce short reports without a wordprocessor. Instructions supplied on disk including demonstration and useful source files.

WANTED TO BUY

8050 disk drive in good condition and/or complete B128 system. Rick Boyer, c/o CEDO, 2601 E. Airport Rd., Tucson, Az. 85706.

FOR SALE

CLASSIFIED AD RATES: $10.00 per line of
110 Characters. Payment with copy.
Display advertising rates on request

B128, 8050, 4023, Zenith Monitor, asst software. Best offer. Mike Osborn, 721 9th Av. Tucson, Az. 85716, 602 624 4534

1	8050 Dual Disk Unit w/B to IEEE cable	$400	1	CRT: orange, with cable	$ 75
1	Swivel CRT Monitor: orange w/cable	$ 80	5	Superscript II Packages	$ 10 ea.
4	Superbase Packages	$ 10 ea.	2	Calc Result Packages w/cartridge	$ 10 ea.
2	Protecto B-128 Programmers Ref. Book	$ 10 ea.			

Warren D. Swan, 1 N 114 Woods Avenue, Wheaton, IL, 60188, (312) 665-1514

Protecto System, will separate, B128 computer not operating. Joseph L Wolfe, 2000 Meredith Rd., Virginia Beach, Va. 23455. 804 460 2704.

Complete Protecto B128 System plus additional software, 8050 physical exam and accounting. $700.00 or best offer. Jutta Bradford, P.O. Box 2085, Dothan, Al. 36302, 800 633 7569.

Protecto B128 package, incl 8050, 4023, Amber monitor, Superscript II, Superbase, General Ledger, Payroll. $500 or best offer. Trident Engineering 301 267 8128.

B128 w/Ctrc par cable, Mntr, dual disk dr, all business sftwr/spdsht, manuals in original box, lk new cond. $750. John L. Smith. 312 456 4462.

Protecto system & asst. software. Drive 0 intermittent. $400 or best offer. Sharon D. Lowe, 560 Perry St., Trenton, N.J. 08618, 609 396 9257 8am-4pm.

Complete Protecto B128 System, SS,SB, Liz Utilities, Knights, other misc from CBUG. Excellant condition original packaging. Best offer over $500 plus freight. Charles Poole 23 Coville Dr., Browns Mills, N.J. 08015, 609 893 6884.

----

2 Standard Protecto Pkgs $595 ea. New 9060 5 meg hard drives $499. SB SS AR GL $50.00. G. A. Katz. 313 352 6566 evenings.

B128 Protecto Package, SS, SB, Knight Copy Util, Teleterm, $625 will ship, Douglas Purdy. 518 793 1690 EST after 5pm.

Protecto B128, 8050, Zenith Amber Monitor, ADA 1800 printer interface, C/R, SSII, SB, Knight's', Casey's. Manuals, most CBUG utilities. Mint. $450 shipping included. Jim Roach, 111 Sherwood Dr. Benton, La. 71006, 318 965 0632.

B128 expanded to 256K, B128 standard, 8050, Amber monitor, ADA 1800 printer interface, 2 300baud manual modems (RS232), Smart Cat 1200 baud modem (RS232), SSII, SB, Programmers Ref Guide, Anderson's 24K cartridge. Best offer(s). Russ Beinder, 1159 Ranger Pl, Victoria, B.C., Canada V8X 3P5, 604 479 8510.

----

2 sets like new in box: B128, 8050, 4023, Amb Mon and all cables; $500/set plus shipping. Also, 1 B128 for $100, and 1 8050 for $300 plus shipping. Ed Repic 714 520 0778.

Tony did it again! The long awaited code to allow use of the cassette port on the B128. ALSO, the previously missing code for operating the drives with drive numbers interchanged. I.E., Drive 0 would be on the left, and Drive 1 on the right. Handy way to equalize wear, and switch drives if one is down, particularly drive 0 since so many programs require drive 0!

For discussion about these programs, read the several articles by Mr. Goceliak in this issue.

* *

```
1 "random file 3/87" 36 2c   9   "cbug contests"     seq  0   "---screen dir---"  del   8    "compare dir's"     prg
67   "b-128 tape code+" prg< 17  "secret b128 key!"  seq  1   "2 col dir"         prg   0    "-----------------" del
9    "tape instruction" seq< 5   "truly secret key"  seq  2   "m/l warning"       sec   1    "--updated auto--"  seq
0    "-----------------" del 15  "monitor & stack"   seq  0   "-----------------" del   8    "---auto inst.---"  seq
4    "zero data snow"   prg  0   "-----------------" del  1   "supplemental art"  sec   34   "auto v1.2"         prg
4    "incrementng snow" prg  0   "basic works too!"  del  0   "-----------------" del   6    "wedge autorecord"  prg
4    "page # snow"      prg  0   "memory diagnost."  del  27  "f key info"        seq   0    "-----------------" del
8    "compare memory"   prg  2   "basic page# snow"  prg  4   "function keys²"    prg   0    "-----------------" del
0    "-----------------" del 0   "-----------------" del  0   "-----------------" del   0    "-----------------" del
0    "-cbug articles--" del  0   "-alternate way--"  del  4   "--archive inst--"  seq   1    "whoopie"           prg
48   "column by tony"   seq  3   "tape loader pgm"   prg< 10  "archive files"     prg   5    "drive denominator  pgm
9    "tape ins."        seq  11  "tape code"         prg< 0   "-----------------" del   1725 blocks free.
```

* *

By: Rev. Doug Gorsuch

This series was appended to this disk just for a place to put it as the general topic disk is full!
An additional copy of this same material will be appended CBUB #M55 being released this issue.

Please read the article in the main body of this issue for the details on the contribution.

```
1    "SS2 CALENDAR"  seq   17  "30/mon"   seq   17  "31/wed"   seq   17  "31/fri"   seq
11   "Instructions"  seq   17  "29/mon"   seq   17  "30/wed"   seq   17  "30/fri"   seq
17   "31/sun"        seq   17  "28/mon"   seq   17  "29/wed"   seq   17  "29/fri"   seq
17   "30/sun"        seq   17  "31/tue"   seq   17  "28/wed"   seq   17  "28/fri"   seq
17   "29/sun"        seq   17  "30/tue"   seq   17  "31/thu"   seq   17  "31/sat"   seq
17   "28/sun"        seq   17  "29/tue"   seq   17  "30/thu"   seq   17  "30/sat"   seq
17   "31/mon"        seq   17  "28/tue"   seq   17  "29/thu"   seq   17  "29/sat"   seq
                                                17  "28/thu"   seq   17  "28/sat"   seq
```

* *

DITTINGER's UTILITIES   CBUG 58   NEW RELEASE   11925 $12.00

Here we have a number of self explainatory programs which Mr. Dittinger assures us are of the lastest and most useful vintages -- at his doing, of course. In addition to the very nominal royalty, if you find there are more than 3 programs of use to you, he has suggested a $1.00 per program not to exceed $7 additional dollars by way of freeware remittance to him.

```
1 "Nick's Programs " 01 2c   8   "tax inst"          prg  24  "baseball stat"     prg  21   "checkbook 1.1"     prg
2   " -------------- "  prg  42  "budget master"     prg  19  "credit card file"  prg  1    "print directory"   prg
9   " menu "            prg  5   "carpet est"        prg  42  "blackjack"         prg  19   "net worth 1.1"     prg
2   "   directory   "   seq  4   "compound int"      prg  14  "fifteen puzzle"    prg  3    "fkeys"             prg
8   " boot      .scn"   prg  18  "coupon organizer"  prg  24  "states quiz"       prg  5    "labelmaker"        prg
8   " intro     .scn"   prg  12  "home inventory"    prg  11  "math worksheet"    prg  1637 blocks free.
18  "tax records"       prg  46  "shopper's list"    prg  15  "interstate"        prg
8   "tax util"          prg  12  "util. bill audit"  prg  15  "kat and mouse"     prg
```

WINTER/SPRING 1987 PRINT FILE   CBUG 59   NEW RELEASE   11978 $9.00

The usual disk file of this issue.

```
1 "w/s 87 master   " w7 2c   19  "de la torre"       seq  19  "SupCrnNov/Dec86"   seq  7    "ricks"             seq
1    "12pt 58col std "  seq  32  "anderson"          seq  15  "mark.ss"           seq  113  "hart"              seq
1    "12pt 58col >2"    seq  32  "berez.bank"        seq  5   "mark.bible"        seq  10   "IEEE connector a"  seq
64   "scratch"          seq  25  'golds.re"          seq  21  "table of content"  seq  6    "cbug local meeet"  seq
17   "mark/cologne"     seq  16  "golds.num"         seq  11  "gorsuch"           seq  2    "lib art format"    seq
129  "matos"            seq  29  "ritter.jcl"        seq  40  "wright"            seq  2    "f+"                seq
107  "swan basic"       seq  17  "newburger"         seq  1   "hints & tips"      seq  83   "54"                seq
35   "swan 4023"        seq  52  "cumfer"            seq  8   "harrison"          seq  13   "55"                seq
36   "ritter.roms"      seq  15  "kennedy intro"     seq  101 "goceliak"          seq  5    "56"                seq
42   "berez.hex"        seq  84  "SupCrnrMay/Jun86"  seq  20  "gardner"           seq  15   "57"                seq
30   "loeffler"         seq  1   "12pt 56 col icpu"  seq  35  "enochs"            seq  7    "58"                seq
26   "faierson.so"      seq  85  "SupCrnJul/Aug86"   seq  8   "rotello"           seq  2    "59"                seq
57   "faierson.cpm"     seq  55  "SupCrnSep/Oct86"   seq  9   "contest"           seq
```

REVISED PRINTER ROMs
4023 COMMODORE PRINTERS 8023P

Commodore 4023 and 8023P printers can now print exactly what the B-128 displays without using special utilities or PRINT CHR$(X) commands. The 4023 printer can also print a small k that looks like a small k and not a capitol K.

These characters [£ ↑ π] and fourteen others have been corrected to print what is displayed by the B-128.
PLUS - two new commonly used characters, a degree symbol (°) and a combined (±) sign were added which are super easy to use with Superscript III's FEATURE capabilities.

Print data exactly as received from DELPHI or other networks. No more graphics characters in place of the brackets.

The B-128 Character ROM has been changed to correct the GRAPHics characters on the C & V keys and the shape of the capital O.

See the article in this issue for a complete description of the printer rom changes by CHR$ code.

Each ROM shipped with easy to follow instructions.

SPECIAL CBUG OFFER

8023P PRINTER ROM	$19.95	4023 PRINTER ROM	$19.95
B-128 CHARACTER ROM	$12.00	SHIP/HNDL	$ 2.00

ORDER YOURS FROM

RITTERBUSCH BUSINESS VENTURES
7201 W. Paradise Dr. Peoria AZ 85345 (602) 878-4832

BUY THE FIRST -- BUY THE BEST

Litsearach $167.00 Pet Speed $99.00

Bing Hart, 17311 E. 40 Hwy A34, Independence Mo. 64055. 816 373 5523

MORE WANT ADS

Complete Protecto System upgraded to 256K. SS, SB, Calc Result, Bterm. Asking $700.00. Robert Gann, 1590 162nd Ave #9, San Leandro, Ca. 94578

B128, 8050, BMC JJ-1201G green monitor, (no printer), al cables, SB, SS, etc. $400.00. Ray Grizzle, 55 Nichols Apts., Davidson Rd., Piscataway, NJ, 08854. 201-932-1399

LEGAL NOTICE

ELOW ARE THE TERMS AND CONDITIONS IMPOSED BY COMMODORE AS TO THE
ELEASE OF B128 DOCUMENTATION TO AND THRU CBUG. PERSONS ACQUIRING
HIS INFORMATION DIRECTLY OR INDIRECTLY ARE HEREBY NOTICED OF SAME

commodore
business machines, inc.

Executive Offices
1200 Wilson Drive
West Chester, PA 19380
(215) 431-9100

February 10, 1986

Mr. Norman Deletzke
Chicago B128 Users Group-International
4102 North Odelle
Norridge, IL 60634

Re: Commodore B128 PM Trademark Computer Documentation

Dear Mr. Deletzke:

Effective February 10, 1986, Commodore Business Machines, Inc. ("CBM") hereby grants to the Chicago B128 Users Group-International ("Licensee") a limited, no cost, non-transferable, non-exclusive license (the "License") to use the CBM owned CBM II Programers Reference Material and the CBM II Programming Support Documentation with updated memory maps (the "Products") for the sole purpose of reproduction of the products by Licensee for distribution to its members as reference material. This License is granted on the express conditions that the Products be held by Licensee and the members of Licensee as reference materials, for their individual and personal use only, and not for any commercial purpose. The term of this License shall be for as long as Licensee continues to provide the products solely for reference purposes and its members continue to hold the products for their individual and personal use and shall automatically terminate when this use ceases, at which time the products and all copies thereof shall be returned to CBM.

In consideration of the granting of this License, Licensee hereby agrees to include with each copy made of the Products, a copy of the applicable introduction sheet attached herewith as Exhibit A which acknowledges Commodore Electronics Limited's ownership of the products and provides CBM's disclaimer as to the products.

IN WITNESS WHEREOF, the parties hereto have executed this Agreement as of the date set forth above.

CBUG, INC.
(CHICAGO B128 USERS GROUP-
INTERNATIONAL)

BY:

Title: Pres CBUG, INC

COMMODORE BUSINESS MACHINES, INC.

BY:

Title: Vice President

HDS:97/mdc

CBM II PROGRAMMER'S REFERENCE MATERIALS

CBM II PROGRAMMING SUPPORT DOCUMENTATION

COPYRIGHT:

This document is copyrighted and all rights are reserved by Commodore Electronics Limited. No part of this document may be duplicated, copied, reproduced or distributed in any form or by electrical or mechanical means, including information storage and retrieval systems, without permision in writing from Commodore Electronics Limited. Duplicating, copying, selling or otherwise distributing this document is a violation of the law.

DISCLAIMER:

THE INFORMATION CONTAINED IN THIS DOCUMENT IS PRELIMINARY IN NATURE AND IS SUBJECT TO CHANGE WITHOUT NOTICE. THE INFORMATION CONTAINED HEREIN IS SUPPLIED IN AN "AS IS" BASIS, AND THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY IS SOLELY WITH THE BUYER. COMMODORE BUSINESS MACHINES, INC. ("CBM") MAKES NO WARRANTIES EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THE ACCURACY, COMPLETNESS, QUALITY, MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OF THE INFORMATION HEREIN CONTAINED. IN NO EVENT SHALL CBM BE HELD LIABLE FOR DIRECT, INDIRECT, INCIDENTAL OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY USE OF THIS DOCUMENT OR ANY INFORMATION CONTAINED HEREIN, EVEN IF CBM HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. SOME LAWS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF IMPLIED WARRANTIES OR LIABILITIES FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES, 30 THE ABOVE LIMITATION OR EXCLUSION MAY NOT APPLY.

HDS:98/mdc

CBUG ARCHIVE ORDER FORM AND NON-DISCLOSURE AGREEMENT

As a condition of receiving copies of CBUG Archive documents I agree and represent as follows without limitation or qualification:

1.) I am a member in good standing of the Chicago B128 Users Group;

2.) I intend to use the materials requested to enhance my and/or other CBUG members' usage of the B128 series of computers and accessories;

3.) I will not disclose, divulge, nor copy the contents of these documents to non-members of CBUG, except as they may be acting for me or on my behalf in the usage defined in #2 supra;

4.) I will cause a signed copy of this agreement to be forwarded to CBUG upon any further disclosure or release of these documents as permitted supra

5.) I understand that many of the documents covered hereunder are either copyrighted by or proprietary to Commodore Business Machines and/or their vendors; that the sole purpose of the provision of these informations by CBM to the CBUG membership is to enhance the usefulness of the CBM products purchased by CBUG members; and that there is no warrantee or assurance of accuracy, completeness or other representation by CBM, CBUG or any other. Any other disclosure is expressly prohibited, and I will not engage in such prohibited disclosures.

6.) I have read the limited no cost license (as printed on the reverse side hereof) extended to CBUG by CBM, and agree to be bound by those same said terms as if I were the named licensee.

Name______________________________ Please print or type

Address____________________________ Phone________________

City____________________ State__________ Zip __________

Country____________________

X______________________________ Date________________

Signature

Qty	Cat #	Archive Description	Price	Extension
______	12370	001 8088 Schematics w/ CCPM listings (printed)(40 pgs prox) <Add $5.00 air mail if not USA or N. America>	$25.00	________
______	12365	002 Software Devel. Information (microfiche)(302 images/pages prox)	9.00	________
______	12351	003 CBMII Programmer's Reference material (3 microfiche)(798 images/pages)	27.00	________
______	12327	from pg ______ to pg ______ of Archive # ______ from pg ______ to pg ______ of Archive # ______ from pg ______ to pg ______ of Archive # ______ Custom produced photocopy of Archive documents.. Allow 45 days for delivery. Add $.05 per page for air mail if not USA or North America. $10.00/order minimum photocopy order	.08	________

AGREEMENT MUST BE COMPLETED AND SIGNED BEFORE WE CAN SHIP ARCHIVE MATERIALS.

TOTAL ORDER THIS PAGE $________ __

you aware that the b, when performing such a round trip, leaves the stack in an unnecessarily bloated condition?

One of the things which always bothered me, as a former 1802 [stack? yeah 64000 bytes for the stack, and let everything else use the rest] programmer, was the seemingly miniscule stack area available. Let me emphasize however, that I have not yet run short of stack space for a bug-free program. Such is—not, however the case when using the b's monitor. [In fairness, perhaps improperly, but the anonymous author of the silver b series user's guide tells us everything about the monitor except how to enter it!]

The 'official' way to enter the monitor, is of course, the good ol' sys 57344. This gives a clean looking stack, the very impressive *** monitor 1.0 *** logo, and it works great, until you wish to exit. There is a wee deficiency there, and the x command just doesn't x. [You can escape to basic with a 'g 8000', but you're not supposed to have to resort to things like that. x is supposed to warm start basic.]

Next try, and by the appearance of the user's guide, at least the 'normal' if not official way to do things, is sys 5 or 6, providing you haven't read or set ti$ recently.

As an aside, and to perhaps pique your curiousity, turn to page 142 of the silver user's guide and look at the example R DISPLAY REGISTERS

Note three things, pc [program counter] = 0007, which indicates where the 6509 brain was going to go next, except that the previous instruction said 'break' (our sys 5!). OK, so even CBM uses sys5, so what? Later!, first squint at the irq display. Try your own sys 5 and compare. This number [yes it's a number displayed in hexadecimal], should never vary unless explicitly changed by you, and you better know what you are doing, or you won't regain control of your b until you power down. Son of a gun, CBM wasn't using the same kernal that we are! Last, cast your peepers on the sp [stack pointer] display. #$71 sounds a mite low, unless that kernal had the same troubles as ours. Now back to the main thrust of the article.

Every time you sys 5, and enter the monitor, the stack grows by five bytes! When you .x, you do indeed warm start basic, all program lines are as they were, and you can even execute the basic command 'cont', if appropriate.

I will save the mildly curious from repeating my next experiment, although you are invited to try if you wish. Sys 5 and .x, over and over, watching the stack pointer get lower and lower, until...lo and behold, the dread 'out of stack' message appears.

To keep things straight, if you are going to run some program and repeatedly inspect it's results via the monitor, either:
1. Always use 'run' for the basic end of your program. Not 'goto 7500' in order to inspect what that portion of the program is doing. Rezootify the stack!
or
2. Include the legitimate but truly underutilized, 'clr' in your program fragment.

I have recently developed a new working hypothesis concerning CBM and the b. There are LOTS of goodies tucked away in there that we were never told about, but in almost every case there is a tiny bug involved. The kind of bug that I can get away with, ...You didn't pay me for it, so you don't seriously expect me to ..., but that CBM couldn't. It is perversely to their credit that they didn't hype the non-bulletproof features.

ON BANK SWITCHING, RPN,...

By: John A. Berezinski

<<While this material is intended for the library section, the potential from Mr. Berezinski's efforts, particularly the bank switching work, merits wider audience. The Bank Switching work is still in preliminary stages. Those with the expertise to assist are welcome to join the efforts. This may not be suitable material for the average member.>>

- - - - - - - - - - -

This disk contains a set of programs for bank switching.

```
4     "bankall4096"\      prg
4     "bankall8192"       prg
4     "bankall16384"      prg
4     "bankall24576"      prg
```

The banking programs are all variations of a master.
The number suffix on each program name indicates address location ahd allows you to choose where you want it.
It is activated by a sys to adress in suffix.
All programs alter irq vector.
Because Calc result finds all memory, none of these programs will function if Calc Result is run.
Bankall####.will function with superbase and superscript running but may leave blinking cursers in strange places on the screen. They do not seem to affect the functioning of the programs.
Programs require 1 meg. I use Andreson's board and put my programs at $1000. If you do not have this area on your board you will also need ram in the cartrige slot $2000-$7fff.
All programs are experimental and may crash at any time. I would not use them for important work until they have been tested out for a while.

