

# USUS NewsLetter

Serving the Pascal, Modula-2, and Portable Programming Community

Vol. 4 No. 4 Jul / Aug 1990

## IN THIS ISSUE

SRC Modula-3	1
Modula-3 Questionnaire	3
Letter About MODUS	6
Modules & Definitions: A New Publication	7
Review of File Archive Formats	9
From The USUS Librarian	14
Zoo Compress Program	15
Zoo Uncompress Program	17
Meeting Minutes (May 1990)	19
Meeting Minutes (June 1990)	20
Treasurer's Report	22
From The Editor	23
Submission Guidelines	23

## SRC Modula-3

A new release, version 1.5, of the SRC Modula-3 compiler and runtime are available now. This is the third public release of SRC Modula-3. The system was developed at the DEC Systems Research Center. It is being distributed in source form (mostly Modula-3) and is available for public ftp. You must have a C compiler to build and install the system.

The primary changes since version 1.4 are:

- many bugs are fixed
- the libraries have been reorganized and extended
- demos and games have been added
- the system was ported to:
  - Apollo DN4500 running Domain/OS,
  - IBM PC running AIX/PS2,
  - IBM RT running IBM/4.3,
  - IBM R6000 running AIX 3.1,
  - HP 9000/300 running HP-UX 7.0
- in addition to the previous ports:
  - VAX running Ultrix 3.1
  - DECstation 3100 and 5100 running Ultrix 3.1
  - SPARCstation running SunOS 4.0.3
- the installation on multiple platforms is easier
- ports are easier

SRC Modula-3 is available without signing any license agreements. If you choose to sign the commercial license, you will be able to use SRC Modula-3 commercially.

Modula-3 is a new language. The goals of its design are best encapsulated in the preface to the Modula-3 Report [1]:

The goal of Modula-3 is to be as simple and safe as it can be while meeting the needs of modern systems programmers. Instead of exploring new features, we studied the features of the Modula family of languages that have proven themselves in practice and tried to simplify them into a harmonious language. We found that most of the successful features were aimed at one of two main goals: greater robustness, and a simpler, more systematic type system.

Modula-3 descends from Mesa, Modula-2, Cedar, and Modula-2+. It also resembles its cousins Object Pascal, Oberon, and Euclid. Modula-3 retains one of Modula-2's most successful features, the pro-

Copyright 1990, USUS INC, All Rights Reserved.

The USUS NewsLetter is published 6 times per year by USUS, the UCSD Pascal System User's Society, P.O. Box 1148 La Jolla, California 92038. The NewsLetter is a direct benefit of membership in USUS.

Tom Catrall Editor  
Robert Geeslin Publisher

vision for explicit interfaces between modules. It adds objects and classes, exception handling, garbage collection, lightweight processes (or threads), and the isolation of unsafe features.

SRC Modula-3 includes a user manual, compiler, runtime library, core library, pretty-printer, and a few other goodies. The libraries include interfaces for X11R4, I/O streams, string functions, access to command line arguments, random numbers, and operating system access.

The compiler generates C as an intermediate language and should be fairly easy to port. Except for the garbage collector and the very lowest levels of the thread implementation, the entire system is written in Modula-3.

The system is available for anonymous ftp from 'gatekeeper.dec.com' [16.1.0.2]. The SRC Modula-3 files are in '/pub/DEC/Modula-3'.

Those files include:

m3-1.5.tar.Z                    the system  
m3-1.5.tar.Z-{01,...,12}    same, in pieces

Report.ps  
Report{1,2,3}.ps  
Release-1.5.ps  
  
m3-mail.<month>.Z

the revised language  
report  
same, in pieces  
the user manual  
(PostScript)  
archive of mail sent to  
m3@src.dec.com

The compressed tar files are about 6.0Mbytes after compression. The entire system requires about 35Mbytes of disk space to build and install.

We are maintaining a mailing list of those interesting in SRC Modula-3. The list is 'm3@src.dec.com'. To be added to the list send a message to 'm3-request@src.dec.com'. We may also be reached at:

Systems Research Center  
130 Lytton Avenue  
Palo Alto, CA 94301

Enjoy,

Bill Kalsow and Eric Muller

---

#### References

[1] The Modula-3 Report (Revised),  
L. Cardelli, J. Dohnaue, L. Glassman, M. Jordan, B. Kalsow, G. Nelson,  
DEC Systems Research Center, Palo Alto, CA and Olivetti Research Center, Menlo Park, CA,  
Nov 89.

VAX, DECstation and ULTRIX are registered trademarks of Digital Equipment Corporation.

Unix is a registered trademark of AT&T Corporation.

SPARC and SunOS are trademarks of Sun Microsystems.

Apollo and Domain/OS are trademarks of Apollo.

IBM and AIX are registered trademarks of International Business Machines Corporation.

RT and PS/2 are trademarks of International Business Machines Corporation.

HP, HP9000 and HP9000/300 are trademarks of Hewlett-Packard Company.

HP-UX is Hewlett-Packard's implementation of the Unix operating system.

PostScript is a registered trademark of Adobe Systems Incorporated.

*The above message was posted to the Modula-3 mailing list. Anyone interested in learning more about Modula-3 may write to the Systems Research Center at the address given above and ask for the Modula-3 report. I have a copy of the version 1.5 Modula-3 compiler but it will take more effort than I am able to expend to get it running on my system. The Modula-3 language is quite well done in my opinion and I hope it gains popularity. If anyone wants to get a copy of the Modula-3 compiler but has no access to the sources listed above, you can contact me. Note that unless you have a BSD derived UNIX and a 32 bit processor system with several megabytes of memory, porting the compiler will be quite a bit of work.*  
Tom Catrall

# MODULA-3 QUESTIONNAIRE

by Sam Harbison

Well, folks, the next year promises to be a big one for Modula-3. BYTE magazine will carry an article on Modula-3 in their November issue, and other articles, a couple of books, and possibly some new implementations are in the works.

My company, Pine Creek Software, is trying to help make Modula-3 a success. We're working with the authors and DEC to publicize the language and to develop services and products for Modula-3 programmers and students. You can help us set priorities for the coming year by telling us how you are using Modula-3 and what we or others could do to support you. (And what you'd like to do to help.)

Please take a few minutes to edit your answers into the following questionnaire and return it to [harbison@tartan.com](mailto:harbison@tartan.com). Don't reply directly to the Modula-3 BBoard, since (I hope) there will be many messages. I'll post a summary of the responses. Don't be limited by the questions; if you want to submit an essay, go ahead.

If it's more convenient, mail your response to

Pine Creek Software,  
Suite 300, 305 S. Craig St.,  
Pittsburgh, PA 15213.

If you have questions, please call me at 412-681-9811.

Thanks,  
Sam Harbison

---

To: [harbison@tartan.com](mailto:harbison@tartan.com)  
Subject: Modula-3 questionnaire response

Questionnaire responses will be summarized and the summary will be posted or released to anyone who asks for it. Your particular responses will not be released unless you give permission. Is it all right to release your response? (yes/no)

->

## Part 1: The Modula-3 Language.

(In this section, please disregard current implementations.)

Generally speaking, how well do you like the Modula-3 language? (e.g., not at all, a little, a lot, tremendously)

->

Which features or capabilities of Modula-3 are particularly good?

->

What features (present or missing) significantly hinder your use of Modula-3? (e.g., multiple inheritance, overloading, generics, case sensitivity, garbage collection, threads)

->

What features or changes are on your "wish list" for Modula-3?

->

Modula-3 is currently described in a research report from SRC and Olivetti. Is this report sufficient to support your programming? Would you like a more detailed reference? Would you like a Modula-3 programming textbook?

->

In your opinion, what other languages compete most directly with Modula-3? How does Modula-3 stack up against them?

->

Are these implementations satisfactory for your current interest in Modula-3? What do they need most?

->

Do you think Modula-3 has the potential to be a major programming language of the 1990's? (If not, why not?)

->

Would these implementations support the use of Modula-3 as your principal programming language? What would they need most?

->

## Part 2: Implementations.

What computer(s) do you run Modula-3 on now?

->

Are the libraries (I/O, OS, strings, etc.) provided with your Modula-3 implementations adequate? How would you like them enhanced or changed?

->

What other computers would have to support Modula-3 for you to use Modula-3 seriously? (PC/DOS? Mac? VAX/VMS? Crays? Other UNIX machines?)

->

Is it important to have more general-purpose libraries, such as support for particular data structures? Which ones do you want?

->

If you've programmed in Modula-3, how much code have you written?

->

Is the documentation with the implementations adequate? What improvements would you like to see?

->

Do you currently use the SRC implementation, the Olivetti implementation, or some other? (If you use more than one, indicate the approximate percentages and which one you like best.)

->

Which of the following Modula-3 implementations would you use? (You can choose more than one.)

- a. An interpreter with excellent diagnostic, interactive, and debugging capabilities.
- b. A fast compiler that generates "OK" code, with a debugger.
- c. A slower compiler that generates highly optimized code, with a debugger.

d. A fully integrated and graphically-oriented system with editor, compiler, linker, debugger, etc.

->

Which currently-available compilers or programming environments (for any language) do you like? Which one(s) do you use?

