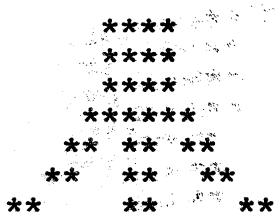


Atari Corp.
1196 Borregas Avenue
Sunnyvale, Ca. 94086

25 July 1986

INTRODUCTION TO GEM PROGRAMMING

T A B L E O F C O N T E N T S

INTRODUCTION TO GEM PROGRAMMING

1. INTRODUCTION

Thanks for your interest and support of the ST series of computers. Enclosed is the documentation and software package you have ordered. Read this section first to learn how to organize your documentation and get quickly on the road to ST product development.

1.1 WHO TO CONTACT IF YOU HAVE A PROBLEM:

Hardware problems	Randy Hain	(408)745-2466
Software questions	John Feagans	(408)745-4923
Package administration and equipment orders	Cindy Claveran	(408)745-2568

2. DOCUMENTATION IN THIS PACKAGE:

The documentation may seem overwhelming at first, but it is really easy to get started. All pages are pre-punched for a three ring binder and the sections you will most commonly access are The Guide (BIOS), GEMDOS, Volume 1--VDI, and Volume 2--AES. To start programming, the first files you should read are on the MicroEMACS disk on how to use the editor. This document contains info on compiling and linking your first program. Later you may refer to other documents in the package for detail on changing parameters of the compiler, linker, and other utilities.

1. Non-disclosure agreement
   Please sign and return to assure future updates.
2. Question and Answer Newsletters
   Commonly asked questions by new developers.
3. Introduction to GEM programming
4. A Hitchhiker's Guide to the BIOS
5. Atari GEMDOS Reference Manual
6. GEM Programmer's Guide, Vol. 1--VDI
7. GEM Programmer's Guide, Vol. 2--AES
8. GEM DOS Programmer's Guide
9. C Language Programming Guide
   (ignore the references for CP/M 68K)

The next series of documents are intended for the advanced developer who wants to exercise the lowest level of operating system:

10. Line-A Technical Reference Manual
11. Intelligent Keyboard (IKBD) Protocol

These next documents can be used by developers of add-ons and those persons who just want to know more detail about the machine:

12. Engineering Hardware Specification

13. Application Notes on the ACSI (DMA port)
14. Chip specifications:
    a. 6850 ACIA
    b. AY-3-8910 PSG
    c. 68901 MFP
    d. Programmable Sound Generator Manual
    e. WD1770 FDC
    f. 128K ROM Cartridge Schematic
    g. 520 ST Schematic

3. COMPUSERVE INFORMATION

We have included a Compuserve Starter Pak with $15 of free time. We hope that you have, or will purchase, a modem to use with your ST so you can join us in our special Developer's SIG on Compuserve. Follow the directions for logging in and type GO ATARIDEV. Type MI for membership and follow the directions. You will have access to restricted data libraries into which we are constantly uploading utilities and example programs. You may leave private messages for John Feagans at 70007,1072.

4. DISKS IN THIS PACKAGE

The ST development package consists of five disks. These disks will enable a developer to compile and link "C" applications or desk accessories for the Atari ST series.

1. C-Compiler.
2. Linker.
3. Utilities.
4. Resource Construction Set.
5. MicroEMACS.

WARNING: It is recommended that the developer make BACKUP COPIES of these disks and use the backups as backups ONLY!

The following is a list of files, programs, and other information that are included on these disks.

4.1 COMPILER

CP68.PRG	Three pass "C" compiler.
C068.PRG	
C168.PRG	
AS68.PRG	68000 asembler.
AS68INIT	
AS68SYMB.DAT	
BATCH.TTP	ST batch processing program.
RM.PRG	ST delete file program.
WAIT.PRG	ST WAIT to return to desktop.
C.BAT	Batch input file.

These files when included within a "C" program enable the use of the AES and VDI predefined messages, object types, GEMDOS definitions etc.

DEFINE.H	Miscellaneous definitions. Ex. NIL, TRUE...
GEMBIND.H	Do-it-yourself GEM binding kit.
GEMDEFS.H	Common GEM definitions.
VDIBIND.H	VDI external definitions.
TOSDEFS.H	TOS file attributes and error definitions.
OBDEFS.H	Common GEM OBJECT definitions.
OSBIND.H	"C" binding to GEMDOS and BIOS.
STDIO.H	"C" standard I/O file.
PORTAB.H	DRI recommended.
MACHINE.H	DRI recommended.
TADDR.H	Tree address offsets.
CTYPE.H	"C" typing functions.
ERRNO.H	DOS errors.
MATH.H	"C" header file.
OSIF.H	DOS functions.
OSIFERR.H	System errors.
SETJMP.H	"C" header file.
STRING.H	"C" header file.
LARGE.S	For compatibility with PC GEM

4.2 LINKER

LINK68.PRG	ST linker with overlay program.
LO68.PRG	ST simple linker.
RELMOD.PRG	ST relocation program.
BATCH.TTP	ST batch processing program.
RM.PRG	ST delete file program.
WAIT.PRG	ST WAIT to return to desktop.
LINKAP.BAT	Batch to link GEM application.
LINKACC.BAT	Batch to link GEM desk accessory.
CLINK.BAT	Batch to link C runtime.
LINKIO.BAT	Batch to link DOS applications.
ACCSTART.O	GEM accessory start file.
APSTART.O	GEM application start file.
GEMSTART.O	"C", VDI, AES library start files.
AESBIND	AES run time library.
VDIBIND	VDI run time library.
GEMLIB	"C" run time library.
OSBIND.O	GEMDOS trap routine.
LIBF	Floating point library.

4.3 UTILITIES

COMMAND.PRG	Command line interpreter.
DUMP.PRG	Hexadecimal file print utility.
FIND.PRG	File string locator utility.
HIGH.PRG	ST medium resolution program.
LOW.PRG	ST low resolution program
KERMIT.PRG	File transfer program.
NM68.PRG	Symbol table print utility.
SID.PRG	Symbolic Interactive Debugger
SIZE68.PRG	Program segment size utility.
APSTART.S	Source to application start.
ACCSTART.S	Source to desk accessory start.
GEMSTART.S	Source to "C", VDI, AES start.
ACSKEL.C	Example GEM desk accessory program.
APSKEL.C	Example GEM application program.
AR68.PRG	Library creation utility.

4.4 RESOURCE CONSTRUCTION SET

RCS.PRG	Resource construction set program.
RCS.RSC	File for RCS.
RCS.DFN	This enables you to see RCS.RSC as an example.
README	Ignore this unless you are converting alpha to beta RCS.
DEF2DFN.PRG	See above comment.
ICON	folder containing:
SE.PRG	Shape editor.
SE.RSC	
SHIC.DOC	Shape editor document.
ATBALL.C	
ATBALL.S	
ATBALL.SHP	
SDASM.PRG	
SDASM.S	
SDC.C	
SDC.PRG	
SDCST.S	
ICONA	folder containing:
ICED.PRG	Icon editor.
ICED.RSC	
RSCREAT	folder containing:
DOC	Documentation about resource create.
STCREATE.C	
STCREATE.INP	
RSC.O	
RSCRLIB.H	

4.5 MICRO EMACS

This disk contains the text editor you will use for constructing your programs and a number of examples which illustrate use of the AES and VDI.