BANKALL#### when loaded and activated will allow you to load a program and even run it and store it in one of your banksets. Set 1 is banks 3,4,5 Set 2 is banks 6,7,8 Set 3 is banks 9,10,11 and Set 4 is 12,13,14. The fifth bank set is our operating system, banks 15,1,2. Bank sets are accessed by shift/ENTER key on numeric keypad then key 1-9 on either keypad.
1 stores in bankset1 3,4,5
2 stores in bankset2 6,7,8
3 stores in bankset3 9,10,11
4 stores in bankset4 12,13,14
5 recalls from bankset1 3,4,5
6 recalls from bankset2 6,7,8
7 recalls from bankset3 9,10,11
8 recalls from bankset4 12,13,14
9 aborts routine and returns you to whatever is running

A quirk in the program is that the number you pick will sometimes be printed when control is returned to current program. It can be deleted. However in superbase it will be taken as a comand, If you choose 2 to store S.B. in bank set2 and you were on menu 2 at the time, S.B. will go into format mode and ask you to reformat. You can either use the escape routine or make sure you were on menu 1 when you use the 2 command for bankall. When bankall surrenders control, S.B. will go to select submenu. You can return from that menu and proceed as normal. So when you pick a bankset number be prepared to act on the menu comand which will result

An alternate and neater way to use bankall#### is to load ypur program and store it in a bank set, then call it from that bankset and run. If you have Casey's scrubber and unprotected Superbase or Superscript, you can proceed as following.
1 Before loading Superbase loader use bankall and store current memory in bankset 1 (shift/ENTER) z1)
2 Load Superbase loader
3 delete line 80
4 run program
5 use bankall and store in a bankset other than 1
6 recall bankset1 if you do not yet need Superbase.
7 when you need Superbase use shift/ENTER # where # is bankset where S.B. is stored plus 4. If you stored S.B. in bankset2 then # would be 2+4 or 6. When S.B is recalled, you will still be in basic. Type "bank15:sys1024" and S.B. will be activated.
8 when you want out of Superbase, first make sure all

bus channels are closed by returning to main menu1 or menu2, then hit shift/ENTER and then 1 . This will return you to basic.

To return to Superbase, simply repeat step 7. You will be starting S.B. from the begining and will have to reopen database and file. However you do not have to wait for disk load.

The same technique applies for Superscript and using bankall will allow you to load S.B. in one bankset and S.S. in another. Then call and sys to program needed as often as you want. If you find yourself using S.S. and needing to refer to information in S.B., this program will save you the time used in loading and exiting these programs.

IMPORTANT any program which disturbs the IRQ vector ($0300) will disengage this program, to reactivate sys to loading address. Standard exits from Superbase or Superscript will do this.

If you use banking from with in Superscript, you will not be able to save or print your letter normally. Keep S.S. memory set on bank2 when switching banksets, when you enter S.S. form a different bankset while it is running, call bank 9 using ESC J 9. S.S. will now use bank 9 for memory and appear to function normally. When you exit through a bankset, first go back to bank 2, ESC J 2.

HEXPRIMER.S is a primer on hexidecimal and microprocessers. It is meant to be the first in a series on assembly language programing.

BLACKBOX is a puzzle game which functions like pool or a pinball game in its actions, but the bumpers are hidden and you must find them. The program will print a graph if you need it, but make sure you shut off then turn the printer back on before you print something else. The cue is controled by the 4 8 9 2 keys with 5 firing the ball. When a ball is fired, the q will be replaced with a letter and that letter will also appear elsewhere on the boarder of the board. Its resulting location will be influenced by what bumpers if any the ball ricocheted off of, if any. There are four bumpers generated on the board. When you think you have located all of them, place the cursor in the lower right hand corner of the board and hit the fire button, the board will be revealed. One flaw of the program is that the q tends to disappear in the corners, it still there and will appear when you move it.

RPN is a series of programs developed from a master. They are loaded in memory according to adress shown in suffix and activated by a sys to same adress.

Stack is 256 deep, but a number takes up five or six spots. RPN uses the B128's software and will recognize most of the math commands available from basic.

Commands are

x exit

d duplicate top of stack

? print top of stack

+ add top two numbers on stack and leave on stack

- negates top number on stack positive becomes negative and negative becomes positive. To subtract, you would enter the first number, then the second number, hit -, return, +, return, ?

sgn I don't remember, but its in the book.

int Leaves the interger half of a whole number.

abs Leaves the absolute of the number on the top of the stack.

sqr Takes the square root of a number.

e erase drops top number from stack

s will swap the top of the stack with the next number down.

r will rotate the third number down to the top, the top number becoming the second one down.

This program should respond to most of the math commands.

It is in machine language and should not affect basic, although when exiting, you will get a syntax error. Ignore it.

<<RPN is a well known scientific method of expressing mathematical formulae vs. the more commonly known algrebra. RPN stands for Reverse Polish Notation having been developed from the work of a famous Polish mathematician. RPM is promoted by Hewlett Packard in their calculator and other product lines. Most of us of scientific bent find RPM far more efficient and easier to use than conventional algebra.>>

John A. Berezinski
512 Apache Ave
Carpentersville, Il. 60110
312 426 2221 6pm to 10pm ONLY

ROUNDOFF ERROR

By: Mathew Goldstein

Although I am no mathematician I think I may be able to help clarify the issue of round-off error raised by Art Kinger in the Winter/Spring edition of Escape. Mr. Kinger asks if Microsoft BASIC can correct round-off errors by converting the number to a string and then recovering the string to it's numerical value as in i$=str$(1/3):i=val(i$). Indeed such a method is helpful under certain circumstances such as the for next loop with a step size of 0.1 as Mr. Kinger points out, and I thank him for bringing this to our attention.

When is this technique helpfull and when is it not and why? First let's take a closer look at how the B128 uses numbers. Although we only (only?) see nine decimal digits on the screen the computer appearantly recognizes numbers internally equivelant to at least eleven digits. I say equivalent to because, as you may know, the computer represents numbers in binary form (1's and 0's). I say at least eleven digits because the nine digits displayed on the screen can be made to round up one in the last (least significant) digit when I alter the tenth and eleventh digits. In other words, the computer "rounds-off" all numbers to 9 digits before displaying on the screen dropping the never seen but nevertheless existing tenth and eleventh digits. I suspect this is done by the operating system software but I don't know. I say round-off in quotes because what can be called the resolving power of the B128 is too limited to allow true round-off to nine digits. For example, the number 3.3333333334 would normally be expected to round to 3.33333333 but in fact it rounds to 3.33333334. This imprecision is an inevitable result of the limitations imposed by binary number representation as explained by Warren Swan in the Summer 1986 Escape and is common to all digital computers. It can never be eliminated but it can be reduced by representing numbers with more binary digits. Therefore computers with 32-bit arithmatic logic units (32 bits means 32 binary digits) are inherently more precise than a 16 bit arithmatic logic unit computer.

Since any nine digit number entered into the computer will be properly displayed as the same nine digit number we can say that the B128 has nine digits precision. This does not mean that every numerical computation is correct to nine digits. It does mean that nine digits precision is the BEST that can be hoped for. For example, try 100*(1/3)-33. The result will be correct rounded to eight digits only. 100*(1/3)-333 is correct to seven digits only, rounded it is correct to only six digits. Similiar results obtain from 1/3-0.333. See if you can find an equation whose results are incorrect when rounded at the first digit (0 digits precision).

Now we can return to the original question. When a number such as 1/3 is assigned to a string as in ot$=str$(1/3) the length of the string will be at most eleven including the initial blank character (space) for the assumed + or the - sign and the decimal point (very large and small numbers will also have an exponential which increases the length to 15). This leaves us with nine digits which is two less than the eleven decimal digits by which 1/3 is numerically represented inside the computer (in binary form, of course), even though we never see more than nine digits on the screen. In effect, the string function assigns to the variable the screen representation of the number (rounded to nine digits). When we return the string to it's numerical form (ot=val(ot$)) the tenth and eleventh digits are no longer there - they are now zeroes.

If we assign i=0.1 and add i five times then multiply by 100 and subtract 50 we get 0 as expected. But if we add i six times then multiply by 100 and subtract 60 we get 1.49...e-08. An error of about 1.5 has been introduced into the eleventh digit. If we printed i+i+i+i+i+i we would still see 0.6 on the screen. But inside the computer we have approximately 6.0000000002. As we continue to add 0.1 we continue to accumulate errors in the eleventh digit which eventually spills over into the tenth digit and then the ninth digit. Sometime before the error spills over into the ninth digit we will see an incorrect result on the screen as the tenth and eleventh digits force the ninth digit to round up one. If we ask the computer if 0.1 added seven times equals 0.7 the computer unambiguously tells us no even though 0.1 added seven times shows up as 0.7 on the screen. If we convert back and forth between alpha. and numerical representations of the sums we will be truncating off the tenth and eleventh digits (where the error accumulates) thereby giving the correct results.

Great! But wait a minute. It is not always helpfull to truncate the tenth & eleventh digits. Although these digits do not resolve well, and accumulate round-off error, they do approximate the correct number somewhat.

For example, 100*(1/3)-33 gives .333333328 but 100*(str$(1/3))-33 gives .333333299. The former has 5e-09 error while the latter has an error of 34e-09. Clearly we do better by keeping the incorrect (but almost correct) tenth and eleventh digits (they are 2 and 8) then dropping them.

Only if we know that the tenth and eleventh digits of a calculated value are always zeroes should we convert the number to an alpha string and then reconvert it to a numerical value to eliminate accumulation of round-off error in the tenth and eleventh digits. Otherwise leave it alone, nothing, as far as I can see, can be done short of writing a double precision ml routine, provided that the hardware supports double precisions. Also note that 100*(val(str$(1/3)))-33 is the same as 100*(int(1e+09*(1/3))/1e+09)-33 although the two methods will not produce the same results under all circumstances. Which leaves me with a question. Has anyone tried the ml routine in appendix B of the Programmers Reference Guide for double precision arithmatic?

INSTRUCTIONS FOR NUMERICAL METHODS PROGRAMS

by Mathew Goldstein

<<Once again this is material normally placed in the library section however the material is appropriate for publication as well. See also the rest of Mr. Goldstein's fine work in the library section>>

These programs are the result of undergraduate coursework at Rensselaer Polytechnic Institute. A textbook "Numerical Methods for Engineers with Personal Computer Applications" by Steven C. Chapra and Raymond P. Canale was the source of the information on which these programs are based. As the title of the book implies these programs are very useful for a wide variety of technical applications and are especially well-suited for the computer but may be of little utility to most other people.

To reduce round-off errors that result from the computers binary representation of fractions numbers are converted to integers by multiplying by an approriate power of ten before subtraction. A brief introduction is included in rem statements at the beginning of each program. Because many people who may need these programs have access to other computers which may have greater precision or be more accessible in times of need the programs were written for the most part using standard ANSI BASIC without machine specific commands.

The polyregress program places the derived equations in the function keys. The equation can then be run as a mini-program with different values of the independent variable. This feature permits interpolation or extrapolation using higher order (and therefore larger) equations, and allows you to see the equation directly but it may make it somewhat more difficult to transport the program to another computer.

One or two of the programs provides you with the option of entering the data manually or reading the data from data statements in the program. Future versions of these programs could include both options all the time plus an option to enter the equation from the keyboard. I am taking another, more advanced course in numerical methods this semester so there may be more programs in the near future. Some of the commands in the JCL workshop extended basic, especially 'evaluate', are well-suited for this type of program, and may find there way into future programs, the ANSI-standard not withstanding.

There is an arbitrary limit of ninth order estimates in the romberg integration program. As the order of the estimate increases so does the time required to do the calculations. If you can set the program running and return 8 hours later you may want to remove that limitation in the program, but remember the B-128 only utilizes nine digit numbers and my experience suggests that the precision is just a few digits as would be expected on an 8 bit machine

Another limitation is the amount of computer memory. The warnings that are displayed are valid for the B-128 , although the numbers are conservative and in most cases can be increased slightly. For example, several programs that allow only 100 data points could work with 110. However, the arrays are programmed for a maximum of 100 elements.

Default values for certain inputs are provided as a guide. Some programs require the function being analyzed to be entered into the program, other programs allow functions to be included in the program as an option. In either case the line number where the function belongs is displayed. Certain inputs must be within a certain range of values. If you try to enter a value outside of that range you will be prompted again for another input with an error message that specifies the correct range.

Freeware notice in 'FROM.MEG' file

JCL INSTRUCTIONS ERRATA

By: David Ritterbusch

The following errors have been found in the JCL Instruction files on the CBUG #51.i disk.

The best way to find the location of the error is to use the ESC G function of Superscript II. For example: ESC G 248 (return) puts line 248 at the bottom of the display. Simple cursor over to the mistake and make the corrections.

CAUTION:
DO NOT PRESS (RETURN) AFTER MAKING THE CORRECTIONS.
unless instructed to do so in the change/to comments.

Pressing (return) in the middle of a line will delete the remainder of the line causing a greater mess.

The number in the line column is the Superscript II document line counter number. The number in the page

column is the page number of the printed document. The page number is given so that you can make pen changes if you have already printed the JCL instructions.

FATAL ERRORS: The missing @

file name	line	page	change	To
700wkshp doca.ss	248	5	Print(x,3)	Print@(x,3)
	249	5	Print(1,6)	Print@(1,6)
	390	8	the '' key	the '@' key
700wkshp docb.ss	71	16	PRINT(h,v)	PRINT@(h,v)
	110	17	PRINT(h,v)	PRINT@(h,v)
	186	18	PRINT(1,5)	PRINT@(1,5)
	366	22	PRINT(7,4)	PRINT@(7,4)
	443	24	PRINT(6,x)	PRINT@(6,x)
	631	28	PRINT(H,V)J	PRINT@(H,V)J
	655	28	PRINT(15,S)	PRINT@(15,S)
	660	28	PRINT(X,X)..	PRINT@(X,X)..
	664	28	PRINT(15,12)	PRINT@(15,12)
700wkshp docc.ss	139	32	PRINT(1,4)	PRINT@(1,4)
	439	38	PRINT(1,25)	PRINT@(1,25)

NON FATAL ERRORS:

file name	line	page	change	To
700wkshp docc.ss	220	34	left bracket	left '('
	405	38	runs of line	runs off line
	413	38	site, and a	site, with a
700wkshp docd.ss	80	45	delete 'In addition' and change 'up' to 'Up'	
	112	46	to be	are
	135	46	delete one of the two 'in's	
	138	46	and a these	and that these
	157	47	FIXMEM	FIXMEM(return)
	529	55	support	with
700wkshp doce.ss	280	65	SYMBOLS	SYMBOL
	304	65	assembly.	assembler.
	307	65	add period to	LIB.
	315	65	These	There
	325	66	is means	is a means

COMMENTS: I have found that the documents appearance is better if right justification (ju1) is not used. If right justification is used, additional forced spaces will be required to keep sentences lined up properly.

SEARCH & REPLACE: These commands will replace the right justification commands in the current document in memory to no justification.

```
        ESC S (ju1) (return)
              (ju0) (return)
        ESC @  M    (return)
```

ADDITIONAL INFORMATION & CORRECTIONS: When the following changes are made to the file document the line number refernece of the original document will change. These should be the last changes made. The line numbers listed are correct if the changes ,are entered in the order listed.

file name	line	page	change	To
700wkshp doca.ss	293	7	delete line (*ln3)	
	325	7	insert a blank line Add 4 spaces and type in the following statement:	

[The display shows cc as 8 but the boot loads 24k of mcode.]*cn1

file name	line	page	change	To
	349	7	remove 5 spaces after JCL (do not press return)	
700wkshp docb.ss	86 87 88 89 90	16	Delete at least 3 spcaes from in front of the :REM's. in this line sequence.	
	110	17	remove all or at least 2 of the periods and all but one space after the PRINT@(h,v) statement.	
	439 440 441 442 443 444	24	Delete at least 3 spcaes from in front of the :REM's. in this line sequence.	
700wkshp docc.ss	516	40	delete one of the two 'with' s	
700wkshp docd.ss	63	45	change *ln10 to *ln5	
	89	45	Insert 2 blank lines Type 17 spaces and then type the following statement.	

[RAMSIZE is $6000 (24K); see pg 7.] (return)

file name	line	page	change	To
	90		Press (return)	
	98	45	Insert 2 blank lines Type 4 spaces and then type the following statement.	

[The byte is at $5FCE (24526); refer to LAST LETTER on pg. 23.]

file name	line	page	change	To
	99		Press (return)	
700wkshp doce.ss	115	61	Insert a blank line. Type ORDER.(return) under the word DISPLAY	
	114		Delete the word ORDER by placing the cursor just past the word ALPHA and press (return).	

COMMENT: SCRINP command : 700wkshp docc.ss lines 222 & 223, pages 34. I do not understand the phrase 'plus space and fullstop'. I can get the following to work for SCRINP field limiters placed in front of the left '('.

Letters only	: Shifted Return
Numbers only	: '#'

I also had to include an extra 'print' command in a program to get it to print what had been input by the SCRINP command. EG:

```
10 Print "Enter a number #( )"
20 scrinp (1)
30 inpfld (1,x$)
40 print            : REM phantom print command
50 print x$
```

NOTE: 700wkshp doce.ss line 452, printed document page 68, refers to an Appendix VI which is supposed to be an explanation of error codes. I have not been able to find this appendix or a list of error codes.

SUMMARY: There are numerous (162) other changes that I have made to the JCL instruction documents, too many to list here in the ESCAPE.

THE OFFER: If you send me your CBUG #51.i disk of JCL instructions I will replace the JCL files with the latest revisions and a printed copy of the 72 pages of instructions with the following two choices:
Disk + Instructions with no binding - $ 9.95
Disk + Instructions with spiral binding & covers - $12.95

Contact: David Ritterbusch
7201 W. Paradise Dr.
Peoria, AZ 85345

<<Dave undertook this entire project on his own. This

type of pitching in is what makes CBUG work. Let's hear it for Dave!>>

COMPRESSED SPACING ON A 4023 PRINTER USING SSII

By: Randolph Newburger

Something interesting happened to me a few short days ago and it might be of some interest to other members also. I will try to put this letter in article form that it might be printed in the next edition of the CBUG Escape. Although I am really not sure if it can be of any use to anyone, maybe the "Gurus" can make some kind of sense out how it can be applied to other printer types.

I was happily preparing a report for my company one day, using Superscript II with my Spell Check disc in drive 1. After I had finished my report and filed it to disc, I exited from SSII by switching off my "baby B" and switching it back on. I then replaced SSII with Kernaghan's Utility disc (leaving in place the Spell Check disc). I then proceeded to dload"bargraph" to create a graph that would spice up my report. Now comes the sexy part, after printing out the graph on carefully spaced paper, I removed Kernaghan, and replaced the SSII disc in drive 0. I then depressed SHIFT/RUN and noticed that the screen was still in the GRAPH mode as SSII started its initial "bump and grind", but before the SSII edited screen appeared. I immediately depressed SHIFT/NORM. SSII appeared to load with no problems. I reloaded my work disc and booted up the file that I wished to header my graph with. Now strange things began to happen. The page seemed to print out ok, it looked like I had misjudged the paper's spacing. I then booted a different page and printed it. Low and behold it only printed in half the space that it should have. I immediately started to look for some explanation. At the top of the page was the legend: *PE:T*. I then got out my not so trusty 4023 operating guide and proceeded to leaf through the error codes (anyone of the real experts out there are way ahead of me at this point!) and found I had created a secondary address printer error! WOW, I don't know even know what that means. I then concluded that I had created "COMPRESSED TYPE" on my 4023. This was impossible according to the SSII User's Manual.

At this point I shut down my system, showed my wife proudly what I had created ("So what! My IBM at the office can do that anytime.") Facing this rejection from my own beloved I proceeded to go back to my "B", load a terminal program, and send Norman a FORUM message via DELPHI. Your answer was to write this letter and see if someone could unravel what I did.

In the meantime I studied the two printouts that I first thought were compressed type and they turn out to be compressed spacing! I immediately went back to the SSII manual because I remembered reading something about line spacing in the Reference section. On pages 123-124 I read sections 9.4.5 Line Spacing and 9.4.7 Lines Per Page, either which may have been a possible solution to what I had accomplished. No banana! The commands didn't work. In fact, even repeating the process didn't work!

Next, I thought that I would make an attempt to solve my mystery with some expert help. I dusted off an issuue of "Escape" and called Progressive Peripherals and Software. Again no joy! They didn't have any information on SSII commands on a 4023 printer. It was too antiquated, they said!!! Now, I was going to end this here with an impassioned appeal for some one out there to help, but I was mad. I know this could be accomplished!!

Again I returned to both my printer and SSII manuals. It was now time for trial and error. After a couple of experiments I had the answer! The proper command should be (RVS)*sa6,22<(RETURN) repeated twice. It won't work if you don't repeat it. This command sequence increases the lines per inch of output to 9 lines per inch. This about doubles the print per page that outputs from a 4023 printer. This means less wasted space and more print per inch. So, my original premise was not correct. You can't have compressed print on a 4023, but you can have more lines printed per inch! I, at least learned something!!