->

### **Part 3: You, again.**

If you are using Modula-3 for a specific project, or are working on an implementation, please tell us something about it.

->

Are you planning to produce any Modula-3 software that could be used by other Modula-3 programmers?

->

Would you be interested in writing an article on Modula-3 for the trade press or for a refereed journal or conference? If so, what subject areas interest you?

->

Would you be willing to teach a conference tutorial on Mod-

ula-3, or organize a conference discussion?

->

Are you planning to use Modula-3 in an academic course? If so: which course, where, and is there anything you need for it?

->

Would you be interested in a "Modula-3 user's group," or in receiving a regular newsletter about Modula-3 developments? If so, please include a postal address below.

->

(Postal address)

What is the single most important thing we could do to support your use of Modula-3?

->

If your job was to make Modula-3 a success, what would you do first?

->

That's it. Thanks for your time.

**MODUS**  
P.O. Box 51778  
Palo Alto, California  
USA 94303-0721

10 August 1990

Dear MODUS Readers,

This letter is to provide you with current information about the Modula-2 Users' Association (MODUS).

My name is Stan Osborne. I am the new "American Administrator" for the Association. Like my predecessors I volunteered for this responsibility. It is my hope that we will be able to publish an issue of MODUS Quarterly every three months.

MODUS was formed to provide a forum for communication between all parties interested in the Modula-2 language. The primary function of MODUS is to publish the MODUS Quarterly. If my memory is correct, MODUS has also sponsored two conferences.

As many of you are already aware, MODUS has not fulfilled its primary function during the last two years. This is no longer the case. Issue #10 of the MODUS Quarterly is ready and being printed. Work has started on getting material for Issue #11. All past and present members are being notified by mail.

Within the next two years the international effort to produce the first Modula-2 standard will be completed. The MODUS Quarterly is one way for you to learn more about the standard and how it affects the future of Modula-2.

You can help with reactivating the Association by:

- submitting a membership application
- sending in letters and articles for publication
- telling other Modula-2 users about MODUS

My sincere thanks to you in advance for your help with the above. If you have questions, comments, criticisms, or complaints about the operation of the Association, please don't hesitate to tell me about them. I am especially interested in suggestions you might have about how any aspect of MODUS can be improved.

Sincerely,



Stan Osborne  
American Administrator

P.S. This letter was included in issue #10.

# Modules & Definitions: A New Publication

The following is excerpted from the inaugural issue of *Modules & Definitions*, a new shareware publication covering Modula-2. Included are the editor's introduction, the table of contents, and information on subscribing. It is reprinted with permission.

## FIO.Open()

Welcome to the first issue of *Modules & Definitions*. This is a magazine designed and dedicated to users of Modula-2.

What you will find here is information about Modula-2 compilers, the Modula-2 language, reviews, programming techniques and information about interfacing with other languages. Eventually, we hope you will find ads for Modula-2 compilers, toolkits and thirdparty add-ons. What you WON'T find here is articles strictly about C, Pascal, Ada or Basic. You won't find reviews about the newest C++ compiler. You won't find ads for the latest Pascal Graphics toolkit. In short, this is a magazine for Modula-2 users, not the rest of the world. Yes, this limits our market a bit, but there are enough of the other guys around. What IS needed is something for us... the Modula-2 community.

In this issue, we start some of the regular features we hope you will come to depend on. Peter Baenziger begins his column on interfacing Modula-2 and Assembly language by explaining why you should bother with low-level programming. David Albert gives us a schematic for a \$10 Analog to Digital converter, shows how to talk to it in Modula-2 and hints about things to come. Jim Singleton gives the first part of his beginners column. John Ribar opens his column on Object Oriented Programming under Modula-2 and Michael Hunt has musings about numbers, rational and otherwise. Greg Walters will be kicking off our SNEAKY TRICKS column with a module to control the DOS Print.Com utility (SNEAKY TRICKS is a monthly feature designed to offer you new and different ways of doing things under Modula-2). We will also be reviewing JPI's new version of their Modula-2 compiler.

We also start our US Modula-2 Watch column. This is an ongoing compilation of books and articles about Modula-2. The name is a bit deceiving at first glance. It really is a jab at the US programming community for the lack of acceptance of Modula-2 as a serious language. Europe reportedly has almost 95% of the colleges teaching Modula-2 as their language of choice. It's time that the US market wakes up to the fact that Modula-2 is just as capable as C and much better when it comes to maintaining software.

What does the future hold for *Modules & Definitions*? Well, in the next few months, we will bring you the latest information on the standards committee, using the BGI (and other) graphics system under Modula-2, using COLORIX .VMG image files in your programs, reviews of the Stony Brook, Logitech and FST compilers as well as Solid Software and

JPI toolkits. In the near future, we will also have articles about using Modula-2 under UNIX, OS/2, Windows, and under other platforms.

Now that I have told you about what we have, let's talk for a moment about what we need from you. The obvious thing we need is more subscriptions. Let your friends and other programmers know about us. The other thing that we need is your ideas, code articles and announcements. You may have never thought about writing an article for a magazine before, or you may have been tempted, but never thought it would be accepted. Give it a try. Our rules are simple. If it is on the subject of Modula-2, it is a candidate for this magazine. We are not opposed to including multiple pages of code or to helping you write your article if you need it. Don't think that you don't have anything to share, everyone has their own programming style and maybe YOU can teach the "old dogs" some new tricks. If you have information about new Modula-2 products, shows where Modula-2 is highlighted, conferences, etc. let us know. We will be happy to put it in.

From the whole staff of *Modules & Definitions*, we hope that you will find us a valuable part of your "standard library".

Greg Walters  
Editor, Electronic Edition

### TABLE OF CONTENTS

FIO.Open()	
(**) The ToolBelt .....	Michael Hunt
Rational.DEF	
Rational.MOD	
TestRat.MOD	
A New Look For TopSpeed .....	Greg Walters
(**) Beginner's Luck .....	Jim Singleton
(**) U.S. Modula-2 Watch	
(**) The OBJECT of this Study .....	John Ribar
WinClass.DEF	
WinClass.MOD	
TestWC.MOD	
(**) Assembly NOT Included .....	Peter Baenziger
(**) SNEAKY TRICKS .....	Greg Walters
Spool.DEF	
Spool.MOD	
SOGEast '90	
(**) Modula-2 for the Real World .....	David Albert
ADC.DEF	
ADC.MOD	
VoltMeter.MOD	
FIO.Close()	

(\*\*) DENOTES A REGULAR MONTHLY COLUMN

## SUBSCRIPTION INFORMATION

Modules & Definitions is published monthly. The subscription rates are as follows:

Electronic \$9.95 annually

Download the magazine from your favorite BBS -

This is a SHAREWARE magazine - you must pay if you use it!

Paper (Printed) \$19.95 annually

Disk (with paper and electronic versions) \$59.95 annually

SHIPPING (Except on Electronic Only orders):

Canadian Orders (\$5) additional/annually

Other foreign orders (\$12 surface, \$20 airmail) annually

### SUBSCRIPTION APPLICATION

Name:

Address:

City: State/Province: ZipCode:

Country: Airmail? Yes No

Circle One: Electronic Paper Paper/Disk

Disk Size: 5.25" 3.5"

Payment:: Check Mastercard Visa

Number: Expires:

Mail To: Modules & Definitions

P. O. Box 7549

York, PA 17404 USA

Or Call: 717-792-5108



# Review of File Archive Formats

by Tom Cattrall

Last winter I mentioned that there was talk of writing a portable file archive utility. The P-system and some other systems have no archive utility available and this would be a chance to give users of these systems such a tool.

The essential characteristics of a file archive utility are the ability to:

- collect several files into a single archive
- compress the archive to save disk space and file transfer times
- save and restore file attributes
- be compatible with utilities available on other systems so that files can be interchanged

To the above, we might add: be written in Pascal and Modula-2 using as much as possible only easily portable language features.

Because there are already quite a few archive file formats available on a wide variety of systems, I feel that we should choose one of them rather than invent something new. This gives us access to files created on other systems, and it saves us the work of inventing something new.

The disadvantage is that the utility might not be as well suited to our needs as one we might design from scratch. This could come from a mismatch between file systems (names and file attributes won't match). Another reason is ease of implementation. Most file archive schemes were designed and written on other systems using C and/or assembly language. As a result, some features might not be especially easy to implement in standard Pascal or Modula-2.

In spite of this, I still prefer to implement some pre-existing standard. The following sections discuss some of the common archive utility file formats. The selection is based mainly on what I am familiar with as well as what is popular.

David Craig has sent a copy of a file archive format design he prepared some time back. Because of the reasons given above, I don't favor using a new unique format. I do plan to publish it however because it is based quite a bit upon MacIntosh and TIFF files. As I have no experience with either of these, I found it interesting as a source of ideas to broaden my knowledge of file sys-

tems.

In the following discussions, the file formats themselves rather than any particular implementation of the utility are what is being covered. It is important to keep in mind that the user interface has little to do with the underlying file format. We are free to implement whatever user interface we want even while following the chosen archive file format.

## ARC

The ARC utility is about the oldest file archiving package still widely being used. It started out as a means for a company by the name of Systems Enhancements Associates (SEA) to distribute software on floppy disk. It provided the basics of gathering a number of files together, and of compressing them to save disk space.

