

---

North Texas PC Users Group

6.7  
July 1987



**North Texas PC NEWS**  
(STARMAIL ADDRESS S1563)

Published monthly by members of North Texas Personal Computer Users Group for their use. Members each receive a free subscription; for others, price of the NEWS is \$2 per copy. Members are requested to notify the Membership Director in writing of address changes. Send all editorial correspondence to North Texas PC NEWS, 2025 Flockcreek Drive, Arlington, Texas 76010.

Editor/Publisher  
John Pnyl (817)275-4109  
Assistant Editor  
Carlisle Phillips (214)348-2345  
Newsletter Exchange Editor  
Tom Prickett (214)690-9087  
Software Review Editor  
Dick Gall (214)234-8888  
Advertising Manager  
open

The opinions expressed herein are those of the authors and do not necessarily reflect those of the Group or its members. Copyright (c) 1987 by North Texas PC NEWS. (Articles without specific copyright notices may be reproduced by other User Groups if credit is given to the author and the publication.)

**Deadline:**

All material for publication in North Texas PC NEWS (articles and ads) must be received by the NEWS staff no later than the 15th of the month.

**Articles:**

Please do not right-justify, indent or otherwise code the copy. If column alignment is critical, send along a hard copy, or written instructions. Article submission is preferred by modem (817/275-4109 or StarText 51583), or disk in ASCII format, unjustified. If you send a disk, please include a printed copy of the article to assure accuracy. Double spaced, typewritten copy is acceptable but must be received a week before the deadline.

**Circulation:**

North Texas PC NEWS circulation is 1200. Member distribution is 898, remaining copies are distributed to PC user groups around the country, and to advertisers, prospective members and others with common interests.

This issue of North Texas PC NEWS was composed using Xerox Ventura Publisher. Repro was printed on a Xerox 4045 Laser Printer and Toshiba P351 dot matrix printer. Typesetters include: Times, Prestige Elite and GTHIC151.

**DEADLINE**  
Copy deadline for August  
PC NEWS:  
Wednesday, July 15th.

Meeting Dates:

July Meeting - 2nd Sat. (11th)  
August Meeting - 2nd Sat. (8th)  
September Meeting - 2nd Sat.  
(12th)

*Editor's Notes...*

We had a visitor from Austin at our June meeting. Karen Willis, Secretary of the Central Texas PC Users Group attended our general meeting. She was quite impressed with Infomart, and with the "basement operation."

Two more happy door prize winners: Jerry Stone won the MicroPort, UNIX package which contained 16.8 megabytes of programs and 2200 pages of documentation weighing in at approximately 16 pounds! Jerry said he just might have to buy a little more memory before he can use it. Lee Morton won a copy of Egil Juliussen's new book, "Computer Industry Almanac."

The ads appearing in North Texas PC News help pay for publishing your newsletter. Volunteer writers prepare the great articles you read every month. Show your appreciation to both of them... patronize the advertisers, and thank the writers.

Remember too, that we like to hear from you about the directions we're taking in the newsletter. If you have constructive criticism or suggestions for improving the content, or would like to lend a hand in preparation of NT PC News, give me a call.

The contest voting results were not in at press time. I think you'll agree we've had more good articles since the start of the contest. Don't stop now... keep those word processors sharpened!!!

**Table of Contents**

President's Message . . . . . 1  
Jim Hoisington  
Personal REXX: A Powerful Alternative  
to DOS Batch Files . . . . . 3  
Andrew Chalk, Ph.D  
On Complexity No. 11 . . . . . 14  
Jim Hoisington  
Selecting Your First Computer . . . . . 14  
Doug Russell  
Frustration with COMMAND.COM . . . . . 17  
Dan Rau  
Marketing Computer Services . . . . . 19  
John Keohane

Editor's Notes . . . . . i  
Agenda . . . . . 1  
Officers . . . . . 2  
Advertising Prices & Policies . . . . . 17  
Membership Application Blank . . . . . 18  
Room Assignments . . . . . 21  
Features:  
Disk of the Month . . . . . 12  
Nerd on the Street . . . . . 15  
SIG Reports . . . . . 20  
SWAP SHOP . . . . . 20

**July 11**

Charles Kroboth, Program Director

9:00 AM to 9:45 AM

AUDITORIUM

**\* NANTUCKET'S CLIPPER \***

Nantucket Software's Vice President, Burt Durant, will be presenting Clipper. Mr. Durant will show us the advantages of using their dBase, compiler, Clipper, over buying a dBase runtime.

10:00 AM to 11:00 AM

AUDITORIUM

**\* NEW VIEWS ACCOUNTING \***

Toronto based Q.W. Page Associates will be sending their Vice President of Sales and Marketing, Ron Benn, to demonstrate New Views Accounting Software. A nested-spreadsheet format is used providing a dynamic interaction between all parts of your company's books via pull-down menus.

**Prez Sez...**

New Secretary.

Laura Murphy, our secretary has resigned. She lives in Dallas and works in Fort Worth. That, plus her additional commitments, were just too much. Thank you, Laura, for being our secretary for a year.

The new secretary is David McGehee. David also leads the hardware SIG.

Group Purchase Director.

Andrew Chalk, our group purchase director and statistician, also resigned last month. I had asked Andrew to do the job for six months and he did more in that six months than most people could do in two years. Besides many group purchases, he got our members a discount at a local computer store and designed and analyzed our membership survey.

The survey has helped to attract advertising to our newsletter and speakers for our programs.

Thank you, Andrew, for a job well done!

The Group Purchase Director position and the Membership Survey Statistician position are both open currently.

Thank Goodness Our Treasurer DID NOT RESIGN!

Joe Brophy, our treasurer, sent me a copy of the letter from the IRS declaring us a tax exempt organization. That means that if you work for a large corporation that would like to donate us a mainframe for us to use as a bulletin board system, you can tell them that any contributions to the NTPCUG are deductible.

Thank you, Joe, for all the work you have been doing on the behalf of the group. Joe is a CPA who special-

izes in tax practice. He's earned three gold stars for his work for the NTPCUG.

I would also like to thank the IRS. That's right, I said thank the IRS. Our reviewer was both helpful and encouraging beyond and above the call of duty. When the Congress passes tax law, it is anything but logical. Our reviewer steered us through the many forms and hurdles to achieve this status.

Forms are in the mail to the State of Texas to achieve tax exempt status with the State.

Coming Up.

When I took over as president, the NTPCUG had the organizational structure for a group of 250 to 300 members. My goal this year is to structure it for 1,500 to 3,000 members. (I did my masters thesis under a professor who was an expert on the structure of organizations so I'm very sensitive to organizational structure.)

Joe Brophy and I have been working on the accounting side. That is almost completed. Next comes the part that you see, our relations with you and the public.

Some of the things that we will be doing this summer are: membership cards, a brochure explaining who we are and what we do, a 24 hour telephone messaging system and possibly, a new logo.

I have set mid-August as a goal for completing these items. Typically we experience our largest growth in the fall and I want us to be ready.

Finally

User groups are starting to become recognized for what they really are: "users helping users."

President's comments continued on page 3.

## North Texas Personal Computer Users Group, Inc.

P.O. Box 780066, Dallas, TX 75378-0066

(Send membership dues, renewals & address changes to Membership Dir. address at bottom of this page.)

### Board of Directors

**Jim Hoisington, Chairman**    **Jim Graham**  
**Reagan Andrews**            **Stuart Yarus**  
**Kathryn Crawford**

The North Texas PC Users Group, Inc., is a non-profit, independent group, not associated with IBM or any other Corporation. Membership is open to owners and others interested in exchanging ideas, information, hardware, predictions, and other items related to IBM Personal and compatible computers. To join the Group, complete the application blank printed elsewhere in this issue, and send it with \$24 membership dues to address shown below. A subscription to the newsletter is included with each membership.

The Group meets once each month, usually on the second Saturday. See cover for date, time and place of the next User Group meeting.

### Officials

President - Jim Hoisington (214)416-3101 h  
 President-Elect -  
     Reagan Andrews, Ph.D. (214)828-0699 h  
 Program Chair. - Charles Kroboth (214)746-5335 w  
 Treasurer - Joe Brophy (214)891-8187 w

Secretary - David McGehee (214)681-0202 h  
 Membership Dir. - Bob Russell (214)422-4269 h  
 Disk of the Month - Tim O'Neil (817)267-8981 h  
 Group Purchases -  
     - Open -

### Special Interest Groups

SIG Coord. - Phil Chamberlain (214)243-5034 h  
 APL - Jim Fiegenschue (214)539-9281 h  
 Artificial Intel. Jim Bender (214)423-3470 h  
 Astrometry - Arlin Collins (214)351-5137 h  
 Assembler - Neil Bennett, Ph.D. (214)517-6854 h  
 BASIC Applic - Ross Carter (214)238-8638 h  
 Beginners - Gordon Baskett (214)247-1873 h  
 Business Applic. Bruce Schubert (214)991-5967 w  
 Enable - Jack Lundberg (214)596-8160 h  
     - Susan Watts (214)416-0077 h  
 C Language - Sid Nolte, Ph.D. (214)233-6178 h  
 Communications - Fred Williams (214)492-1315 w  
 Databases - Chris Morgan (214)746-5335 w  
     - Bob Monaghan (429)3245 w  
 DBase - David Hayden (214)380-8172 h  
 DOS - Jim Hoisington (214)416-3101 h  
     - Reagan Andrews, Ph.D. (214)828-0699 h  
 Genealogy - Minnie Champ (214)341-6507 h

Graphics - Mike Durbin (214)271-8779 h  
 Hdw Solutions - David McGehee (214)681-0202 h  
 LOTUS - Peyton Weaver (214)462-0552 h  
     - Mark Gruner (214)373-3147 h  
 Pers Fin Plan - Hugh Christensen (214)484-3072 h  
 Programmers - Neil Bennett, Ph.D. (214)517-6854 h  
 Science/Engr. - Sam Leven (214)991-7642  
     - David Lamb (214)931-3068 h  
 Turbo Pascal - Warren Ferguson (214)692-2506 w

**BBS: SYSOP:      Tom Prickett (214)418-6969**

#### Address Changes, etc...

Payment of dues, address changes, and inquiries about membership should be directed to  
 NTPCUG Membership Director  
 135 Skyline Drive  
 Plano, Texas 75074

(Check newsletter mailing label for your renewal date..)