Randolph Newburger
529 Clemson Drive
Pittsburgh, PA 15243

GRAPHICS FINALE

By: Neil Cumfer

This report is the last in a series of three reports explaining how you can print B-128 graphic characters on the 4023 printer when using SuperScript II. Go back and re-read the two earlier reports. Three essential changes are required. Line 3140 is incorrect; replace the second, fourth, and eighth items of data so that the line will read as follows:

3140 data 172, 192, 162, 93, 164, 165, 167, 95, 187, 175

Then run the program; it will replace the "ss2 tables" file on your disk in drive 1 with the correct one needed for the 4023 printer to print out all of its characters and graphics.

The table presented in the second report is reprinted here with corrections.

keystroke sequence	screen graphic	effect on 4023 printer
ESC $	reverse $	prints the British Pound sign
ESC (	reverse R	prepares printer for reverse field printing
ESC)	reverse r	cancels reverse-field printing
ESC [(left bracket)	reverse W	prepares printer for enhanced printing (W=wide)
ESC] (right bracket)	reverse N	cancels enhanced printing (N=narrow)
ESC '	reverse left-arrow	forces a carriage return on the printer without a line feed
ESC >	reverse u	puts printer in temporary Upper case & graphics
ESC ;	reverse 1	returns printer to Lower-case & upper-case mode
ESC 1	reverse 1	prints graphic shown on key V (this graphic is normally produced in graphics mode by control-C)
ESC 2	reverse 2	prints graphic shown on key 2
ESC 3	reverse 3	prints chr$(223), text mode version. See page 117 in your gray manual.
ESC 4	reverse 4	prints graphic shown on key 4
ESC 5	reverse 5	prints graphic shown on key 5
OESC6	reverse 6	prints graphic shown on key 6, only after ESC >
ESC 7	reverse 7	prints graphic shown on key N
ESC 8	reverse 8	prints graphic shown on key 8
ESC 9	reverse 9	prints graphic shown on keypad's / key
ESC 0	reverse 0	prints graphic shown on key 0

Use CTRL (numeric) for the graphics on the odd-number keys of the main keyboard (1, 3, 5, 7, 9), and use ESC (numeric) for the graphics on the even-number keys (2, 4, 6, 8, 0). In the case of 5, either CTRL or ESC will work.

Use the new chart included in this report if you have trouble finding the right combination of keys to produce a desired graphic. First, find the graphic you want to use in the gray manual, page 117. Note the number identified

with it in the CHR$ column. Then find that number in the **left column** of this chart, and press the keys it tells **you. When** use of the control key is called for, you must **hold down the** control key while pressing the next key; **don't let it** up too soon! Use the keys on the keypad only **when it specifically** tells you to use them instead of the **same key** on the main keyboard.

The first part of the chart shows the keystrokes you can use when the printer is in its lower-case mode.

The second part of the chart shows the keystrokes you can use for the graphics that can be produced only when the printer is in its graphics mode. The sequence ESC > is in parenthesis to remind you that the printer must be in graphics mode. Do not key in the parentheses! You do not necessarily have to key in the ESC > sequence if you are using several of these graphics together. Once you have set the printer to the graphics mode, it will stay there until you reset it to lower-case, or until the end of the print line, WHICHEVER COMES FIRST. You will have to use ESC ; after these graphics to get back to lower-case mode on the same line, where you need to be for printing graphics shown in the first part of this chart, or for text.

Remember, SuperScript will command the printer to use its lower-case mode a‡ the beginning of every print line, so choose your margins carefully. If you have a print right-margin of less than 80, AND you key in ESC > once followed by several of these graphics, AND the printer reaches that right-margin in the middle of these graphics, the ones that are forced to a new line will not be printed as the graphics you want because the printer is forced into lower-case mode at the start of the new line.

****

91	control keypad 3
92	escape $ (escape shift 4)
93	see below
94	pi
95	control left-arrow
161	control 1
162	escape 2
163	control 3
164	escape 4
165	control 5
166	control pi
167	see below
168	control '
169-text column	control + (control shift =)
169-graphics	see below
170	control keypad -
171	control keypad +
172	escape 0
173	control keypad 0
174	control keypad 2
175	escape 9
176	control keypad 1
177	control keypad 4
178	control keypad 5
179	control keypad 6
180	control keypad 7
181	control keypad 8
182	control keypad 9
183	control keypad ?
184	control keypad CE
185	control keypad *
186-text column	control right-bracket
186-graphics	see below
187	escape 8
188	control -
189	control keypad
190	control 7
191	control 9
192	escape 1

193-218 see below; push shifted alphabetical keys to produce capitalized letters on the printer only if you are in lower-case mode

219 see below

220	pound sign (shift left-arrow)
221	escape 3
222	see below
223-text column	escape 7
223-graphics	see below

****

93	(ESC >) shift right-bracket
167	(ESC >) escape 6
169-text column	see above
169-graphics	(ESC >) control =
186-text column	see above
186-graphics	(ESC >) control right-bracket

193	(ESC >) shift a
194	(ESC >) shift b
195	(ESC >) shift c
196	(ESC >) shift d
197	(ESC >) shift e
198	(ESC >) shift f
199	(ESC >) shift g
200	(ESC >) shift h
201	(ESC >) shift i
202	(ESC >) shift j
203	(ESC >) shift k
204	(ESC >) control left-bracket
205	(ESC >) shift m
206	(ESC >) control < (control shift ,)
207	(ESC >) shift o
208	(ESC >) shift p
209	(ESC >) shift q
210	(ESC >) shift r
211	(ESC >) shift s
212	(ESC >) shift t
213	(ESC >) shift u
214	(ESC >) shift v
215	(ESC >) shift w
216	(ESC >) shift x
217	(ESC >) shift y
218	(ESC >) shift z
219	(ESC >) control keypad 3
220	see above
221	see above
222	(ESC >) pi
223-text column	see above
223-graphics	(ESC >) control left-arrow

****

One additional graphic can be printed on the 4023 from SuperScript. It is shown in the gray manual, page 115, as screen-display poke-code 94 (set 2). To produce it, simply push the up-arrow key (shift-6).

Some graphics can be produced in either lower-case or graphics mode, and some can be produced using a different combination of keystrokes than the ones shown. If you find a more convenient sequence for a particular character, jot it down on this chart so you'll be able to remember it the next time you need to use it.

Now for some technical notes on the maketables program. The data in program lines 2710, 2810-2910, and 3000-3060 correlate to each other as shown in the following table. Values shown are those recommended to set up SuperScript for printing graphics on the 4023 printer.

cbm-ascii identifying which key press after ESC		poke codes for the character presesented on the screen	cbmascii value sent to printer on output
(	40	210 reverse R	18 reverse on
&	38	128 reverse @	0
<	60	128 reverse @	0
[left bracket	91	215 reverse W	1 enhance on
;	59	140 reverse 1	17 cursor down (lower-case)

, **comma**	44	128	reverse @	0	
' **apos-trophe**	39	159	reverse back-arrow	141	shift return/no line feed
)	41	146	reverse r	146	reverse off
!	33	128	reverse @	0	
>	62	149	reverse u-	145	cursor up (graphics mode)
] right bracket	93	206	reverse N	129	enhance off

Line 3140 contains the cbm-ascii values sent to the printer when the sequences ESC 0 through ESC 9 are used. These poke a reverse digit to the screen.

In the second report, I failed to tell you non-programmers exactly how to delete lines 2810-2920. It's quite simple really! Just type this in on a blank line:

```
        delete 2810-2920 <return>
```

To replace it, just type this in, again on a blank line:

```
        2810 data 210, 128, 128, 215, 140, 128, 159,
146, 128, 149, 206, 128
```

and end the line with a carriage return.

Before running the program, you will want to check it to be sure the changes you made were incorporated correctly. Just type in (on a blank line, of course):

```
        list <return>
```

and watch the program scroll down the screen. Freeze the scrolling by pressing down on the Commodore key so you can read the statements. Then, resume scrolling by touching any key.

I am submitting a file named "finalgrafxtables" to the CBUG library. This file is a replacement for "grafx maketables", which should be discarded. It will contain all the corrections to "ss2 maketables" which have been described in this series of reports. Remember, you only have to make these changes once--the program will create a file "ss2 tables" on your disk, which you can use over and over, every time you load SuperScript--so give it a try! You too can have graphics today!

Still doubtful? I am also submitting a sequential file created in SuperScript, named "4023graphicsdemo". Load this file into SuperScript, then output it to the printer. It will print the graph you saw when you ran the "printer demo" on the 8050 demo disk. Study the screen, comparing it with the printed graph, until you understand all the ins and outs of switching back and forth between graphics mode and lower-case, enhanced and reverse printing, and overprinting on the same line.

Vertical graphs are just as easy to create! Here's how I made the quarterback salaries portion of that file. I defined the row of graphics for the chart at the top, then used the ESC A (append) command to copy it several times, as explained in the SuperScript tutorial on page 25. Finally, I cut off the tops of the columns, each in turn, by defining a row of blanks and doing a column move down until each column was the desired height. The command *sa6,21 tightens up the printer's line spacing so that the graphics will connect. It's used at the end of the second line before you want it to go into effect. *sa6,36 will get you back to the normal 6 lines per inch. Thus text alongside the chart should be double spaced. As you can see, the 4023 is not too precise about lining up the vertical lines, so some graphics will be more suitable for this application than others. Have fun decorating your documents with fancy output like this, courtesy of SuperScript II!

CASEY'S SCRUBBER
NOTES FOR NON PROGRAMMERS

By: Marilyn Gardner

Jack Casey has given us a set of programs which allow machine language programmers among us to modify our copies of Superscript and Superbase. The blurb for Jack's disk states that it also allows us to "make unprotected disks of these programs in a manner that makes loading both easier and faster." This statement may have led some non-programmers to purchase Casey's Scrubber, and they may have been disappointed. For one thing, the loading isn't much faster. The following load times were timed with a stopwatch:

Casey's Superscript	42 seconds
Superscript original	53 seconds
Pre-Superscript plus Superscript	64 seconds
Gociliak's Superscript with Pre-Superscript	55 seconds

The most time saved by Casey's Scrubber is 22 seconds. Not a whole lot for $19.00, is it? But if you use Superscript often, it could be worth it. <<Remember too, that you get an unprotected copy, (or atleast a copy in the case of Superbase). Most of us anxiously shelled out $20.00 for Knight's copy all. For Jessie, his program was quick work and we gladly paid for his expertise. In Casey's case it was a huge project involving hardware and software from which his royalties might represent 1897 minimum wage rates...a few pennies a day!>>

If you made a disk containing both Superscript and Superbase, you may have noticed that Casey's Scrubber didn't make a loader program for you, as it did when you made either Superscript or Superbase separately. You can make a separate disk for Superbase, with the one line loader program as the first program on the disk (10 dload"sb7",d1), or you can use the following program for a loader program as the first program on any disk and it will load Casey's Superscript or Superbase:

```
 10 print"Superbase?";
 20 get a$: if a$ = "" then 20
 30 if a$ <> "y" then dload"ss2",d1
 40 printchr$(147)"when logo comes on screen, open door
      on drive 0"
 50 print:print"after error message shows on status line,
      close door and press return"
 60 dload"sb7",d1
```

This program can then be loaded with shift/run. It will ask if you want to run Superbase. If you type y for yes, it will print on the screen the reminder about the anomoly and then and load Superbase. If you type n for no or any other letter, it will load Superscript. This short program takes no more room on the disk than the one line loader programs.

A more serious problem with Casey's Scrubber version of Superscript is that it leaves the function keys unprogrammed, and there is no Superscript II program on the disk to add or modify these keys per the Superscript manual. To add function keys to Casey's Superscript program, do the following (Things to type in are shown in capital letters. If you push the graphics key before you start, your typing will look like what is here.):

(1) Put Casey's Superscript (or the combination disk) in
    drive 1 and the Superscript original in drive 0.
(2) DLOAD"SS2",D1
(3) LIST
(4) DLOAD"SUPERSCRIPT II" (use the shift when
    typing the II)
(5) Move the cursor up to line 4 of the listing and press
    return. Continue to press return until the cursor
    is below line 80.
(6) Move the cursor down to the blank part of the screen
    and type 3 GOSUB 160
(7) DELETE 100-150
(8) DELETE 1480-
(9) 1470 RETURN
(10) SCRATCH "SS2",D1
(11) are you sure? Y
(12) This message will be on screen: 01,files
     scratched,01,00,1 You then type, DSAVE"SS2",D1

If you later want to modify the function keys as it describes in the manual, you can load ss2 from Casey's Superscript disk, make the modifications, and scratch the original and save the new definitions. There will then be no need to have a program to load separately as the manual describes.

Casey's Scrubber does set up both Superscript and Superbase to allow less disk manipulation, since a loader program on your data disk in drive 0 loads the program from drive 1 and you could just leave the disks in the drive all the time and not have to switch them around all the time. If you've already done something similar, like putting pre-Superscript on a data-disk and left that in drive 1 and preceeded all your file names with 1: then you might not be as impressed.

Jack Casey did a fine job of releasing the code for these two programs, but Casey's Scrubber is of most value to programmers. The above notes are primarily for non-programmers who may have purchased Casey's disk.

Marilyn Gardner
1630 Madison Street
Evanston, Illinois 60202
(312)866-9159

Independent Commodore Products Users Group (ICPUG)
Superbase Corner Files Notes

By: Jim Kennedy

1. Superbase Corner was renamed Super* Corner when Superbase and Superscript for the C128 were released as many items in the column covered the use of the two programs together.

2. All files on this disk were prepared with Superscript on the C128 and filed to disk on an 8250 drive

3. Since some people have had problems reading long Superscript files into memory with certain word processing programs I have split some articles into Parts A and B as a precaution. Also I have removed virtually all formatting commands to simplify use with other programs.

4. Feel free to use any of these articles but please credit the ICPUG Newsletter.

Jim Kennedy
Chairman & Super* Corner Editor
'Lost Battles'
57 Gunnersbury Avenue
Ealing
London W5 4LP
England

Tele: 011 44 1 993 2634 (from the U.S.A.)

P.S. I'm a retired U.S. Army Lt. Col. living here in London; born in Evanston Illinois (1936), raised in Palo Alto, California (1941-1954), educated at M.I.T. (class '58), and now spending my time involved with bicycle racing and ICPUG. I do some racing myself as a veteran (times are nothing to write home about), organize races, serve on God only knows how many committees and bodies and am Racing Secretary for the Hounslow and District Wheelers cycling club. I also spend a lot of time (about five hours a day) helping to keep ICPUG up and running, review software and hardware in articles for the ICPUG Newsletter and sometimes other commercial publications, and write the odd sports event computer programs usually as a Superbase applications package. I'd be happy to see any visiting computer types who happen to pass through London (particularly if they want to sample some of the British real ales also known as Bitter).

- - - - - - - - -

A NOTE OF APPRECIATION

By: Norm Deltzke

ICPUG was the first of the co-op contacts made by CBUG, or rather by Mark Schwarzbauer while in London last August persuant to earlier contacts over CIS and DELPHI with Mike Todd an ICPUG officer and earlier by phone and the earlier Superbase files via John Tranmer, President of Precision Software also of London. We have agreed to exchange any information of mutual interest -- on disk whenever possible. Jim Kennedy forwarded about 600K of Superbase/Superscript files plus copies of their past issues. While ICPUG has but a few B128 members, much of the Precision software information is highly transportable.

The materials provided by ICPUG regarding Superbase are in part reproduced on the following pages. Those articles which are more obscure to B128 usage are available on library disk -- check the listings in the library section. We've elected not to edit those selected for printing as many of our members have multiple CBM interests.

ICPUG has a membership similar in size to CBUG. With the now pending agreements with the B128 group in Cologne W. Germany, the mutual efforts can't help but be exceptionally successful.

CBUG extends our greatest of appreciation to Messrs. Kennedy and Sehumacher in our mutual efforts to enhance CBM users, both B128 and others of mutual interest.

Thanks again, Jim, for the cooperation.

P.S. ICPUG Subscription rates:
26 Pounds Sterling for Air Mail
15 Pounds Sterling overseas Surface Mail
1 Initial membership co-op share (UK legal req.)

SUPER* CORNER May/Jun 86
Abridged

Edited: By Jim Kennedy
Chairman, ICPUG, London U.K.

<<Note, while the discusson is centered around the C-128 and C-64 versions of the mentioned software and machines (and the 128 is nearly always meaning the C-128, nearly all operations between Superbase for the C versions are similar or identical with the B versions. In the case of Superscript, the program version is similar to our Superscript III.>>

INTRODUCTION

Since Hugh de Glanville had more than enough on his plate with the editorship of the Newsletter, it became necessary to find someone else to take over the editing of Superbase Corner for each issue. Having 'messed about' with this database program for a couple of years I volunteered to try and carry on this really useful column which Hugh started. All should be aware that I have neither the programming abilities of people like Simon Tranmer, Tom Cranstoun, Jack Cohen and Brian Leighfield nor the writing finesse of Hugh and Henry Velleman. Bearing these points in mind the 'Corner' continues.

Policies shall remain the same as they were under Hugh with a couple of minor additions. First, and time permitting, I intend running most programs submitted prior to publication to insure they work as claimed. Now this is normally a luxury which few editors can afford. However I am going to try doing it and see how long I can keep at it. My caveat to authors is therefore to use the PRINT:LIST:DISPLAY command to produce output from a working program.. Better still, if you want to convert a Superbase program into a word processor file on disk then use the trick of fooling the system into diecting the output to disk by

```
pdev 8,8,8:print"filename":list
```

followed by

<reproduced by permission of ICPUG>

```
  pdev 4,7,0:display
```

to redirect the output to where it normally belongs. You can then cleanup the resulting disk file with a word processor. This methods avoids keying in errors which frustrate even the old hands and make newcomers give up in disgust when things crash.

My second addition is to include Superscript in the column where appropriate. Hence the name change. What with the older Superoffice and the new 128 versions of Superscript and Superbase it is becoming difficult to separate these two packages. By the way don't try as I did to store a file on disk with an asterisk in the name such as SUPER* CORNER. Asterisks can wipe out all files on a disk. Better wprd processors such as Wordcraft and Superscript don't accept them thereby avoiding these perils as Hugh pointed out last year in a reply to a reader.

COLLATING SEQUENCES

Superbase assigns characters to their ASCII value sequence during both key field entry and sort operations with two exceptions. First, upper and lower alphabetic cases are ignored. Second, a shifted space is treated as the number 9. Both exceptions arise from the fact that Superbase subtracts 128 from CHR$ values greater than 127. I discovered this when reviewing a suggestion for making key fields unique by entering Jones , Jones. , Jones.. , etc. The problem with this is that one must develope a ruotine to strip away the full stops prior to printing. Two easier methods of achieving unique fields are available. One method I use is to add a number or letter to the last column of the field and then truncate the printed output by one column. Example:

```
     Jones          1
     Jones          2
```

in a 15 column field become Jones when the output is truncated by the command print&14[keyfield].

The other method (I think I learned it from Will Light but apologize in advance if it was someone else) is to use shifted spaces following the entry. These shifted spaces show up on the screen as thick dark bars but do not print. Letting - indicate a shifted space means Jones- and Jones-- print as just Jones. If you look at the ASC value of a shifted space in Superbase you will find it is 185. We can use the following code to do this:

```
  100 a$=[field]:as=asc(a$)
  120 display @2,2a$@2,4as
```

As you probably know CBM ASCII lists the shifted space as CHR$ 160 which is not visible on screen. Superbase traps 160 and reassigns it as 185 (a thick dark bar on the screen) and when we subtract 128 from 185 we are left with 57 which is the same as the CHR$ value for the number 9. This is why Superbase treats shifted spaces as the number 9. Records with a combination of 9's and shifted spaces would sort in the following order: 91, -2, 93, -4, -5, 96, etc. We can think of our man Jones--- as being Jones999, the 9's not printing.

DATE HANDLING

I could not get John Barr's date manipulation program (Jan/Feb 86 p68) to run as listed using Superbase 64. The problem is in lines 110 to 140. Once all the records have been deleted one cannot do a Select Add. Rather one needs to at least one record in the file at all times. It turns out it was easier for me to keep two, call up the last one with Select Last, do a Select Delete on it, clear the one permenant record which came up automatically following the delete, and then do the Select Add. This second temporary record is then deleted the next time the routine is run. The rest of the code works as written and is quite useful. The revised code is as follows:

<reproduced by permission of ICPUG>

```
  100 rem Diary Writer
  110 file "temp"
  120 select 1:select d:clear:select a
  150 a=[date]:b$=[time]:c$=[category]
  ......
```

While we are on date handling I'll mention something I find useful. The SET and DUMP commands may be used to call up and display the date on which a program was last used where it may be either acepted or modified and dumped back to disk as a "date" file. This "date" file may be again Set in later modules and used to automatically date reports or lists. The accept or modify feature means if you last used the program on say yesterday 11 May 86 that date would be displayed and all you need type in is 12 and <return>. To add it the "Start.p" in Superbase 64 one could use the following code:

```
  ......
  170 display @6,18 "Database Selected  : demo"
  180 for i=1 to 500: next i:rem displays name on screen
then continues
  190 database "demo",8,0:rem disk 8, drive0
  200 set "day"
  210 display @0@1,18"
  220 display @0@25,18dt$
  230 ask &11@5,18"Enter Today's Date  :";dt$
  240 dump "day": rem stored as dt$
  250 file "test"
  260 load "prog1"
  400 rem *** set system parameters ***
  ......
```

In the program called "prog1" above one would then place lines such as

```
  10 set "day" : rem recalls the variable dt$
  .....
  2000 print @60dt$ : rem prints today's date at column 60
```

When running the program for the first time type RUN 230 while the "Start.p" program is in memory to create a disk file called "day". Without the file on disk the Set "day" command will result in a "File Not Found" error. Notice I also have "hardwired" the database name into the code which makes it easy to select a database (assuming only one per disk) by inserting the disk and typing Execute "start" (eX"start) which Hugh mentioned in the Jan/Feb 86 issue.