CARDS	folder containing:
MAKECARD.BAT	
CARDS.C	example dialog box and field usage.
CARD.H	
CARDS.H	
CTYPE.H	
CTYPE.O	
CARDS.RSC	
DOCUMENT	folder containing:
NEWFORM.DOC	form_keybd( ) and form_button info.
DISK.TXT	how to address 1772 through DMA port.
DOODLE	folder containing:
LINKDO.BAT	example draw program
DOODLE.C	
DOORSC.C	
DOODLE.H	
DOS.H	
TREEADDR.H	
DOODLE.INP	
DOOSTART.O	
DOODLE.PRG	
DOODLE.RSC	
DOOSTART.S	
EMACS	folder containing:
ME.TTP	the editor.
TUTOR.DOC	how to...
TEXT1	
TEXT2	
TEXT3	
FORMDO	folder containing:
FRMTST.C	
FORM.DEF	
FORM.H	
FRMTST.PRG	
FORM.RSC	
MENU	folder containing:
MLINK.BAT	example menu program
MENTST.C	
MENTST.DEF	
MENTST.H	
MENTST.RSC	
TEMPLATE	folder containing:
TEMPLATE.C	put your application in the middle and have a functional GEM app.

5. HARDWARE REQUIREMENTS

The minimum system is 512k and one floppy disk. A recommended configuration is a 520 ST and a double sided drive or a 1040 ST. On a 520 ST a second drive is useful but on the 1040 ST you may want to use a RAM disk program available from several third party sources. For professional developers, the Atari hard disk is a good choice. For a choice of monitors we recommend that the serious developer have both monochrome and color monitors available for testing all possible configurations of user machines.

6. COMPILING A PROGRAM ON THE ST.

The three development disks have been arranged so that compiling and linking on the ST is a very simple process. The following is an example on how to compile and run the sample APSKEL.C program supplied on the compiler disk.

WARNING:
Remove the write protection tab from the compiler and linker disks before compiling and linking.

IMPORTANT NOTE:

The following instructions apply to a dual drive system. Some developers will have a single drive system. There may not be enough disk space on a single drive system to compile, assemble, and link. We urge you to get a second drive if your first drive is single-sided. If you have a double-sided drive, or a 1040, all necessary files should fit on one disk.

1)After making backups of the three development disks copy APSKEL.C and ACSKEL.C to a newly formated disk.

2)Place the compiler disk in drive A: and the disk containing APSKEL.C and ACSKEL.C in drive B:.

3)Select the BATCH.TTP program and open it.

4)Type "C B:APSKEL" in the open application dialog box and select "OK". At this point the screen will turn white and the batch program will begin compiling the APSKEL.C program. The compile is finished when the program askes for a carriage return. Hit carriage return to return to the desktop.

5)Replace the compiler disk with the linker disk and double click on the BATCH.TTP program. Type "LINKAP b:APSKEL" in the open application dialog box and select "OK". The screen will turn white while the batch program links the proper files together. The link is finished when the wait for carriage return prompt is displayed. Hit the return key to return to the desktop.

6)Now just double click on the file APSKEL.PRG on drive B:. The move, size, close, and full boxes are all active in this program. Selecting the close box returns you to the desktop.

To compile and link the desk accessory ACSKEL.C program do the following.

1)Follow the same procedure to compile and link the ACSKEL.C desk accessory that you moved to the new disk. Remember to use "LINKACC ACSKEL" when linking the desk accessory together.

2)Next rename the ACSKEL.PRG program to ACSKEL.ACC. CONTROL.ACC and VT52.ACC are the Control Panel and VT52 Emulator included on your system disk.

3)The final step is to transfer the ACSKEL.ACC accessory you have just created to your system disk and reboot the system. (Rebooting the system installs the accessory under "DESK" on the menu bar of the desktop.

4)To run, move the mouse to DESK on the menu bar and click on "Sample Accessory". If you resize the window you will see disk icons and other windows on the desktop. Selecting the close box will remove the desk accessory.

If you type C.BAT and LINKAP.BAT you will be able to see the proper order of instructions necessary to compile or link a program file. For more information on compiling and linking please consult the ATARI ST developers manuals.

The two remaining link batch files -- LINKIO and CLINK -- are used for creating either TOS or TTP type applications with GEMDOS and the C Run-time libraries respectively. No examples are provided for this discussion.

INCLUDE / LINK TABLE:

The following table represents the proper "include" files necessary to link to different portions of the TOS operating system.

SYSTEM	INCLUDE	LINK TO...
VDI	<define.h> <vdibind.h>	VDIBIND library
AES	<define.h> <gemdefs.h>	AESBIND library

	<obdefs.h>	
GEMDOS BIOS XBIOS	<define.h> <osbind.h>	OSBIND.O object
"C" library	<stdio.h> <math.h>	GEMLIB library LIBF library

All of the above libraries can be mixed depending on the needs of the developer.

6.1 TO CREATE A GEM APPLICATION:

You must link to APSTART.O as the first file in your link statement.

```
link68 [u,s] file.68k = apstart,program,vdibind,aesbind
```

6.2 TO CREATE A GEM DESK ACCESSORY

You must link to ACCSTART.O as the first file in your link statement.

```
link68 [u,s] file.68k = accstart,program,vdibind,aesbind
```

6.3 "C" RUN TIME LIBRARY

Using GEMSTART.O you MAY link to AESBIND or VDIBIND libraries and the run time library.

```
link68[u,s]file.68k=gemstart,program,
                    vdibind,aesbind,gemlib,libf
```

7. BATCH PROCESSING ON THE ST

7.1 BATCH.TTP

This is a batch program for the ST. To use, you must create a file with the extention .BAT.

To execute the C.BAT file you must open BATCH.TTP and enter "C PARM1" in the open application dialog box.

Where C.BAT is defined as:

```
cp68 %1.c %1.i
c068 %1.i %1.1 %1.2 %1.3 -f
rm %1.i
c168 %1.1 %1.2 %1.s
rm %1.1
rm %1.2
as68 -l -u %1.s
rm %1.s
wait
```

PARM1 is the C file to compile. (B:PARM1 to compile files on drive B:.) PARM1 is substituted for %1 in C.BAT. No file extension is needed -- it is included in C.BAT)

There is also a LINKAP.BAT file which is defined as:

```
link68 [u] %1.68k=apstart,%1,vdibind,aesbind
relmod %1
rm %1.68K
wait
```

Again, to execute just open BATCH.TTP and type "LINKAP APSKEL".

REMEMBER: The linker disk must contain link68.prg, rm.prg, batch.prg, link.bat, and relmod.prg.

7.2 RM.PRG

This is a delete file program necessary for the BATCH.TTP program so that files can be deleted from drive B:.

7.3 WAIT.PRG

This program waits for a carriage return before

continuing.

7.4 RELMOD.PRG

Converts the output file from LO68 or LINK68 into a form that is executable by GEMDOS.

The full form of the commandline is:

RELMOD [-]inputfil[.68K] [outputfile[.PRG]] ...

(stuff between brackets is optional)

If the input file has no extension, ".68K" is automatically appended. If there is no output filename specified, the name of the input file (with a ".PRG" appended) is assumed. If an output file IS specified, it has the default extension of ".PRG".

If the input filename begins with a dash ('-'), the dash is ignored, except for the fact that it indicates an input filename. (If that's confusing, don't worry -- read on).

EXAMPLES

RELMOD FOO

Will look for a file called "FOO.68K", and produce a file called "FOO.PRG".

RELMOD BAR -BAZ.ZIP

Will look for a file called "BAR.68K", and produce a file called "BAR.PRG". Similarly, it will look for "BAZ.ZIP" and produce "BAZ.PRG".