# Personal REXX: A Powerful Alternative to DOS Batch Files

Review by Andrew J. Chalk, Ph.D.

Copyright (C) Andrew J. Chalk, 1987. All Rights Reserved

## Quick Summary

**Personal REXX from Mansfield Software is three things in one. A powerful alternative to the DOS batch facility, a good programming language for beginners, and a tool to write macros for Mansfield's editor KEDIT. This month I review Personal REXX. Next month we will look at KEDIT and how Personal REXX can enhance it.**

Anyone who has used batch files knows that the DOS batch facility is rarely dignified with the description "programming language". It limits the user so much as to prevent him doing some very basic programming tasks. For example, you cannot ask the user a question in a batch file and then (make your program) act on the response. You cannot access system information (such as the time of day) and act on the response. You cannot make a batch file switch from an arbitrary directory, load a program, and switch back to the original directory. There is virtually no "structure" to batch files and elementary program control structures like "DO WHILE" loops and "SELECT" statements are missing.

It is no surprise that a myriad of utilities designed to do just these things have appeared on bulletin boards, or that PC Magazine could entitle a recent program-

---

Prez Sez... continued from page 1.

We are not a bunch of teenage hackers! Andrew Chalk's membership survey proved that. We are getting more and more mail from vendors wanting to sell us something or to make a presentation because they realize that we use our computers for more than playing games.

In a recent issue of PC Magazine, the editor said that he liked to attend user group meetings. So, I sent him an invitation to attend one of our meetings. I also enclosed a picture postcard of the INFOMART to show him our fine meeting place.

Our program chairman, Charles Kroboth, suggested that I tell him that we went in with AppleCorps and bought the building as a meeting place because "people in New York will believe anything about people in Dallas!"

Jim ▲

ming article "The Utility that DOS Forgot". The fact is that, while applications software and development tools for microcomputers have advanced by leaps and bounds since the introduction of the PC, the basic system utility programming tools included with DOS have remained neglected by Microsoft and IBM.

Mainframe programmers are the most aware of this situation, since it is the rule rather than the exception that mainframe operating systems have a systems utility language. UNIX has its shell scripts, MVS has job control language, and TSO has CLISTS. People whose computer baptism was into microcomputers may miss it less, much as the person who has always used floppy disks does not miss a hard disk. The fact that few users survive a trip in the other direction, from hard to floppy disks, suggests that hard disks are useful. Likewise, the fact that mainframe programmers miss their systems utility language when they move to the microcomputer should give us pause for thought that maybe we ought to see what they mean.

This review is about such a systems utility language -- REXX, the language that IBM mainframes use under the VM/SP operating system. It has been implemented, in full (with the exception of some VM/CMS-specific features), for the PC by Mansfield Software in "Personal REXX". Mansfield also produces an editor, KEDIT, that is a PC implementation of the IBM mainframe editor XEDIT. Personal REXX is integrated with KEDIT in a functionally similar way that mainframe REXX is integrated with XEDIT. Both editors use powerful macros written in their respective version of REXX.

Let's start with an introduction to the REXX language, then deal with some details of interest mainly to programmers. Next we will look at Mansfield's implementation, and we will finish with a discussion of how Personal REXX can be used. Given the suitability of REXX as a first language, the language description is aimed at the novice. Needless to say, a review of this length can only skim the language and convey a few basic concepts.

## An Overview of the REXX Language

REXX was developed by M.F. Cowlshaw at IBM's U.K. labs during 1979 through 1982. Early versions were distributed only internally to IBM over the company's VNET internal network. REXX spread throughout IBM like a kind of subculture long before its eventual release in 1983, supplanting IBM's official utility languages EXEC and EXEC2. At one point, Cowlshaw was replying to 350 pieces of electronic mail a day from IBM employees in 20 countries. REXX took on new significance while this review was



being prepared. IBM announced at their Systems Application Architecture (SAA) presentation that REXX will be the standard "scripting language" across their product line from PC to mainframe. The objective is that a REXX program written for one system will run unamended on all others.

REXX is best explained with examples. In all cases below, the computer's screen responses are shown in bold type. Here is the simplest possible REXX program:

#### Example 1 -- The World's Simplest REXX Program.

```
say "Hi y'all world"
```

All this program does is print to the screen

```
Hi y'all world
```

This illustrates one of the most important design principles behind the REXX language. Make the programming statements descriptive of their function. The equivalent statement in DOS is "echo Hi y'all world" which, one might say, is not as intuitive but is a reasonable substitute. Note that single or double quotes may delimit a string literal, but if you want to use one literally inside the string, then you must use the other as the delimiter. To take matters a step further, REXX is not only expressive of its purpose in its statements, it also allows variables the user defines to be descriptive as well. Variables can be almost any length, so if you want to name a variable "encyclopaedia", it is quite legal to do so. Let us build on the above example to introduce variables.

#### Example 2 -- Creating Variables in REXX

```
say 'Enter a number' pull number say 'The
number is' number
```

The screen output is

```
Enter a number
```

```
5
```

```
The number is 5
```

There are several new things here. In line 1, the program displays the statement within quote marks on the screen. This is just like Example 1. The second line illustrates one way of soliciting input in REXX. The statement "pull" pauses program execution for input, in this case from the user but, in other instances, from elsewhere (a facility called the stack, to be discussed below). The user enters the digit "5" and REXX assigns it to the variable "number". Notice two things. First, the variable is given an intuitive name. Second, the variable "number" is created simply by assigning it the value typed in by the user (in this case, 5). There is no need for type declarations or variable declarations in REXX. Just name new variables "on-the-fly" and REXX creates them.

The final line of the program looks superficially like just another invocation of the "say" statement. However, there is more to it. The variable "number" has been assigned to "5" in the previous line. REXX replaces the variable with its value and prints out the digit "5". Notice that the single quote marks in the "say" statement are closed before the word number.

Two variations on the last line illustrate another general principle behind REXX. If the third line were changed to

```
say 'The number is number'
```

then the program would display the line

```
The number is number
```

showing that symbols within single quotes are not interpreted by REXX. Instead, they are passed through as a literal to the operating system. In this case the REXX statement "say" displays the line on the screen, but if you had entered the following

```
'c:\wp' the REXX program would start WordPerfect
from the root directory of your "C" disk.
```

Consider another way of expressing the last line in Example 2. Suppose we leave out the single quotes all together, so that our line becomes

```
say The number is number
```

what happens then? The computer displays the line

```
THE 5 IS 5
```

on the screen. This is just the opposite of the previous case. Since no symbols are within single quotes, every symbol is interpreted. It so happens that the default in REXX is uppercase. Therefore, all symbols in the example are converted to uppercase by the REXX interpreter and passed to the "say" instruction for display. All single spaces between symbols are maintained and (if we had any) multiple spaces are converted to a single space. The last word in the program is converted to uppercase, but REXX has already assigned the digit "5" to "NUMBER" and hence proceeds with the interpretation before the "say" statement displays it to the screen.

Consider example 3:

#### Example 3 -- Local Variables and Subroutines

```
say 'Enter a number' pull number
say 'The number is' number
say 'Twice the number is' 2*number
say 'The square is' square(number)
say 'What power do you want to calculate?'
pull exponent
say 'The answer is' power(number,exponent)
exit
```

```

square : procedure
arg number
return number**2

power : procedure
arg num,exp
calc = 1;
do exp
calc = num*calc
end
return calc

```

This program might proceed as follows:

```

Enter a number
3
The number is 3
Twice the number is 6
The square is 9
What power do you want to calculate?
4
The answer is 81

```

The first four lines contain nothing beyond the material in Example 2. However, line 5 introduces the use of functions in REXX. When the expression "square(number)" is reached, the opening parenthesis abutted onto the word "square" signifies a function. The interpreter jumps down to the line "square : procedure" and begins execution there. The "arg" instruction informs the function that there is a single parameter being passed from the calling (main) program and REXX proceeds to evaluate the function. In this example, the calculation is simple. Square the parameter. The statement "return number\*\*2" achieves this and returns the result to the main program.

Seasoned programmers will notice that I have glossed over an important concept. The value of the REXX variable "number" was "3" on entry to the function "square". Has this changed as a result of the function's action? The answer is that it has not, because the variable identified as "number" inside square is local to that function.



SAY "LET'S HAVE AN OFFICE PARTY."

This is achieved through the use of the word "procedure" following the label "square:". Procedure makes all variables created within or passed to a function local to that function. The variable "number" in the main program is not affected by anything we do to the variable "number" in the function "square".

Experienced programmers know that functions and the closely re-

lated concept of local variables are an important part of a programming language because they help us organize our thoughts. Intellectual Titans don't need structure because they can carry the values of all variables (and much else) in their heads. However, for those of us without such gifts the ability to write programs quickly and to ensure their reliability is enhanced, because we can break down the program into its logically separate tasks. Thus, calculating the square root is a logically separate operation from getting input from the user. For this reason we break it out into a function in Example 3. This ability to break down a long program gives the language "structure", and REXX can rightly be called a structured language.