SUPERBASE: THE BOOK

I received a draft copy of this book and this column is probably the place to review it. The book's complete title is SUPERBASE: THE BOOK, A Guide to Database Applications and is written by Dr. Bruce Hunt of Precision Software Ltd. (From now on I'll refer to it as THE BOOK.) It should be published in the near future.

My advice is BUY IT as soon as you can. This is the book which should have been written and bundled with Superbase as part of the manual. It bridges the gap between the three beginner's level tutorials and the reference and programming portions of the manual. If I only had this book when I first bought Superbase it would have saved be dozens or even hundreds of hours of work. It should also reduce the number of phone calls to Precision asking for Superbase help by half.

THE BOOK assumes you have read and worked through all of the tutorials. It starts off in the first of three sections discussing what is a database and how one should be designed. It explains how to organize various files and what sort of field layouts are required for each. It then goes through a thorough discussion of how to use all the options selectable from the various menus such as Select Add , importing data, "list" creation and use, pattern matching, the Batch operation and so forth.

It goes into command line operations in the second part and this in turn leads into simple programming. At this point some experience with Basic would be useful but I

<reproduced by permission of ICPUG>

suspect many will be able to actually learn the necessary parts of Basic from the book. It has lots of examples and shows the output of each one. An example of a useful technique, many of which are scattered throughout THE BOOK, is how to do an ascending and descending sort together. Normally one may do one or the other but not both. The problem arises when one wants to sort something by say last [name] from a to z and within each name the amount [owed] from largest value to smallest. The solution? Create a [result] field where the amount [owed] is subtracted from an arbitrarily large figure, say $10000.00, and sort on this field along with name field. Display or print from the sorted list the [name] and [owed] fields.

The third section goes into advanced programming techniques including the Set and Dump commands, the Do/Perform metacommand and reorganizing the database. And I suspect a lot of people will be doing a lot of reorganizing of databases after they read this book. It has some 'neat tricks' scattered throughout such as how to avoid the 'division by zero' error which was mentioned last issue.

Finally, it has an appendix of almost 30 pages entitled Troubleshooting. This section lists every known bug in every version of Superbase ever produced for any machine. Precision Software Ltd. are to be applauded for their forthright and open policy of telling users what bugs existed and of providing updates and fixes for them. This is the kind of documentation and service one hopes to receive from mainframe manufacturers and software houses. To find it for Superbase is very welcome indeed. Well done Precision.

The price will be <<IS $15.95 U.S. from Progressive Periferals, Denver Colorado>>. My only recommendation to Precision Software Ltd. is that it should be incorporated as part of a series of three manuals furnished each purchaser. (But sold separately it still represents very good value for money.) Such a series should include a tutorial, THE BOOK, and a reference manual with the contents of the present Programming and Reference sections of the existing manuals. My review copy did not have an index but I have been assured one will be in the published version.

I had contemplated writing a couple of Super* Corner columns reviewing for new members what was contained in previous year's Newsletters about Superbase. I don't believe that will now be necessary, most of what I would have written is in THE BOOK. Get one! It has almost put me out of a job before I've properly started it.

SUPERBASE AND SUPERSCRIPT FOR THE 128

I received my 128D the other day and with Superbase and Superscript up and running it was a case of love at first sight. Unfortunately, my machine would not boot the CPM+ disk which came with the machine although it seemed to load in 128 and 64 modes OK. Something was wrong with the drive so back to Corby it went. Consequently I didn't have a lot of time to experiment with it and my initial reactions may change with additional use.

When both Superbase and Superscript are loaded together Superscript has only one of the two text areas available for use (area 2 in fact) and it is not terribly large, about two A4 pages was my quick estimate. This size however may be changed in a trade-off with Superbase program variable space by means of the command Varspace X where X is a number between 1 and 12. Most letters could be written within this space. What would be exciting is some add-on RAM for the machine with a version of Superbase and Superscript utilizing the additional memory. One could then have two text areas in Superscript, lots of variable space in Superbase and maybe some the extra memory in use as a printer buffer and/or RAM disk. A 128D with 512K RAM along Superbase and Superscript would be a very attractive business machine at a very reasonable price.

<<SuperOffice for the B128 (exclusively from N.W. Music, see ad this issue) is much expanded from the C versions. In a minimum 256K B machine (either factory or upgrade) you have 2 full text banks of 577 lines each -- not quite 9 pages of text!! Better yet, if you have a 1 Meg upgraded B, you have 7 banks of 577 line text. When switching between banks, to or from Superbase/Superscript all data remains in memory when you switch between banks or programs!>>

I'll hopefully have my 128D back in time to cover the 128 version differences in more detail next article. In the meantime I leave you with some comparative times. A Superscript spelling check which took 4min 46sec on my SX64 took only 1min 58sec on the 128D. Loading text of 66 blocks took 47sec versus 18 on the 128D. More next time. <<Ah, but the lowley B-128 with 8050 drive loads 66 blocks in 12 seconds, and re-formats it in 14 additional seconds using Superscript II>>

CBM PRINTERS AND SUPERSCRIPT

<<These instructions appear to relate to the Superscript III version for the B128 which are virtually identical to the C128 Superscript.>>

There has been a lot of discussion recently on Compunet and in correspondence from readers about Printer Definition (PD) files, ESCape codes and secondary addressing as well as using the Feature Feature and Printer Feature commands in Superscript. As I use both a CBM DPS 1101 daisywheel and MPS 801 dot matrix printer I'll attempt to pass on some tips about using them with Superscript and Superbase. I won't attempt to answer questions about other printers as I have no way of checking what works and what doesn't.

With Superbase most people will want to do things like change the characters per inch settings on their printers. Many (most?) people use only one printer and it is tempting to place control codes for it in the Superbase "Start.p" system parameters. However problems arise if you place something like

```
 435 print chr$(27)+chr$(31)+chr$(11)
```

to effect 12 cpi printing with a DPS 1101 and ever try to run the program without the printer attached. Things grind to a halt with the "I/O Error No. 5" message displayed as Superbase was trying to send the control codes to the nonexistant device. Place this kind of code in a subroutine executed just before printing is to begin. Changing chr$(11) in the above line to chr$(9) would cause the printer to print at 15 cpi and chr$(13) would return it to its 'normal' value of 10 cpi.

Turning to Superscript we find there are many ways of controlling the printer. I'll cover three of them below. Choose whatever method works best for you.

1) Modified Feature Lines in the PD File. With the MPS801/803 printers Superscript normally prints in Business or Cursor Down Mode (set by line 9 of the PD File). To change and print in Graphic or Cursor Up Mode we can use one of the unused Feature lines in the CBM Dot Matrix PD File. These lines are numbers 18 to 31 in the file (you have to count them carefully). Since two lines for bold print are not used by Commodore dot matrix printers we can 'redesignate them Graphics On (chr$ 145) and Graphics Off (chr$ 17). To do this enter the numbers 145 before before the colon on the Bold On line (line 19) and 17 on the Bold Off line (line 26). Save the modified defaut file to disk and next 'install' it by F1-P-I-P. Now F1-F-B-S and F1-F-B-E sequences will turn the Graphic Mode on and off respectively. When Graphic Mode is on you obtain the graphic symbols on the right side of the keys by typing a shifted key (eg. shifted 's' will print a heart while the unshifted 's' prints the uppercase S). You won't be able to see the graphoic characters on the screen so you'll have to keep making mental swaps between what is on the screen and what will appear in print. It may be useful to have your printer manual handy (pages 23 and 24 in the case of the MPS 801).

2) The Feature-Feature-Number Facility. To obtain graphic symbols with chr$ values higher than 127 (ie.

<reproduced by permission of ICPUG>

<reproduced by permission of ICPUG>

those on the left side of the keys) requires a different approach. To print these symbols, which actually run from chr$ 161 upwards, we place their chr$ value before the colon in one of the ten Feature lines in the PD file (lines 32 to 41). These symbols may be entered by typing F1-F-F-7 if 7 was the number chosen for a particular symbol. Michael Hulme wrote me saying he uses this method for underlining text by printing the symbol corresponding to chr$ 183 on the line below that he wishes to underline. Of course this precludes text on this line but is effective for titles and such which are normally separated from the main body of text. Again don't forget to save changed defauts file to disk and 'install' it.

3) Sending ESCape Sequences. Here we really have two methods of doing this. We may use either the Layout-Enter-Feature command or the Feature-Feature-Escape command followed by the relevant character. An example will help illustrate this. The DPS 1101 has 6 'unprintable' characters on the daisywheel which apparently cannot be invoked by any key on the keyboard. They are printed in the Printer Test at power on with the Form Feed switch depressed however. To access them we use the Feature-Feature-Escape facility. Enter F1-F-F-E-Y to print one of them while z, H, I, J, and K replace Y to print the other five. These 6 characters could also have been defined as Features 1 thru 6.

ODDS AND ENDS

These are a few (hopefully) useful bits I don't know where to place except in this section. It is a sort of follow-on to Mike Todd's DYKT (Did You Know That) contributions but of less importance.

DYKT Loading errors with Superscript and Superbase (and possibly other programs) on an SX64 can be reduced by turning down the brightness control. Simon Tranmer pass this tip on to me. Don't ask me why it works, I haven't got a clue. Probably a hardware bug somewhere. <<See note elsewhere this issue on monitor placement interferance with 8050 disk drive operation.>>

CLOSING BIT

I leave you with a modification of one of the little gems from Bruce Hunt's SUPERBASE: THE BOOK. The full length algorithm was listed last month to avoid the division by zero error. This one is shorter and is useful for averages. Suppose we have a result field we will call [average] which receives the average calculated from the contents of other fields which are summed into a field we call [total]. Another field records the number of actual entries made to these fields we will call [count]. The usual calculation method would be:

[average] = [total]/[count]

But if the [count] field is zero a divide by zero error occurs. They way around the problem is the following:

File definition : example/average

#	name	type	Format/Calculation
1	name	Key	Length 9
2	f1	Numeric	+####.##
3	f2	Numeric	+####.##
4	f3	Numeric	+####.##
5	c1	Result	+# ([f1]=0)+1
6	c2	Result	+# ([f2]=0)+1
7	c3	Result	+# ([f3]=0)+1
8	total	Result	+#####.## [f1]+[f2]+[f3]
9	count	Result	+## [c1]+[c2]+[c3]
10	average	Result	+####.## [total]/(([count]=0)+[count])

Here the terms ([f1]=0) and ([count]=0) assume the value of -1 if true and 0 if false under normal Boolean expression rules in Basic. (THE BOOK has a complete explaination about these matters if you are not familiar with them.) If we actually counted something and placed a valur in [f1] then ([f1]=0) is false and is set to 0 while [c1] will then have 1 entered in it. On the other hand, if we didn't count anything and left [f1] blank or entered 0 then [c1] will also have 0 entered (-1 + 1 = 0). Likewise if we have a finite count at the end of things then [count]=0 is false, its value is 0 and we are left with only the term [count] in the denominator in field 10, just as in a normal calculation for averages. However, when we have a zero count then the term ([count]=0) becomes -1. We will then aslo have a zero [total] but the expression has become 0/-1 and this won't cause an error as we are dividing by -1. The book contains the general algorithm which is longer but also works for cases where [total] may have a finite value with the [count] field at zero. All useful stuff in the right circumstances.

<reproduced by permission of ICPUG>

SUPER* CORNER Jul/Aug 86
Abridged

Edited: By Jim Kennedy
Chairman, ICPUG, London U.K.

64 to 8296 SWITCH

The Corner for this issue starts off with a contribution by Jack Cohen on changing the format of a Superbase database from the 64/128 type using 1541/4040 disk drives to the 8096/8296 type using 8050/8250 drives. For example, such a utility would be useful for a business using Superbase on an 8096/8296 in its office for mass mailouts which has employees working from home keying in names and addresses on 64/128 machines. It has not yet been tested on 1570/1 drives but Jack says it should run alright on those machines. The program is in Basic and is 'run' on the 64 data disk before trying to read it on the 8296 machine. The code is as follows:

```
  1 rem      This program changes the format of a database
in Superbase
  2 rem      from the 64 type to the 8296 type and vice
versa automatically
  3 rem      Load superbase on the 8296/8096 using
8050/8250 drives.
  4 rem      before pressing 'return' when
instructed--switch off 8050/8250
  5 rem      Switch on 4040 (connected to 8296/8096)
  6 rem      If you have used this program then Superbase
will recognise the
  7 rem      Diskette created on the '64' and you can
access the files.
  10 rem     (c) Jack Cohen ICPUG ---for
8050/8250/4040/1541
  20 open15,8,15:gosub660
  30 print#15,"m-r"chr$(252)chr$(255)
  40 get#15,dos$:dos=asc(dos$+chr$(0))
  50 bu=18:if dos=213or dos=183 then t=39:goto80
  60 if dos=43 then t=18:goto80
  70 t=18:bu=4
  80 b=0:i=0:j=0:k=0:j1=0:s=1
  90 print"[cur home]insert data disk in drive '0'"
  100 input"[cur down][cur down]name of database ";db$
  110 gosub600
  140 print"[cur home][cur down][cur down][revs
on]working[revs off]"
  150 dim b$(255)
  160 open3,8,3,"#1"
  170 gosub280
  180 for b=0to7:b$=""
  190 for j=5to20
  200 if b$(32*b+j)=chr$(160)then220
  210 b$=b$+b$(32*b+j)
  220 next j
  230 if b$=db$then
t=asc(b$(32*b+3)):s=asc(b$(32*b+4)):b=7:goto270
  240 next b
  250 t=asc(b$(0)):s=asc(b$(1)):if t=0then print"[cur
home]d'base not found":close15:end
  260 goto170
  270 gosub280:goto350
```

<reproduced by permission of ICPUG>

```
  280 print#15,"u1:"3;0;t;s
  290 print#15,"b-p:"3;0
  300 gosub660
  310 for i=0to255
  320 get#3,b$(i)
  330 if b$(i)=""then b$(i)=chr$(0)
  340 next:return
  350 rem alter parameters from 64 Superbase to 8296
Superbase or vice versa!
  360 if b$(14)=chr$(96)then
b$(14)=chr$(64):b$(15)=chr$(64):goto520
  370 if b$(14)=chr$(64)then
b$(14)=chr$(96):b$(15)=chr$(96
  520 for i=0to255
  530 print#15,"m-w";chr$(i)chr$(bu)chr$(1)b$(i)
  540 next i
  550 print#15,"u2:"3;0;t;s
  560 gosub660
  580 close2:close15
  590 end
  600 for i=1to len(db$)
  610 p$=mid$(db$,i,1):if asc(p$)>127then630
  620 p$=chr$(asc(p$)+128)
  630 r$=r$+p$
  640 next
  650 db$=r$:return
  660 input#15,en$,em$,et$,es$
  670 if en$<>"00"then close15:print"[cur home][cur
down][cur down]attempt failed":end
  680 return
```

A description of how the program works may interest some readers. Line 20 opens the command channel and checks for errors in 660. Lines 30-70 read memory location $FFFC to obtain the byte which identifies the disk drive, select the correct directory track & sector as well as the numbers of such tracks & sectors. Line 150 dimensions an array to the size of a disk sector. Line 160 opens a channel using buffer #1 and goes to 280 to read the first block of the directory into memory. Lines 310-340 get all 256 bytes of this block into memory. The program continues at 180 with an examination of the directory block which is divided into 8 sections of 32 bytes. In lines 180-240 each name is concatenated and when a shifted space [chr$(160)] is reached then the name is complete. Line 230 checks whether the name is correct as requested in 100 and, if not, the remaining ones are examined. Line 250 checks for any more directory blocks and either terminates or repeats the process. If found the program reads the sector pointed to by the appropriate bytes in front of the name. The identification values on this block are then changed to the 8296 type or vice versa. The amended sector is written to disk by lines 520-580.

DATABASE TO DATABASE TRANSFERS

Those of you who purchased Superbase: The Book will find a routine on page 149 to effect database to database transfers of data using two disk drives. This route will possibly be the one most 64/1541 users upgrading to a 128D or 128 and 1570/1571 will use to simultaneously reorganize their database file format and port across all their data. The idea behind the routine is quite simple really. It calls on one drive with the old database and file and reads the contents of each field of a record into a variable. It then opens a new database and file on the second drive and places the variable contents into the various renamed and reorganized fields of the new record and goes back to the first database to repeat the process over again from the next record. You may simply use your old 1541 as drive device 9 and keep your new 1570/1 as device 8.

The program has one small problem in that once you call up another database and then go back to the first one Superbase has lost all knowledge of what is the current record. If you run the program, as I did straight from the book, you will end up with an endless number of copies of the first record from the old database in the new database. There are two solutions to this problem.

<reproduced by permission of ICPUG>

One solution is to delete the record from the first database once its contents have been converted into variables. Then when you return to the first database to get the succeeding record you only have to do a select first to get the next record since the original first one no longer exists. Use a backup copy for the actual transfer as the old file is completely deleted by the process. This does have the advantage that if the transfer is interrupted then one can restart the program again from the first of the remaining records. The additional/changed code is as follows:

```
 35 select d
 ....
 60 database "db.one",8,0:file "old": select f:eof menu
```

John Barrs suggested a second solution. His method is to go back and select the record in the old file by its key and then do a select next. However this works only with unique key field files but by now you should be convinced from what all of us have been saying in these pages over the last couple of years NEVER allow duplicate key fields if there is any alternative. John's code is as follows:

```
 60  database  "db.one",8,0:file  "old":select  a$:select
n:eof menu
```

While on the subject of transferring data comes a timely reminder from our editor and past Superbase Corner suzerain Hugh de Glanville that 'export' and 'output to' are not interchangeable commands. One should use the former command to transfer data to an identical format file. The later has two forms; 'output to' for data transfer to a file with changed format or 'output ... fill to' for word processing with EasyScript or very early versions of Superscript (ICPUG South East variety).

The 'output to' forms have carriage returns between fields as well as between records and it is these latter ones which have to be 'trapped' as explained in the Superbase manual. All of these various commands insert different things in with the raw data in their sequential files. For instance, 'output across all to' and 'output down all to' include the trailing spaces in each field but the 'down' form also inserts a carriage return following each field. The 'fill' form will omit these spaces but inserts an extra carriage return following dates as well as following each record. All forms of 'output to' store a date as 14APR36 while 'export' stores it as the number 13254. Additionally, the 'output to' forms truncate numeric fields to two digits following the decimal point unless the &x,y format command is used while 'export' option leaves numeric fields as they are. Using the 'output to' forms which include trailing spaces can result in files many times larger than ones created by 'fill' or 'export'. This was what happened when Hugh used 'export' instead of 'output to' with a nearly full data disk and he ran out of space which triggered his note to me reminding others of the perils of not rereading the manual.

MORE ON DATES

Most of you probably know that Superbase starts with 1 Jan 1900 which is stored as 1 and continuing through to Friday 31 Dec 1999 which is stored as 36525. Now some people would like to access dates beyond that. The problem lies in the fact that the computer cannot tell 1 Jan 1940 from 1 Jan 2040 in its format of 01JAN40. There is a way around this however. Look at an ASCII code table. Following the number 9 (decimal 57 or hex 39) appears the semicolon (58 or $3A). Now if you key the figures 1jan:0 into a [date] field it will accept them quite happily, display it as 01JAN:0 and tell you that day falls on a Saturday. The value actually stored is 36526. In other words Superbase carries on past the year 1999 as follows:

```
   :0, :1, ...... :9, ;0, ...... ;9, <0, thru =0, .. >0,
.. ?0, .. ?9
              which translates into:
 2000, 01, ...... 09, 10, ...... 19, 20, thru 30, .. 40,
```

<reproduced by permission of ICPUG>

.. 50, .. 59

Using this technique one can continue up to Wednesday 31 Dec 2059 (number 58440) after which point Superbase will refuse to recognize the date. Again, thanks to John Barrs for mentioning that Superbase could recognize date numbers up to 2060.

DICTIONARY

For those of you interested in such things the The Superscript 64 and 128/dictionary is organized as follows:

symbol	letters	symbol	letters
a	a	i	jkl
b	b	j	m
c	c	k	no
d	d	l	p
e	e	m	qr
f	f	n	s
g	gh	o	t
h	i	p	uvwxyz
		u&us	USER

(i.e. dictionaryo file holds words starting with the letter 't')

You may observe the entire dictionary by searching using an *. Note the first word in the user dictionary is the 'word' z'p. This is apparently a marker denoting the start and if deleted will effectively erase the entire user dictionary (at least it did on my 64 version). I include this table as I have somehow 'lost' parts of dictionary files when copying disks and using it means only one bad file has to be scratched and replaced rather than the whole set.