### Advantages

It is widely used not only on MSDOS systems but is also available on UNIX, VMS (DEC's operating system for the VAX computers), Amiga DOS, MacIntosh, and many others. If you put something into an ARC file, there is a good chance that most other computer systems will be able to read it. Because of its dominant position for so many years, there is a huge amount of material available in ARC format. For this reason alone, ARC is worth considering when deciding on a file archiving utility to implement.

### Disadvantages

The ARC utility was written to run on MSDOS and its archive structure only knows about short MSDOS file names. Thus, abc.txt is OK, but abc.text isn't. Nor is LongFileName.def. So any system that uses ARC format must map its names to the 8.3 character upper case only file name format. No file directory structure is allowed.

Compression performance is not as good as some more recent algorithms.

Over the years, the authors came up with improved compression schemes and added them to ARC's vocabulary. Each ARC file has a coded compression type that tells the decoding utility how to go about the decompression process. This means that while the compression performance of ARC has improved over the years, any ARC utility needs to be written to support this wealth of different schemes (9 or so). This makes any coding that

much more difficult.

### Availability

SEA, the authors and owners of ARC, have distributed C source code of the ARC utility. They have a very reasonable set of restrictions on using the ARC code and name. Very roughly speaking, the high points are:

- If you call it ARC, it had better be a well written and completely compatible program. If it isn't compatible with what they call ARC, you can't call it ARC.
- You should send them a copy of the source of any ARC utility you develop based on their source code. If you plan on making money from it, you must do so via shareware. They let you keep the first revenues (up to \$100K) and they then get a 6.5% cut of everything over \$100K.
- The ARC format itself is freely available for anyone to use.

SEA has been very helpful in making the ARC format available to all.

## Zoo

The Zoo archive format was developed by Rahul Dhesi approximately 4 years ago. Like the others, the Zoo format allows files to be collected and compressed. Its structure also allows such extras as long file names, case sensitive file names, and file comments. The ability to handle a wider range of file names, along with its availability on many systems, makes zoo my favorite for being widely accessible and easy to implement. In fact, I already have generic Pascal code written for the compression and decompression algorithms used by Zoo.

### Advantages

Simple structure with only a single relatively simple compression scheme. All files are stored either in uncompressed format, or in the single available compression scheme.

File names supported include: case sensitive, long names, directory names.

Files may have comments.

Multiple versions of files may exist in a Zoo archive.

Available on MSDOS, VMS, UNIX, MacIntosh, etc.

### Disadvantages

Compression performance not as good as some newer schemes.

### Availability

The Zoo format is freely available. As with ARC, the author of Zoo has distributed C source for the Zoo utility. There are some restrictions on using his source code but in general they aim to make sure that the source remains freely available, retains the copyright notices, and remains compatible with the Zoo format.

## LHarc

LHarc was developed in Japan and uses a new compression scheme that is quite a bit better than ARC or Zoo. It supports longer file names. It has become the dominant file archive utility for Amigas.

### Advantages

Better data compression than ARC or Zoo.

Handles long file names and directory structures.

Available on MSDOS, Amiga DOS, UNIX.

### Disadvantages

The documents describing the LHarc file format and compression scheme are written in Japanese. That would certainly slow down my ability to work with it.

The compression speed is rather slow.

### Availability

I'm not sure what the restrictions are on using the format. I have C source for a UNIX version of LHarc. There are around 4000+ lines of C code.

## ZIP

The ZIP utility was developed by Phil Katz in 1989. It is quite fast and has very good compression. It handles long file names and directories. ZIP looks like it might be becoming the dominant file archive utility for MSDOS.

### Advantages

Good compression.

Handles long file names and directories.

Has a robust file structure.

### Disadvantages

No source to use as a guide in writing a Zip utility.

No Zip utility available for UNIX, VMS, MacIntosh.

### Availability

A document describes the Zip file format and compression scheme. Use of the format itself is not restricted. Since there is no Zip source code to use as a model, writing a Zip utility from just the description would require extra effort to figure out the algorithms. (There are several different compression schemes that can be used in Zip files.)

### Others

The UNIX community uses compressed tar files and tapes. Compression factors are pretty good. The code has been finely tuned by many people and the result is very fast times for compress and decompress operations. Almost all UNIX sites can handle them, and utilities exist for many other systems (including MSDOS) to use compressed tar files.

Stuffit and Packit are used by MacIntosh systems. I'm not really familiar with their characteristics or performance. Utilities on MSDOS and other systems can be used to unpackage these files but in my experience they don't always work properly.

### Summary

As stated above, Zoo is my first choice because of its power and simplicity. It is relatively easy to implement and has good performance and is available on a large number of systems.

ARC would be my second choice because it allows access to so many existing files. The biggest disadvantage is that there are so many compression schemes to deal with.

Many of the routines written when doing the Zoo utility would be reused in doing ARC or any of the other utilities. These reused routines would include the CRC

(cyclical redundancy check) routine, user and file system interfaces, and date routines.

### Benchmarks

The following tables show the results from testing times and compression ratios for ARC, Zoo, LHArc, Zip, and UNIX Compress. I used Modula-2 source code for the text test, and 68020 object code for the binary test. These are just a sample to give an idea of where the various formats stand. Depending on the actual files being processed, times and compression ratios vary quite a bit. The relation between the different formats doesn't change by a lot though.

The times are in seconds, the sizes in bytes.

#### Modula-2 Source Code

	Size	Compress Time	Decompress Time
Original	180597		
ARC	71180	36	23
Zoo	68722	18	10
LHArc	48792	78	21
Zip	47497	35	10
Compress	67189	5	4

#### 68020 Object Code

	Size	Compress Time	Decompress Time
Original	134110		
ARC	76088	29	19
Zoo	73424	17	8
LHArc	54024	62	19
Zip	54163	52	8
Compress	74351	5	3

### ZOO Archive Format

The format of zoo archives is given below. A master header starts the zoo archive and then for each member of the archive, a file header plus the file data is appended.

```

Master Header
file 1 header
  file 1
file 2 header
  file 2
.
.
.
file n header
  file n
EOF

```

All multiple byte numeric fields are stored in Intel byte order (low byte first) necessary.

The text string in field 1 is used to tell users that the file is a zoo file if they should inadvertently try to view it as if it were an ascii format file. The headers are designed so that older versions of the zoo utility can decode files archived in a newer format. Any newer features will of course be ignored by the older code. The fields major and minor version needed to extract are the minimums

Several of the fields in both header types can be ignored. These include the comment fields and the file version fields. Using the pointers in the headers (field 3 in the master header, field 5 in the file header) will let you access the portions of interest: the headers and the actual data file.

To find the beginning of the first File header use the

Zoo Master Header				
Field	Bytes	Size	Value	Description
1	1 .. 20	20		20 bytes text string terminated by control Z
2	21 .. 24	4	fdc4a7dc	Hex tag to find headers
3	25 .. 28	4		Byte offset of where data starts
4	29 .. 32	4		Previous field negated for consistency check
5	33 .. 33	1		Major zoo version needed to extract
6	34 .. 34	1		Minor zoo version needed to extract
7	35 .. 35	1	1	Type of archive header
8	36 .. 39	4		Byte position of archive comment
9	40 .. 41	2		Length of archive comment
10	42 .. 43	2		Data about versions

pointer in the master header (field 3). The beginning of the file data is given by field 5 in the file header.

file leader is: "@)#(" followed by a null.

The tag fields are used to verify that a zoo archive is OK, and as an aid to finding headers if a file is corrupted. In addition, before the start of any data file, a 5 byte "FILE\_LEADER" is used to flag the start of data. The

The following hex dump is for a test zoo archive containing 2 1 line files. Since the files are so short, there is no compression.

The zoo listing of the test archive is:

Zoo File Header				
Field	Bytes	Size	Value	Description
1	1 .. 4	4	fdc4a7dc	Hex tag to find headers
2	5 .. 5	1	1	Type of header, always 1 for now
3	6 .. 6	1		Packing method: 0 = none, 1 = LZW
4	7 .. 10	4		Byte position of next directory entry
5	11 .. 14	4		Position of this file
6	15 .. 16	2		MSDOS format date
7	17 .. 18	2		MSDOS format time
8	19 .. 20	2		CRC of this file
9	21 .. 24	4		Original file size in bytes
10	25 .. 28	4		Current file size in bytes
11	29 .. 29	1		Major zoo version needed to extract
12	30 .. 30	1		Minor zoo version needed to extract
13	31 .. 31	1		Deleted Flag: 1 = file deleted, 0 = not
14	32 .. 32	1		File Structure (if any)
15	33 .. 36	4		Byte position of file comment (0 if none)
16	37 .. 38	2		Length of comment in bytes
17	39 .. 51	13		Filename (null terminated MSDOS format)
18	52 .. 53	2		Length of variable part of file header
19	54 .. 54	1		Time zone where file was archived
20	55 .. 56	2		CRC of file header entry
21	57 .. 57	1		Length of long filename (bytes)
22	58 .. 58	1		Length of directory name (bytes)
23	59 .. xx	xx		Long filename
24	xx .. xx	xx		Directory name
25	xx .. xx	2		Filesystem ID: 0 = UNIX, 1 = MSDOS, 2 = Other
26	xx .. xx	4		Filesystem attributes
27	xx .. xx	2		File version flag bits
28	xx .. xx	2		File version number (if any)