8. ALLOCATING MORE STACK SPACE

Stack space for applications and desk accessories has been preallocated to 1K. We realize that for some application this may not be enough stack room. For this reason we have included the source files APSTART.S and ACCSTART.S on the Utilities disk. You now can make appropriate changes to these files to create the necessary environment for your application or desk accessory. For C-runtime library applications, check the source for gemstart.s and the comments for increasing stack space.

9. GLOBAL VARIABLE NAMES TO AVOID

Certain global variable names are present in the binding files which you will link to. The linker takes the most recent definition of anything as being the real definition. An example of this would be if your program has an array names max[] and you link to the VDIBIND library. The linker will confuse your max[] array with the function max( ) in the VDIBIND. With that in mind, also avoid the following names associated with the named modules.

apstart.o
accstart.o
---
crystal
ctrl_cnts

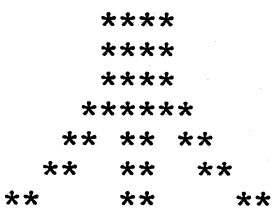
gemstart.o

_main
_exit
_start
__cpmrv
_base
sw
_sovf
brk
__BDOS
blkfill
index
strchr
__pname
__tname
__lname
__xeof

osbind.o
---
gemdos
bios
xbios

aesbind
---
c
control
global
int_in
int_out
addr_in
addr_out
gl_apid
ad_c

vdibind
---
i_ptr
i_ptr2
m_lptr2
mul_div
MUL_DIV
smul_div
SMUL_DIV
umul_div
UMUL_DIV
gsx1
gsx2
iioff
iooff
pioff
pooff
vdi
vec_len
max



Atari Corp.
1196 Borregas Avenue
Sunnyvale, Ca. 94086

1 July 1986

Errata

GEM PROGRAMMER'S GUIDE VOLUME 1: VDI

3-16 vst_load_fonts
is: WORD additional;
change: delete this reference.

5-17 Set character height
Note: This is a table of magic numbers which select the internal system fonts:

4	6x6
6	8x8
13	8x16

5-36 Set user-defined fill pattern
is: Note that the writing mode must be set to replace (mode 1), when using a multiplane fill pattern.
add: ,and the foreground color must be set to 1.

6-14 v_get_pixel
is: (handle,x,y,pel,index)
change: (handle,x,y,&pel,&index)

7-15 Input string, request mode
additional comment: This will not work if you are using the AES! In all request modes, it will just return with no input.

7-17 Input string, request mode
(see comment for 7-15)

8-5 Inquire color representation
useful ST info for RGB color conversion:
from hardware to VDI: VDI=(HDW*125)+62
from VDI to hardware: HDW=VDI/142

9-19 Escape 16
is: This escape returns ... tablet, mouse, joystick, or similar device.
delete: mouse, joystick, or similar device.

Appendix D Standard keyboard

Please refer to IKBD document for this information.

Appendix F Character sets
Please refer to appendices of BASIC and LOGO manuals packed with your ST.

GEM PROGRAMMER'S GUIDE VOLUME 2: AES

4-8 AC_OPEN
should be: word0 = 40
word4 = me_rmenuid

4-8 AC_CLOSE
should be: word4 = me_raccmenid

6-5 Object library data structures
additional info: For type G_IMAGE, the ob_spec is a pointer to graphic data.

6-8 Iconblk structure
additional info: The word containing ib_char has the foreground+background color encoded in the high byte. Here is how to decode it--

```
fg = (ch>>12) & 0x000f
bg = (ch>>8) & 0x000f
ch &= 0x00ff
```

6-20 OBJC_DRAW
additional info: This call automatically turns off the mouse while drawing and restores its state when finished.

6-27 OBJC_EDIT
The sample call shows this function as having 6 parameters--delete &ob_ednewidx to generate the proper call with 5 parameters.

7-13 FORM_DO
is: ... -1 if the form ...
should be:... 0 if the form ...

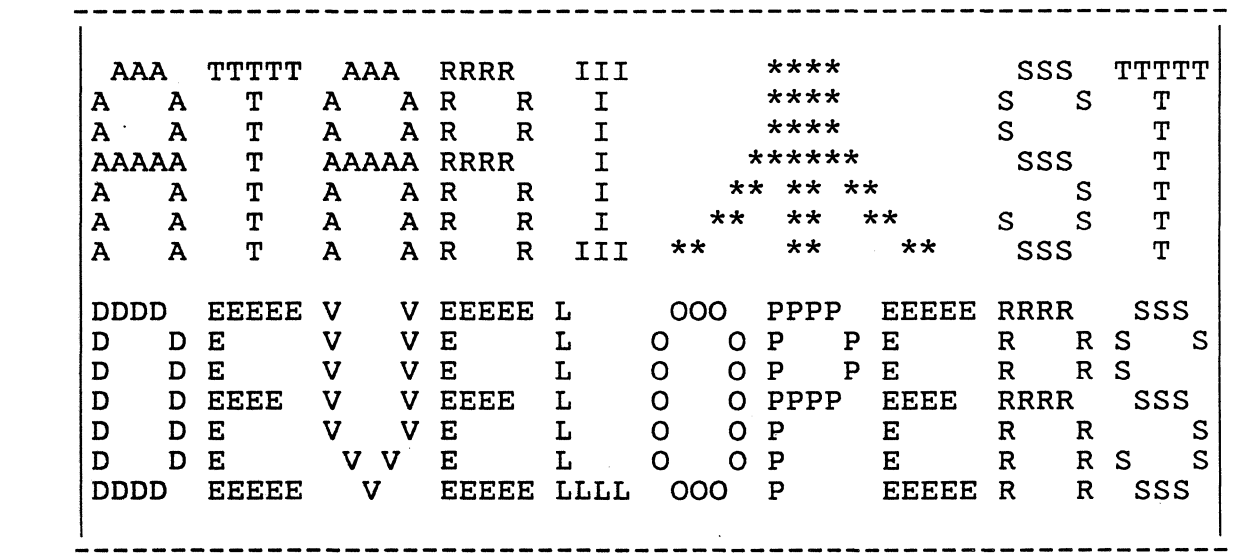
7-15 FORM_DIAL
Needs to have nine parameters:

```
fo_diflag
fo_dix
fo_diy
fo_diw
fo_dih
fo_clipx
fo_clipy
fo_clipw
fo_cliph
```

8-14 GRAF_HANDLE
Note: gr_hwbox and gr_hhbox are the width and height of the boxes containing mover, sizer, closer, etc. icons in a window.

10-5 FSEL_INPUT
Note: The value of fs_iexbutton will be -1 if there is a file selection error.

11-22 WIND_SET
Note: The string used for the name is not stored in the AES. A pointer to the name is all that is stored. If you wish the name to be saved then keep a copy in your data space.



Question and Answer Bulletin February 1986

1196 Borregas Ave., Sunnyvale, Ca. 94086

Here are the latest questions from the Atari developers mailbag as answered by John Feagans, Director of Software Technology. Leave questions on Compuserve for PIN 70007,1072 or GO PCS57 for Atari developer SIG information.

Q: How can I have a different background color for text. There does not seem to be anything I can set to make this happen.

A: A way we recommend to make this happen is to do a rectangle fill of the area behind the text then use v_gtext with a transparent writing mode.

Q: I have a GEM application that I am porting from the IBM PC. Everything compiles O.K. but I get a number of undefined symbols in the link step. Some of these symbols are dos_free, dos_gdrive, dos_lseek, and dos_open.