BUGS

Various bugs have been reported when using the 128/1571 or 128D. Some are restricted to Superscript 128 (SS) and/or Superbase 128 (SB) but others are more widespread. Pete MacLaurin of Precision Software Ltd (PSL as you all know by now) has furnished ones known to him and I've added a couple more.

1) Some older versions of SS 128 had dictionary creation problems. PSL have fixed this one.

2) Backup with SB 128 and SS 128 sometimes appears not to work properly especially when disk is half full or greater. PSL are still investigating this one although Superbase 128 v2.06 has solved it for most users but a few others still report problems. These problems may actually arise while the file is being created or written to rather than when it is being backed up.

3) SB Utility64 Program won't work with files whose parts are on both sides of a 1571 disk. This may be related to the problem above and PSL are working on it.

4) ROM trouble with 1571. PSL can do nothing here. I understand CBM has a revised version which has had some limited distributed on EPROMs.

5) Tripler IEEE interface causes all sorts of errors and won't work properly with 1571. Not much PSL or CBM can do here; suggest using another interface. (PSL recommend Brain Boxes.)

6) 1571 goes into 1541 mode unexpectedly. Apparently this one hasn't been fully sorted out yet.

7) DPS 1101 printer will cause SB 128 and SS 128 backup programs to 'hang' or fail. PSL have been notified along with CBM but the only thing to do is disconnect the printer while loading and backing up.

8) 29, Disk ID mismatch, xx, xx error message in SB 128 occurs when a disk has been removed and reinserted in a 128D (and also, so I'm told, in a 1571 and some later model 1570s). One must either use the Database command and reselect both the database and file or execute the Start program.

The bug listed under item 8) means that one cannot transfer a program file from one disk to another without having to reinstall the database. I have been informed that this was an 'improvement' made to the 1571 drive. Commodore have shot themselves in the foot over this one.

<reproduced hv permission of ICPUG>

Won't they ever learn. The 'improvment' was done because some stupid twits ignored the manual, used an identical ID for all their disks and then complained that after swapping disks and writing to them that the BAM got mixed up. CBM created a pain in the backside for serious users.

A note of caution. It is early days in the life of the 1571 drive and more bugs may be discovered in this machine. Consequently I take precautions and have been keeping four, yes four, copies of each data disk. I examine carefully the directory of each backup disk and if I find that the copy has more than a couple of extra blocks free compared to the original there is almost certain trouble in store. For example, a couple of times with early versions of Superbase 128 I have found almost 200 extra blocks free after a backup. Everything seemed OK with the copies until I tried to backup these copies; these backups of backups were faulty and gave ID mismatch errors and lost records in the database. Further, I am not yet fully convinced that the 1571 found in the 128D and its stand-alone version being sold in the United States are identical drives. It is nice to have the double sided capacity, but be cautious for the next few months as a sort of insurance policy.

MONTHLY CALENDAR

From Peter Miles in Dorset comes an interesting series of Superscript macros for the 128. He developed it as a sort of D.I.Y. SASCO calendar but it is useful to illustrate a couple of important points about macros. First point is that with Superscript 128 you may call one macro from another and return to the calling macro. Think of this as a subroutine which may be called by many different macros.

Peter's macros, with some slight modifications, are as follows:

<<CBUG NOTE: The symbol ² is actually an up arrow. We've used the ² since it is a direct Superscript keyboard symbol of Up Arrow mis-reproduced by our printer.>>

```
   *s=z²m/eap/sd0²m/cz²&g
   *f=²i   /cal²m/cr Mar
   *g=²+f²+f²+f²+f²+f²+f²+f²&m
   *m=²m/gn8/²&i
   *i=²+f²+f²+f²+f²+f²+f²+f²&n
   *n=²m/gn16/²&v
   *v=²+f²+f²+f²+f²+f²+f²+f²&b
   *b=²m/gn24/²&u
   *u=²+f²+f²+f²+f²+f²+f²+f²&y
   *y=²m/gn32/²&o
   *o=²+f²+f²+f²m/gn39/²&p

*p=*****************************************************
***********²m
   *cm:Press RUN/STOP or ESC & then 's' to start
```

The beginning macro 's' does three main things. It clears the screen, sets decimals to zero (i.e. integers), and sets the calculator to zero. But notice the very beginning bit where it adds a letter z to text, if any, in memory. This is necessary because if there is text in memory then the command F1-E-A is followed by the Proceed/Abandon confirmation submenu. On the other hand, if no text is present the command executes directly and this results in an extraneous letter p being inserted. Adding the letter z (or anything else) forces it into the 'with text present' mode. We next call 'g'.

Macro 'g' is the bit where we first call on macro 'f' and this is our subroutine macro. To call a macro 'f' as a subroutine rather than a successive operation one uses the format ²+f in lieu of ²f, the '+' denoting it is a subroutine call. This facility does not appear to be mentioned as such in the Superscript manual but it does appear in the macro at the start of the Wide Table file included on the Superscript 128 program disc. Precision Software should amend their manual accordingly to make more users aware of this quite useful feature.

Macro 'f' simply tabs to the first of the standard default tab setting (²i), inserts 3 spaces, adds 1 to the calculator (/cal²m), displays the result (/cr) followed by the month (March in this instance). Notice there is no code at the end of macro 'f' to call another one, it

<reproduced by permission of ICPUG>

merely returns as would a subroutine to macro 'g' which finishes after 7 days have been calculated by calling 'm'.

Macro 'm' sends the cursor to line 8 (/gn8) for the next 7 day line and the process continues. Obviously the spacing achieved by (/gn8) can be set to whatever you wish by changing the 8 to say 10 or 15 etc. The other successive macros continue to call the 'f' macro until the calendar is complete and the asterisks are printed.

SUPERDESK <<C128>>

Superdesk is a single disk Superbase/Superscript program for the 128 which I wrote to make my 'office' operations and ICPUG duties a bit easier. Frankly, I designed it to do most things Sidekick and other desk top utility programs do on an IBM PC or clone while running both Superbase and Superscript along with having the dictionary available on the same disk. Ideally it requires a 128 and 1571 or 128D as it fills more than one side of a disk when the database has more than a few records in it. One may also use a 1541 drive as device 9 with the program. This allows one to keep the one Superdesk 'master' disk in the 1571 or 128D and text files on separate disks in the 1541. Disk swapping is almost eliminated except for backups.

Superdesk is designed around a Superbase file containing names, addresses, telephone numbers, birthdates, and other things I find useful like sending out Christmas or Holiday greeting cards. The program does various tasks like displaying the telephone number of the person being called, the elapsed length of time of the call along with the running cost based on dailling code charge letter (local, a, b, b1, A, B, C etc). It then stores the data to keep a record of costs under personal or business categories with end of month itemized totals and printouts. It allows for quick letter writing by automatically entering Superscript, loading a predetermined letter heading from disk, entering the current date along with the complete name and address of the person or organization being written to from the selected Superbase record followed by the salutation such as Dear John. One need only start typing after the Dear John bit and of course, you may finish with one of the three macros which close a letter and print the envelope as covered in my Jan/Feb 86 article on Superscript pg 23. A collection of other macros including the paragraph numbering one (Mar/Apr 86 pg 159) are included in the defaults file.

Superdesk also provides a short memo facility, a dairy which is maintained as a Superscript document, a miscellaneous expenses file similar to the telephone call file, and it provides printed or screen listings of birthdays, Christmas card lists, club members, and separate business and personal telephone number lists. Finally there is a CBM ASCII table which I find useful when programming. The dictionary is available whenever Superscript is entered, is on the same disk as all the other files and avoids the disk swapping problems. The program, which is menu driven, is loaded in the usual manner by loading Superscript first followed by Superbase.

A ReadMe file is included as the first one on the disk directory and explains the whole program along with the macros used. TThe program is really too long to cover in the Newsletter so a copy has been placed in the ICPUG 128 library. Members with a 128 and both Superscript 128 and Superbase 128 should order a copy from the appropriate librarian as Superdesk. Those with 1570 drives may wish to delete the dictionary files so as to provide more space for names and addresses. Superdesk also acts as a normal program disk for both Superscript and Superbase when either are loaded without the other.

I don't mind copies of Superdesk being given to anyone but would ask all ICPUG members to strongly encourage nonmembers to join if given a copy. All should feel free to make modifications and improvements to Superdesk and I'd be pleased to receive copies of these for possible inclusion in future versions.

<<CBUG will be persuing conversion of these and other programs for B128 usage --do I hear any volunteers to take on the task?!?!>>

<reproduced by permission of ICPUG>

A PLEA

Please send all items for possible inclusion in Super* Corner on 1541/1570/1571/4040 format disks. <<Send via CBUG if you do not have ability to convert from 8050 format -- CBUG will transfer to 4040 format and forward both original and duplicate. Please conspicuously mark the disk if it is not duplicatable via Copy All (i.e. database files, etc.>> Include not only your programs but the TEXT file of your letter/notes as well. Nothing causes more frustration than to have a word processed letter explaining about the program on the enclosed disk without the text file on the same disk. I just don't have the time to spend rekeying in the author's explanation of what his program does and how to use it. I am reasonably certain that Hugh de Glanville and Henry Velleman feel the same way (and how! -Eds.)

SUPER* CORNER Sep/Oct 86
Abridged

Edited: By Jim Kennedy
Chairman, ICPUG, London U.K.

ACCOUNTS ON SUPERSCRIPT

Pete Miles from Dorset who developed the Monthly Calendar macros mentioned in this column last issue of the Newsletter has come up with a quite amazing Superscript application package; one that keeps the accounts for a small business. His package is based entirely on Superscript <<Superscript III for the B128>> unlike most others which are designed around databases or spreadsheets. Having an accounts package running under a word processor must make Superscript almost unique in its field. Pete described the package to me in a letter (kindly placed on a disk for which I am very grateful) which is extracted below. I was talking the other day to Simon Tranmer who developed Superscript. Even he is amazed at some of applications of Superscript such as this one and the paragraph numbering problem of Gary Benneworth (covered in the Mar/Apr Newsletter pg 159). Full marks to Pete for coming up with a really useful package.

Pete and an electrician friend of his, both users of Superscript on 128D's, decided that an electronic cash book might be a good idea providing it was more simple than the professional programs currently available. They felt that there had been an increasing complexity of accounts programs based on combined databases, spreadsheets, etc. They took the view that most small businesses are interested in ending the year with more money than they started and if a simple, no frills program in 80 columns existed then they would be reasonably happy. Pete's electrician friend has been using the system for some months with complete success.

The package consists of a set of Superscript document formats and macros and It assumes that the user is reasonably knowledgeable about using Superscript on the 128. It should not be thought of as a way of getting around professional accounts programs as it does not carry out conventional double entry procedures. Rather, it should be seen as a specific way of achieving a set of well laid out and understandable cashbook & ledger procedures. The system has not been approved by the Customs & Excise and nor is it likely to be. However, all normally required information may be extracted from this system and in some instances, such as the Analysis by Account Number module, more information is available than many other 8-bit driven accounts programs.

The procedures are based on the standard audit trail format; that is to say, each receipt, invoice sent out, cash spent, banking made, simple payment or whatever, are entered in the Day Book format. However, it does make life a bit easier if all of one type of payment are entered together, payments first, then receipts and so on.

Whilst the Day Book is the heart of the system, as many other detail formats as desired may be created depending

<reproduced by permission of ICPUG>

on requirements. The sample formats included in the package are as follows:

1. Day Book
2. Receipts
3. Sales Ledger
4. Purchase Ledger
5. Purchases
6. Statement.
7. Bankings
8. Analysis by Account No.

Each format is stored on disc as a separate document and is called to text area 2 as required, with the exception of the Day Book which is always in text area 1. Various macros (command strings) are used and considerable use is made of the '²+' feature wherein one macro calls another 'subroutine' macro. As mentioned previously this feature is not explained in the Supercript manual but is used in command strings for the Wide Table file on the Superscript demonstration program.

The package will be placed in the 128 Software Library and members may obtain copies in the normal manner. A READ ME file is on the disk which contains instructions on use and the address and phone number of Pete Miles. Pete welcomes any suggestions and improvements from other users.

MORE ON SHIFTED SPACES AND COLLATING SEQUENCES

Super* Corner in the May/Jun Newsletter pg 240 contained some suggestions for achieving unique fields in Superbase where one wished to use something like a person's last name as the key. One technique I mentioned was to use shifted space(s) following the last name in the key field. There is one precaution I must now add to that suggestion which shows up when doing export or import operations. Unlike screen operations, export/import operations do not treat a shifted space as the number 9. Rather, in these operations it is converted to an ordinary space and, if your key field is unique, this leads to problems. Two records with Jones in the key field, one being just Jones and the other being Jones- (- denoting a shifted space) will both have the same key following an export and subsequent import. If the key field in the imported database is unique it will reject the second and subsequent Jones.

However this conversion from shifted space to ordinary space does not occur when transferring data from Superbase to Superscript when both are running side by side. In this instance the shifted space will appear as the number 9 in the Superscript document. While it is no great problem to cleanup the Superscript document by removing the unwanted 9's life would have been easier if these two phenomena were reversed, the shifted spaces being preserved during export/import and converted to ordinary spaces when inserted into Superscript documents.

To successfully export and then import a database with shifted spaces one would have to make the imported file non-unique. One could then go through this imported file making changes to the key fields by adding shifted spaces or whatever where necessary to make all records unique and then reformat the file as a unique one. This procedure is probably somewhat dodgy; I don't know what would happen if two records with the same key were overlooked and the file reformated as a unique one.

At this point I suspect that most people would either start using the truncation option I mentioned or using a full stop in lieu of the shifted spaces as being a relatively unobtrusive addition to someone's last name. (Paul Jones. looking rather better than Paul Jones4 .)

To save others from treading down the same path I should mention that I tried to place the key field on screen 0 and place a replica of this field on screen 1 but truncated by a couple of columns. By then selecting screen 1 before printing I hoped to eliminate the number from a record like Jones 2. It didn't work. When the replica field was truncated so the number 2 didn't display it was still present in the printout. As far as I can tell one must either truncate the output by a command PRINT @1&15[KEYFIELD] where the normal field is say 16 characters long or make both a key field which will contain Jones1, Jones2, etc. along with a separate last name text field used for printing if one wishes to export and import to a unique key field file.

<reproduced by permission of ICPUG>

AT REPLACE BUG?

Whilst I cannot prove it I believe that the 1571 disk drive may have this problem. A few times one file has corrupted another after being replaced on a disk which is more than half full. I have therefore changed my old standby method of saving updated versions of Superbase programs. The old method was:

```
1 goto 3
2 save "programname":menu
3 rem
```

I merely typed RUN 2 to save the revised program; Superbase automatically converting the Save command into a Save with Replace one similar to the Basic command of SAVE "@0:PROGRAMNAME",8. I used this system successfully for over two years with my 64 version of Superbase. However with my 128D I have been plagued with these overwrite and corruption problems. Consequently my revised Superbase routine is now as follows:

```
1 goto 4
2 maintain o"s0:programname.p"
3 save "programname":menu
4 rem
```

This insures that the no overwriting occurs and I have not experienced any further problems (yet). Similar changes could be made to ordinary BASIC programs.

A 128 AND 1571 VERSUS A 128D

There still seems to be one bug lurking in the background when I use Superbase on my 128D whereby backups of a Superscript/Superbase disk which is more than half full (that is has data on the second side) show many more Blocks Free than the original disk. This first backup works OK but a second backup made from the first backup won't allow access to Superbase files. Testing is still going on to try and find out what is happening. To date the only difference between systems which have exhibited these strange effects and those that have not have been that one was a 128 with separate 1571 and the other a 128D, the 128D showing the symptoms. I hope to have this sorted out by next issue of the Newsletter.

1571 DISK DRIVE AND SUPERBASE

As I mentioned in the last Super* Corner if you open the latch on the 1571 disk drive while running Superbase you get the error message

```
  29,Disk ID Mismatch,xx,xx.
```

This means you cannot save a Superscript document to a separate disk of say correspondence and the go back to the Superbase record you were working from. You are forced to reselect both the database and file and start from scratch so to speak. Simon Tranmer and Tom Cranstoun both inform me that this problem has now been solved and a fix should be available in the near future in the form of Superbase 128 Version 2.07. This new version has added a command to the Maintain Other set of commands. This new command, as explained to me by Tom Cranstoun who 'fixed' it, is of the form:

```
  maintain o "UOL"
```

This command will allow one to log back into Superbase 128 when one has opened the drive door. <<In the B128, if you open the drive door the same problem exists, however, with an 8050 or 8250 this is no problem in Superoffice as you simply read and address to the other drive. I've not checked to see if the B128 SuperOffice releases the drive in use when you go to SuperScript. It re-enter Superbase,

<reproduced by permission of ICPUG>

for those of you who don't know how, at main menu just type ' database"filename" ' or ! dA ' instead of the whole word.

TWO OTHER BUGS AND A PLAUDIT

Two other bugs in Superbase 128 have been fixed in Version **2.07.** First one was that Utility 64 would not work on a **1571** database when data was present on the second side of **the** disk. The second was that Utility 64 would not read a **record** greater than 246 characters without a lock up.

Tom Cranstoun covered these problems and fixes in some detail in a letter to me along with a recovered database of mine; I had sent him a series of my disks to help him finding out where the beasties were resident. My thanks to him for this. I know many of you will say that as editor of this column Precision Software go out of their way to be helpful to me. But I have sat in their offices and overheard them dealing with other customers. The replies I heard appeared reasonable. Bear in mind I was only hearing one side of the conversation and then only a few out of the thousands of calls they must have received over the years. However, I compare these experiences with some others of mine that I have had with firms like Microsoft which have not been very good (such as how to get both upper and lower case letters out of Multiplan on the 64 - they just didn't want to know). I'll stick with the helpful firms like Precision anyday.

It is essential for me to stick with reliable reliable products from reliable companies from whom help is available. What would happen if I lost all the ICPUG exhibition files and data covering things like which members volunteered to help on which show days, their telephone numbers, what the show expenses were and so on? Also those members sending disks to me with programs would be very upset if one got corrupted and I couldn't recover the results of their hard efforts. I cannot afford to use a program that is not supported by the manufacturer or importer.

AND A CLANGER

The disk drive door problem mentioned above wasn't really of Precision Software's making; Commodore changed the drive ROM in an 'improvement' exercise. I also heard a rumour that Commodore 'improved' the new model 1541 disk drive (a 1541C I believe it is called) which goes with the new repackaged 64C to such an extent that it won't work with GEOS, the new Mac type interface which is to be bundled with the new models computer. Apparently some other protected software programs won't work on the new drive either. My sources tell me that the problem is that the number of tracks were changed. Now the old standard **1541** had tracks 1 to 35 but some software uses higher **number** tracks (36-40) for protection schemes. The 'new' **1541** has apparently added track 0 and deleted the ability **to** access the tracks above 35. Hence, its alleged **incompatibility** with the old standard. Superscript and Superbase apparently do not use these tracks above 35 and so probably should work but this has not yet been tested. **Advice?** Try before you buy.

I'll make Commodore an open offer: "We of the Independent Commodore Products Users Group here in the United Kingdom will volunteer to beta test all major Commodore end products for 30 days prior to final fixing of the design and operating system". Such tests would almost certainly prevent the 'shoots oneself in the foot' syndrome which Commodore is currently experiencing on a **monthly** basis. Anyone at Corby (soon to be Maidenhead), **West** Chester, or other Commodore headquarters should feel **free** to contact me direct (address inside front cover). I **feel** reasonably safe in saying dozens of ICPUG members **will** volunteer to test each new item.

SUPER* CORNER Nov/Dec 86
Abridged

Edited: By Jim Kennedy
Chairman. ICPUG, London U.K.

Due to the long review article on Superbase for the Amiga the Corner in this issue will be very short. You will read elsewhere in the Newsletter about the 1571 ROM bugs which affect Superbase and Superscript on the 128, which I made mention of in previous articles. Readers are referred to my column entitled Jim's Jottings for a detailed discussion of these points.

Karsten Hunger from Denmark furnished me with a printout of a Superscript Defaults file for a Seikosha SP1000VC Printer which he says works for him. SAE to me if you would like a copy. Also, Graham Johnson of Sutton Coldfield sent me his Superscript Defaults for a Smith-Corona D200. Since they are on disk I can either furnish a printout (again SAE please) or I can place them on a formatted disk sent to me with return postage and label in a reusable mailer.

Pete Miles in Dorset has come up with another series of Superscript macros for daily recording of materials and labour used on various jobs such as electrical, plumbing, carpentry contracts, or anywhere else one uses a variety of materials and spends varying amounts of time on projects during the day which must be accounted for. Rough notes can be made at work and transferred to a Superscript document later. Each item is saved as a separate 'document' on the disk so it is limited to somewhat less than 144, the maximum number of files on a 1571. Again, a formatted disk to me with return postage, label and reusable mailer. But don't forget to enclose a note about which item you want! I cannot read minds.