Archive test.zoo:

Length	CF	Size	Now	Date	Time		
18	0%	18	22 Jul 90	15:11:58	7100	file1.txt	
18	0%	18	22 Jul 90	15:12:22	b205	file2.txt	
36	0%	36	2 files				

```

000000 5a4f 4f20 322e 3030 2041 7263 6869 7665 ZOO 2.00 Archive
000010 2e1a 0000 dca7 c4fd 2a00 0000 d6ff ffff .....*.....
000020 0200 0100 0000 0000 0001 dca7 c4fd 0200 .....
000030 8300 0000 7100 0000 f614 7d79 0071 1200 ....q....}y.q..
000040 0000 1200 0000 0100 0000 0000 0000 0000 .....
000050 6669 6c65 312e 7478 7400 0000 000a 001c file1.txt.....
000060 a5c2 0000 0000 b601 4000 0000 4029 2328 .....e...e)#(
000070 0046 696c 6520 3120 3a20 4142 4344 3031 .File 1 : ABCD01
000080 3233 0adc a7c4 fd02 00dc 0000 00ca 0000 23.....
000090 00f6 148b 7905 b212 0000 0012 0000 0001 ...y.....
0000a0 0000 0000 0000 0000 0066 696c 6532 2e74 .....file2.t
0000b0 7874 0000 0000 0a00 1c06 c800 0000 00b6 xt.....
0000c0 0140 0000 0040 2923 2800 4669 6c65 2032 .e...e)#(.File 2
0000d0 203a 2041 4243 4430 3132 330a dca7 c4fd : ABCD0123....
0000e0 0200 0000 0000 0000 0000 0000 0000 0000 .....
0000f0 0000 0000 0000 0000 0000 0000 0000 0000 .....
000100 0000 0000 0000 0000 0000 0000 0000 0000 .....
000110 0000 fc83 ....
000114

```

The contents of the files are given below within quotes. The quote marks aren't included in the file contents.

```

file1.txt: "File 1 : ABCD0123"
file2.txt: "File 2 : ABCD0123"

```

Zoo has the capability to handle long filenames and to include the names of the directory paths for files. The following is an example using the same 2 files but with the full directory path names used.

Archive test.long.zoo:

Length	CF	Size	Now	Date	Time		
18	0%	18	22 Jul 90	15:11:58	7100	/u3/usus/9007/file1.txt	
18	0%	18	22 Jul 90	15:12:22	b205	/u3/usus/9007/file2.txt	
36	0%	36	2 files				

```

000000 5a4f 4f20 322e 3030 2041 7263 6869 7665 ZOO 2.00 Archive
000010 2e1a 0000 dca7 c4fd 2a00 0000 d6ff ffff .....*.....
000020 0200 0100 0000 0000 0001 dca7 c4fd 0200 .....
000030 9100 0000 7f00 0000 f614 7d79 0071 1200 ....y.q..
000040 0000 1200 0000 0100 0000 0000 0000 0000 .....
000050 6669 6c65 312e 7478 7400 0000 0018 001c file1.txt.....
000060 e742 000e 2f75 332f 7573 7573 2f39 3030 .B.../u3/usus/900
000070 3700 0000 b601 4000 0000 4029 2328 0046 7....e...e)#(.F
000080 696c 6520 3120 3a20 4142 4344 3031 3233 ile 1 : ABCD0123
000090 0adc a7c4 fd02 00f8 0000 00e6 0000 00f6 .....
0000a0 148b 7905 b212 0000 0012 0000 0001 0000 ...y.....
0000b0 0000 0000 0000 0066 696c 6532 2e74 7874 .....file2.txt
0000c0 0000 0000 1800 1c07 d900 0e2f 7533 2f75 ...../u3/u
0000d0 7375 732f 3930 3037 0000 00b6 0140 0000 sus/9007....e..
0000e0 0040 2923 2800 4669 6c65 2032 203a 2041 .e)#(.File 2 : A
0000f0 4243 4430 3132 330a dca7 c4fd 0200 0000 BCD0123.....
000100 0000 0000 0000 0000 0000 0000 0000 0000 .....
*
000120 0000 0000 0000 0000 0000 0000 0000 fc83 .....
000130

```

## ARC File Format

The format of an ARC file is given below. Each component file is appended to the end of any previous members in the ARC file. Each member file is preceded by a file header.

```

file header 1
file 1
file header 2
file 2
.
.
.
file header n
file n
EOF

```

The elements in the header are:

Field	Bytes	Size	Value	Description
1	1 .. 1	1	26	ID Byte (Hex 1A)
2	2 .. 2	1	1..9	Compression Method Used
3	3 .. 15	13		Filename (null terminated)
4	16 .. 19	4		Compressed File Size in Bytes
5	20 .. 21	2		File date (MSDOS format)
6	22 .. 23	2		File time (MSDOS format)
7	24 .. 25	2		File CRC
8	26 .. 29	4		Uncompressed File Size in Bytes

Field 8 is omitted if the compression method is 1. In this case, the header is 25 bytes long rather than 29 bytes.

Numeric values in fields 4 .. 8 are stored in Intel (low byte first) byte order.

The compression method is identified by a value of 1 to 9 as described in the following table:

NAME	METHOD	DESCRIPTION
Stored	1	Old style, no compression
Stored	2	New style, no compression
Packed	3	Compression of repeated characters only
Squeezed	4	Compression of repeated characters plus Huffman Squeezing
crunched	5	Lempel-Zev packing of repeated strings (old style)
crunched	6	Lempel-Zev packing of repeated strings (new style)
crunched	7	Lempel-Zev-Welch packing with improved hash function
Crunched	8	Dynamic Lempel-Zev packing with adaptive reset
Squashed	9	Dynamic Lempel-Zev packing, larger hash table

The beginning of File Data is 29 bytes after the beginning of the file header. To get to the file header of the next file, skip forward by 29 bytes for the current header, and by the compressed size of the current file (field number 5). Use 25 bytes rather than 29 bytes for the header length if the compression method is 1.

MSDOS dates are stored as a single 16 bit word in the following format:

Bits	Description
15 .. 9	Year - 1980
8 .. 5	Month
4 .. 0	Day

MSDOS times are stored as a single 16 bit word in the following format:

Bits	Description
15 .. 11	Hour (24 hour clock)
10 .. 5	Minute
4 .. 0	Second / 2

The format for the 2 test files in arc format is given below in a mixed character and hex format dump. There was no file compression because of the short files used.

The arc utility report for the test archive is:

Name	Length	Storage	SF	Size now	Date	Time	CRC
file1.txt	18	---	0%	18	22 Jul 90	3:11p	7100
file2.txt	18	---	0%	18	22 Jul 90	3:12p	b205
<b>Total</b>	<b>2</b>	<b>36</b>	<b>0%</b>	<b>36</b>			

```
000000 1a02 6669 6c65 312e 7478 7400 a818 0012 ..file1.txt.....
000010 0000 00f6 147d 7900 7112 0000 0046 696c .....}y.q....Fil
```

```
000020 6520 3120 3a20 4142 4344 3031 3233 0a1a e 1 : ABCD0123..
000030 0266 696c 6532 2e74 7874 00a8 1800 1200 .file2.txt.....
000040 0000 f614 8b79 05b2 1200 0000 4669 6c65 .....y.....File
000050 2032 203a 2041 4243 4430 3132 330a 1a00 2 : ABCD0123...
000060 1400 ..
000062
```

Pascal versions of the Zoo compress and decompress routines are given below. I translated it from Rahul Dhese's C code. As shown, it works exactly the same as the C code and can be used to compress and decompress zoo archive or any other files.

I started translating this code on nearly a line by line basis from the C code. The result was a lot of work and quite a mess. Switching gears, I studied the C code and then used the ideas to write Pascal that did the equivalent function. But I was unable to completely forget the C code while writing Pascal. So the Pascal is still influenced quite a bit by the C code, and it wasn't a positive influence.

Now that I've been away from this for a couple of years, I can see things I'd like to change. In particular, the main loops could be structured better to remove the need for GOTO's, and to make the flow cleaner. Also, the routines to pack the variable length bit fields: WrCode in the compress routine, and RdDecode in the decompress routine, could be changed to be cleaner and faster running.

## From the USUS Librarian by Stephen Pickett

I recently inherited the USUS Apple ][e, and am now in a position to fill orders for the following formats: Sage, IBM (5.25 and 3.5"), Apple ][, 8" single and double density. This is made possible by the weird and heterogeneous Corvus Omninet strung through my house. I even have an Atari ST hooked to it, though no Atari p-System, so if anyone can accept TOS-hosted volumes then there's no problem there either. The whole lot is nicely archived on a 150MB streamer tape and I can run copies off in minutes.

Problem is, no one called!

In an effort to make the library worth calling for, I am about to review and add a bunch of software for inclusion. As I my expertise is limited to Pascal, our brave Newsletter Editor, Tom Catrall, will be assisting me with Modula-2.