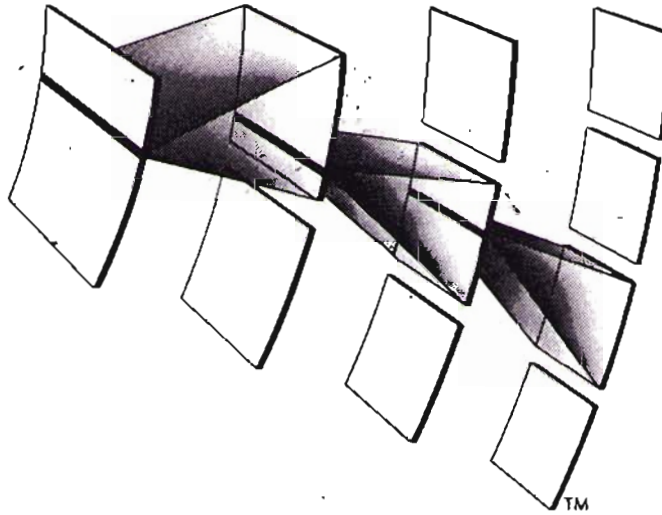
The best known example of a language without structure is probably BASIC. It is with good reason that much BASIC code is referred to pejoratively as "spaghetti". A major innovation of modern implementations of BASIC such as Quickbasic and Turbobasic is the introduction of local variables and other concepts borrowed from structured programming languages.

The last step in Example 3 is the function "power". The main reason for introducing this function is to show that REXX embodies program control statements. In this case the "do loop" is used, but this is only by way of example. The REXX language has as rich a set of program control statements as any other high-level language, including "do...end", "do...until", "do while", "if ... then", "else", "select... when...otherwise" and (my favorite) "do forever" (The last of these only stops when Atari's share of the PC marketplace exceeds that of IBM). The "leave condition" instruction causes an exit from an infinite loop when "condition" is true. The function "power" returns the result of raising "number" to the given power. It is a more syntactically complex than "square", but no more complex conceptually.

The final feature of REXX that I want to discuss is its string handling abilities. Cowlshaw's treatise "The REXX Language" begins with a section on fundamental language concepts. The first principle he chooses to mention here is readability. The concept pervades his book in that Cowlshaw wants a language that can be used to get things done quickly, and he sees part of that as the ability for the user to read the source code and quickly translate the language concepts into their human language equivalents. This approach is quite different from the terseness prevalent in the "C" language, where programs can be made very unreadable (at which point they are sometimes referred to as "elegant"). The readability of REXX code extends beyond the literal sense of the term to the language's ability to handle statements from conceptually the same standpoint as the English language. Consider the following example: ►

# NEW VIEWS™

## The Visual Approach to Accounting



Unlike menu-driven batch systems with separate modules, NewViews opens windows on an integrated hierarchy of accounting documents that are related in real time.

## No Menus · No Modules · No Batches

NewViews is a complete double-entry accounting system for the IBM personal computer family. It uses a new visual approach that is creating unprecedented excitement and interest in accounting software. Now carried by Softsel, NewViews represents the first generic accounting system on the market.

NewViews is a clear indication that software is becoming easier to use. And its clean design and ease of use are getting noticed.

Nancy Trespasz; Computer Reseller News, April 20, 1987

The developers of NewViews threw away all the time-honored assumptions about accounting software and came up with a unique and better way of doing it.

Bill Machrone; PC Magazine, December 9, 1986

This is a generic software package that can be adapted to a variety of applications - the limit is only the user's ability and imagination.

Robert Williams; The Bottom Line, April 1986



### SPECIAL PRESENTATION

Ron Benn, V.P. Sales & Marketing and Co-Developer of NewViews will address the next General Meeting.  
Saturday July 11, 1987  
10:00 A.M. - 11:00 A.M.  
Infoman  
Alexandria Auditorium  
Mike Troxell and Mike Bernard  
Versatile Systems Inc.  
NewViews Master Dealer  
and Authorized Training Center  
Tel: (214) 746-5333

Complete Financial Accounting

# \$695.00

Major credit cards, check or money order accepted.

NewViews is a trade mark of Q.W. Page Associates Inc.  
© 1986 Q.W. Page Associates Inc.  
IBM is a registered trade mark of International Business Machines Corp.  
Softsel is a registered trade mark of Softsel Computer Products Inc.

## (416) 923-4567

**VERSION 1.10**



Example 4 -- String Handling

```
say 'Type in a sentence'
pull response
say 'Number of words is' words(response)
say 'The length of the sentence is' length(response)
say 'Here are the first three words'
say word(response,1)
say word(response,2)
say word(response,3)
say 'here are the fifth through twelfth characters'
say substr(response,5,8)
```

Here is a sample session using Example 4:

```
Type in a sentence
I just can't think of anything
Number of words is 6
The length of the sentence is 30
Here are the first three
I
JUST
CAN'T
Here are the fifth through twelfth letters
ST CAN'T
```

Here we see that REXX can split up a sentence into its constituent words, or even letters, with great ease. These and other features make the language well-suited to text-intensive applications. Note the use of the four built-in functions `words()`, `length()`, `word()` and `substr()`. `Words(string)` returns the number of words in a string, `length(string)` returns its length, `word(n)` returns the word at position `n` in the string, and `substr(string, start, length)` returns the characters in "string" from position number "start" for "length." These built-in functions do not need to be defined in the program, and like all REXX functions they can be nested. My favorite "one-liner" is:

```
say center(strip(reverse(string)),80)
```

which reverses the order of the letters in a string typed on the command line. The order of execution is to first reverse the string, then strip the blanks, then center the result on a line of 80 characters (This is a useful sanity tester for people on networks).

## REXX for Programmers

The foregoing was aimed mainly at non-programmers and therefore went to some lengths to spell out some basic language concepts that apply to REXX. This section is aimed at the programmer and can be ignored by the person with no background in programming concepts. REXX does not require knowledge of programming concepts or jargon. In fact, as I will say below, it is a good choice for the beginner to learn programming.

REXX is a structured procedural language that is designed for fast development and easy debugging. All the implementations of the language available at present are interpreted and Mansfield Software Group says that they have no plans at present for a compiled version. Given this limitation, REXX is not a good choice for processor intensive applications such as statistical software. Furthermore, although Mansfield has gone to some length to provide access to hardware functions and the operating system, Personal REXX does not have the system access found in, for example, C, or implementations of Pascal such as Turbo Pascal. The language includes no hardware-specific functions since it is designed to be portable. The Personal REXX implementation has 10 PC hardware information functions that return such things as the current video mode, the equipment flag, RAM size and ROM date. In addition, there are 10 functions to change features of the PC hardware. These include delays, sound effects, direct keyboard read without echo, access to ports, and RAM access by memory address. The DOS-specific functions in Personal REXX provide REXX versions of DOS commands that execute more quickly than a direct use of quoted DOS commands and return a return code to the calling REXX program. Some of these have extensions to the functions available from the basic DOS command. There are also screen handling functions making the development of color menus straightforward. One feature that I would like to see in a future release is REXX functions to make interrupt calls and DOS function calls.

The REXX language itself is high-level (as the examples above make clear) and syntactically is close to `@ascal`. It does have built-in functions to handle bit-level operations (`BITAND()`, `BITOR()`, `AND BITXOR()`) and similar functions to convert between character, ASCII ordinality, and hex representations of bytes. The language provides all the usual program control structures expected of a high-level language. The ability of the language to treat variables as strings but also interpret numeric strings as numbers is extended to real numbers as well as integers. The default precision is 9, but can be altered at will. Arrays are implemented using a stem-leaf syntax, i.e. "element.1" and "vararray.i.100" are valid references to elements of a one-dimensional and a two-dimensional array respectively. Since no explicit variable declaration is required, it is legal to increase the size of arrays just by assigning values to new elements. The language supports recursion.

The Personal REXX implementation includes an Applications Programming Interface (API) that allows a user application to invoke REXX, and for the user to extend the range of functions in the REXX language. The distribution disk contains example interfaces to 8086 assembly language, Microsoft C and Pascal.



## The Personal REXX Implementation of REXX for the IBM PC/XT/AT

### The Personal REXX System

Regular REXX users will wonder whether the PC implementation of REXX is as seamless as mainframe REXX. In this respect, Mansfield has done a creditable job. The user can choose the degree of integration he desires. The tradeoff is that each component of Personal REXX uses memory. At a minimum, a resident interrupt handler occupying 2k must be resident in memory at all times. The main REXX program occupies 100k but must also have a storage area allocated for variables it creates. The size of this is set by the user, but must be at least 10k to run a REXX program of reasonable size and can extend up to 40k. To run applications under REXX, Mansfield recommends a minimum of 384k but someone willing to accept less functionality could use the program on a 256k machine. If memory is available, the REXX interpreter can be made resident. In its base configuration, a REXX program is invoked from the DOS prompt with the command "REXX ogramname." Most users will also load the REXX batch manager, a resident 2k file, enabling them to invoke REXX programs the same way that batch files are invoked (i.e. ogramname).

One of the most useful features of REXX is the program stack. This permits one to store instructions for later execution and has no counterpart in the DOS batch facility. To use the stack in Personal REXX requires another resident utility, this one occupies 2k plus the reserved stack area (typically user set at 8k), and may be defined as a device in the CONFIG.SYS. The driver for this facility occupies about 500 bytes.

To summarize what we have so far, full-featured REXX occupies 16.5k to 70.5k (subject to user preferences) while it is not running. Add 100k to for the interpreter and 10k to 40k for a (required) internal storage area when REXX is running. In practice, I decided to run REXX from a ramdisk (VDISK in extended memory) rather than having it permanently resident, and to load the stack manager (8k) and set the internal storage area at 40k. This occupied a total of 16.5k when not running a REXX program, and 156.5k when running REXX.

On IBM mainframes, the interface between REXX and the system is virtually seamless. In particular, several CMS commands and REXX work closely together. One of these is the CMS global variable facility, which is a more powerful analog to the DOS environment "set" command. Since this would not otherwise be available in Personal REXX, Mansfield has implemented the CMS global variable facility as a separate program that one can invoke within REXX. Likewise with the CMS LISTFILE, EXECIO,

MAKEBUF, DROPBUF and DESBUF commands. All of these commands can be called from within REXX and function in a similar manner to their CMS counterparts.