If any of you are reaching a point where you are running out of room for a Superbase database on a disk then it may pay you to do some quick calculations. Superbase stores data in 128 byte blocks of which only 123 are useable. If the records in a file contain an average of say 135 bytes (allow one byte per character in a field plus one for each field as a field separator) then your database is using two Superbase blocks of 128 bytes for each record which is equivalent to one CBM disk block of 256 bytes. However, if you were to do things like cut down the number of fields in the record and the amount of information stored in the fields to say an average of 110 characters per record you would be able to almost double the number of records on your disk. It doesn't matter if the odd record goes over the 128 byte limit and uses two Superbase blocks, just try to keep the average down so most records use only the one. Savings can made by use of abbreviations, omission of commas and full stops and concatenation of many 'tick the box' fields into only one field using a series of letters as a code. Don't be alarmed that you will not see any extra blocks free after doing this 'condensation'; once Superbase has allocated a block for database use, it stays committed. But the space is there, free to be used for the next record to be entered. A backup makes an identical copy of the existing disk so it will also not show increase in the number of blocks free. These savings will, however, show up after export and import as these operations rebuild the files in the imported database. This is also a useful thing to do once a database has been in use for some while.

Superdesk, which is available from the 128 Library, may be useful to members just learning to use Superbase and/or Superscript on any of the Commodore machines including the C64. Whilst on the C64 one cannot use the integrated parts of the program where Superbase calls Superscript other parts may prove useful. Things like the Set and Dump commands, modifications to the Start.p program and Help screens for menus are used so one may see how these are employed and incorporate them into other programs. I have placed Superdesk in the Public Domain so everyone is free to use and modify it as they desire. It was not written as the 'final version' of a desktop utility but rather as something to stimulate others into doing something really useful.

Those of you who have purchased Superbase: The Book may have noticed something missing from the bottom of page 30 and the top of page 31. The missing words are:

<reproduced by permission of ICPUG>

...output or other processing. You can copy, rename, **or delete** lists.

Memos

These are created with the 'memo' option. They...

Thanks to Mike Hatt for bring this to my attention.

<reproduced by permission of ICPUG>

SPEEDING UP LIFE IN THE FAST LANE!

By: Mark Schwarzbauer

If you use your word processor like I use mine you probably already use alot of short cuts. Sometimes though it takes someone to beat us on the head before we realise that we can customize our programs to better fit our needs.

For example, in SUPERSCRIPT the function keys are preset. However, they can be easily changed. Using the "Casy's scrubber" deprotection disk helps. Not only does a deprotected disk load faster but it also is easier to customize. I got tired of out-putting to video and then having to push "c" at the end of every page to continue. So I changed the "output to screen(esc ov)" "F1" to "esc ocv" on "F1". This is a very straight-forward easy to do change. I did not even realise that the SUPERSCRIPT manual details this change in appendix F. If you find yourself using certain routines rather often, then customize your function keys for your personal use.

Another wonderful time saver is using SUPEROFFICE rather than SUPERSCRIPT AND SUPERBASE. Purchasing SUPEROFFICE is not just getting 2 programs on one disk. The two programs are integrated. This means that you can switch from one to another instantly. It also means that you can directly incorporate SUPERBASE FILES into SUPERSCRIPT MAILMERGE. This means no more crazy fill files!

SUPEROFFICE is even a better deal than ever before because the new SUPEROFFICE contains SUPERBASE II+. The version of SUPERBASE II IS 2.07. SUPEROFFICE-SUPERBASE is version 2.08! There appears to be improvements in the SUPERSCRIPT contained in S.O. too. Of course, you do need a 256k machine to run SUPEROFFICE. But upgrading is both easy and inexpensive. Upgrades are available from people advertizing here in the ESCAPE, but you can do it yourself if you know how to solder. It only costs about $25.00 in parts to upgrade to 256k.

<<IMPORTANT NOTES: $25.00 in parts if you know where to buy at rock bottom prices, and still get decent quality. Major national chains likely will charge over $100 for the same bill of materials, but not always the best quality. I know of members who have used sockets for which they overpaid and were sold inferior sockets, who nearly lost their computers! It may be advisable to buy a kit from one of our advertising members such as Jim White (S&W supply) than to risk buying from your local merchant.

SECONDLY, Mark Schwarzbauer and many of us have had experience in chip soldering, desoldering filled holes, and have the proper miniature irons, solder suckers, etc. to do the job. Sometimes even we do oversimplify. Please don't attempt soldering a circuit board if your prior "experience" was installing copper water pipe in your summer cottage.

It's really rather easy, if you have proper equipment and adequate experience. If not, send the drive to one of our members offering the upgrade or better yet, just go straight to one of the plug in memory add on boards such as the 1Meg offered by Anderson.>>

Even if you don't use SUPEROFFICE, the extra ram will be another great way of saving time. Having 256K means you can switch to other banks and be working on 3 documents at once! You can edit, save, switch and load and do it so much quicker. There is information available on upgrading on the "B-128 users group" disk available in the CBUG library.

In the mean time- keep up the speed, so you can slow down a bit.

THE BIBLE ON DISK?

Yup, the whole thing if you want it bad enough. We will have the N.T. available in the NEW KING JAMES VERSION on two very packed disks. It is taking many hours to reformat this from IBM to B-128. It will all be saved in Superscript files by individual chapters. The price will be $30.00 for the New Testament. The Old Testament will only be released if there is sufficient demand as the amount of work that goes into this effort is quite extensive.

Obvious uses will be for pastors and Sunday school teachers that prepare their work on the computer. Having it on disk will also allow you to use Superscript to search for phrases as well as words. All the files are linked together and can be placed on one 8250 disk instead of 2 8050 disks. If you have questions or comments let me know.

Mark Schwarzbauer- 414-743-4151

SUPERSCRIPT III
AN ADVANCED REAL ESTATE APPLICATION

By: Galen Enochs

Since Precision Software kindly forewent any software protection of their new Superscript for the B, it is possible to customize it to your own requirements. My major use of Superscript is for filling out real estate appraisal forms. In so doing, it is often necessary to enter the same information in a number of blanks. For instance, the square footage of a house is entered in four different blanks. I decided to list the entire program and attempt to modify it so that it would duplicate any necessary information in the proper locations automatically as soon as the first item was entered. In so doing I learned a number of helpful things which I am passing on to the club in the hopes that some of you programming gurus will consider making some more important changes.

First, I will address modification of the basic loader program, then cover some salient points of how Supercript operates, and then how to insert your own routine.

Since bload searches only the specified drive, (unspecified = d0), error traps will force it to search both drives. This allows Superscript to be loaded from either drive.

```
10 trap 30
20 bload"logo",b15,p53248:goto40
30 bload"logo",d1,b15,p53248
40 trap 60
50 bload"3.2",b2:bload"3.15",b15:goto70
60 bload"3.2",d1,b2:bload"3.15",d1,b15
70 sys1024
```

In the above code, d0 is checked first, and if the file is not found the traps send the program to the statements to search d1.

The function keys can be changed to simulate frequently used run-stop commands. The purpose is to have one-keystroke commands instead of two. This is not too important unless you use a lot of run stop commands. For what I do, it is vital. There are two useful ways in which this can be done. First, chr$(28) + a string of characters can be assigned to a function key to perform commands, such as key2, chr$(28)+"gd"+chr$(19). This would move down 1 screen and place the cursor in the home position. However, this command would be forever fixed once you are in Superscript. A more flexible way is to define a function key as a run-stop command. This can be done with chr$(27), which is the chr$ code for esc. If you use key2,chr$(27)+"1" in your loader program, for instance, and then define 1 in your defaults file as /dl+" ", pressing f2 will catalog the current drive. Changing the definition of 1-key will change the definition of f2.

So much for the preliminaries. How do you place your own code in Superscript and get it to execute it? The key is to locate a safe place to put your code and then point Superscript to it. I have placed my code from 20210 to 202c0. You need to use supermon to locate safe areas. First of all, you need to know some general things about how Superscript operates.

The program ss3.2 resides in bank 2 from 0400 to 82da, and ss3.15 resides in bank 15 from 0400 to 058d. The code in bank 2 is the guts of Superscript, and the bank 15 code gets you back to bank 2 after kernal calls, etc. We don't need to worry about bank 15 for this operation.

The main text area is located in bank 1 beginning as 0003. The 2nd text area is in bank 2 starting at 85b2. Text area 1 appears to end at fcff and area 2 appears to end at dfb2. When ss is first run, there is a short pause between the reverse line appearing at the top of the screen and the menu for selecting whether to create a workdisk, etc. This pause is for ss to fill the two text areas with $20's (spaces). It seems ss clears more space for text than is available to the user. I have placed a brk command in the program in a location which would stop the program as soon as a certain key is pressed during text mode, then filled both text areas full. Pressing the required key breaks into the monitor. Examination of memory shows 20,s above the highest location my text reached. Also, interspersed in these regions above the text areas were other numbers, indicating that the program must use some of these locations for temporary workspace.

The best way to find a safe place is to use supermon (e000 version) to fill bank 1 and bank 2 with 0's, except for the locations which supermon uses (don't put anything in locations 0 or 1). Next, from supermon, l"ss3.15",b15, and l"ss3.2",b2. Then display memory location 25464. There will be a 6c 37 00 there which is jmp(0037). This is a very important location, because it acts like the pivot man in basketball, passing off the ball to a teammate breaking for the basket. It is Superscript's the pivot man which directs the program to various sections of code, depending on which key is pressed during text mode. Replace the 6c with 00. Next g20400, and ss will begin. As soon as you reach text mode, the first key pressed will break into the B's monitor. Enter bank 2 with the v02 command and examine memory for your 0's. These blocks of 0's should be safe areas, within certain limits of risk.

These "safe" areas are relatively small and you may need to break your code in chunks and tie them together with jmps. A preferable alternative would be to find the routine which fills text area 2 with 20's and limit the amount of memory it uses. I have not yet attempted to do this. If anyone else does, let me know.

Location 25464 is a good place install your detour sign to point to your routine. First of all, every keystroke passes through this location, and secondly, it is a 3-byte instruction, which allows you to substitute a JMP command without disturbing subsequent commands. My routine is located at 20210, so I will use it as an example.

At this point you have to be careful to make sure that after your routine is executed, you get back to your detour point in exactly the same shape you left, which means that in front of your own code you must make provisions to save all the information in the various registers. Therefore, the first 6 bytes of your code should be something like: pha; tya; pha; txa; pha; php; likewise, following your code, right prior to your JMP, you have to restore everything in reverse order with plp; pla; tax; pla; tay; pla. Specifically, your procedure should be as follows:

1) at location 5464 place JMP $0210
2) at location 0210 save your registers to the stack
   (pha; tya; etc.)
3) write your code to perform whatever you need
4) recover your registers from the stack (plp; pla; etc.)
5) JMP ($0037)

The JMP ($0037) is an indirect jump which depends on what is located at loctions $37 and $38 for its target. The target address changes, depending on what key is pressed. By knowing the contents of these two locations, you can determine the entry points for various functions of the program. Determining this information is not as hard as it might seem. Using the above method you can point the code to $0210. At $0210 code as follows: LDA $38; LDX$37;BRK. When you press any key, you will enter the monitor. You can then read the target address from the A and X registers. Restart ss by entering g 0400, and repeat the process by pressing another key. You should find the list below to be the entry points for various keys.

1) All alpha and numeric keys and all function keys
   defined as alpha or numeric keys point to $5473.
2) All functions keys defined as a menu command
   (beginning with chr$(28)) point to $5866.
3) Tab or function key defined as tab point to $6b6f.
4) Return points to $6072.
5) Stop and Esc point to $5c12.
6) Clr points to $6029.
7) Home points to $6039.
8) Cursor down - $5eda.
9) Cursor up - $5ef8.
0) Cursor left - $5ebc.
1) Cursor right - $5e97.
2) Undefined keys, i.e. off/rvs, norm/graph, have no
   pointers.

So, what is the benefit of all of this? How about SUPEROFFICE! Superbase resides at 20400 and f0400 like Superscript. Put Superbase in bank 3, changing those instuctions that have to do with bank switching so the B can find Superbase and so Superbase files can be put in bank 4. Put a ramdisk in bank 5 for quick file shuffling between the two supers. Put the bank 15 codes for both programs in bank 6, to be alternately shuffled into bank 15, depending on whether Superscript or Superbase is operative. Which leaves banks 7-14 for huge files.

If anyone is interested in making this a community effort, contact me at (618)656-5103, or write

Galen Enochs
311 Douglas Ave.
Edwardsville, IL 62025

P.S. If a ram disk is too tough, just change the code in Superbase to make it stick its export and get its import from text area 2 of Superscript.

CURLY BRACKETS ON SUPERSCRIPT II

By: Joe Rotello

If you have the CITOH/CBM 6400 equipped with the Letter Gothic (Diablo) 12 Pitch Wheel, you can print the "curly brackets" that many of us need or would like from time to time.

To print them, imbed the left bracket in-text by typing "ESC 2" (ESCape key followed by the number 2 from the numerical keypad) and imbed the right bracket by typing "ESC 3". The numbers will highlight in the EDIT mode as you type them.

On the video OUTPUT display, the "special characters" will not be highlighted - that is, the example below will appear "This is the 2curly bracket3" when viewed in the OUTPUT VIDEO mode:

This is the 2curly bracket3

Due to where the printwheel places the "curly brackets", the printed output character spacing may seem a little odd and you might want to insert an extra space in the text or even remove a space as per your tastes. Like just about any other printable character, the "curly brackets" can be double struck (bold), shadow, underlined or any combination thereof.

<<There are numerous special symbols on various wheels. Using the ability to send special characters as described above is well explained elsewhere in this issue. Each printer has its own unique requirements for special characters, print styles (such as quad high, italics, etc on newer dot matrix printers). The escape sequences are usually well documented. Often one must experiment with ASCII cods from 1 to 255 to locate the special character codes on daisy wheel type printers.>>

Joe Rotello
4734 E 26th St
Tucson AZ 85711
(602) 790-6333

DISK DEVIL

By: Ralph Ricks

Getting lots of read errors?

After moving my B128 with 8050 to a new built-in cabinet, the disk drove me crazy until I happened on a curious phenomenon. If the monitor is placed next to the drive, it causes read errors!

Move it away just 2 inches and the read errors are gone. Moving the monitor back just four inches from the front of the drive works too. Killing the power to the monitor also fixes the problem, but that is less convenient. I haven't tried it with monitor above or below drive, but I suspect the results would be similar.

The deflection yoke on the neck of the monitor's picture tube is no doubt the cause of the problem. That yoke is an electromagnet that pulls the beam of electrons around so that the electrons can strike the phosphor on the front of the tube. The phosphor particles light up when struck. That is what gives you the picture. A color monitor may well have a more powerful yoke, depending on the particular technology.

Why does that affect the disk drive? The read/write head of the drive is a very small electromagnet. It just gets swamped by the magnetic field from the monitor if it is too close.

ADVICE: Keep all magnets, away from your magnetic media (disks). It is no doubt possible to scramble your disks by placing them on, under, or beside the monitor.

Ralph Ricks,
5362 Old Pirate Lane,
Huntington Beach, Ca 92649

213-834-5208 days.

COMPILERS & PetSpeed

By: Bing Hart

A few of you have called about PetSpeed for the B, enquiring where to purchase it, and reporting that the company in Colorado advertising it in CBUG ESCape were not in fact selling it! Well, I contacted England repeatedly and finally worked out a "manufacturing" (copying for sale) license with them. So I now can sell to you. They liked having me distribute it in the States because I apparently have more experience using it than anyone else. Anyway, because there appears to be limited CBUG interest in other compilers and even Petspeed, I'll spare CBUG the expense of publishing my extended article on PetSpeed compiler limitations and idiosyncrasies. If you already own PetSpeed for the B, I'll be happy to send it to you free (include a SASE and currently 80 cents for xeroxing).

Systems Software, LTD has also agreed to send me what source code and documentation that they can in order to help us alter the code and remove limitations of Petspeed 700. Not only is Petspeed the best for the B and the first optimizing compiler generally, but we'll make it better still!
The End.

Know & Control the World, Bees
or
ANALOG to DIGITAL, & appliance CONTROLLER

By: Bing Hart

For your Bee 128 to know the world ... sense the world, it needs sensors; then it needs a device to convert the typical analog signal from the sensor to digital information. Controlling the world (control electrical devices such as household appliances, laboratory instruments, etc) it needs to switch off 120 VAC or perhaps lower voltage DC.

For years I had sought a high quality low cost device (which of course is why we all purchased the B in the first place).

I found it. Its the ADC1 from Remote Measurement Systems. It interfaces directly to the RS-232c port (also available with an interface to user port for C 64 or even Vic).

Most of the low cost Analog to Digital Converters ("ADC"'s) don't interface simply to Commodore, don't have 12 bits (resolution sufficient even for a decent temperature sensing - 8 bits is about as accurate as viewing frost on the window), and don't have the sensitivity (a low maximum input voltage such as 0.4 volts or 0.04 with amplifier) for many of my applications, particularly scientific laboratory instruments).

In addition, the ADC1 has 16 analog to digital sensors, 4 digital inputs, 6 digital (TTL) outputs for controlling devices' settings, BSR transmitter (option), low power CMOS components, onboard temperature sensor, and it comes from a very helpful group of people who have been in business for some time.

My applications include:
  Weather station, environment (temperature & humidity) control
  Interface & controller for UV-VIS spectrophotometer &
    HPLC & Fluorometer & pH meter & ...
  RS-232 analyzer (not tried yet); poor man's IEEE analyzer
    (better than a voltmeter)

Voltmeter (with some differences)
Electrical energy consumption monitoring for household & lab
Burglar alarm

I had hoped to monitor and Fourier analyse brain waves and voice (e.g. so called lie detection), but I suspect I'm going to have some difficulty there. The maximum sampling rate is 120 readings per channel per second, although at this rate there are some additional limitations such as lowered resolution.

I'm writing more software for it. Pretty easy in BASIC, plus I can use much of the same subroutines that I developed for LitSearch, and for speed I compile it with PETSPEED. The immense volumes of data produceable from it can fully utilize Anderson's Megaram board and display it with High Res Technology Dan Deconinck's hi res board, mass storing on the 9060 harddisk from Northwest Music.

The manual that comes with it is extensive, including instructions for interfacing to other computers (even if the B should become outdated I can use it on virtually any other computer). The manual includes circuit diagrams and suppliers so that if one wished to, he/she could make their own (which I presume would be illegal as well as more expensive than buying it assembled).

It has features that would allow it to be a portable or field instrument, although I don't plan using it that way. Remote operation via modem tickles my fancy. Some 'friends' wanted to dial ahead & have it warm the water on their hot-tub until shortly after I called them yuppies.

ERRATA etc.

By: Bing Hart

Commodore B128 Programmers Reference Guide, p121, shows the RS232 CR and CMR register addresses at $DD02 and $DD03 respectively. Actually it is exactly reversed.

Warren Kernaghan's very useful memory map contrasting Pet/CBM to the B128, on page 4, shows CBM address decimal 196-197 corresponding to 200-202 on the B, and CBM address 216 "line # where cursor is flashing" as being 207 on the B. Correct this to 200-201 in the first case, and change the 207 to 202. Otherwise you may find your cursor off by 1 screen line.

INPUT# from SCREEN

Liz Deal explained a very clever way of forcing the B to use INPUT# from the screen work in the B like it used to (e.g. in the 8032). Caution: It may not work the same way depending on which line the cursor is on (near bottom).

VERSIONS of the B128

Ever notice that some B's have fuses and some don't? Likewise different weight, power supply, screen for power supply, different amounts of RF interference to your TV? These two versions lack the distinctions in serial number, peek values, etc, as far as I can tell, that people have talked about in CBUG ESCape. Thus it appears there are even more versions of the B than we previously assumed. When specifying your B, note whether or not there is a fuse sticking out in the back.

IEEE anomalies

Anyone notice voltage differences coming out of the B compared to the old Pet/CBM? Perhaps major difference is contact ("pin") 17. I suspect numerous consequences to this - including some lockup problems and idiosyncrasies of the operation of DS and DS$ in certain circumstances, including the operation of some peripherals, e.g.:

8300P Commodore printer:

This printer is essentially a Diablo 630 (or Xerox 630) printer with IEEE bus (some models built in) and Commodore labels on it. I've had at least one for years now, and love it. Best printer I've ever had, probably the best CBM has put out. BUT. Finally had a problem with it - refusing to respond to IEEE commands - so had a board replaced. Then it worked, but at some time after it began printing, it locked up the IEEE. When? Well, you know I'm a believer in electronic devices sometimes possessing a free will. When the lockup occurs cannot be predicted even by the full moon. So, tried same board in another 8300 printer. Same result. Tried a couple more boards ... you guessed it, same free will problem. To date even "The Printer Works, Inc" has not been able to fix it. I had to send them a B to test out different ROMS (Warren Kernaghan discusses the multitudinous variations known to the above company in a previous ESCAPE). The problem apparently doesn't occur when the 8032 is driving it. Using the original ROM in the new board prevents the lockup, but the printer simply halts at the FWM (Free Will Moment).
Conclusion: Anybody want to buy a couple of 8300 printers? Or sell one that works on the B? They truly are wonderful printers ... if they have the proper board for working on the B (picky, picky, picky). I'd love to hear from anyone with such a printer.