The material under review comes from the following sources:

1. MUSUS material which has not been hitherto in the disk library. I have scavenged some interesting-looking stuff, including a utility for Apple ][ pcode which allows the system to do many interesting things under Version II which were pre-

viously only possible under Version IV, which has never run very well on the ][.

2. Some material of my own making churned out at odd moments over the years

3. (at least promised) a program that has been used to run the totaliser at a race track, using an Apple ][.

Please try to add to the above list. I should be able to turn out one volume per month for about a year before I run out of stuff, but your hot submissions will probably improve on my scrapings of the proverbial barrel!

Stephen Pickett, Librarian

CompuServe: 71016,1203  
 Phone days: (604) 439-1858  
 US mailing address: PO Box 1279,  
 Point Roberts, WA  
 98281-1279

```

PROGRAM lzcompress;
LABEL
CONST
1;
maxBits      = 13;
maxMax      = 8192;
maxMax10    = 8202;
clearCode   = 256;
eofCode     = 257;
firstFreeCode = 258;
checkGap    = 4000;
ratio (*)   = 512;
bufferSize  = 512;
inName      = 'testIn';
outName     = 'testOut';
TYPE
atableIndex = 0..maxMax10;
atableEntry = RECORD
    first,
    next      : -1..maxMax10;
    zCh       : 0..255;
END;
aStatus = (found, nextUse, firstUse);
aBuffer = PACKED ARRAY [ 1..bufferSize] OF CHAR;
VAR
freeCode    : 0..maxMax;
nBits       : INTEGER;
maxCode     : INTEGER;
bitsOut     : INTEGER;
bitInterval : INTEGER;
bytesIn,
ratio       : INTEGER;
ratFlag     : BOOLEAN;
inOffset    : INTEGER;
inputLine  : string[255];
inFile     : TEXT;
outFile    : FILE OF aBuffer;
outPtr     : INTEGER;
outByteTel : INTEGER;
outBit     : INTEGER;
table      : ARRAY[atableIndex] OF aTableEntry;
nextCh,
prefixCode,
k           : INTEGER;
where      : INTEGER;
status     : aStatus;
PROCEDURE PutByte(b : INTEGER);
BEGIN
    outFile[outPtr]:= CHR(b);
    IF outPtr >= bufferSize THEN
        BEGIN
            Put(outFile);
            outPtr:= 1;
        END
    ELSE

```

```

END;
outPtr:= outPtr + 1;
PROCEDURE OpenFiles;
BEGIN
    Reset( inFile, inName);
    ReadLn(inFile, inputLine);
    inOffset:= 0;
    Rewrite( outFile, outName);
    outPtr:= 1;
END;
PROCEDURE CloseFiles;
BEGIN
    Close ( inFile );
    PutByte(outByteTel);
    IF outPtr > 1 THEN
        Put(outFile);
    Close ( outFile, lock);
END;
PROCEDURE wrCode (code : INTEGER);
VAR
    i : INTEGER;
BEGIN
    bitsOut := bitsOut + nbits; (* total number of bits written *)
    bitInterval := bitInterval - nbits;
    IF (bitInterval < 0) THEN
        ratFlag := TRUE; (* time to check ratio *)
        FOR i:= 1 TO nbits DO
            BEGIN
                IF ODD(code) THEN
                    outByteTel:= outByteTel + outBit;
                IF outBit = 128 THEN
                    BEGIN
                        PutByte(outByteTel);
                        outBit := 1;
                        outByteTel:= 0;
                    END
                ELSE
                    outBit:= outBit + outBit;
                    code:= code DIV 2;
                END;
            END;
        END;
    PROCEDURE Initctab;
    VAR
        i : INTEGER;
    BEGIN
        bytesIn := 0;
        bitsOut := 0;
        ratio := 0;
        ratFlag := FALSE;
        bitInterval := checkGap;
        nbits := 9;
        maxCode := 512;
        FOR i := 0 TO maxMax + 1 DO
            BEGIN
                table[i].zCh := 0;
                table[i].first := -1;
                table[i].next := -1;
            END;
        freeCode := firstFreeCode;
    END;

```

```

(* ch char to look for *)
(* * where last entry looked at *)
VAR count : INTEGER;
VAR t : astatus;
BEGIN
  where := index;
  index := table[index].first;
  t := firstUse;
  IF (index <> -1) THEN
    BEGIN
      WHILE t = firstUse DO
        BEGIN
          IF ((table[index].zch MOD 256) = (ch MOD 256)) THEN
            BEGIN
              where := index;
              index := table[index].next;
              IF (index = -1) THEN
                t := nextUse;
            END;
          ELSE
            BEGIN
              where := index;
              index := table[index].next;
              IF (index = -1) THEN
                t := nextUse;
            END;
          END; (* end WHILE *)
        END; (* end IF *)
      LookupCode:= t;
    END;
  BEGIN { LZcompress }
  OpenFiles;
  outByte1:= 0;
  outBit:= 1;
  InitCtab;
  WrCode(clearCode);
  IF NOT RdCh(nextCh) THEN
    BEGIN
      WrCode (eofCode);
      GOTO 1;
    END; (* normal return from compress *)
  (* compression loop begins here with nextCh holding the next input char *)
  WHILE TRUE DO
    BEGIN
      IF ratFlag THEN
        CheckRatio;
      nextCh := nextCh MOD 256;
      REPEAT
        PrefixCode := nextCh;
        IF NOT RdCh(nextCh) THEN
          BEGIN
            WrCode (prefixCode);
            WrCode (eofCode);
            GOTO 1;
          END;
        nextCh := nextCh MOD 256;
        k := nextCh;
        status := LookupCode (prefixCode, nextCh, where);
        IF (status = FOUND) THEN
          nextCh := where;
        UNTIL status <> found;
      (* reach here with status := firstUse or nextUse *)
      AddCode (status, nextCh, where);
      WrCode (prefixCode);
    END;
  END;
  (* normal return from compress *)
  nextCh := nextCh MOD 256;
  k := nextCh;
  status := LookupCode (prefixCode, nextCh, where);
  IF (status = FOUND) THEN
    nextCh := where;
  UNTIL status <> found;
  (* where found *)
  (* reach here with status := firstUse or nextUse *)
  AddCode (status, nextCh, where);
  WrCode (prefixCode);

```

```

FUNCTION RdCh(VAR c : INTEGER) : BOOLEAN;
VAR count : INTEGER;
BEGIN
  bytesIn:= bytesIn + 1;
  RdCh:= TRUE;
  IF (inOffset > length(inputLine)) THEN
    BEGIN
      IF eof(inputFile) THEN
        RdCh:= FALSE;
      ELSE
        BEGIN
          ReadLn(inputFile, inputLine);
          inOffset:= 1;
          if length(inputLine) > 0 THEN
            c:= ORD( inputLine[inOffset] )
          ELSE
            c:= 10;
          END;
        END;
      ELSE
        BEGIN
          inOffset:= inOffset + 1;
          IF inOffset > length(inputLine) THEN
            c:= 10
          ELSE
            c:= ORD( inputLine[inOffset] );
          END;
        END;
      END;
    END;
  PROCEDURE CheckRatio;
  BEGIN
    bitInterval := CHECKGAP;
    bitsOut := 0;
    bytesIn := 0;
    ratFlag := FALSE;
  END;
  PROCEDURE AddCode (status: astatus;
    ch, index : INTEGER);
  VAR t : INTEGER;
  BEGIN
    IF freeCode >= maxMax THEN
      t:= -1
    ELSE
      t:= freeCode;
    IF (status = nextUse) THEN
      table[index].next := t
    else
      table[index].first := t;
    IF (freeCode <= maxMax) THEN
      BEGIN
        table[freeCode].first := -1;
        table[freeCode].next := -1;
        table[freeCode].zch := ch MOD 256;
        freeCode:= freeCode + 1;
      END;
    END;
  END;
  FUNCTION LookupCode (index, ch : INTEGER;
    VAR where : INTEGER) : astatus;
  (* index where to start looking *)

```



```

nextCh := k;
IF (freeCode > maxCode) THEN
  BEGIN
  IF (nbits >= MAXBITS) THEN
    BEGIN (* To continue using table after it is full,
           remove next two lines *)
      wrCode (clearCode);
      InitCtab;
    END
  ELSE
    BEGIN
      nbits := nbits + 1;
      maxCode := maxCode + maxCode;
    END;
  END; { IF }
END; { WHILE }
l: CloseFiles;
END.