A: The problem here is some differences in name between what you are using, what is described in the GEMDOS spec, and what is actually in the bindings. Here is a table which may explain what is happening:

Your code	GEM DOS manual	osbind.h
dos_free	d_free	Dfree
dos_gdrive	d_getdrv	Dgetdrv
dos_lseek	f_seek	Fseek
dos_open	d_open	Fopen

Q: From the VDI there doesn't seem to be a call to set pixels. How do I do it and be compatible with all resolutions?

A: You can do it by drawing a 1 pixel wide line with a length of 1 pixel. There is also a line-A call for setting pixels. The latter method is much faster, however, it does not clip.

Q: I just installed my ROMs and the system doesn't come up. It is just a white screen. It is like this every time I reset. Oh...there it is...what happened?

A: If you have just installed your ROMs and do not have a disk drive attached, the TOS is going to spend considerable time attempting to read some files from disk. It is looking for desk accessories, an auto folder which may contain drivers to load, and a desktop.inf file created if you saved a desktop. If you want to shorten the power-up process, then have a disk attached with a valid floppy disk in the drive.

Q: I just ported my IBM PC version and have the symbol UMULDIV undefined.

A: VDIBIND defines MUL_DIV, mul_div, and umul_div, but the symbol UMUL_DIV is missing. If you are programming in C perhaps you could #define it in a header file. In assembler, you could equate UMUL_DIV to umul_div.

Q: Can I boot another operating system if I have TOS in ROM?

A: Yes. The system has been designed so that a disk with a boot sector can have another operating system which can be loaded in. Another way to load another system would be to put a driver in the auto folder on the system disk. A third interesting method would be to make a desk accessory which loads the new system. It is also possible for a program to perform an extended bios call PUNTAES which frees up the memory used by the AES and desktop while preserving the bios,dos, and VDI.

Q: How can I load in code from BASIC? Where is it put? How do I allocate space for it?

A: To use the BLOAD function you must first allocate a string array big enough to hold your code. Next use the varptr function to get the address of the storage area. Use BLOAD with this address and the CALL function to execute your code. One final point--make sure the code is relocatable and completely relative as BLOAD does no fix-ups.

Q: How are the icon outlines dragged on the desktop and what is the shape?

A: Icons are dragged on the desktop by moving a polyline XOR box. The vertices are a fixed set that resemble the edges of the mask data for the icon.

Q: How do I stop BASIC from drawing windows on my output screen?

A: Windows are redrawn when BASIC calls a routine in the event library. Some examples of these routines are evnt_multi, evnt_keyboard, and evnt_button. It is possible to stop the evnt_multi calls because there is a BASIC system table called SYSTAB. Check the documentation for particulars. At an offset of +24 there is a variable called GEMFLAG. A POKE of 1 in this location turns the event calls off and a POKE 0 turns the event calls on.

Q: When I double click on AS68 from the desktop, the screen turns white and immediately returns to the desktop. How do I type in the name of the file to assemble?

A: It is posible to run AS68 from the desktop. The procedure is to install it as a TTP type, (TOS takes parameters) and save desktop, or show-info and rename it as a .TTP. Next time you double click on AS68, a dialog box for parameters will come up. You can then type "-u -l myfile.s" and click on OK to run the program. Object files fill be generated and control returns to the desktop upon completion of the assembly.

Q: I have a program where I am drawing my own dialog boxes and menu by constructing the object tree and calling OBJC_DRAW. The problem is that when I draw the dialog box, the menu pops down. I have tried everything to prevent this. What is happening?

A: We suspect what is happening here is that you may be passing the address of the menu rather than the dialog box. If you have the level set to MAXDEPTH, everything including the dialog box over a popped down menu may be drawn. Double check what you are passing to the objc_draw.

Q: I have a dialog box with an editable field. When I try to put string to initialize the data, everything is messed up. It seems like the pe_text pointer is pointing to the wrong place!

A: The OBSPEC field of the editable object points to a TEDINFO structure. In this structure there are three pointers: in order they are 1) to a string containing the actual text, 2) to the template, and 3) to a character validation field. Beyond getting the proper pointer, another problem you may encounter is that the string is a fixed length. You may be moving data into it which is longer than was created at the time you defined the text in the resource

construction set.

Q: I am running the batch program and get a bus error after c0.prg has been running a little while compiling my program. Could this be a bug in the compiler.

A: A bug is always possible but here are a few other things to suspect when a problem like this occurs. The batch program will abend if it cannot find a specified program to load. You may also have an include file with something illegal. That might explain why things proceed through the pre-processor in abort in the first compile step. Another thing to check is if you are running out of space on the disk from which the compiler is running. Each step of the compile creates some intermediate files which require space. It is possible to specify to the compiler which drive to which it writes the intermediate files.

Q: I am transferring files from my IBM PC to the ST using kermit. ASCII files come over o.k. but binary files are messed up.

A: When you transfer binary files you must tell the server to SET FILE TYPE BIN. You must also tell kermit on the ST to gi (get image) or si (send image).

Q: I am using ICED.PRG to design my icons for use in my resource file. How do I load these icons into the RCS?

A: As an example, let us consider that you are in the RCS and have created a dialog box. Drag the icon type from the parts box into your work area. Select the work area and select the icon you have just moved down. Bring down the menu under the file heading and select the open. At this point the open item will not load a new resource but will load in and link the data for the icon file you select in the file selector.

Q: What documentation errors do you know about in the Hitchhiker's Guide to the BIOS?

A: The current Hitchhiker's Guide shows the initial PC being set from $FC0000 and the initial SP from $FC0004. These values are reversed.

Hitchhiker's Guide is wrong in two places about the GEMDOS call $20 to set supervisor mode. The flag to merely test the current mode is $1 and not the $FFFFFFFF specified. The latter will crash the system. Second, the returned value from this function is $0 for user mode and $FF for supervisor mode, not $0 and $1 as documented.

Q: To get started writing my program I need some more example programs. Where can I find some?

A: We recommend that you get onto Compuserve and take a look at the Atari 16-bit SIG and also the Atari developer's SIG. For the latter, type GO PCS57, and read the membership information. We are constantly uploading example programs into the data libraries as well as other developers. These programs are available for download to your system.

Q: I am trying to redirect standard output by using the Fforce function. Gemdos is checking the handle for a number less than 6 and reporting an error. How do I redirect the console?

A: The new handle which you pass this function must be greater than 6 or less than 0. Values of -1, -2, -3 correspond to CON, AUX and PRN.

Q: How can I make a listing of my assembler program?

A: There is a way to make AS68 write a listing file to disk. To do this you must add the parameter -P and specify the output file name following the input file name with >filename. Here is an example which may work for you.

```
AS68 -P -L -U file.s >list.txt
```

You can do this either from a batch file or you may install AS68 as a TTP and do this from the desktop.

Q: I am trying to compile the simple example C program that just uses printf to output "hello world". I keep getting undefined symbols during the compile and the object will not link. What is wrong?