I have managed to make some of the control sequences to work like they are supposed to in the manual/documentation, 'tho most don't. Another reason for my wanting contact with any 8300P owners, albeit trivial compared to its working on the B.

TELECOMMUNICATIONS -- COST OF SEARCHING

By: Bing Hart

Assuming you've read the first article in the last ESCape, and assuming you wish to use online searching, you've then contacted some or all of the vendors and determined which ones have the databases of interest to you. Cutting costs is the primary theme of this article, and you've likely already determined that you can cut costs by not using a professional online searcher nor expensive "middle men" like Delphi or Compuserve for any significant amount of information.

It is additionally less expensive (to bypass middle-men) for some vendors because some will give you so many dollars credit when you acquire a password from them.

You've noted that their rates vary regarding startup fee if any, annual fee (paid at end of year) if any, royalties, connect time charges for various database files, monthly minimum if any, different amounts of usage, etc.

You'll likely note that more than one vendor/system has the same databases or same types of databases. Which is the least expensive, and which serves your purposes more?

The hourly rates for the same kind of database can easily vary 10 fold from one vendor to another etc. But I assume you've considered and selected with the "up front" costs in mind.

Prepare for more confusion than answers. Many different considerations will vary the cost, they interact with each other, and only you (and your software if it is high quality) can make the best decisions.

File-Vendor selection:

Many different databases seemingly relate to the same subject matter, e.g. commercially available microcomputer software can be sought in multiple databases for a given vendor and many others are offered by another vendor. In such cases I have no pearls of wisdom for your choosing amongst them. But if the exact same database is offered by more than one vendor, I do have some caveats to consider:

Some vendors may have a database divided into multiple files according to date range, yet another file may have the entire database. Another vendor may have only the database divided. The time it takes to search out the records you want ("develop a SET") for a given time period will be greater in the undivided file; but if you wish to receive relevant records from all time periods, its definitely faster to use the combined file rather than each of the divided database files. In either case, how much faster you develop the SETS depends in part on your software (whether your software allows quick editing and quickly and accurately sends your commands, etc. like LitSearch), plus you may need built into your software a means of measuring the speed with which the set is developed).

If the undivided file is faster, it probably has higher hourly $ rates. Whether faster times higher rates saves you money or not may require precise numbers that only only software like LitSearch can provide.

If the number of records is identical for a given database offered by two or more vendors, there are additional reasons to anticipate differences in the time to develop a set: Different vendors have different languages (and you may be able to use it to alter your strategy), each with their own speed, and different computers, each with their own speeds.

Price of receiving records:
If this hasn't been confusing enough, accessing different vendors may require different telecommunications networks (Telenet, Dialnet, Ma Bell, et al.). Even if you use the same network for the various vendor systems, the rates for that telecommunication system may be different according to the vendor you choose. Thus you might save money by picking a different telecommunications network for each vendor. Your software should remember and honor such choices.

And the speeds of transmission (in characters/second, not just baud rate) may vary. This is important when it comes to calculating how fast the host system and telecommunications system is going to send you the records you want, once the sets have been developed and you have edited & sent your message commanding them to be transmitted.

Some vendors will append to each record some routine information (typically useless) or padded leading spaces for indentation or padded spaces for purposes of right margin justification. Thus the amount of useful information in the exact same record will vary from one vendor to another. This too may be an appreciable cost consideration. (P.S. I have software for removing some of this useless information and thereby saving disk space - enquire).

Thus you need to be able to determine how much time it took to develop a set and how much time for the host to send you some records before you can compute the least expensive way to do a search. I know of only one program that offers such benefits (fortunately, it runs on the B).

A vendor system may have additional charges for each record they transmit to you. These charges may vary with the file you use as well as the format (format refers to what part or parts of the record you wish to receive, e.g. titles only, full record, etc).

Other vendor related costs:
On pricing, beware that the pricing policies change, so that the least expensive now may be the most expensive later. A few months ago I was downloading hundreds of titles each month for "free" (given that I simply used up my $12/month minimum at a flat rate of about $14/hr) from BRS After Dark system. Then they changed policy and began charging 3 cents for each record downloaded in addition to the hourly rate. For me that was the equivalent of about $35/hr. It became much cheaper to use Knowledge Index system which costs $24/hr. However, If I download lots of full records (not titles alone), the "Per Record" charge is insignificant and BRS After Dark is still cheaper, assuming I do >$12 worth of searching each month.

SDC used to require a startup fee, but no longer does. Dialog didn't, but now requires a $25 payment at the end of each year. Considering the time value of money, calculations of expense can be complicated. I'll prepare a spreadsheet if there are enough interested CBUG members.

If you do alot of searching with a given vendor, you may qualify for rate discounts. However, this generally is true for heavy users that use already expensive vendors at the more expensive times of day.

Generally the major vendor systems charge hourly rates for telecommunications network in addition to their other charges. Knowledge Index and BRS After Dark systems do not charge extra for the needed telecommunication network (and are much less expensive) because they are less powerful search languages and offer the service during low usage times (e.g. after 5 pm). BRS also has a compromise between their professional service and their low cost service in their BRK THRU system, available anytime of day, but with less powerful search language and charges rates between their other two. Which reminds me, calculation of the connect time charges usually includes royalty payments, but sometimes not. In order to be certain, call customer service.

Sometimes the service is "free". For example the vendor Searchmart had/has a database of available microcomputer software, and they charged nothing. But for me, the AT&T charges to their Florida location were well above what I would pay for similar but better databases from commercial vendors. Too, their search language was so inflexible that much more of my time and attention was required, more connect was required, and they still didn't usually have the records of interest.

(Once again you may better appreciate some of the complexities of comparing connect time charges. That's why LitSearch does the computation for you).

Free service continued: Dialog doesn't advertise it, but you can usually call them and get a free password for their Ontap files (abridged versions for education purposes), and there is not even telecommunication charges. Totally free? Yes, unless you value your time. It takes time to learn any system command language, and once you've learned a system, such as Dialog, you are likely to use that system rather than a less expensive one. That's where they make their money. Such situations inspired me to have LitSearch translate from the language of one vendor to that of another. Of course the vendors change language from time to time, and that's another reason LitSearch was made to be very flexible in its translation of a strategy from one vendor language to another.

Sometimes there are hidden or unexpected charges. Lots of Dialog users misread the original contract which offered in-depth information on selected databases, only to discover they had to pay dearly for it.

SOFTWARE for online cost reduction:
Many online searchers requested more than a few records to be telecommunicated, and when their RAM buffers filled and then emptied to disk (providing they were lucky enough to have this done automatically rather than locking up), there were incoming records totally lost (money down the BUFFER RATHOLE). You can appreciate why people appreciated LitSearch simultaneously downloading to disk

and screen (and printer if desired) rather than playing with inadequate buffers. Those with software that locked up were charged for all the records, even if the user hungup the telephone promptly. Let me supply a similar but more specific example: you request a 100 records with a command such as "t 1/5/1-100" but instead hit the "00" key at the end. Your buffer fills up at say the 20th record, so you hangup, only to discover later that you were billed for the full 1,000 records at $0.25 each ($250.00)! Or suppose you make a typo and develop a large set such as might be found with the prefix "re*". It might take many minutes just to develop the set, and if you didn't find out until later ... well, you get the point. Simple mistakes can be costly, and if you are using ordinary terminal software designed for bulletin boards etc., there will typically be lots of them.

Typo's are always problematical, but if you can type your commands offline, check the spelling etc. before going online, it can be a tremendous savings in time and money and errors. That's why LitSearch (and some of the "Johnie come lately" imitation software) must have this feature. Of course one cannot always anticipate which commands one wants to send. So your communications software needs to have a very easy, reliable fast editor for those last minute changes. In fact many of these last minute changes require some fast and accurate math calculations .. and I for one usually make mathematical errors at this time (I'm in a quite adrenergic state while online and the host is busily greedily calculating how much money they are going to make off my mistakes) ... so LitSearch now performs these calculations for me. Quickly.

In summary, you want to do as much as you can pre-search (especially in writing your search STRATEGY (sequence of commands to issue online), but have the fastest most reliable way of changing (editing parts) when online, and you want to be able to download to disk and screen (and printer if fast enough) even very long ("infinite") uninterrupted (without wasting time with buffers) online sessions.

When I first began searching, I asked Dialog about available software. They told me it was impossible to use a microcomputer to access their online service (what a croc). I spoke to some professional online searchers, who refused to think about good software, saying that what I was trying to do (cut costs automate) was unethical. I later developed the software (you know, the impossible unethical thing) and informed the major vendors, asking if they had any interest. They didn't, but 2/3 of them now sell competing software with some of the same features. Of course it doesn't run on any CBM, and you know what I think of the quality.

If you don't have this quality of software to execute online searches for you, it can be a big help to have a friend, colleague or client watching the screen with you and helping to prevent or correct for errors. Hopefully your software prevents your cohorts from observing your password. LitSearch (and recently some of the vendors' mainframe software) offer that protection. Rip-off can be expensive.

When you become certain of exactly how to perform a given search, or parts thereof, the least expensive way to do it will include the entire search to be performed automatically. Indeed you way have a whole sequence of searches to be performed automatically, possibly each with different vendors & telecommunications networks & baud-rates & ...

Short of automating your searches, you will find that there are times when you want more than a simple inconspicuous prompt printed on your screen. An AUDIBLE prompt from your software can spare you those times when the host is ready for you but you don't know it. But if you are ALWAYS attentive to the screen, and this doesn't drive you to join the Moonies, you have no opportunity to discuss your next proposed tentative message to send with a colleague, or read manuals put out by the host, and you loose money anyway.

thus your software can save you money in oh-so-many ways.

Your software may require a lot of time to learn how to use. Once learned, you may forget aspects of it by the next time you use it. If you have to study your manuals while online, this can be expensive. If you $ value your time learning and relearning it, this can be expensive. If not knowing your software causes errors online, this can be expensive.

Post-Search expenses:
After you logoff you may need to perform any number of things with the downloaded file. One is to inspect it to determine what further online searches are needed. Almost invariably there is room for improvement. The easiest inspection is from hardcopy. Ideally the hardcopy is already there (the very fastest way to get hard copy is to have it already accomplished as soon as you logoff - which means downloading to printer WHILE it was also downloading to screen and disk - "triple downloading" - but that requires high quality software, and a fast printer if you are using 1200 baud or higher while online with host).

Often-times a post-search inspection of the online session reveals a need for another search (e.g. to get a few missing records). If you can do this quickly, you may save appreciable connect time with some vendors who will still remember your previous search and not require you to repeat much of it prior to execution of your revised search. Another difference between vendors is in the amount of time it takes to logon, get past their menus and news bulletins etc. If you do a lot of short searches, this expense can add up.

Given a need for making a quick hardcopy post-search, there are numerous ways to best and quickly accomplish this. Insure that your software allows selected portions to be made as quickly, easily and reliably as possible.

Another variety of inspection of the downloaded file calls for editing of the downloaded file. The software designed for this is wordprocessing. Most terminal software allows for downloading &/or converting into wordprocessing readable format. If you have Superscript, you should be in good shape.

Rarely is there software available to do some fancy analysis of the disk downloaded online session records, especially in CBM ASCII etc. There is some available for LitSearch, but this whole matter goes beyond the scope of the present article.

After the search you may wish to analyze which parts of the session cost what. This can be very useful toward determining how to do your searching better next time. For example receiving the same records at the "same baud rate" from two different vendors and two different telecommunications networks will often take different amounts of connect time. There are many reasons for this, e.g.:
baud rate specifies amount of time to send A single character - but says nothing about the amount of time between characters. Such characters per second (cps) rates may vary. If the vendors don't vary it, the telecommunications network will. It also depends on how many other people are time-sharing at the time, and my experience is that the vendors won't reveal such information. You must determine it yourself, empirically.

Sometimes you will be able to think quickly enough online that no follow-up search is needed (no postsearch analysis). But to do this, you typically will need to inspect, while online, the records that the vendor is sending you. If you are already saving costs by using at least a 1200 baud modem, any appreciable amount of information whooshing by on your screen will be difficult to read. And your spouse will find you very irritable afterwards because it is so frustrating for you to read lines that keep bumping up each time there is a scroll up due to a carriage return on line 24 of your screen. You think its frustrating trying to read this article? Reading 1200 baud for awhile is something else. Point is, most people can't do it well and think at the same time. Its even more difficult if your software doesn't give you a full 24 lines of screen. Ideally the incoming online information would appear n your printer too, but then you probably don't have fas enough printer. Anyway, the software to facilitate these problems is rare indeed but can save you lots of money (even alimony?).

If you have a printer that uses less expensive fabric ribbon, and allows one to put highly legible print in small pitch and lines/page (e.g. 15/inch, 88 lines/page), such as the CBM 8300P, you can save on paper as well as the time it takes you to flip back and forth visually from page to page. Of course your software should accommodate all of this easily. As in professional wordprocessing, it is sometimes nice to have settings for a draft-quality printer and another for letter-quality. Ideally both settings should be available without having to reload your software.

Hardware:

The B has features near ideal for online searching: a real number pad and cursor control and INS and DEL and HOME keys (without requiring a NUM lock), function keys, simple fast RUN and operating system, good bus, lots of available disk space, ESC and CTRL keys, good BASIC and optimizing compiler, good wordprocessing and online software available, a real RS-232C port, quality monitor, ...

We plan additional software that will perform additional analyses on the incoming online information ("Eavesdropping" on the RS232) while another B128 runs LitSearch. This can provide POWER that the most intricate software or 68000 chip cannot achieve. And at B128 prices. And without your having to learn another keyboard or DOS etc. Some computers brag about the speed of their processors. Two B's can do more than 1 computer with nearly twice the CPU speed.

A 2400 baud modem, accessing your vendor from certain cities, can save considerable money on some searches. I am confident the future will make 2400 baud more ubiquitous and reliable (it is already quite reliable in many cases).

If you do alot of searching on select databases and have the capital, you may be able to purchase a Compact Disk with that database on it. The savings per record are tremendous (e.g. equivalent $3/hr vs $87++/hr for Fisheries; $0.50/hr vs $24 or $14.30++/hr for minimum Medline). Capital outlay may be $1500 and up. This is a rapidly changing area, but the casual user will not be able to justify it for quite some time.

In summary thus far, the online costs depend on selection of vendor, telecommunications network (including AT&T sometimes, especially depending on where you live), database/files, software to operate your B online and offline, hardware, when you search, and the abilities used and charges of the searcher(s). Quite important in some cases is the elegance of the search strategy/profile (sequence of commands you issue to the host/vendor system). A good search strategy in turn depends on most of the above, but more importantly it depends upon the cleverness and experience of the searcher(s). A good discussion of this is beyond the scope of this article, except to say "when in doubt, use the free 800 telephone number to check it out with customer service". You can often write a better search strategy than they will suggest, and it is really fun and challenging in addition to paying off.

Searching can be easy, inexpensive, invaluable, and fun. And there is always the challenge of doing it better and cheaper.

Bing Hart
17311 E 40 Hwy #A34
Independence, MO 64055
(816) 373-5523

DOCUMENTATION FOR SUPERSCRIPT II CALENDAR

By: D. Gorsuch

<<This again is of both general interest and library material>>

When one uses Superscript II as much as I do, you don't want to have to boot up another program to do your other chores. As a pastor, a calendar is a necessary tool for planning and scheduling. I thought it would be nice to be able to have a calendar that I could use in S.S.II, having it available any time I needed it, for changes, additions, or printing.

It is very simple to use. Having loaded in S.S.II, all you need to do is load the month in which you want to work. This is done by entering in two variables; 1)the number of days in the month followed by a slash, and 2)the three letter abbreviation for the first day of the month. For example, lets say you want the month of August 1986. There are 31 days in August and the first day of the month begins on Friday. So you would load: "31/fri".

Now that you have the calendar loaded, you will notice that at the top there are some question marks across the top. Move the cursor to the first one and begin to enter the month of August. In this case you will have three "?" marks left in the first set of "?" marks. Just use the space bar to erase them until you come to the second set of ? marks and enter the year, 1986. Now you are ready to use and save this month. In a matter of 5 or 10 minutes, you could make up all the months for any given year.

There is one caution in using this calendar. You must do all of your cursor moving with the cursor arrows. Do not use the INS/DEL or RETURN keys as they will mess up the calendar itself. Also, do not work in "insert" on. I have virtually no knowledge in programing, and I don't know if this could be improved upon. If you can, please do so! Also, due to the graphics I had to use, if you write next to a line, you might erase that line. It will take some getting use to, but I have found it to be very beneficial and simple to use.

I made this for working on my 8023P printer, so you may have to change some formats if you have a different printer.

****

If you find this beneficial, please write me and let me know! If you have improved on it, I would also like to benefit from your expertise. Write me at:

Pastor Douglas E. Gorsuch
P.O. Box 123
Wilson Creek, WA 98860

THE FALL 1987 CBUG NAME CONTEST RESULTS

By: Norm Deltzke

As you may recall, the specification of an eligible name to be found was in this format "name.town" where the 'name' was a person's last name and 'town' of residence. These were tossed into The ESCAPE at random subject to certain mechanical considerations. The names were chose from drawing cards for a series of five numbers and the match or next highest zip code in our membership database selected the name to be inserted. Here are the entries:

Author of Article Containing Name	Page #	Name.Town
Rose	14	Springer.Cordell
Cumfer	17	Ranch.Spokane
Kernaghan	25	Godwin.Chestertown
Kernaghan	30	Totter.Edgewood
Lacey	34	Thompson.Helix
Deal	36	Oloughlin.Duncan
Bogart	40	Carothers.Alamo
Kloft	46	Warner.Lowell

Contrary to the terms of the contest, anyone who won, even if late or beyond the eligible number was held fully eligible. The winners were:

James Katzfey, Texas
S. Twenge, Iowa
Stuart Harner, S. Dakota
Randolph Newburger, Pa.
Tom Rehm, Il.
Bruce Croucher, NY
John Lemkelde, Pa.
Irvan Hildreth, Co.

Bob Huffman, In.
Norman Clay, Oh.
James Bogart, Ia.
Mary Hoyt, Tx.
V.K. Vance, Ca.
John A. Murray, Ct.
Roger Basham, Ca.
Paul lAilstock, Va.

Why did we limit the contest to only one month? Well, if one was going to accept a bit of aid from the good ol "B", just do a SEARCH for ????.???? to find possible entries. This might produce some wrong data, but it would find all the entries. Interesting application of the SEARCH function -- instead of searching memory, you search the linked files -- all the files on the CBUG print file disk are linked!!

If anyone did not receive their prize promptly, let us know. BUT we will need aproximate dates as well as your name, address, and zip.

Maybe next Christmas we can find some other crazyness!

ON MODIFYING B-LINE & SUPERBASE

By: Lt Col John A. Wright

Now that the Holidays are over and I've managed to hire a few new people to help out with the office work, I can get down to doing some computing. The first thing I would like to do is mention a problem I had with my version of Jack Cassey's program to unprotect Superbase. The program works great for Superscript but it seems to have a flaw in the Superbase portion. Maybe I have done something wrong or have a bad copy, I don't know. I followed the directions as they were printed (in fact three times on three different disks) and got the same problem all three times. Here's the problem. After Superbase is loaded, instead of the usual request for a data base name appearing, I am left at the main menu. This is no big deal in itself and can be worked around, but it would be nice to have the program work the same as the original. I suspect that something has gone haywire in the ML portion and the "Start.p" program is not being executed. I don't know anything about ML programming (yet) so I can't look into that aspect.

The second problem I found is more serious. When I tried to download a program when in Superbase (I have written a few to simplify file manipulation) I get garbage. When I try to look at the program which should have downloaded all I find is some strange looking characters. When I try to run the program, bye bye Superbase and reload time.

Any help that Jack can provide, or if someone else has had the same problems and has found a fix, I would sure appreciate some information. My converted Superscript works really great and I have written an "Autoboot" program that lets me use both disk drives. I would like to get Superbase working so I can put them both on the same disk and ,using the "Autoboot" have a "poor man's" "Superoffice". I've included a listing of my "Autoboot" here if anyone wants to use it.(This could probably be converted to ML then F-Keys used to go to line 100 or 200 for faster loading from either SS or SB.

```
10 rem
20 rem        start program for all data disks
30 rem
40 print"S";spc(15);"NrSSSSSS select the program you
      want to load SSSSSSR"
50 print"qq";spc(20);"1. superscript";spc(10);"2.
      superbase"
60 getk$:if k$="" then 60
70 if val(k$)<1 or val(k$)>2 then 90
80 on val(k$) goto 100,200
90 print"qq";spc(5);"you must select either the number
      '1' or the number '2'. try again!":goto 60
100 dload"ss2",d1
200 dload"sb7",d1
210 end
```

On to more encouraging things. I want to thank Liz Deal for her great efforts on Key Trix. This algorithm has saved me hours of time writing programs. (One thing she doesn't mention, or maybe I didn't understand it, was that with Key Trix loaded Superscript doesn't work right. Guess that's because of where Key Trix and SS reside? Anyway I used it to modify my Beeline program to allow me to modify my setup files. Yep, the goto............ and gosub........... numbers didn't get changed when I re-numbered. But thanks to someone else finding the same thing, I found the cure in the last CBUG. Thanks to you.