In operation I found Personal REXX to be reliable and easy to use. I used the program with SIDEKICK, CED, and a resident clock all simultaneously in memory. This setup functioned fine, despite Mansfield's warnings about memory resident programs "clashing". Mansfield allows the user to change the Personal REXX Interrupt (by default, 60H) through a DOS environment variable, so I suspect that Personal REXX will prove a well-behaved resident program. The only program that refused to coexist happily with Personal REXX was "Taskview", a multitasker. At one point within Taskview, the machine started to reboot itself while I was using Procomm! To be fair, Taskview has its problems independently of whether Personal REXX is loaded or not.

I would recommend running Personal REXX off a ramdisk, and having that as the first drive in your path. That way, the hard (or floppy) disk does not have to grind away searching for REXX before loading the REXX program. In this setup, REXX programs load instantly and run far faster than DOS batch files. If you write cpu-intensive tasks in REXX (such as sorting programs, or a program to find every occurrence of the word "the" in a large file) then you might notice the processing time. For instance, Example 3 above iterates once for each power you calculate. It took REXX five seconds to tell me that 2 raised to the 1,000th power was 1.07150863E+301. The only deviation from the official REXX language, as specified by Cowlishaw, was in Example 3 (above). Personal REXX required parentheses around the operand to the "return" instruction in the function "square".

During the preparation of this review, Mansfield sent a beta release of version 1.6. The enhancements include a windowing package (including an example using spreadsheet style menus), new functions to read and write to the screen (including the ability to specify the whether the BIOS or direct screen addressing should be used, and whether "snow" should be accounted for), and several new utility functions.

### Personal REXX Applications

That brings us to the question of what might one use Personal REXX for? As mentioned at the start, Mansfield's KEDIT can use REXX programs (called macros) to perform almost any task one could ask of an editor. We will discuss this next month when KEDIT is reviewed. The other two uses of Personal REXX are, first, as a replacement for the DOS batch facility. Second, as a language for applications that are beyond the power of the dos batch facility.

As an example of the first, consider the steps one goes through to invoke the Microsoft Macro Assembler

(and similarly for most compilers). Typically, a DOS batch file contains a set of commands to assemble, link, and exe2bin the source file. The DOS batch facility cannot even prompt the user to ask whether he wants to exe2bin. In Personal REXX, a typical assembly language programmer's .REX file could do the following. First, check the file existed, then call the built-in REXX function to get the time. If all checks out, assemble the file and act on the return code, making sure to return the finishing time and the elapsed time. At the exe2bin stage, the REXX program prompts for a response and then branches on the result -- either creating the .COM file and cleaning up the disk or going straight to the cleanup. Again, REXX could give the programmer timing statistics if desired. A simple program like this provides more functionality than the DOS batch facility and takes only about 15 minutes to write.

There are a multitude of examples of applications programs written in REXX (including several on Mansfield's bulletin board). An easy one that takes about five minutes to write is a utility to hide and unhide files. In an office environment this would be very handy. Just in case you forget the name of your hidden file, the REXX listfile utility can list out all files, hidden or not.

A slightly more complicated example is a program called TRAVERSE supplied on the Personal REXX distribution disk. Traverse is a program that looks like it came from the "Chuck Norris Utilities." It goes through every directory on a given disk and performs the same command. The obvious application is deleting those infernal .BAK files that so many programs seem to create. Just enter "TRAVERSE erase \*.bak" and every single .BAK file is erased.

At a slightly more advanced level, how about a program that automatically keeps a record of your computer usage (remember all those IRS regulations)? Let's call it REXXLOG. The first time you run REXXLOG it creates a file with the title shown below. Every time you log on, REXXLOG automatically saves a record of your logon time in a global variable on disk. When you finish, you type LOGOFF and the starting time, finishing time, and elapsed time is appended to a usage log file as a new record thus:

```
REXXLOG -- THE REXX COMPUTER USAGE LOG
Version 1.0 Copyright John A. Doe, 1987. All Rights Reserved.

Date      Time Logged on  Time Logged off  Total Time On
1 May 1987. On at 03:56:11. Off at 03:56:22. Time on: 00:00:11
2 May 1987. On at 04:35:13. Off at 06:31:04. Time on: 1:55:51
```

Of course, every now and again you get logged off involuntarily by power outages, etc. In this case, REXXLOG is smart enough to know that it should not update your usage log as you will be logging on

straight away. The only manual input is for the user to remember to type "logoff" when logging off as that updates the usage log. REXXLOG also uses the file to count your total time on during the month and displays it when you log on or log off. At the end of each month, REXXLOG changes the logfile name to archive it and starts a fresh file.

This is a medium-sized REXX application, yet it would take someone who had used REXX for one month about one evening to write. One reason for the rapid development time is the "trace" facility built into REXX and Personal REXX. It can be set to trace as little as nothing, or as much as every single intermediate calculation. It makes error tracking a cinch. Other examples of REXX applications would be applications menus (color full-screen and Lotus menus are easy to write using the built-in screen addressing functions), a file backup utility, and a utility to list two sorted directories on the screen side by side for comparison. By far the most ingenious application I have seen is a four-function calculator that uses a grand total of 8 lines of REXX source code!

#### Technical Support, Documentation, Upgrade Costs and the License Agreement

Mansfield's technical support is exemplary. They provide daytime phone support and a bulletin board. I left several messages on the board and was astonished at the regularly fast, helpful responses. One bug I found was with the LISTFILE utility (an enhanced equivalent to the DOS 'dir' command). Under PC-DOS 3.2 it would terminate with a "Stack overflow" message. I left a message on Mansfield's bulletin board one evening and the following morning received a call from the program's author. The problem was that the Microsoft C compiler (which was used to compile Personal REXX) sets a default stack size of 2k. This stack is used to pass a copy of the environment to the application. If the user's environment is large (and mine is over 900 bytes), the stack overflows. Mansfield sent me a new version of the program and also uploaded the correction to the bulletin board so that I could get it more quickly. Now that's impressive.

Interestingly enough, I had been having the same stack problems when I try to sort the directory in the communications program, Procomm. Just for grins, I reduced my environment drastically and tried the Procomm sort utility. Hey presto! It worked.

Mansfield's upgrade policy is very reasonable. Minor updates to Mansfield products are posted on the bulletin board, so that they are easily available. There has been one major upgrade to Personal REXX thus far, and that was distributed free to registered owners. The forthcoming version 1.6 will include a complete

new manual and cost \$15, although the retail price will go up \$10. There are also dozens of REXX files on the bulletin board contributed by the user community. All are free, save the cost of the phone call.

Personal REXX documentation is comprehensive. The manual is a 180 page letter-size binder. This takes you through installation, configuration, writing REXX programs, advanced usage, and special features such as the KEDIT interface. The text is very clear, the manual is indexed, and the examples are plentiful. All of the example programs are on the distribution diskette. There is a section on compatibility with the REXX language and the mainframe CMS version of REXX (most programs are portable subject to minor modifications), a list of error codes returned by REXX, and a language summary. My only complaint about the manual is that textual references are to chapters but the chapter number is not at the head or foot of each page.

There are two types of software license agreement in the world. One type is exemplified by WordPerfect and Borland International. Both firms license the user, not the machine. I.e. You can use more than one copy of the software, provided there is no chance that both copies are used simultaneously. This type of license agreement allows you to make a copy to use at work, as well as one to use at home. The other type of license agreement is exemplified by Ashton-Tate, which expressly states that it is a violation of the license agreement to use Framework at home, and at work (even if you manually transport the disks!). Mansfield's agreement is of the Borland/WordPerfect variety, so you are not forced to buy two copies of Personal REXX in order to have it available.

For the language itself, Mansfield has recognized that books on REXX are not as readily available as books on other high-level languages and thoughtfully included the "REXX Bible" --Cowlshaw's "The REXX Language". I would recommend that the beginner also get the official IBM user's guide to REXX (referenced at the end of this review) which is, contrary to form, an epitome of clarity.

### Summary

Before I was asked to review Personal REXX, I had used the language extensively in a mainframe environment. In the mainframe world, even people who are otherwise averse to programming become REXXaholics. Everyone who tries it seems to love it. The language is inherently intuitive, so programming quickly yields tangible results. I was always baffled that IBM had not chosen to distribute a REXX interpreter in lieu of the BASIC interpreter with PC-DOS. Fortunately, Mansfield Software has stepped into the breach with the "program that IBM should have written" and done a fine job.

### References

1. Cowlshaw, M.F., The REXX Language (Prentice-Hall:Englewood Cliffs) 1985.
2. Mansfield Software Group, Personal REXX User's Guide (Mansfield Software Group: Storrs, CT) Version 1.50, June 1986.
3. IBM Corp., System Product Interpreter User's Guide (IBM Corp.: NY) SC24-5238-1. Release 3, February 1984.
4. Abacus, A., "REXX: A beginner's alternative", Computer Language, v3 n6 (June 1986).

### Hardware Configuration Used in This Review

The machine used in this review was an AT clone running at 8 Mhz. with zero wait states and the Phoenix v3.01 BIOS. The machine had 3 megabytes of RAM and Personal REXX was run from VDISK in extended memory. All applications were loaded from a MFM encoded hard disk with a 28 ms access time. The operating system was PC-DOS versions 3.3 and 3.2.

### Product Details

**Name:** Personal REXX v1.5.

**From:** Mansfield Software Group P.O. Box 532 Storrs, CT 06268 (203) 429-8402

**Details:** One diskette (5.25 inch or 3.5 inch) Not copy protected.

**Requires:** IBM PC/XT/AT with 256k (384k recommended), DOS 2.1+, one floppy or 3.5 inch drive. Supports LIM EMS.

**Warranty:** Mansfield will replace if REXX is incompatible with your equipment

**License Agreement:** You may use REXX on more than one machine, so long as it is not being used on more than one machine simultaneously.