A: The most likely cause of the compile errors is that you are missing the header file STDIO.H. Be sure that at the beginning of your source file that you put the following statement in:

```
#include "stdio.h"
```

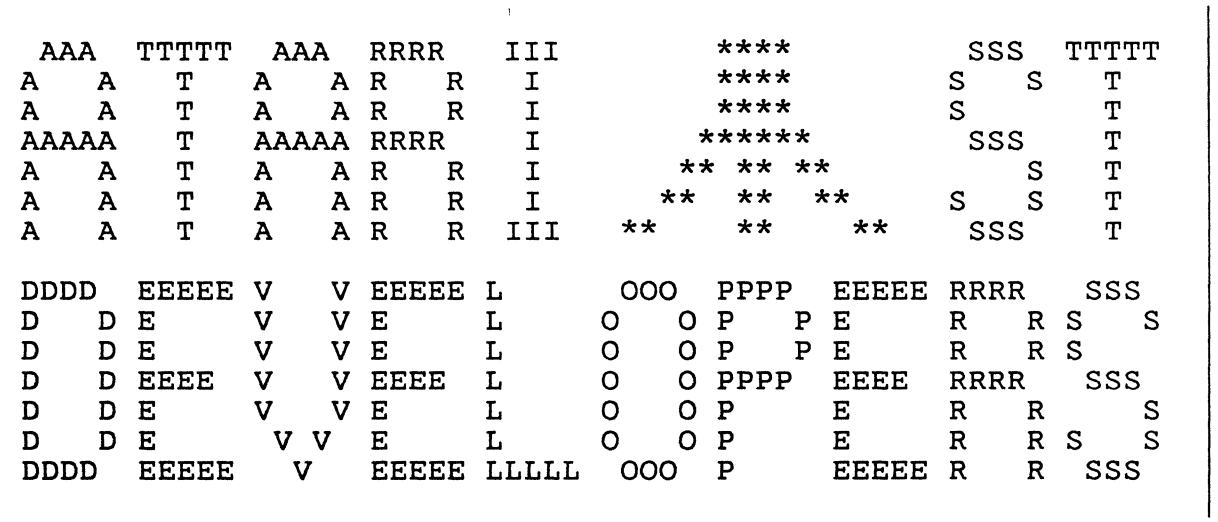
Problems with the link may be caused by not specifying all the files that are required, or placing the files in the correct order. Here is the bare minimum you should have for a C program that is not a GEM application:

```
gemstart.o,(your object),gemlib,libf
```

Q: Why is the return key not returned by the getchar function?

A: When Digital Research wrote the 68k run time libraries for the Alcyon C compiler they used the Unix new line definition for terminating input. In a generic system the return key would generate the sequence CR LF. The run time libraries throw away the CR and count on the LF coming

through. The only way to get around this problem is to use the unfiltered bios or dos input from console. The scanf function is similarly afflicted. Instead of forcing the user to terminate input with control-J, build a buffer of characters terminated by a null character using the bios and bdos. Pass this string to sscanf which acts only on the string and does no keyboard input.



Question and Answer Bulletin March 1986

1196 Borregas Ave., Sunnyvale, Ca. 94086

Here are the latest questions from the Atari developers mailbag as answered by John Feagans, Director of Software Technology. Leave questions on Compuserve for PIN 70007,1072 or GO PCS57 for Atari developer SIG information.

Q: How can I print the special characters in the Atari character set between codes 0 and 31?

A: You can use either the bios or the VDI. Device 5 on Bconout is a raw character output device that does not do an interpretation of character codes. You can alternate between output to console and this device to do cursor positioning and new lines. The VDI call to use is v_gtext. This routine accepts character indices between 0 and 255. You should be careful in using these character codes because they are only defined in the three system fonts: 6 x 6, 8 x 8, and 8 x 16. The VDI specification does not require loadable fonts to have characters in these positions.

*

Q: When I dump a file to my printer, the first line prints across and then printing hangs up at the right margin. What is going on?

A: Check if your printer has a switch select position for generating a line feed after a carriage return is received. Some software will generate only a return and count on the Bios to generate the line-feed--TOS does not do this. If your printer does not have this facility, some programs such as STWriter and 1st Word have configuration files which

define the end of line sequence which you can set for proper operation with your printer.

*

Q: Why don't the numeric pad and the cursor keys work in the VT 52 emulator?

A: The layout of the ST keyboard is patterned after the DEC VT 220. The VT 52 is an older but still popular terminal. The Enter key is half size and the cursor keys are in the positions of *, -, + and half of the Enter key on the numeric pad. Look for an update of the desk accessory in the future--meanwhile here is a brief summary of cursor escapes. The full list is contained in the Hitchhiker's Guide.

ESC A	Cursor up
ESC B	Cursor down
ESC C	Cursor forward
ESC D	Cursor backward

*

Q: I am writing a terminal emulator program. How can I smooth scroll the screen?

A: Refer to section 6 of the VDI manual on raster operations. You can use the copy raster functions, vro_cpyfm and vrt_cpyfm, to move blocks of data on the screen. You may also want to double buffer the screen. Furthermore you can move the pointer to the start of display memory.

*

Q: My C program compiles fine but I get the following undefined symbols in the link68 step -- getscanf, printf, fpmult, and iob. I am linking with GEMLIB and LIBF.

A: There is no symbol getscanf. Perhaps you mean getchar and scanf? To resolve the other undefined symbols you must be sure to #include stdio.h and also be sure the start file you link with is gemstart.o

*

Q: How can I write boot sector information? Is there an entry in the DOS or BIOS for doing this?

A: Use the protobt( ) extended BIOS call to create a boot sector. The disktype parameter should be 2 or 3 for 1 or 2 sided media respectively. The serialno parameter should be a random number. The execflag should be zero unless the prototyping buffer contains code (such as a copy of the loader) that you want executed when the disk is booted. Write the boot sector protyped in the buffer to track 0, side 0, sector 1 of the new disk. Use the extended bios function flopwr( ).

*

Q: I am using funtion 7 of line-A. I am trying to use a pattern but I don't know what to put in the variable pattern_mask.
pattern_address seems to be set up correctly because my pattern seems to almost come through.

A: The pattern_mask variable is actually the length of the pattern. That should be what will make your pattern work.

*

Q: How can I put an icon on the desktop?

A: There is no way to place an icon other than those built into the system onto the desktop. The desktop is an application program and it was not designed to interface the other way from GEM AES. If you write your own application and create a new desktop, there is nothing to prevent you from using your own icons. A good example of this is the Resource Construction Set. If you want to represent a desk accessory by an icon you could open a nearly invisible window and print the icon on it. The limitation on this method would be that you could not drag other icons to it and be able to tell what was moved. If you want to be able to drag things to your program you must use the system icons. The procedure is to build a loadable driver and change the desktop.inf to contain the proper label information.

*

Q: In the resource construction set, what is the difference between a free-tree and a dialog box?

A: A dialog box is a fixed format object list whose net result when you draw it is obviously--a dialog box. Its format is designed to work with the form library. The free tree is used to define other object lists. Of course you could define a structure for a dialog box with it--but you could also define a dialog box that had a different border and worked with your own form handler.

*

Q: P_exec has two parameters which the documentation does make clear. They are the pointer to an environment string and a pointer to a command tail. What are these strings and how do I construct them?

A: The command tail is an array of strings terminated by a zero. If you do not use it then point to two consecutive zeros. The command tail is used by the run application dialog box for a TTP application. What you type in is inserted in the command tail. The environment string is something that is inherited by a process. It is just a simple string.

*

Q: How can I display two dialog boxes at the same time?

A: The form library routines are designed for handling only one dialog box at a time. If you want to simultaneously display two of them then you will have to write your own form handler. They can be displayed using the object draw commands. You may also want to try form_button and form_keyboard which do not grab control of the screen. Look for documentation on these functions in this space soon.

*

Q: My menus seem to have so many entries that they are of a certain size which prevents the AES from restoring what was under them when they pop back. What are the maximum size menus that I can have?

A: The total width times height in pixels cannot exceed 8k bytes for hi-res, 4k bytes for medioum res, and

2k bytes for low res. Assuming and 8 x 8 font or 8 bytes per character, a maximum for low res might be 256 characters--16 entries of 16 characters each.