```

```

PROGRAM Lzdecompress;
LABEL
CONST
  stackSize = 4000; (* size of decode stack *)
  maxBits = 13; { max nb code bits }
  maxMax = 8192; { max code + 1 }
  maxMax10 = 8202; { maxMax + 10 }
  clearCode = 256; { code to clear table }
  eofCode = 257; { marks end of compressed data }
  firstFreeCode = 258; { first free code }
  inSize = 512; (* size of input buffer *)
  outSize = 519; (* size of output buffer *)
  inName = 'testout';
  outName = 'testoutd';

TYPE
  aTableIndex = 0..maxMax10;
  aTableEntry = RECORD
    next : INTEGER;
    zCh : INTEGER;
  END;
  anInputBuffer = PACKED ARRAY [1..inSize] OF 0..255;

VAR
  i, t : INTEGER;
  table : ARRAY[aTableIndex] OF aTableEntry;
  curCode,
  oldCode,
  inCode : INTEGER;
  freeCode,
  nbits,
  maxCode : INTEGER;
  finChar, : INTEGER;
  inByte1 : INTEGER;
  inBit : INTEGER;
  outputOffset : INTEGER;

```

```

stackPointer : 0..stackSize;
stack : ARRAY[0..stackSize] OF INTEGER;
inFile : FILE OF anInputBuffer;
outFile : TEXT;
inPtr : INTEGER;
inBuffer : anInputBuffer;

PROCEDURE OpenFiles;
BEGIN
  Reset(inFile, inName);
  inPtr:= inSize + 1;
  inBit:= 8;
  Rewrite(outFile, outName);
END;

PROCEDURE CloseFiles;
BEGIN
  Close(inFile);
  Close(outFile, LOCK);
END;

PROCEDURE Push(x : INTEGER);
BEGIN
  stack[stackPointer]:= x;
  IF stackPointer <= stackSize THEN
    stackPointer:= stackPointer + 1
  ELSE
    BEGIN
      WriteLn('Stack Overflow');
      GOTO l;
    END;
END;

FUNCTION Pop : INTEGER;
BEGIN
  IF stackPointer >= 1 THEN
    stackPointer:= stackPointer - 1;
    Pop:= stack[stackPointer];
  END
  ELSE
    BEGIN
      WriteLn('Stack Underflow');
      GOTO l;
    END;
END;

FUNCTION GetByte : INTEGER;
BEGIN
  IF inPtr > inSize THEN
    IF NOT EOF(inFile) THEN
      BEGIN
        READ(inFile, inBuffer);
        inPtr:= 1;
      END;
  IF inPtr <= inSize THEN
    BEGIN
      GetByte:= inBuffer[inPtr];
      inPtr:= inPtr + 1;
    END
  END

```

```

ELSE GetByte:= -1;
END;
(* RdDecode reads a code from the input (compressed) file and returns
its value. *)
FUNCTION RdDecode : INTEGER;
VAR
  t,
  w,
  i
  : INTEGER;
BEGIN
  w:= 0;
  t:= 1;
  FOR i:= 1 TO nbits DO
    BEGIN
      IF inBit >= 8 THEN
        BEGIN
          inByte:= GetByte;
          IF inByte < 0 THEN
            BEGIN
              WriteLn('Unexpected EOF on input, corrupt compressed
              file');
            END;
            GOTO 1;
          END;
          inBit:= 0;
        END;
        IF ODD(inByte) THEN
          w:= w + t;
          inByte:= inByte DIV 2;
          t:= t + t;
          inBit:= inBit + 1;
        END;
        RdDecode:= w;
      END; (* RdDecode *)
    END;
  PROCEDURE InitDTab;
  BEGIN
    nbits := 9;
    maxCode := 512;
    freeCode := firstFreeCode;
  END;
  PROCEDURE wrdChar (ch : INTEGER);
  BEGIN
    IF ch <> 10 THEN
      Write(outfile, CHR(ch))
    ELSE
      WriteLn(outfile);
    END; (* wrdChar *)
  END;
  (* adds a code to table *)
  PROCEDURE AddCode;
  BEGIN
    table[freeCode].zCh := ch;
    table[freeCode].next := oldCode;
    freeCode:= freeCode + 1;
    IF (freeCode >= maxCode) THEN
      BEGIN
        IF (nbits < MAXBITS) THEN
          BEGIN
            nbits:= nbits + 1;
            maxCode := maxCode + maxCode;
          END;
        END;
      END;
    END;
  END;
  (* save suffix char *)
  (* save prefix code *)

```

```

END;
BEGIN
  { LzdCompress }
  nbits := 9;
  maxCode := 512;
  freeCode := firstFreeCode;
  stackPointer := 0;
  outputOffset := 0;
  OpenFiles;
  InitDTab;
  (* initialize table *)
  WHILE TRUE DO
    BEGIN
      curCode := RdDecode;
      IF (curCode = eofCode) THEN
        GOTO 1;
      END;
      IF (curCode = clearCode) THEN
        BEGIN
          InitDTab;
          curCode := RdDecode;
          finChar := curCode;
          k := curCode;
          oldCode := curCode;
          wrdChar(k);
        END;
      ELSE
        BEGIN
          inCode := curCode;
          IF (curCode >= freeCode) THEN (*IF code not in table(k<=>k<=k) *)
            BEGIN
              curCode := oldCode;
              push(finChar);
            END;
          WHILE (curCode > 255) DO
            BEGIN
              push(table[curCode].zCh);
              curCode := table[curCode].next; (* push suffix char *)
            END;
          finChar := curCode;
          k := finChar;
          Push(k);
          WHILE (stackPointer <> 0) DO
            wrdChar(Pop);
            AddCode;
            oldCode := inCode;
          END; { WHILE }
        END;
      END;
    END;
  1 : CloseFiles;
END.

```

# Board Meeting Minutes (May 9, 1990)

By Keith R. Frederick

Minutes of the Board Meeting of USUS, Inc., held in room 1 of the MUSUS forum teleconferencing facility on the CompuServe Information Service May 9, 1990.

Present at the meeting were:

User ID	Name
71016,1203	<b>Stephen Pickett (sfbp) BoD</b>
73760,3521	Keith Frederick (KeithF)
70260,306	Hays Busch (AHB)
72767,622	<b>Tom Cattrall (TomC) BoD</b>
72747,3126	Bob Clark (BobC)
73447,2754	<b>Henry Baumgarten (Henry) Bod</b>
74076,1715	Felix Bearden (felix)
73007,173	William Smith (Wm)

The meeting started off with Tom Cattrall asking William Smith if he wished to chair the meeting. William obliged with Stephen Pickett seconding; Henry Baumgarten, Felix Bearden, and Hays Busch agreeing.

## I. Apple IIe and Library Disks

Tom Cattrall moved to accept Hay's offer regarding his storing of the Apple IIe and library disks until there is reason to send them elsewhere. However, Stephen Pickett had an amendment. Since he is the new Keeper of the Library he thought that it would be sensible for him to be entrusted with the Apple IIe.

After some discussion regarding the details of bringing the Apple IIe across the border (into Canada) and the exact procedure needed to amend the motion, the mover (Tom Cattrall) accepted the amendment.

The Board then voted on the amended motion. All voted in favor and William stated that Hays and Stephen will arrange for the transfer of the Apple IIe.

## II. Changes to the bylaws

Since Felix Bearden had not yet written the changes to the bylaws, regarding meeting attendance by the Board of Directors, he suggested that the discussion be tabled.

## III. Placing an advertisement in the Journal of Pascal, Ada, and Modula-2 (JPAM).

Felix suggested a budget of \$1100 for the advertisement, indicating however, that the actual quoted cost was less but he included money for doing a good "mat" for JPAM. Henry Baumgarten commented that the currently planned ad had serious deficiencies and was difficult to understand. William stated that there are three main questions:

- 1). Do we place an ad,
- 2). how much do we spend, and
- 3). what goes in the ad.

Also noting that the content of the ad can best be discussed offline. Felix moved to

- 1). Place an ad in JPAM,

2). budget \$1100 for the ad, and

3). that Tom Cattrall and Henry approve the ad before it is placed.

Tom stated that there should be work done on the ad offline and have an ad presented in the next meeting for approval. However, now a vote to place the ad, if a satisfactory one is made, should be done.

Henry agreed but thought that the format of the advertisement could be settled while online. Henry suggested that a vote on whether the ad should be placed should be made and then the topic can be taken from there.

Stephen commented that there is no point in placing a single ad but advertising should be looked at in terms of policy and an ongoing commitment to promulgate USUS. Secondly, with the substantial changes in the makeup of the board and officers, he believes, that the expenses will be kept under control.

Stephen then asked for the treasurer (Bob Clark) to comment. Bob stated that he couldn't see USUS be more frugal than it had been in the past and that currently USUS has \$7,088.23 in the bank with a newsletter to pay for and the JPAM subscription costs. However, Bob mentioned that USUS has not paid for any of the JPAM subscriptions and doesn't know how or when USUS will be billed.

William asked if an estimate on the JPAM subscription costs could be made. Felix replied that an estimate can't be made yet as the renewals are just starting to come in. Tom Cattrall calculated that current USUS expenses seem to run about \$600 per month, which implies that current funds would last roughly ten months. However, Tom noted that his figure is quite rough but probably still in the ball park.

Bob Clark noted that he would be out of touch from mid June through mid July, so any arrangements must be made money-wise that they be handled before then.

Henry asked if there is a seconded motion on the floor. William replied yes and asked for a vote. Felix and William voted yes. Bob Clark abstained. Tom Cattrall and Henry voted yes with the condition that offline discussions approve the copy before it is submitted. Stephen reminded them that an amendment must be moved. Henry agreed and Tom then suggested that the motion be removed and resubmitted in a new form to make sure the offline discussion part is included. Stephen agreed and stated that the following wording (or something close) be added to the beginning of the motion:

"WHEREAS USUS agrees that it is in the society's best interest to promulgate itself, that a program of advertising be undertaken, to include but no necessarily be limited to the following advertisement: <advertisement text follows> Also that all members of USUS and its board exhort one another to publish details of its activities by any means possible AT NO COST TO USUS, and that those preparing the present advertisement be required to present an ongoing publicity plan to the August Board Meeting."