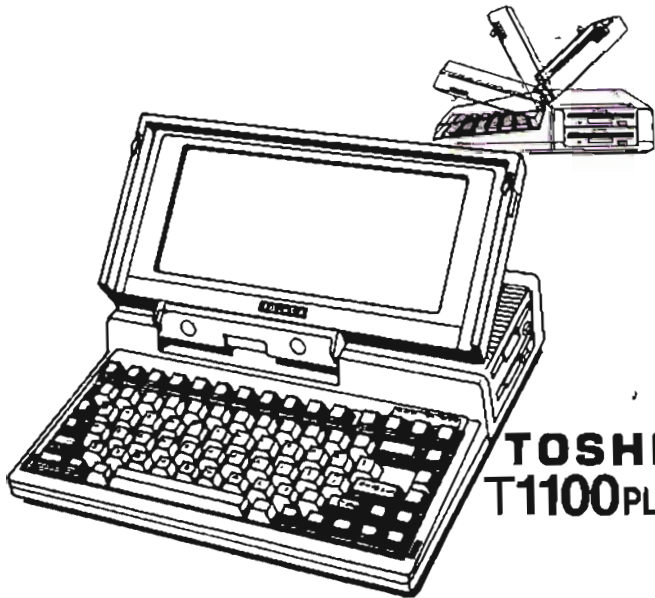
**Price:** \$125.00 + \$3 shipping.

I would like to acknowledge the comments of Warren Ferguson, Ph.D. and John Pribyl on this review and, in particular, their discovery of an error in one of the examples!

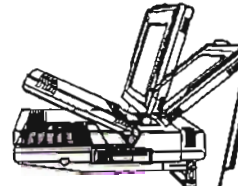
# DISCOVER THE DIFFERENCE

Discover the difference of dealing Direct. Direct Systems Corporation works directly with manufacturers to bring you top quality products at rock bottom prices.

Why pay inflated retail prices ever again? Direct Systems is now featuring the Toshiba laptop PC's. Another difference worth discovering.



**TOSHIBA  
T1100PLUS**



**TOSHIBA  
T3100**



- IBM® PC compatibility in less than a 10 pound package.
- 80C86 CPU, operates MS-DOS 3.2 software.
- Fully portable: runs on batteries and AC.
- Sharp, clear supertwist LCD.
- 640KB memory, 2 built-in 720 KB 3.5" disk drives.
- Optional 1200 bps Hayes® compatible modem

- Fully IBM compatibility in a 15 pound portable package.
- Power of a desktop AT - 80286 processor.
- 10MB hard disk built-in.
- One 3.5" diskette drive (720KB).
- High resolution gas plasma screen.
- Optional 1200 bps Hayes compatible modem.

**THIS IS A LIMITED OFFER! ORDER YOURS TODAY!  
CALL FOR USER GROUP PRICING.**



**DIRECT  
SYSTEMS CORPORATION**

10414 Miller Road (just off L.B.J. Freeway) Dallas, Texas 75238

IN DALLAS:  
214 343-0800

IN HOUSTON:  
713 932-9500



## Disk of the Month

Howard B. Hamilton, Jr., Ph.D.

The Disk of the Month for July 1987 is the latest version of GT PowerComm, one of the premier communications programs for the IBM PC and compatibles. This program and the introductory README file have been contributed by a new club member and recent California transplant, Ed Reardon.

We want to thank Ed and the other volunteers who reviewed software and wrote README files for May and June: Kathy Crawford, Ken Loafman, Dwight Neal, Howard McCalla (another new member of our user group), Tim O'Neil and James Turnock. We also want to thank the volunteers who have helped person(?) the DOM desk and pass out the thousand disks that you have purchased in each of the last five months.

### Review of May and June Disks

Two of the programs are from Jim Button, PC-File + (the DOM for May) and XD (Extended DOS). PC-File + (Disk 0158 A and B) is a complete rewrite of one of the most popular shareware programs ever, which was reviewed by Tim O'Neil in the May NT PC News. XD is discussed below.

Disk 0159 provides some aids for those who program in BASIC. One, **BASICAID** will expand/compress BASIC programs for readability/processing speed. **BRENTBAS** will translate a structured BASIC program into line numbered BASIC. **TinyBASIC** is based on the original version written for the Intel 8080, when space was at a premium.

**Utility and file management** disks include 0160: Vernon Buerg's Utilities (including the popular **LIST** and **SORTF**); 0161, **DOSamate**, a DOS shell and task-switching utility; 0162: **FileFriend** Evaluation Copy (memory resident file manager); 0164: **LQ** (stand alone or resident program for printing several fonts on dot matrix printers); 0172: **PC Magazine Utilities** (DOM for June); 0173: **Archival programs** from three sources, SEA (System Enhancement Associates), Vernon Buerg, and Phil Katz, plus two utilities that help index or catalog files/disks that include **.ARC** or **.LBR** files; and 0179: **XD**.

**XD (Extended DOS)** is an (optionally) menu driven tool with on-line help that can aid in the selection of groups of disk files for mass operations. Using wild cards (\* and ?) files are proposed and selected for copy, delete, move, etc. operations, with several opportunities for re/de-selection before commitment

to the operation; all the checking does slow things a bit. Operations include **KILLing** a directory by deleting all the files in it and then deleting the directory itself, with options to list the files first. The **LIST** function (optionally) heads each page of a file printout with filename, date of the file, and page number, and provides several other options, with a revisable default configuration file. **MOVE** operations include moves across drives (most move utilities revise pointers in File Allocation Tables and hence restrict moves within one drive). An **UPDATE** function considers all the files in a given directory selected (with or without wild cards) and looks for and replaces them with any more recent file found anywhere on the current path; i.e., a backup function oriented to the destination instead of the source directory.

**Disk 0163: Instant Recall** is a memory-resident (or stand alone) free-form database with its own word processor. This version is labeled **TRYWARE** and limits the text database to 80K and restricts the size of imported files. But they can be rejoined with the cut and paste functions and all other features of the registered version are present. Retrieval of data is immensely flexible: by words **AND** phrases, **WITH/WITHOUT \*/?** wildcards, **AND** creation, last modification **AND** reminder dates (however, keying in the date criterion is not convenient), **AND** logical **AND/OR** combinations of all of the above! I am impressed; you will be too.

**Professional aids.** Disk 0165: **ManageX III** provides timekeeping, billing, and bookkeeping software designed to help attorneys, consultants, and other professionals manage their businesses. A more recent version will be distributed as soon as someone volunteers to review it. Disk 0166: **Markis Presentation** is a demo disk of productivity aids for sales management. Sample files have been included and you can add, change, and delete up to 10 records of your own. 0177: **PC-GENERAL LEDGER** is a complete ledger and record system.

Disk 0167: **Outline!** is a RAM-resident or loadable outline and thought processor. It has up to 9 windows and is rather flexible in the display and movement of the outline levels. I was disappointed when I tried to use it to prepare viewgraph slides, as I could find no command for changing the line spacing and no way to print the outline title at the top of each slide (page).

Disk 0169: **TURBO EXPERT** is a demo of an expert system toolkit. While I have only looked at a few such tools, the system was menu driven and did not impress me as very flexible or professional. Disk 0170 A/B are **ARC'd** files of information from IBM on PS/2. ➤

**Programmer Tools.** For programmers (better look at the README files, yourself), 0168: **The Window Boss** provides C language support for creating tiled windows (whatever they are); 0171: **A86/D86** is an assembler and debugger and, according to the author, much has more readable tutorial/documentation than in earlier versions (the README file lists the new features); 0175 A/B: Kent Cedola's **EGA Graphics** provides graphics tools for Turbo Pascal and MicroSoft C programmers; 0176: **Icon Maker** (by Sid Nolte) is software to help programmers who want to include graphics in their programs in a simple manner and aid for the counted cross-stitch hobbyist.

Two application packages. 0178: **Transtok** translates stock information downloaded from the Source: StockCheck or Unistox databases; and 0180: **ZAPCIS** minimizes connect time on CompuServe and *MsgVu* manages those files downloaded from any source.

### Other Information

Quite a few people have volunteered to review some of the disks available. We will have to see which ones get done and have README files ready in time to be distributed at the July meeting. More volunteers are needed. If you would like to help review software or assist with the DOM table operations, send me a message on the club bulletin board. If you don't have a modem, get one (that's a joke, son); or you can call me at 214-644-5721, evenings until 10:30.

If you have a software to donate, please write down on the label or an enclosed piece of paper: your name, telephone, and the source of the programs, to the best of your knowledge. I have some neat looking software, labeled "Neat Programs for Disk of Month," that somebody turned in a couple months ago. We can't

use them because we don't have any idea where the programs came from. If you like the software you are donating, why don't you review it and write a README file; instructions are in Catalog II for June and I can send them to anybody on the club bulletin board. Software that already has a README file by somebody who has used it, can be distributed the next month, instead of having to wait until a volunteer selects it, tries it out and writes it up.

Howard

### DOM Particulars

The North Texas PC Users Group copies these programs as a service to the club and its members. We try to test all the programs, but we do not warrant the programs in any way. You must decide if a program is suitable for your system and use. If you ask, we will tell you what we know about any program, but the final decision to buy and/or use these programs is yours. We will gladly and without question exchange an unreadable diskette for one of the same program.

**EXCHANGE:** All members of the club are encouraged to contribute copies of public domain programs to the club library. For each new diskette of software contributed, you may select any diskette in the club library in exchange. The contributions will be reviewed before credit is issued at the next meeting.

**MAIL ORDERS:** At prevailing prices plus \$1.00 for mailer and postage. Mail your order to Tim O'Neil, Box 396, Bedford, TX 76021.

**PRICE:** Members: \$2.00 per diskette (if the program is on two diskettes the price is \$4.00) Non-members: \$3.00 each diskette

**CATALOG DISKETTES:** Currently this is a two volume set priced at \$4.00. This has all of the readme files from each diskette in the club library.