*

Q: What is the mechanism by which ST Writer reads the directory and prints the file names?

A: STWriter uses the F_sfirst and F_snext to read the directory entries. Furthermore, it uses some special data at the beginning of each file to indicate that it is an STWriter file.

*

Q: How do I get the command tail from my C program?

A: TOS passes your program a count of parameters and a pointer to the array. Here is an example of how to declare them in your program:

```
main(argc,argv)
     int  argc;
     char *argv[];
     {
      /*  your program  */
     }
```

*

Q: Are there any variables around so I can get the position of the cursor?

A: Since the orignal line-a variables offsets were frozen, we decided to expand the list. At a negative offset from the line a address there are two variables. v_cur_cx is at -14 words and v_cur_cy is at -13 words.

*

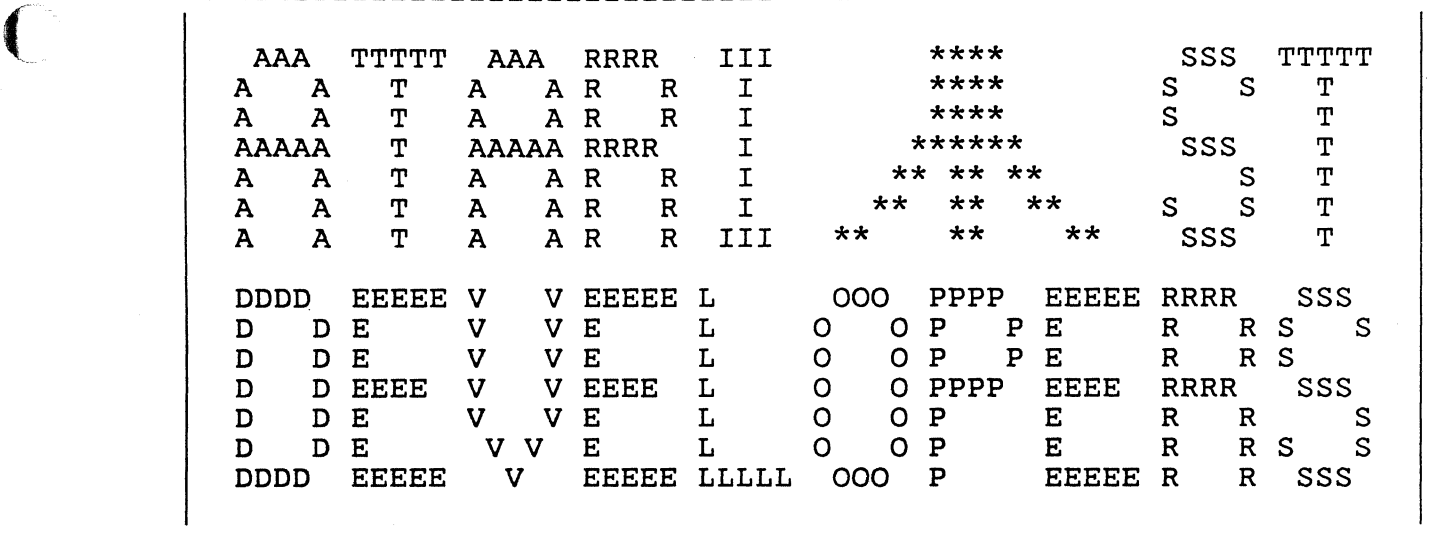
Q: How can I ring a bell without going through the trouble of do_sound?

A: Send a control-G to console output: Bconout(2,7).

*

Q: How do I access the data in an editable field from a dialog box that was created by the Resource Construction Set?

A: When you create the dialog box, use the Name function to give both the tree and the object a name which you can use later. Have the RCS create a .h file for your resource and #include it in your program. Define a pointer of type OBJECT and use rsc_gaddr with this pointer and the label for your tree to get the address of the start of the object list for the dialog. Later, you can index [label of object].ob_spec to get the pointer to the tedinfo structure. Do one more indirection to get the pointer to the actual string.



Question and Answer Bulletin April 1986

1196 Borregas Ave., Sunnyvale, Ca. 94086

Here are the latest questions from the Atari developers mailbag as answered by John Feagans, Director of Software Technology. Leave questions on Compuserve for PIN 70007,1072 or GO PCS57 for Atari developer SIG information.

This months contents

---

1. Corrections

In the March 1986 issue I mis-stated a description of the real VT-52 keyboard. This device actually does have a numeric pad and cursor keys. The difference between the numeric pad that it has and the VT-220 style that the ST has is that the cursor keys are located in the positions of the *, -, +, and Enter key on the numeric pad. The enter key itself is split into two smaller keys.

---

2. BIOS

*

Q: How do I make the buffer larger for communication over the MIDI port?

A: Use the xbios call Iorec(MIDI) to get the address of a structure which contains a pointer to the existing buffer. Change it to point to

your new and bigger buffer. Be sure to restore things when you exit your program for the benefit of the application that follows yours.

---

3. DOS

*

Q: How do I get the address of the DTA buffer for use with the Fsfirst( ) command?

A: There are two ways that you can do this: 1) Allocate a buffer yourself and use the Fsetdta(ptr) to tell the system about it, or 2) ptr=Fgetdta to use the one already defined.

*

Q: I am running out of stack space when I link with apstart.o. Where do I edit to increase the size?

A: Look for the label ustk in apstart.s. Directly preceding it you will find a statement ".ds.l 256". This statement allocates 256 * 4 words, or about 1k of stack. To increase the stack to 8k, make it ".ds.l 2048". Remember that declaring local arrays in C will eat more stack space--allocate accordingly. To make the apstart.o you must assemble your new source by as68 -l -u apstart.s.

---

4. VDI

*

Q: How do I transfer a color image from memory to the screen without copying the background around the image?

A: You need to construct a mono-plane mask with zeros where the image in memory will be transparent and ones where the image in memory will occupy space. Use the copy raster transparent (vrt_cpyfm) with transparent mode to cut a hole into the screen. Second use vro_cpyfm with "or" mode 7 to move the color image to the screen. See section 6 of the VDI manual for further details.

---

5. AES

*

Q: How do I hide and show the mouse?

A: Use the graf_mouse call. A useful thing to include in a header file are the following definitions:

```
#define HIDE_MOUSE
        graf_mouse(256,0x0L)

#define SHOW_MOUSE
        graf_mouse(257,0x0L)
```

*

Q: In the mentst.c example program, why are so many parameters in the evnt_multi call zeroed out?

A: The reason for this is that in this example program we are only interested in returning if there was mouse event. We are not interested in keyboard, timer, or button events, hence we have a dummy pointer as place holders in those parameter positions of the evnt_multi.

---

6. Desktop

*

Q: How does the AUTO folder work?

A: If you create a folder named AUTO, the system looks for it on power-up boot and will execute any programs it finds inside. If there is more than one program, the programs will be executed in the order that they were created in the folder. These programs are named *.prg but it is not possible to have an auto-starting GEM application. However, the programs may use the VDI, DOS, and BIOS. A good example to try is to place STWriter inside an AUTO folder and it will start running on power-up. When you exit STWriter, initialization resumes and execution brings up the desktop. AUTO folders are generally used for drivers such as those for the hard-disk and printers.

Q: How can you get the name of the data file that initiated a program with an installed document type?

A: For programs that have undergone the install application option at the desktop, it is easy to get the name of the document. The desktop program places the file name string in the command tail. The command tail starts 80 hex into the base page. You can get this information by accessing the base page directly, or by main(argv,argc) from a C program that uses gemstart.o.