Felix agreed to withdraw the original motion and resubmit it

with Stephen's wording up to but not including "Also..." and adding that the advertisement be approved offline by Tom and Henry before it is placed.

Tom then asked if anyone had any expertise in publicity and if so could they help. Felix replied that he had a little experience and even though the return on an ad of this nature is low it does state that USUS is alive. Felix further commented that direct mailing that is being covered at no cost to USUS is a better return on investment. Stephen suggested that perhaps his wife (Cathy), being the senior account executive at the Vancouver Office of Canada's largest PR/advertising firm, could help USUS in this matter. Tom Cattrall expressed interest in the offer and stated that the actual submission of the ad be held off a bit. William stated that the board is now voting on the current motion. The vote was unanimously approved.

The full motion that was passed is as follows:

"WHEREAS USUS agrees that it is in the society's best interest to promulgate itself, that a program of advertising be undertaken, to include but no necessarily be limited to the following advertisement:

- a. that USUS place the ad in JPAM
- b. that USUS budget \$1100 for the ad
- c. that the ad be approved offline by Tom Cattrall and Henry before it is placed."

#### IV. Selection of a Board of Director Chairperson

Henry quickly nominated William Smith for the position and Felix seconded it. However, Stephen Pickett noted that this was not possible as the Chairperson cannot be a non-Board member. Stephen then asked Henry if he would serve as Chair. Henry stated though, that being "nominally retired" he intends to spend as much of the summer at his isolated cabin three miles south of the Canadian border in northeastern Minnesota and thus would hardly be the best choice for Chairperson.

Tom Cattrall suggested that since USUS doesn't have a Vice President, that William Smith be elected VP, Henry the chairperson, and retain Alex as president so that there would be three chances for someone to lead meetings.

Felix and Stephen agreed. Stephen went on to say that in the short term it isn't important to determine the board chairperson. Henry stated if it is understood that he wouldn't be readily available from about mid-June to mid-September then he would

agree to be chairperson. Stephen suggested that USUS only have one board meeting during the summer period because, in his opinion, not much happens in the summer and it would limit the number of meetings that Henry would miss.

Felix Bearden then nominated Henry Baumgarten with Tom Cattrall and Stephen Pickett seconding. William Smith commented that he is willing to stand in for Henry to lead the meetings, like tonight, but not be a Vice President at this time. The vote for Henry to be Chair was made with all voting yes except Henry, who abstained.

#### V. Renting the USUS mail list

Tom Cattrall stated that there has been a request to rent the USUS mail list and questioned what is the procedure for handling the request.

Henry suggested that Felix contact Hays, who probably had a definite procedure for handling the situation. Otherwise, Henry mentioned, that he favors having the staff handling the request while still keeping the board informed in case policy decisions need to be made. William Smith indicated that Henry's suggestion was the way it has been previously handled but noted that he didn't know the current price for renting the list.

Stephen Pickett asked if USUS still respected the flag on the membership forms indicating that the member didn't want his/her name to be sold. Felix replied that yes, USUS still respects that request.

William then came up with a price from 2/1/88, which was 10 cents per address plus materials and shipping. The labels would be just like the News Letter labels except, of course, some are "culled" since some members don't want the mail.

Stephen then stated that he would support the existing procedures; indicating that there is no need for the board to interfere at this time. Tom agreed and said he would put Felix in touch the person interested in the list.

#### NEXT MEETING

The Board agreed to adjourn at 9:08:06 PM PST and meet again at 6:30 PM PST / 7:30 MST / 8:30 CST / 9:30 EST June 13th 1990 in Room 1 of the MUSUS conference facility.

Minutes submitted by Keith R. Frederick

## Board Meeting Minutes (June 13, 1990)

By Keith R. Frederick

Minutes of the Board Meeting of USUS, Inc., held in room 1 of the MUSUS forum teleconferencing facility on the CompuServe Information Service May 13, 1990.

Present at the meeting were:

User ID	Name
71016,1203	Stephen Pickett (sfbp) BoD
73760,3521	Keith Frederick (KeithF)
72767,622	Tom Cattrall (TomC) BoD

73447,2754	Henry Baumgarten (Henry) BoD
74076,1715	Felix Bearden (felix)
73007,173	William Smith (Wm)
73030,2522	Ron Williams (RONW)
72230,1601	(gary)
76702,513	Harry Baya(Harry)

The Board Meeting started at 6:32 PM PDT with Henry Baumgarten as chairperson.

## **I. MUSUS (in)activity and changes to the bylaws**

Harry iterated a note he received from the CIS people. The note stated that MUSUS has until August to make significant changes in message volume or else they would close down the forum. However, Scott Cress of CIS did offer help to find MUSUS another home if need be. Harry pointed out that significant volume could be regarded as a steady increase from the near zero-level that MUSUS currently has.

Stephen Pickett commented that CIS is being extremely reasonable and if MUSUS can't generate some message volume the organization will die quickly. He also mentioned that he has some ideas to generate activity based on his plans regarding the USUS software library.

Harry mentioned that he, unless someone opposes the idea, plans to bring in a Modula-2 SysOp for MUSUS. Also, he stated that he prefers "go frog" instead of "go codeport" for the new "go word" of MUSUS. Felix agreed with Harry on "go frog."

Harry suggested to return to the MUSUS problem later and concern ourselves with the matter of attendance by directors.

Felix apologized for not having the bylaws amendment ready, indicating that he had been out of town mostly since the last meeting. Tom Cattrall moved to table the bylaws issue until the next meeting. Felix Bearden, Harry, and William Smith agreed to table. Stephen Pickett disagreed with tabling. Henry declared the issue tabled until the next meeting.

Stephen stated his disfavor with the tabling decision. Mentioning that there won't be a meeting with a quorum until Henry gets back and if CIS has no place to put MUSUS there may be no place for a meeting. Thus, USUS has to act now on almost everything or else USUS won't be able to do anything.

Felix commented by saying that the bylaws issue requires a membership vote. Felix suggested that a special meeting be called. Otherwise the topic can be handled online. Tom Cattrall liked Felix's suggestion and asked whether a proposed bylaws change could be drafted in two weeks. There was no response other than Felix stating that he would relinquish the responsibility to any volunteers.

Stephen stated that he did not agree with the amendment. Stephen stated that USUS could survive nicely, as in the past, on twice-a-year meetings (if necessary by conference call). Furthermore he believed that any board member asked to resign because of lack of attendance would do so. Lastly, the amendment is no substitute for someone taking the time to phone a board member in question. Stephen said that he was almost a victim of the amendment without even realizing he had been elected. Noting that we wouldn't want others to be treated as he was.

Henry responded by saying that previously the phoning idea was used without much success and that Tom's reminder post-cards appear to be more effective.

Stephen moved to replace the current topic with a motion that "enshrines in the expectations USUS has of directors, the idea of attendance at meetings." Furthermore, he stated that removal of the bylaw change be made. Felix stated that he would try to post proposed changes next week.

Stephen went on to state that the people have been elected, not appointed to this board. Therefore it doesn't seem reasonable for the few other individuals to have any sort of arbitrary power

to remove them. Also that it is grossly unfair to the electorate and would put the board in jeopardy with its constituents. Tom reminded everyone that it would be the members of USUS that would be the ones that decide if the board should have this power. The decision would not be a purely arbitrary one.

Henry and Tom both commented on personal experiences with other groups; both illustrating how an attendance rule can work and be fair. Felix stated that he would work out the wording for the bylaw change. Stephen indicated that there is a big difference between missing a meeting because one doesn't want to bother with attending and missing because one cannot attend for legitimate reasons. Furthermore, the amendment must insist on a fresh election for any vacated position.

Felix reminded everyone that the motion was tabled. Henry agreed and decided to go to the next topic.

## **II. Advertisement in the Journal of Pascal, Ada, and Modula-2 (JPAM)**

Felix started off by saying that he had not yet received the advertisement package from JPAM who promised it was on its way. Felix also stated his dissatisfaction with the current planned advertisement and that he will post another as soon as possible. Henry said he had given some comments to Felix regarding the advertisement.

Tom stated his approval of the comments made to Felix. Felix again stated he is going to post a revised copy of the advertisement.

## **III. USUS Library**

Stephen stated that he had received the Apple IIe with only minor glitches and is in the process of gathering ideas in order to put together some more useful disks for the library. Furthermore, he also mentioned that the MUSUS library only contained recent files and wondered if older files had been purged.

Tom commented that he intended to talk about trying to get a more user friendly library submission form that would encourage public domain and/or more freely distributable submissions.

Felix told Stephen that there are no outstanding library orders and that Hays had taken care of them. Harry, commenting on Stephen's statement, stated that he had only removed about seven items from the library and that the bulk of the library should be still online. Also, Harry said, that he agreed with acquiring a new submission form.

Henry asked Tom if he would draft a new submission form. Tom said he would do what he could and would work offline with Stephen Pickett on the library issue.

## **IV. More on MUSUS**

Harry, reiterating the earlier discussion, stated that he is pursuing John Ribar as a Modula-2 SysOp and that he needs a new "go word" for the forum. Harry restated his preference for "go frog" but said he would go with "go codeport" if there was no support for "go frog."