**MEDIA:** DSDD 5 1/4" Formatted as 9 sector data diskettes. Public domain software only, standard full disclaimers

**AVAILABILITY:** We will do our best to have all past diskettes at each meeting. DOM sales will begin at the DOM counter around 9:00, and continue until 2:00 PM

**IBM EXCHANGE NEWSLETTER:** The EXCHANGE for the current month will be available at the auditorium AFTER the main meeting, at no charge to paid up members of the NTPCUG.

---

## HERITAGE CONSULTING SERVICES

## FINANCIAL SERVICES

---

\* Over \$49,000 savings to client on two W/C Insurance audits \* Almost \$9,000 saved on payroll tax lien negotiations with IRS \* Inventory accounting system development that saved \$500,000's in losses \* Two terms as member of board of directors of NTM \* In charge of registration for Annual Gathering being held in July \* Member of the North Texas PC Users Group \* Member of the Dallas Chamber of Commerce \* Certified as a minority (woman owned) business by the City of Dallas \* Authorized by IRS as client representative \*

. 2536 Cattail . Carrollton, Texas 75006 . (214) 416-9000 .

---

## On Complexity

No. 11 in a Series

Jim Hoisington

What about the new IBM PS/2 family of microcomputers? Are they any good? Will they be accepted in the marketplace? Here is my evaluation of the new machines.

To assess the value of the new machines, we need to think about what we will expect of any new microcomputer. And, then to understand what we will want in the future, we need to look at what we have done in the past with microcomputers.

The three solutions to problems that the early microcomputers provided were word processing, spreadsheets and data bases.

Word processing solved the problem of how to correct and rearrange documents multiple times without having to manually retype the entire document with each revision. Word processing also gave us the ability to handle repetitive documents by using the mail merge and boiler plate features.

The second problem that early microcomputers solved was managing data. The early database programs had lots of limitations but they gave people a way to keep data in the computer and to access it in several different ways.

Finally, early microcomputers gave us the ability to redo large ledger sheets quickly when one of the underlying assumptions changed.

Since the days of the early microcomputers, we have moved through at least two generations of software. To understand where we are headed, we need to look at what happened to our original solutions to problems.

As the software changed, it generally did more for us than the previous release or program. Spreadsheets now include a limited database and limited graphing of data. Word processing has added multiple fonts, indexing and spellchecking. Databases now let you rearrange your data in any order, graph it, and export it to a spreadsheet or your favorite word processor. In fact, there are now "integrated" programs that allow you to do all three functions from within one program.

With each generation of software, we have asked our personal computers to solve more and more complex problems easily. It is my premise that we will ask future personal computers to solve increasingly more complex problems easily.

The easily part is important. If the computer can solve the problem but not do it easily, then it doesn't qualify as a personal computer.

From a software developer's standpoint, what is needed to solve an increasingly complex array of problems easily is something called "bandwidth." It has to be able to move more data faster.

The machine has to be able to move more data faster between the CPU and main memory because the CPU will be faster and the main memory larger. The computer also needs more "bandwidth" between main memory and the hard disk drive because it will also be larger. And, it has to move more data faster between main memory and the screen because the screen will present more information (pixels) at one time.

The new PS/2 machines from IBM have all the hardware requirements. The new VGA screens display more information. The processors or CPU's are faster. The micro-channel moves data around the system faster and allows for more than one processor to be running in the system. The keyboard has 12 function keys versus 10 keys on the original PC. Add on memory boards and larger disk drives have already been announced by the same vendors who make add-on products for our PCs.

Overall, the new computers have the "bandwidth" needed to solve a new level of problems easily. What they are lacking is the software that takes advantage of their capabilities.

Jim

---

## Selecting Your First Computer

Doug Russell

I'm new to both PCs and North Texas PC Users Group. I bought my PC only a few months ago and joined the group about the same time. I am completely satisfied with my PC and I really enjoy using it. I would like to share some ideas and procedures that made my first purchase of a PC a relatively painless experience.

I was attending school for a business degree, had a small business and needed a PC to write research papers, business correspondence and crunch numbers. I felt I was OK in the software department; I had a good word processing program and an interesting package that is a combination of Lotus 1-2-3 and dBase III.

Hardware was another story; I needed to find an adequate and inexpensive PC. I wanted an IBM compatible or clone. I felt a clone could do the same job as an IBM PC for less money.

My research started at the local library. Computer magazines provided a good source for reviews on



computer systems, floppy disk drives, monitors, keyboards, etc. Brian Starfire's article in each Monday's business section of the Dallas Morning News was helpful. I also picked the brains of the PC Users Group members and anyone else who looked like they had ever been near a computer. I asked them where they bought their PC and if they were satisfied. They were a great help and I want to thank them all. I also obtained past issues of North Texas PC News and read them. Through these and other sources, I got an idea of what I would need in my PC and why. Here are a few of the things I found along the way:

1. **Basic System.** A basic system is normally advertised as having 256K internal memory, keyboard, a 360K floppy disk, but no monitor. They may also include a warranty. Depending on what they include, the prices range from \$350 to \$650.

2. **RAM.** A lot of sophisticated software requires nearly 256K (or more) of Random Access Memory (RAM). Add memory used by DOS and you might not have much left over for anything else. I consider 640K RAM a minimum. An upgrade from 256K to 640K costs in the neighborhood of \$40 to \$60.

3. **Disk Drives.** Stores normally include one disk drive in their basic system. With one disk drive I felt I would spend most of my time changing disks, so I bought two. You should get double-sided, double density (DSDD) disk drives, which store approximately 360K bytes of data. A second disk drive costs \$80 to \$200.

4. **Parallel and Serial Ports.** The parallel port allows attachment of a printer. The serial port provides for a modem to send or receive data over the telephone. Both cost about \$50.

5. **Monitor.** The monitor is not normally included in the price of a basic system. A monochrome monitor costs \$70 to \$150. A color monitor costs \$200+ more than a monochrome and displays the same information. My choice was the monochrome monitor with green screen.

6. **DOS.** You cannot operate the PC without a Disk Operating System (DOS). DOS sets up your PC so you can use it. It acts as a translator between you and the computer. In some cases it comes with your basic system. If it comes extra, plan to pay between \$100 and \$150. As with any software, get the latest version. I bought MS-DOS Version 3.2.

7. **Keyboard.** It generally comes with the basic system but if you want one that is more versatile they cost around \$100 to \$150. I'm happy with the one I got with my PC.

8. **Power Supply.** The power supply unit should be rated at least 130 watts. The larger the power supply, the less possibility of it konking out if you add a hard

disk, internal modem, or other add-in cards at a later date.

9. **Warranty.** The warranty should be no less than one year on parts and labor. The place where you buy your PC should also have a service department where they service what they sell. If you have a problem within the warranty period, you can take it in and they will fix it free.

Something else that's very important... I mentioned earlier that I had some software. To be sure that my software would work on the PC I purchased I took the software with me to try on each system that looked interesting. Most salesmen don't mind you doing this. Most claim that the PCs they sell are compatible with software that can be used on the IBM PC. It is a strong selling point. If my software worked, I included the system on my list, otherwise I left it off. By the way, if the store won't let you check out your software on their machine, go somewhere else. You'd hate to buy it, get it home and find out your software wouldn't run.

Once I knew what I wanted on my PC, I had to find out where to get it. The local newspapers carry computer advertisements in the business section with most ads appearing in the Saturday and Sunday papers. Also, there are other periodicals you can pick up at local computer stores. It was initially confusing because the companies never advertised so I could compare prices! One company advertised a price for a basic system with 640K internal memory, without a monitor or ports, while another priced their basic system as having 256K, a monitor and parallel port.

Armed with my list and software, I visited the stores I had selected and asked them questions. How much for the basic system? How much for each of the components I wanted to add? How much to assemble and test the system? Did my software work? I wrote it all down. When I got back home I had a LOT of data that had to be organized.

To help solve that problem, I got a ledger sheet, also called an analysis pad. In column 1, I listed what I wanted my PC to have: monochrome monitor, two disk drives, etc. To the right of column 1, I headed up other columns with names of the stores I had visited. I then filled in the matrix with component prices for comparison purposes, added sales tax and totaled each column. When I got through, it looked something like the table shown on the next page.

You may want to take a ledger sheet with your list and fill it in as you go. You might get a better price if the store knows you are going to check other places.

I purchased everything on my list from the same store for less than \$900 including assembly, testing and taxes. I had visited them a couple of times and was impressed with the crowd of customers and the long line at the checkout counter. I figured they must be doing some-

LEDGER SHEET

1	2	3	4
Company	Fred's Comp	Joe's Comp.	Carla's Comp.
Basic System Name	Smart	SST	PC
Basic System	\$ 550.00	\$ 460.00	\$ 600.00
640K	50.00	80.00	75.00
2nd Disk Drive	100.00	95.00	85.00
Parallel & Serial Port	50.00	75.00	inc.
Monochrome Monitor	inc.	120.00	80.00
DOS	inc.	80.00	inc.
Keyboard	inc.	inc.	inc.
Warranty (1 year)	inc.	inc.	inc.
Assembly & Testing	50.00	inc.	inc.
Power Sup (130+ watts)	yes.	yes.	yes.
Software work?	yes.	yes.	yes.
	-----	-----	-----
Subtotal	\$ 800.00	\$ 910.00	\$ 840.00
Tax (6.75%)	54.00	61.42	56.70
	-----	-----	-----
Total System	\$ 854.00	\$ 971.42	\$ 896.00

(bing right! The salespeople were willing to spend time with me even if I was just looking.

While their system price was not the lowest, it was low enough and I liked the way they appeared to do business. They also threw in a box of floppy disks and a surge protector. (I think you should also add a surge protector to your list. It may protect your new PC from harm due to electrical fluctuations that sometimes occur during thunderstorms.)