On Beeline, and the modification I mentioned above. I use many Bulletin Boards here in the Omaha area as well as Delphi and a Military network. I thought it would be easier to have the program ask me what BBoard I wanted to use, then let the program load the files. All I really needed were the F-KEYS and the SETUP file. (Baud, parity,echo,etc.) The phone numbers file will work for all my BBoards. Well, after a few hours I had what I wanted. It works fine, and saves some little time, not to mention disk space, as I put these all on my backup to my backup Beeline disk.

I know that there is a new version of Beeline out and it really sounds like an improvement on an already great routine (Why can't I download/upload program files, only seq files?). So maybe this won't work on the new program. Anyway here is what you do:

```
-----------------------------------------------------------
!  ***CAUTION: MAKE SURE THAT YOU USE YOUR BACKUP TO       !
!   YOUR BACKUP DISK BEFORE STARTING THIS PROCESS,         !
!  UNLESS YOU HAVE MONEY TO BURN AND DON'T CARE ABOUT      !
!  TRASHING YOUR ONE AND ONLY MASTER DISK.****             !
-----------------------------------------------------------
```

1. Load and run Key Trix

2. Load the basic portion of Beeline. (dload"beeline1.4. basic")

3. Now use Key Trix to renumber the lines (Be sure to use the correction found on page 3 of the Fall CBUG ESCAPE.

I used the following: 'ren5, 100'. You should do the same as I will use MY new line numbers from now on. You have to do this, as the Beeline program contains 1254 line of code with no line number spaces between lines, ie 101,102,etc. Why did you do that Ken?

4. Type in the following:

```
    a.      6375        input        "Enter        +BBOARD*
NAME:DELPHI1.1########"
                 ;sf$:print"$"
          ("+"=rvs key)
          ("*"=off key)
          ("#"=back arrow key)
          ("$"=down arrow key)
```

***Note: "DELPHI1.1" will show up everytime you elect to modify/change the setup file and when you first load the program. Just press return to load your DELPHI file or enter your own BBOARD name in this spot. (More about building your files in a minute)

```
    b. 6380 sp$(4)=sf$+".fkeys"
    c. 6385 sf$=sf$+".setup
    d. 6390 return
    e. 5583 gosub 6375
```

5. That's it. All we've done is directed the program what function keys and terminal/modem parameters (ie. baud,parity etc.) to set so that these parameters match the BBoard you want to work.

6. Now save the program to a new Beeline file (dsave"beeline1.6.basic")

7. Now you have to build the "F-KEY" and "SETUP" files for each BBoard you want to work.

8. Follow the directions you got with the Beeline program starting on page 12 to modify your SETUP file, except don't replace the disk. (Almost forgot, you need to load and run either the original Beeline or your new one to get to this position)

```
------------------------------------------------------------
!   MAKE SURE THAT YOU ARE NOT USING YOUR ORIGINAL          !
!   DISK FOR THIS AS THINGS CAN, AND ALWAYS DO GO           !
!   WRONG WHEN YOU USE THE MASTER DISK:  THAT'S             !
!   ORPHAN'S COROLLARY TO MURPHY'S LAW***)                  !
------------------------------------------------------------
```

9. When you select 2 (as directed on line 1, paragraph 4, page 12) you will be asked for a file name. Replace the reversed "beeline1.4.setup" with your file name and delete any extra characters with the space bar.

**** Remember in step 4.A. (line 6375) I used "DELPHI1.1" and said that it will show up all the time? Well, most of the time I use Delphi when I first power, so all I have to do is press return and my files load. I can now call Delphi and go to work. If you don't use Delphi that often or have another net you use, then change that (DELPHI1.1) to the one you do use. The thing to remember here is that whatever you use in step 4.A will cause the program to load the fkey and setup files for that BBoard unless you change it at the prompt. PS. Nothing magic about the "1.1" its just my file version convention.****

10. O-K, almost done, it's faster to do than to write or read believe me. Now we need to modify the function keys. If you don't need to change them ever, then you can delete line 6380 in step 1. above. If you're like me you use these little beauties to reduce your time on line. Don't go back to Menu 1 as directed by the Beeline instructions in paragraph 4 page 12. Instead go to Menu 2. Now modify your function keys as desired. Save them to the SAME FILE NAME as you did your SETUP file. (Use the same procedure outlined in 9. above)

11. The above must to be repeated for each BBoard you want to use. If you look at the code for Beeline you will see that all the files are "Beeline1.4... " All we are doing is renaming the Setup and F-key files so we can address different files for each board instead of using a new disk for each board as the directions would lead you to do.

12. You can do the same thing with the phone list if you really want, it requires another line of code at 6387, but if you really want to do it, let me know and I'll figure it out for you.

13. O-K now reload Beeline.basic1.6 and add one last line of code. Here's where the feint hearted falter add at 50000 dsave"@Beeline1.4.basic". This replaces the program you have been using before now. (See why I said use your backup to your backup)

14. Now run the new pgm and see if it works. It should, if not retrace your steps and check everything out. If still no luck give me a call.

This has gotten a lot longer than I expected. If Norm thinks that it's worth the time to publish then I guess it is worth the time to write. But one last thing. In Pre Release #4 there is a great little pgm for making F-key templates. Really comes in handy now with several F-Key configurations for different Boards! (this too needed some correction so I modified it somewhat to print out properly. It too is on the disk I sent Norm).

I have included the some utilities and a Superbase program that will create and catalog, in Library fashion, and a modest file to let you find articles in your favorite magazine or book.

I'll keep plugging away at writing Basic programs that do what I want, until I find a good book or some help on how to write ML programs. If you need some help writing programs let me know. I think I'm done traveling for a month or so and really enjoy computing.

Lt Col John A. Wright
818 Juniper Dr., Papillion, Ne.
402-339-5728

Chicago Area B128/256 User's Meeting Information

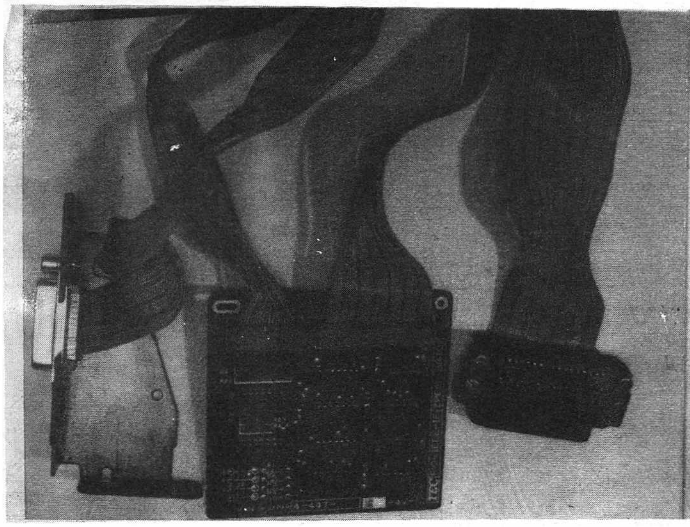
If you are traveling any distance be sure and check with the coordinator as to meeting contents and possible rescheduling or weather cancellations. Meetings are educational/tutorial in content from beginning to end + Question and Answer sessions. We do have systems at the meetings.

CBUG EAST: Bethelem Lutheran Church, 1334 Wesley, Evanston Il. Fourth Sunday (frequently rescheduled). 2:00 p.m. Location is 1 block N. of Dempster St (Rt. 14), 1 block E. of Ridge Ave. Expressway access Rt. 94, exit east; Rt. 294 (N. bound exit only), exit east.

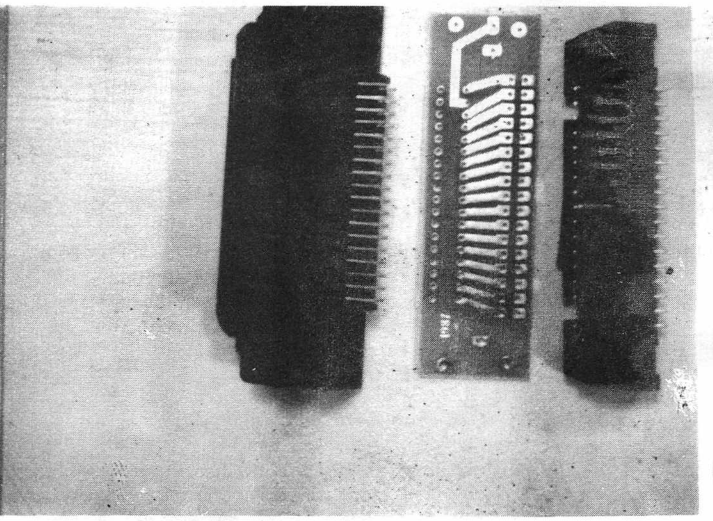
Meeting Coordinator:
Marilyn Gardiner
(312)866-9159

CBUG WEST: First Congregational Church of West Dundee, 5th & Main streets, West Dundee, Il. 7:30 p.m. Main street is Rt. 72. The church is located a few blocks east of Rt. 31 and about a mile west of Rt. 25. West Dundee is immediately north of Elgin Il.

Meeting Coordinator:	Alternate contact:
Warren Swan	Herb Gross
(312)665-1541	(312)695-1316



IEEE to Centronics Adaptor Board
order #11221 $35.00



Connector type changer (comes assembled)
order #11236 $15.00

USE THE 6400 IEEE CONVERTER WITH OTHER PRINTERS

A couple of issues back Warren Kernaghan suggested that the 6400 converter board was a superior method of IEEE to Centronics interfacing. So why not rewire the adaptor board's flat cable to a standard 36 pin Centronics connector? Easy said, not so easy to do even for fairly good technicians. So, CBUG has made up a little circuit board and rounded up the necessary mating connectors to produce a type changer adaptor. Plug the type changer into the Centronics header on the IEEE adaptor, and the other side of the type changer into your Centronics ported printer. Walla, instant Centronics with all the added features Mr. Kernaghan wrote about.

For about a year, CBUG has been offering the IEEE internal converter designed for the CBM 6400 printer. It is a 4" square board with two long flat cables, one of which has a standard IEEE 488 connector, and the other has a 34 pin dual row 1/10" header to connect with the logic board in the 6400. Unfortunately the header with not mate with standard Contronics connectors. The circuit board is double sided plated thru, the connectors prime quality with gold plated contacts.

These adaptors are believed to work with most common printers, though they will not work with my large Daisywriter 2000. They do work with the Star and Cannon printers and many others which now follow the standards. If by chance the adaptor does not work with your printer, we'll refund the full purchase price.

The adaptors have jumper positions to select device numbers other than 4. They are not enclosed so you will want to mount or tape them securely out of the way or install them in a protective box. In some cases, the connector on the printer uses bailing clamps to keep the connectors latched together -- so you may need to use a short Centnronics extension cord compatable with the bailing clamps which are readily available locally or via mail order (see Matos's article).

PAGE 2, June 6 1987 - - - - PREVIOUS RELEASES ORDER FORM

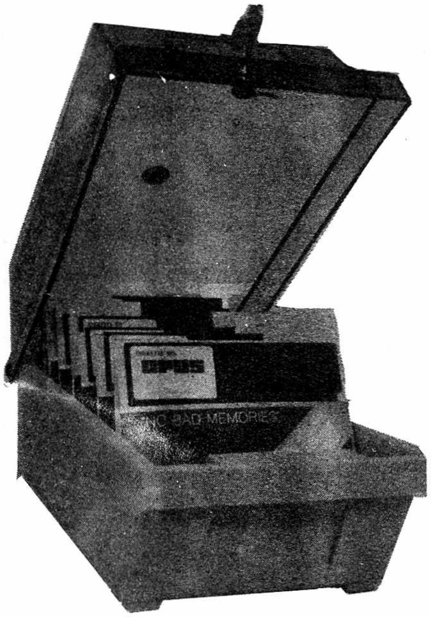
Please repeat your name and zip code per chance your order sheets are separated:

Shipping Name: ______________________________ Shipping Zip Code: __________________

Description	Quantity	Stock #	Price	Extension
DSDD Premium Opus Disks /Pkg of 10 w/ sleeves & labels		10021	6.00	
Illinois residents add 7% Sales tax above item only				
RR1 Norm's Utility v1.2		12862	9.00	
CBUG #3 Swan's Utility #1		12881	14.00	
CBUG #6 CBUG/TPUG #1		12913	9.00	
CBUG #47 dFile database pgm--Available ONLY to US members		11856	16.00	
CBUG #7 Northrup's Superbase Applications		12932	9.00	
CBUG #13 Superbase tutorial pgms & Leighfield aids texts		12787	9.00	
CBUG #31 Superbase Corner & Hints		12538	9.00	
CBUG #15 Friendfam (Superbase application pgm)		12716	14.00	
CBUG #33 Medical Accounting (Superbase application)		12540	9.00	
CBUG #49 Medical Finance #2 (Superbase application)		11875	9.00	
CBUG #50 Educational Records (Superbase application)		11889	9.00	
CBUG #8 Sermons		12946	9.00	
CBUG #9 CABS GL pro forma #1		12951	9.00	
CBUG #11 Terminal Pgms w/ BTerm		12257	14.00	
CBUG #11a Terminal Pgms w/o BTerm		12261	9.00	
CBUG #12 Scott's B-Mon		12984	14.00	
CBUG #16 Swan's Basic Course		12773	19.00	
CBUG #17a Liz's Utility v1.2a		12670	16.00	
CBUG #46 Liz's SECOND Utilities		11841	14.00	
CBUG #18 Games and Education		12792	10.00	
CBUG #19 Old BUG texts and programs		12805	9.00	
CBUG M20 CBUG Utilities etc #2		12768	9.00	
CBUG M45 CBUG Utilities & Misc. #3 (mislabeled #2 fall 87)		12837	9.00	
CBUG #21 Retail News Distribution pgm		12699	9.00	
CBUG #22 Math Education Programs		12701	9.00	
CBUG #23 Bible Games		12735	15.00	
CBUG #24 8432 Emulator Disassembled		12720	9.00	
CBUG #27 Goceliaks Gold Mine - disk utilities/engineering		12492	9.00	
CBUG #28 Casey's Scrubber		12504	19.00	
CBUG #29 CBUG TPUG P1 & P2		12519	9.00	
CBUG #30 Harrison's Assembler		12523	35.00	
CBUG #32 Kernaghan's Utilities v3		11536	10.00	
CBUG #36 London Sampler		12561	9.00	
CBUG #37 SUPERPRINT		12576	19.00	
CBUG #40 Public Domain Math A		11771	10.00	
CBUG #41 Public Domain English A		11785	10.00	
CBUG #42 Public Domain GHBT		11790	10.00	
CBUG #43 Public Domain Science A		11803	10.00	
CBUG #44 Public Domain Science B		11818	10.00	
SET of 5 Public Domain CBUG #40 thr #44 inclusive		12822	45.00	
Physical Exam for the 1541		12223	35.00	
Physical Exam for the 4040		12238	35.00	
Physical Exam for the 1571		12242	35.00	
CBUG #48 CBM Diagnostics adapted for the B128		11860	9.00	
PR1 Pre Release #1		12824	9.00	
PR2 Pre Release #2		12839	9.00	
PR3 Pre Release #3		12843	9.00	
PR4 Pre Release #4		12749	9.00	
PR5 Pre Release #5		12542	9.00	
PR6p Pre Release #6 partial		12557	6.00	
CBUG #10 Fall 1985 ESCAPE and prior files - disk		12965	9.00	
CBUG #25 Winter/Spring 1986 ESCAPE print files - disk		12665	9.00	
CBUG #26 Jan. 1986 Telecom issue and CBUG #25 overflow		12651	9.00	
CBUG #38 Summer part 1 1986 ESCAPE print files - disk		12580	9.00	
CBUG #52 Summer part 2 1986 ESCAPE print files - disk		11906	9.00	
CBUG #53 Winter/Spring 1987 ESCAPE print files - disk		11911	9.00	
NOTE: Fall '85 ESCAPE and Jan '86 Telecom issues are out of print. Substitute print file disks.				
Winter/Spring 1986 ESCAPE, copy of publication		12449	6.00	
Summer 1986 ESCAPE Part 1, copy of publication		12468	4.00	
Summer 1986 ESCAPE Part 2, copy of publication		12473	3.00	
Fall 1986 ESCAPE, copy of publication		12346	5.00	
TOTAL THIS PAGE ---- Extend to main order form				

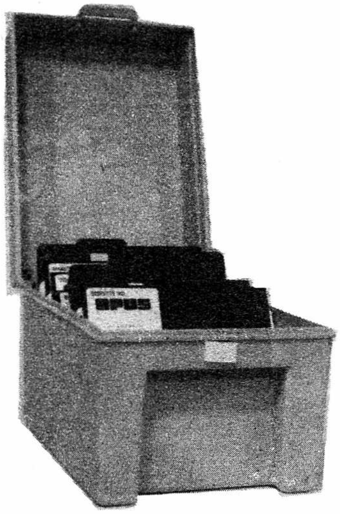
DATA CASE ORDER FORM

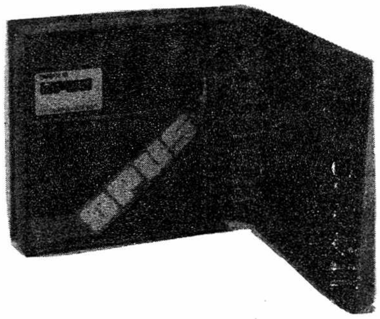
DD120L Holds 120 5¼ Disks
Locking Clear Top, Beige Body
10 Dividers w/ Labels
Order #10960



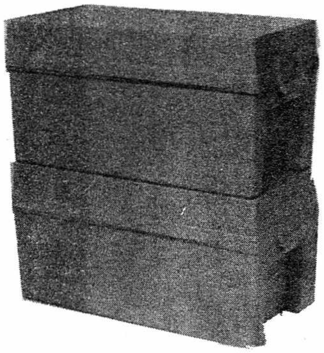
DS100L Holds 100 5¼ Disks
Locking Clear Top, Beige Body
seable Hinged or Lid Nested Underneath
Dividers w/ Labels Order #10975

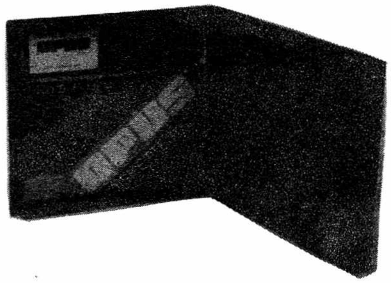
Extra Heavy Duty Monitor Swivel Stand
Holds Monitors up to 12" on Center Feet
Can be used locked in positon or Adjustable
Beige MS1212 Order #10937





Library Shelf Box, Holds 10 5¼ Disks
Living Hinge Design. Pastel colors
- we'll ship all same in an order
DD10 Order #10989

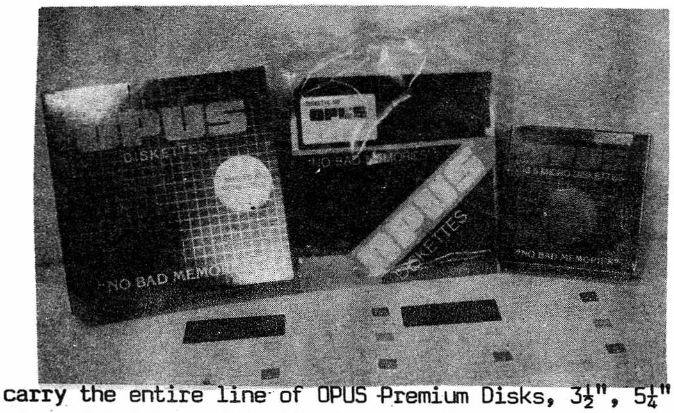




DD5 Mailer/Tote Box, Holds 5 5¼ Disks
Living Hinge Design. Pastel colors
- we'll ship all same in an order
Order #10941

Heavy Duty, Compact, Stackable, Non-Breakable.
One Piece Living Hinge Design. Six Dividers
Factory Installed. Beige body and cover.
D55 Holds 60 5¼ Disks Order #10994
D33 Holds 40 3½ Disks Order #10918

NOTE: Due to shipping and Packaging costs, our data case products are not available outside N. America!



We carry the entire line of OPUS Premium Disks, 3½", 5¼" & 8" Single and Double sided, Single & Double Density; PLUS, quad and High Density AT compatable 5¼ disks.

Item	Description	Quanty	Price	Extension
10960	DD120L locking 5¼ case		$16.00	
10975	DS100L locking 5¼ case		12.50	
10922	DD50L same as 10975; holds 50		8.00	
10994	D55 stackable 5¼ case		6.45	
10918	D33 stackable 3½ case		5.00	
10989	DD10 5¼ library case		1.50	
10641	DD5 5¼ mailer/tote case		1.00	
10937	MS1212 Monitor Stand		16.00	========

Name____________________ Zip Code _________ TOTAL __________
Prices and specifications subject to change without notice.