Tom Cattrall agreed on John Ribar as a Modula-2 SysOp and that his preferred "go" choices are: Program, or CodePort rather than frog. Stephen said he would like to see PORTABLE, PORTSOFT, GENERIC, or even STRUCTURED as code-words. Also, in regards to John Ribar, that an agreement is writ-

ten down which makes it clear to the new SysOp what the issues and terms of his relationship with USUS is.

Felix gave his support for the forum to be named "Portable Language Forum," with "go PLanFor" as the codeword. His second choice is "codeport." Harry stated that he would prefer not to have a written agreement with John Ribar, indicating that he nor William nor Eli had signed agreements with USUS. Henry also suggested that "Codeport" is the best compromise so far.

Tom Catrall agreed with Harry about not having a written agreement. Tom also said that USUS needs to upload as much useful Modula-2 and Pascal code and to publicize MUSUS in any way possible.

Henry said he would like a simple name like "PLF" for short, and "Portable Language Forum" for long. Felix suggested having an Ada section since it is keeping with the past advertisement and with JPAM.

Harry commented that CIS has offered to help MUSUS with publicity and that in the SysOp forum (on CIS) he is getting a lot of cooperation. Also, Harry mentioned that he is adding an

Ada section to MUSUS. Harry then asked for final comments regarding "codeport" as being the new "go word." Stephen suggested "portcode" (the real name for pcode) and Tom Catrall felt "go Pascal" and "go Modula" will be used more often but "codeport" was fine with him. CODEPORT ended up as the agreed upon name.

#### V. Next Meeting

Felix indicated that there would be a special meeting on June 27 and then the regular meeting on July 11. Stephen pleaded that it be one or the other but not both. Tom Catrall suggested meeting on June 27 and then in August. Stephen couldn't promise to be available in August but Felix and Henry stated their approval.

The Board agreed to adjourn at 8:40:16 PM PST and meet again at 6:30 PM PST / 7:30 MST / 8:30 CST / 9:30 EST June 27th 1990 in Room 1 of the MUSUS conference facility.

Minutes submitted by Keith R. Frederick

## Treasurer's Report by Robert E. Clark, Treasurer

### May 1990

### June 1990

Bank Balance	\$6,893.51	4-30-90
<b>Income - May 1990</b>		
Dues:		(new/renew)
Student	25.00	0/1
General	140.00	3/0
Professional	0.00	0/0
Institutional	0.00	0/0
Other Income:		
CIS	25.81	
Library fees	66.00	
Publications	0.00	
PowerTools	0.00	
	-----	
Total Income:	\$ 256.81	
<b>Expenses - May 1990</b>		
Administrator:		
CIS	0.00	
Telephone	0.00	
Postage	0.00	
Photocopies	0.00	
Supplies	0.00	
Other:		
Bank charges	2.00	
Newsletter	385.75	
Mail from La Jolla	11.28	
Reimbursements	61.33	
	-----	
Total Expenses	\$ 460.36	
<b>Bank Balance</b>	<b>\$6,689.96</b>	<b>5-31-90</b>

Bank Balance	\$6,689.96	5-31-90
<b>Income - June 1990</b>		
Dues:		(new/renew)
Student	0.00	0/0
General	0.00	0/0
Professional	0.00	0/0
Institutional	0.00	0/0
Other Income:		
CIS	0.00	
Library fees	0.00	
Publications	0.00	
PowerTools	0.00	
	-----	
Total Income:	\$ 0.00	
<b>Expenses - June 1990</b>		
Administrator:		
CIS	0.00	
Telephone	0.00	
Postage	0.00	
Photocopies	0.00	
Supplies	0.00	
Other:		
Bank charges	2.00	
Newsletter	0.00	
Mail from La Jolla	5.35	
Reimbursements	44.00	
Del. Agent fee	70.00	
	-----	
Total Expenses	\$ 121.35	
<b>Bank Balance</b>	<b>\$6,568.61</b>	<b>6-30-90</b>

# From The Editor

by Tom Cattrall

Well, due to a lot of things coming up, this issue is about a month late. I didn't want to delay the article on file compression utilities any longer and writing it took quite a bit of my time. So, the moral is that newsletter issues will appear more promptly if plenty of material shows up in my mailbox (hint hint).

MODUS, the Modula-2 User's Society is making a comeback from a period of inactivity. See the letter on page 6. If you are interested in membership, write to the address given in the letter to get an application form. That, and the appearance of Modules & Definitions and the upcoming burst of activity with Modula-3, may be a sign that Modula-2 (and 3) are coming of age.

## Status of MUSUS

MUSUS (the Compuserve forum sponsored by USUS) has been undergoing changes and problems recently. There is a chance that by the time you read this, it will have been closed down.

Over the last couple of years, users of the P-system have wandered away from MUSUS and no real effort has been made to change and attract a new user base. To put out that effort requires a lot of time and energy, something that those of us in the computer field often have in short supply.

The forum has been reorganized somewhat with new message sections, announcements, and a new name: The Portable Programming Forum. It can now be reached by any of the following:

GO PASCAL, GO MODULA, or GO CODEPORT

As I write this, we are trying to work on new ideas for making the forum a gathering place for Pascal and Modula-2 as well as UCSD P-system users. The forum could use some fresh ideas and one or more new sysops with a lot of energy to aid the current staff. If anyone reading this thinks they might have something to offer, get in touch with me or with the forum sysop, Harry Baya.

## When Was Your Last File Backup?

Just last week two friends of mine reported losing all of their hard disks. On Thursday, the first person came up and said that his computer (a 286 clone) had lost touch with its hard disk and wouldn't boot. After talking with him, it sounds like there might be a chance that the problem is just that the battery that keeps the BIOS configuration has gone dead. So he was going to get another one and see if that helps. I haven't heard back yet.

The next day, I heard from a second person that has an Amiga with a harddisk. She wasn't too clear how it happened, but apparently she tried to format a floppy disk and ended up formatting her hard disk instead.

As it is for anyone that suddenly loses their hard disk, their first thought was "when was my last backup". The answer for both of them was "a long time ago".

The second reflex action upon hearing stories like this, is for anyone owning a computer system to want to drop everything and do a backup. So that a little good might come of these unfortunate stories, I'm passing them along in the hopes that others will be motivated to keep up to date backups and avoid losing a lot of data.

---

## Submission Guidelines

Submit articles to me at the address shown on the back cover. Electronic mail is probably best, disks next best, and paper copy is last. If your article has figures or diagrams, I can use encapsulated Postscript files in any of the disk formats listed below. If you can't produce encapsulated Postscript, then paper copy is probably the only practical method for submitting graphics.

You can send E-Mail to my Compuserve ID: 72767,622, or indirectly from internet: 72767.622@compuserve.com. For disks, I can read Sage/Stride/Pinnacle format disks. Also, any MS-DOS 5.25 or 3.5 disks, and 3.5" Amiga disks. If anyone wants to send Mac format disks I could probably get someone to translate them into something I can use. Whatever you send, please mark on the disk what format it is. That will save me a lot of guesswork.

Text should be plain ascii rather than a word processor file. It

can have carriage returns at the end of all lines or only at the ends of paragraphs. What you send doesn't have to look pretty. I will take care of that. My spelling checker will take care of spelling errors too. If you want special formatting use the following conventions:

1. Underline, put an underline character at each end of the section to underline.
2. \***Bold**\*, put a star at each end of the section to **bold**.
3. ^*Italics*^, put a caret at each end of the section to be set in *italics*.
4. ??Special requests??, such as ??box next paragraph?? should be surrounded with "?? ??".

**NewsLetter Editor :** Tom Cattrall  
Amity Software Inc.  
7600 Seawood Road SE  
Amity, Oregon 97101  
503/835-1613  
Compuserve : 72767,622  
Internet : 72767.622@compuserve.com

**NewsLetter Publisher :** Robert H. Geeslin Ed.D.

**USUS Board of Directors**

Henry Baumgarten	73447,2754
Tom Cattrall	72767,622
Frank Lawyer	72401,1417
Stephen Pickett	71016,1203
A. Robert Spitzer	75226,3643

**USUS Officers**

President:	Alex Kleider	71515,447
Treasurer:	Bob Clark	72747,3126
Secretary:	Keith Frederick	73760,3521

**USUS Staff**

Administrator:	Felix Bearden	74076,1715
Legal Advisor:	David R. Babb	72257,1162
MUSUS Sysop:	Harry Baya	76702,513

**USUS Membership Information**

Student Membership	\$ 30 / year
Regular Membership	\$ 45 / year
Professional Membership	\$ 100 / year

\$15 special handling outside USA, Canada, and Mexico.

Write to the La Jolla address to obtain a membership form.

**NewsLetter Publication Dates**

<u>Issue</u>	<u>Due Date For All Newsletter Material</u>
Sep / Oct 1990	September 1, 1990
Nov / Dec 1990	November 2, 1990
Jan / Feb 1990	January 4, 1991
Mar / Apr 1990	March 1, 1991

**USUS  
P.O. BOX 1148  
LA JOLLA, CA 92038**

**ADDRESS CORRECTION REQUESTED  
FIRST CLASS MAIL**