There is a lot of used equipment out there. You may be able to purchase your system in this manner and save 20% to 30% of the retail price. However, I feel that this is sort of like buying a used car. You have to

really know your stuff about what you're buying, or have a friendly mechanic to help you out.

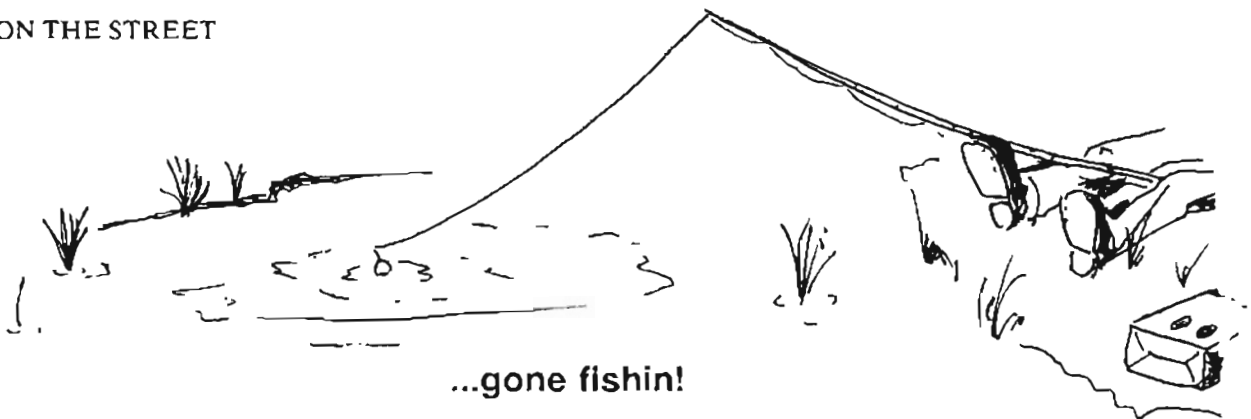
The same goes for software. If you can check out the software on a PC before you buy, great. If you can't, ask other people who have used it what they think.

In the short time since I bought my PC, basic system prices have gone down about 10% and hard disk prices about 20%. Because of the tremendous competition out there for our computer dollars, there will always be better prices for better systems. However, you can wait forever for lower prices and in the process lose out on the advantages and enjoyment of having a PC now. Good luck!

Doug



NERD ON THE STREET



...gone fishin!

## Frustration with COMMAND.COM

Dan Rau  
Silicon Valley Computer Society

Have you ever tried to exit an application and gotten the DOS message "Put disk with COMMAND.COM in drive and hit any key to continue?" If you think that's annoying, imagine how it felt to see that message when I was using an electronic RAM disk. There is no way to put a disk with COMMAND.COM in a RAM disk if it's not already there (Catch-22). If you try to do anything else it just keeps repeating the message. All I could do was reboot the system and lose whatever was in RAM. That's frustration!

After experiencing that frustration on more than one occasion, I got motivated to find a way to avoid it. Another motivating factor was generated due to use of IBM's PE2, which is an upgrade to their superb Personal Editor. PE2 has an attractive feature which allows access to DOS from within the editor. All DOS commands are available while in this state. When exiting DOS, you're right back in the editor exactly where you were before entering DOS. However, it looks for COMMAND.COM in Drive A: when making the transition to DOS and again when quitting the editor. This I couldn't tolerate, because I don't always have a COMMAND.COM disk in Drive A: and I didn't want to give up that neat PE2 feature. I don't have a hard disk. I have a dual floppy IBM PC.

I added a command to my AUTOEXEC.BAT file to copy COMMAND.COM to my RAM disk during bootup so that the RAM disk could always contain COMMAND.COM. Then I set the COMSPEC environment parameter so that DOS would always look for COMMAND.COM in my RAM disk. This worked fine for entering DOS from within the editor, but it still looked for COMMAND.COM in Drive A: when I quit the editor. This turns out to be a peculiarity of DOS 2.0 which I was using, but there is a workaround. The command COMMAND/P causes COMMAND.COM to be permanently resident and only costs about 3K of memory.

The following sequence at the end of an AUTOEXEC.BAT file solves the problem nicely.

```
COPY COMMAND.COM C:
SET COMSPEC=C:\COMMAND.COM
COMMAND/P
```

C: is the RAM drive. The batch file aborts after COMMAND/P so it has to be the last command. You can check your current environment status by entering the SET command by itself. Look up SET or

ENVIRONMENT in the DOS reference manual. If you want, experiment with a small batch file before committing to the AUTOEXEC.BAT file.

If you use a RAM drive and you've been bugged by COMMAND.COM as I have, try it. It's a pleasure to no longer be intimidated by COMMAND.COM "where are you" messages. Why it was designed that way in the first place, I don't know. A



## Advertising Prices & Policies

North Texas PC NEWS

### Ad Prices:

Full page (7 x 9) \$120  
Half page (7 x 4 1/2) \$ 80  
Cir page (3 1/4 x 4 1/4) \$ 50  
Business Card (2 x 3 1/4) \$ 30

### Shop & Swap Prices:

\$ 0.15 per word  
(80 words maximum per ad)

### Deadlines:

All advertising copy must be prepaid and received by the NEWS staff no later than the 15th of the month.

### Policies:

- . Shop & Swap ads must be typed, or neatly hand lettered.
- . Commercial ads must be in repro form, final size, ready for printing
- . No ads will be run on cover.
- . No bleed pages. No color.

### Payment:

Payment must accompany ad copy.  
Make checks payable to North Texas PC News.

### Address:

Mail all advertising copy to:  
Editor, North Texas PC News  
2025 Rockcreek Drive  
Arlington, TX 76010

### For further information ...

Call J. P. Pribyl at 817-275-4109.

MEMBERSHIP APPLICATION

North Texas PC Users Group

The NTPCUG is a non-profit, independent organization of individuals learning to apply personal computers to practical problems. For additional information about the Group, call (214) 746-3297.



Application Status: (Check One)
>>>> \_\_\_\_\_ NEW MEMBER
>>>> \_\_\_\_\_ RENEWAL
>>>> \_\_\_\_\_ CHANGE OF ADDRESS

(Please Print Clearly or Type)

NAME: (Last) \_\_\_\_\_ (First) \_\_\_\_\_ (MI) \_\_\_\_\_

OR Company/Organization: \_\_\_\_\_

ADDRESS: \_\_\_\_\_ (Suite/Apt) \_\_\_\_\_

CITY: \_\_\_\_\_ STATE: \_\_\_\_\_ ZIP: \_\_\_\_\_

PHONE: Home (\_\_\_\_) \_\_\_\_\_ ; Work (\_\_\_\_) \_\_\_\_\_ (Ext) \_\_\_\_\_ (Check Preferred #.)

Do you want access to the Club RBBS \_\_\_\_\_?

Please initial here \_\_\_\_\_ if you wish to have your address included on member lists sold for the club's benefit to advertisers of IBM compatible products.

The NTPCUG expects and encourages volunteer participation by members in helping put on the monthly meetings at INFOMART. This usually consists of a few hours of your time each year. If asked, would you consider assisting the Group with one or more of the following activities:

[A] Working with NTPCUG Volunteer committees?. (See below, and description of activities on reverse.)

Volunteer Areas from [A] above (Please check all that apply.)

- [IB] Information/Registration [MM] Membership [GP] Group Purchase
[NL] Newsletter [ES] Equipment Setup [FB] Financial/Bookkeeping
[DM] Disk of the Month (DOM) [PR] Publicity/Public Relations [ST] Startext NTPCUG Column

- [B] Giving a talk or demonstration to a small group?.
[C] Giving a talk or demonstration to a large group?.
[D] Being a volunteer, informal "consultant" in your area of expertise for NTPCUG members?

Would you be interested if the Group arranges instructional courses (at various levels) in any of the following areas at a cost per student of approximately \$5/classroom hour?

(Please circle or specify, indicating level preferred, i.e., beginning, intermediate, advanced, etc.)

- [A] Spreadsheet software -- Lotus 1-2-3, SuperCalc4 etc., (Please specify) \_\_\_\_\_
[B] Data Base software -- dBase, RBase, Reflex, etc., (Please specify) \_\_\_\_\_
[C] Word processing software -- Word Perfect, Wordstar, etc., (Please specify) \_\_\_\_\_
[D] Integrated software -- Framework, Symphony, etc., (Please specify) \_\_\_\_\_
[E] Programming Languages -- APL, Assembly, Basic, "C", Fortran, Forth, Pascal, (other) \_\_\_\_\_

Do not write in this area -- for use by NTPCUG

Annual Dues are: \$24.00 (Regular Membership) \_\_\_\_\_ \$12.00 (Student Membership with ID) \_\_\_\_\_

Applications should be mailed to: North Texas PC Users Group, Inc.
(Make checks payable to NTPCUG.) P.O. Box 780066
Dallas, TX 75378-0066

Received: \$ \_\_\_\_\_ Check No. \_\_\_\_\_ Date: \_\_\_/\_\_\_/\_\_\_ BY \_\_\_\_\_

## Marketing Computer Services

John Keohane

A recent poll of members of the North Texas PC Users Group indicated that 24% of us are consultants. Perhaps there is an additional number that do consulting on the side, or would like to be consultants/contractors. If the people I know in the NTPCUG are typical, the consultants in our midst are very strong technically. If I am at all typical, we are not near as good as we might be at marketing.

Here are a few thoughts about marketing the consultant. These views are my own as an independent consultant. Operating as an independent is, I think, a strong point, for I believe there is really no such thing as a consulting or programming firm, only individuals who are good at consulting or programming. At any point in a project, a good consultant knows all about the project, and his or her client benefits by getting expert help in all phases.