---

7. Accessories

---

8. BASIC

*

Q: The accountants in my 50 million dollar company do not like my basic programs as large numbers will end up with a few pennies added to the end. What can I do?

A: You are seeing the effect of a single precision scientific math package. The maximum number of significant digits that can be stored is 6 to 7. Decimal fractions cannot be accurately represented by a binary floating point structure. To overcome the latter limitation, you can store all numbers as pennies, only inserting the decimal point when inputting or outputting them. To gain extra significant digits you can save each number as an integer whole and fractional part. To really gain significant digits you might consider writing some math routines which operate and store numbers in strings.

---

9. LOGO

*

Q: I keep getting an edit buffer full error. I have tried turning off buffered graphics and not loading desk accessories to get more space. Will getting ROMs help me?

A: The edit buffer is a fixed size. You need to break up your program into smaller

segments and load them in. Getting the ROMs would only give you more program storage and would not increase the size of the edit buffer.

---

10. Development Tools

*

Q: How do I get the C-Compiler to make a listing of my program with line numbers

A: The Alcyon C currently supplied in the developers kit does not have an option for generating a listing. Errors that are reported can be found by using the editor to step to that line in your source. I hope that someone out there will write a print/header and line number generating program to share with us in the data libraries in the developers SIG.

*

Q: I am including stdio.h and linking with gemlib and libf, but I still have undefined symbols etoa and ftoa in the link step when I use printf.

A: The order of the library files in the link are important. Make sure that libf is listed after gemlib. If you have any doubts you can also list it as libf gemlib libf.

*

Q: I have a RAM disk program. I would like to speed up my links by putting all the linker created temporary files on the RAM disk. How do I command the linker to do this?

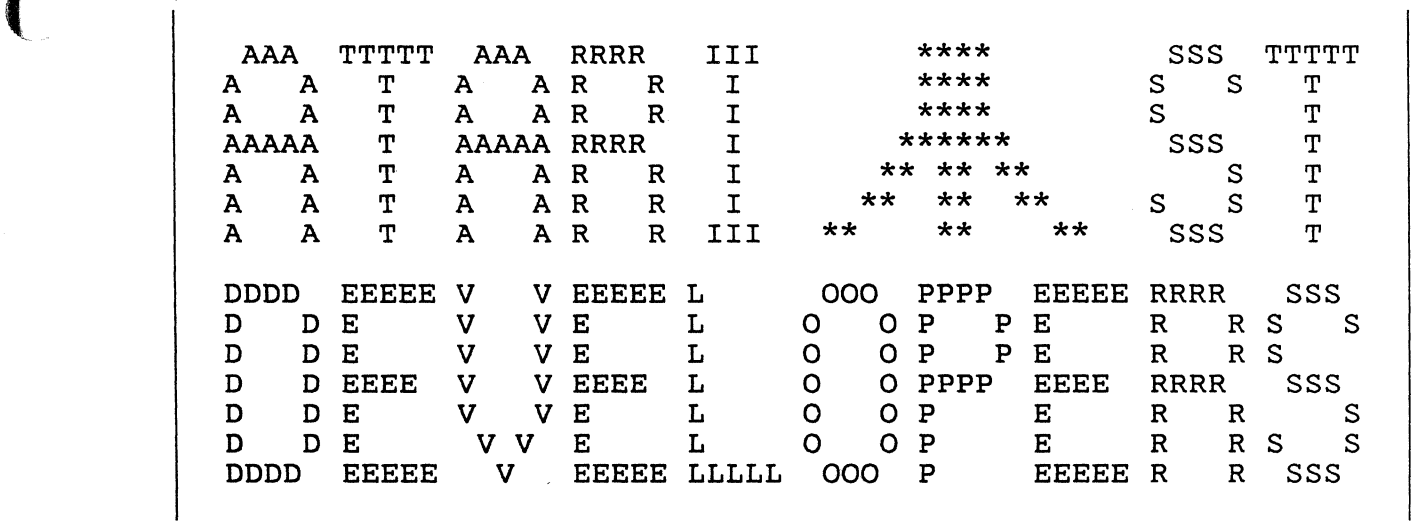
A: For LO68, -F d: The -F option specifies the drive on which temporary files are created. The variable d: is the drive designation. For LINK68 TEM[d:].

---

11. New On Compuserve

In data library 7 (for registered Atari Developers only) in the Atari Developers SIG on Compuserve, the following files are new this month:

WRITE.O	C RTL fix
LSTOUT.O	" "
CC.BAT	" "
GEMDOS.DOC	New improved.
BCD.DOC	Math package.
GETDOB.C	" "
MATH.OBJ	" "
PRINTD.C	" "
RELMOD.C	New improved.
SLDSET.PRG	Image service.
SLIDE.DOC	" "
SNPSHT.TOS	" "
SNPSVE.PRG	" "
TEMPLA.C	GEM skeleton.
FED.PRG	font editor.
FEDFIX.TTP	" "
FEDFIX.C	" "
FED.DOC	" "
FRDME.DOC	" "
FONTUSE.DOC	" "
FONTUSE.S	" "
STHDW.DOC	New document.
LARGE.S	Porting aids.
QA2.DOC	Previous Q/A.



Question and Answer Bulletin May 1986

1196 Borregas Ave., Sunnyvale, Ca. 94086

Here are the latest questions from the Atari developers mailbag as answered by John Feagans, Director of Software Technology. Leave questions on Compuserve for PIN 70007,1072 or GO PCS57 for Atari developer SIG information.

This months contents

---

1. Corrections

In this issue we return to the single column format. In trying to make better use of space available by going to double-column format, many readers could not follow the columns when reading this newsletter on-line. Remember, your suggestions are always welcome as are your questions.

2. BIOS

*

Q: Are nulls deleted by TOS when communicating over the RS-232 port?

A: When you write a communications package it is important to use the raw-unprocessed BIOS conin instead of a higher level which may intercept control characters including control-C. There is no reason for nulls to be deleted.

*

Q: How do Europeans access the special characters in the Atari charcter set?

A: The Atari character set is standard in all models of the ST around the world. There are special versions of TOS for each country which contain resources translated to the local language and keyboard translation tables which decode the scan codes to the character codes. The scan codes are always the same for the same key positions because the IKBD controller is the same in all models. The legends on the keys in those positions may vary by country. Some countries have a dead-key function driver which is in the auto folder at boot time. This driver installs in Trap 13 and monitors Bconin for the pressing of an accent mark. When the next vowel is pressed, the correct character code is generated for the application software.

---

3. DOS

*

Q: How can I prevent alert boxes from printing when I get a DOS error?

A: See the BIOS function setexc(vecnum,vec). Vecnum is the number of the vector to get or set. Vec is the address to setup in the vector slot else a get is performed if the value is -1L. Vecnum $101 is the critical error handler. Insert your own routine there and the alert boxes will cease printing. Be sure to restore the old vector when you are finished.

---

4. VDI

*

Q: What is the 6 x 6 font used for?

A: The smallest font is used for labeling icons in medium and high resolution.

5. AES

*

Q: How can I access the menu from a desk accessory?

A: There is no way that this can be done. Menu messages are given to the desktop program (or currently running application that generated the menu. You could generate your own pseudo menu in your window but you would have to do everything yourself.

*

Q: Do you need more than one resource file in your program so that multiple resolutions are handled correctly?

A: It is only necessary to design one resource. The RCS stores all the object positions and width and height in terms of byte alligned character coordinates. When the file is loaded by the rsrc_load function, the x and width parameters are multiplied by the current global character width and the y and the heigh by the current global character height. Thus all the numbers are adjusted to the current resolution you are working in.