Some basics of marketing cross over various industries. I've found some books very helpful, including Miller's book "Strategic Selling" and Wolfe's "Sell Like an Ace, Live like a King!", and Girard's "How to Sell Anything to Anybody". OK, some parts of them are bokey, and they are not written by PhDs, or University of Chicago MBA's, and perhaps some of the things are elementary, but not elementary things that have always come into my consciousness. I understand that Zig Ziglar is a marvelous marketer, and bought his book, but found it too disjointed to benefit me.

When I first started consulting, I thought clients would knock on my door and ring my telephone, because I do good work, and stand behind it. Well, I have learned that simply is not so. As an independent consultant, you have to market your work. That includes meeting people, phoning people, and telling people what you do.

When I started marketing, I thought I was pretty knowledgeable. After all, I'd been successful as advertising manager for my college yearbook, and as a participant in phonothons for my university, and my MBA degree includes marketing as a major. However, as a marketer, I was missing A LOT.

I also thought I might market without ever contacting individual people. I thought teaching computer

courses, leading the BASIC programming group for NTPCUG, writing a weekly computer column for a local newspaper, and writing articles for this newsletter would bring business by osmosis. It should operate similarly to the junior high school chemistry experiment, where, combining highly salted and unsalted water, simply by pouring one in the container of the other, results in water with equal saltiness without even stirring one into the other.

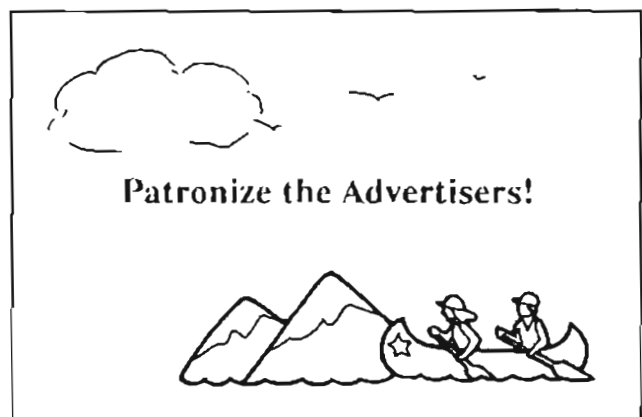
Not so, Charlie! None of that has worked.

What has worked is contacting people. Some months ago, I asked a friend in the banking business how he finds new loan customers for his bank. He told me that he contacts people he knows. Unwisely, I wrote him off as being passive. "Passive?" I would now say, "Smart". You have to start with your friends and acquaintances. You have to let them know what you do well. It's like infield drilling in the oil patch. You're much more likely to strike pay dirt if you do drilling near where oil has already been discovered rather than wildcat out in the open country, far from where any oil has been discovered.

You also have to specialize. A multinational firm may provide all kinds of service in all kinds of computer consulting, and they may have enough experts in enough areas to pull this off, but that's like multiple orchestras. As an independent consultant, you're playing solo, and cannot expect to perform well in the multitude of things that a Big Eight accounting firm might do. However your costs can be lower, you can be much more accessible to your clients, and, if you specialize, you can provide as good or better technical expertise and service.

John

■



## Selected SIG Happenings

### General Special Interest Group (SIG) Information

A reminder that any SIG news items for this newsletter must be received by noon of the 14th of the month -- regardless of the date of the next meeting.

### BEGINNERS SIG

July will be the first part of a new 3-part Beginners series. These sessions are for the new, or relatively new, PC user, and cover the fundamentals of hardware, DOS, and "how to get started".

Gordon Baskett

### DOS SIG

"Back to the basics" might have been the theme for the June DOS

SIG meeting. Jim Hoisington, NTPCUG President, and Reagan Andrews went back over DOS's DIR command, pausing to discuss volume labels, erroneous size estimates as a result of DIR reporting bytes used in creating files rather than the clusters occupied by files and other topics.

This theme will be continued in the July DOS SIG meeting with emphasis on commonly-used DOS internal and external commands, and the surprises not documented by either IBM or Microsoft awaiting the unwary DOS user. Also in July, attempt will be made to compare the advantages of "moving up" to later DOS versions, i.e., from 2.1 to 3.3, while recognizing disadvantages involved.

Reagan Andrews

### LOTUS SIG

The topic for the June meeting was Lotus Graphics and Freelance. The presentation was informative and many participants had some useful comments.

The subject for the July meeting will be 1-way and 2-way data tables ("what-if" tables in Symphony). We also plan on discussing the results of the June survey to help plan our future meetings.

We generally have time to answer some user's questions about Lotus 1-2-3 and Symphony. If you have questions, or are just learning, come by and join us.

Mark Gruner

## SWAP SHOP

*Four lines free each month to members; 5th through 10th lines at 15 cents per word. Larger ads at commercial space rates. Send check to the Editor for words exceeding the four-line limit. Free ads are on a space-available basis. Mail ads to the Editor.*

Sr. Accounting Position Wanted. Member with 16 yrs. exp. in Analysis, Budgeting and Reporting; PC exp using LOTUS, and FRAMEWORK. James Turnock - Use BBS or call (214)867-0655

Going through a computer conversion? Facing hard decisions on which package is for you? Get fast relief, call Bruce Schubert, CPA 991-5967

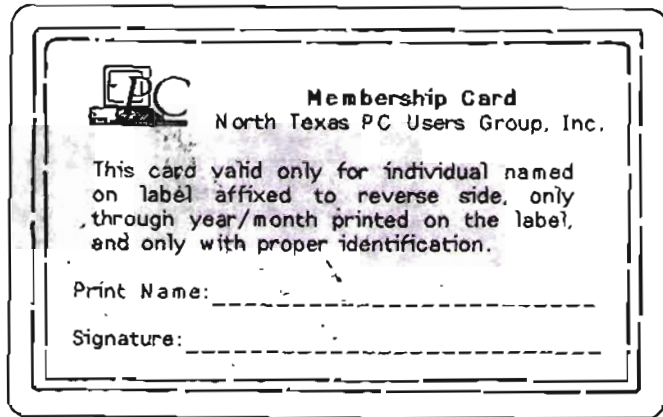
WANTED: Advertisers for North Texas PC News. Select audience of computer-enhanced people who are constantly in need of more hardware, software and peripherals for IBM and compatible computers. Only \$120 for a full page ad. See price list in this issue for other sizes. Contact the Editor, or send your ad with payment, to North Texas PC NEWS, 2025 Rockcreek Dr., Arlington, TX 76010 by the fifteenth of the month.



### MEMBERSHIP CARD

This is your membership card in North Texas PC Users Group. You will need it for identification at Disk of the Month sales, group purchases and other activities. This card is valid only for you, the person named on label on reverse side. It is valid through expiration date shown on the label.

When trimmed, the card will fit the holders previously furnished for Infomart cards which are no longer required. Wear your membership card instead. Additional holders will be available at a nominal charge.



Trim card to wallet size.

## Room Assignments



Saturday, 11 July 1987

9:00 AM to 9:45 AM  
AUDITORIUM

**\* Nantucket's Clipper \***

Nantucket Software's Vice President, Burt Durant, will be presenting Clipper. Mr. Durant will show us the advantages of using their dBase compiler, Clipper, over buying a dBase runtime.

10:00 AM to 11:00 AM  
AUDITORIUM

**\* New Views Accounting \***

Toronto based O.W. Page Associates will be sending their Vice President of Sales and Marketing, Ron Benn, to demonstrate New Views Accounting Software. A nested-spreadsheet format is used providing a dynamic interaction between all parts of your company's books via pull down menus.

*Scheduled SIG times could change. Check the Bulletin Board just before the meeting. Check room numbers in lobby at INFOMART.*

*Note: No APL meeting in July.*

<u>8:30 - 9:55</u>	<u>9:00 - 10:55</u>	<u>1:00 - 1:55</u>
Beginners	Genealogy (w/Apple)	Artificial Intelligence
<u>9:00 - 9:55</u>	<u>9:30 - 9:55</u>	Business Applications
Assembler	Orientation	Communications
Astrometry	<u>11:30 - 11:55</u>	Data Bases
DOS	Orientation	<u>2:00 - 2:55</u>
ENABLE	<u>12:00 - 12:55</u>	BASIC Applications
Graphics	C Language	Advanced Programmers
Hardware Solutions	LOTUS	dBase Programmers
Science/Engineering	Personal Finan. Planning	
	Turbo Pascal	



North Texas PC NEWS

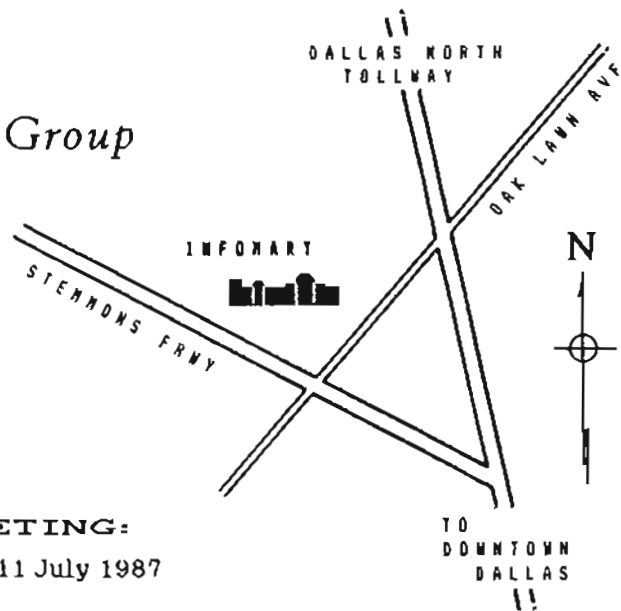
2025 Rockcreek Drive, Arlington, Texas 76010

BULK RATE  
U. S. Postage  
Paid  
Arlington, TX  
Permit No. 823

Address Correction Requested

1088 Expires 8804  
C. Frank Seay, Jr.  
6939 Joyce Way  
Dallas TX 75225

North Texas PC Users Group



NEXT MEETING:

11 July 1987