---

6. LOGO

*

Q: Are SETWRITE and SETREAD working?

A: These functions are currently un-implemented in ST Logo.

*

Q: How can a user define his own fill pattern?

A: The example in the Logo manual has a typo. You must define 16 and not 15 values as shown to make the fill pattern work. The last value reads "1280" and should be "128 0".

---

7. Development Tools

*

Q: How do I increase the stack size in gemstart?

A: In the file gemstart.s you will find a comment about allocating 1k of stack space next to a add.l #500,d0. $100 is alocated for the base page and $400 (1k bytes) is allocated for the stack. To increase the stack to 2k, make the number $900. Assemble this module and link the resulting object. There is an interesting point to be made about the C run time library function malloc here. malloc allocates space off this stack. A new and improved gemstart will be available soon to allow you to malloc more than 1k. Keep in mind that this malloc is not Malloc

the DOS function. The latter Malloc works fine if you have previously done an Mshrink to return unused memory to the operating system.

*

Q: My program to spawn processes get the message "Not enough free store." What is happening?

A: Sounds like you did not do an Mshrink to give memory back to the DOS when your first program started running. (This was verified later in the thread.)

*

Q: How do you get wild cards to work in Kermit?

A: You must enclose the file name in quotes.

*

Q: How do you make the menu box below the title in the RCS larger?

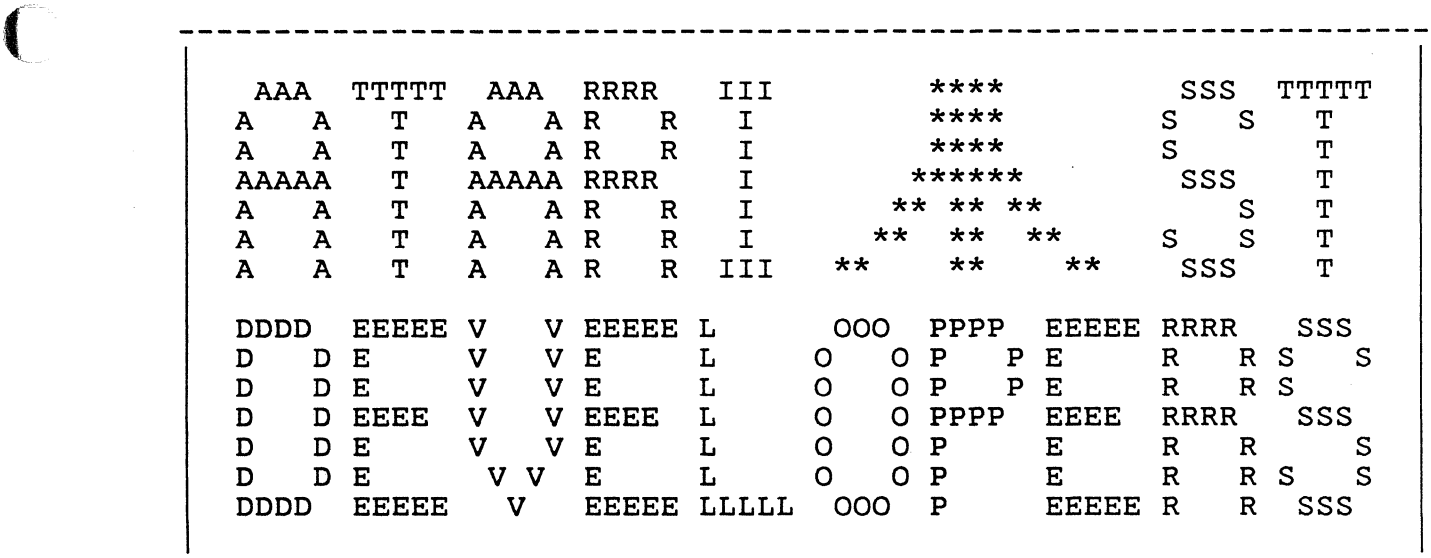
A: Point the mouse at the lower right hand corner and drag the box to the desired size.

---

8. New On Compuserve

In data library 7 (for registered Atari Developers only) in the Atari Developers SIG on Compuserve, the following files are new this month:

lo68.prg	the other linker program
ahdi.prg	rev 5 hard disk driver.
hdx.prg	hard disk partitioner.
hdx.rsc	
wincap	
ship.prg	hard disk head parking.
invite.let	UK show announcement.
newfrm.doc	form_button documentation.
ctype.o	character typing module.
f2s.doc	ROM making software.
f2s.prg	
glue.doc	
glue.prg	
send.prg	
end	
math.h	C header file support
string.h	
qa3.doc	previous Q/A bulletin.



Question and Answer Bulletin June 1986

1196 Borregas Ave., Sunnyvale, Ca. 94086

Here are the latest questions from the Atari developers mailbag as answered by John Feagans, Director of Software Technology. Leave questions on Compuserve for PIN 70007,1072 or GO PCS57 for Atari developer SIG information.

1. Discussion

We would like to clear up a bit of the confusion on the use of the Alcyon version 4.14 C compiler. The programs which are actually new in the system are cp, c0, and c1, the as68, and ar68. The Motorola FFP (libf) and the double precision math pack (libm) are upgraded and a new addition respectively. All the other tools and programs on the disks are the standard issue in the regular developer's kit. GEMLIB was modified slightly by Alcyon to include procedures needed by the new math libraries. In the future, everyone will be updated with a new linker.

The files on the disks have been organized in a two disk system for C- development. There is some redundancy on the linker and compiler disks with regard to batch.ttp, wait, and rm. The header files for the AES and VDI are not included since you already have them on the standard kit. You may configure the .bat files any way you wish, but be especially vigilant to cf.bat and linkf.bat which are intended to work as a pair. cf.bat has the -f option which is telling the new c-compiler to use the Motorola fast floating point. If you mix cf.bat with link.bat, the first symptom you will see is "printf.o multiply defined." I have done this accidently with my keyboard that has a sticking "k" key. If you want to be safe, just make a batch file that compiles and links your file in one operation. Remember also

that you must add stdio.h and ctype.h as before if you use those functions.

---

2. BIOS

Q: How can you initiate a screen dump from the keyboard?

A: Hold down the <Alternate> key and press <Help>. Be sure you have done the correct printer installation if you are using a non-Atari printer capable of doing raster dumps. Repeat the key sequence to halt the print at any time. The best part of this feature is that you can activate it from any program even if it does not use GEM.

Q: How do you turn off the cursor in a TOS application?

A: Send the VT-52 escape to turn off the cursor. Look in the Hitchhiker's Guide to the BIOS for a complete description of all the escape codes.

---

3. DOS

Q: How do I install partitions I have created with the hard disk format program?

A: Select one of the drive icons on the desktop. Move the mouse to the Options menu and select Install drive. Change the drive identifier to your choice and select OK. Be sure to do a Save desktop so that the installed drive will appear each time you boot.

Q: How can I recover some files I accidently trashed?

A: There is hope -- but not much. If you have not created a new file since your accident, you may be able to use a disk utility to view sectors and piece it back together. However, those sectors may be scattered all over your disk. The best medicine is prevention. Leave the confirm deletes option connected.

---

4. VDI

Q: Looking into the VDIBIND library, I saw that there were many functions which couldn't be found in VDIBIND.H. e.g. vro_cpyfm, vs_clip. What is the rason for not including these in VDIBIND. Can we include them or don't they work well?

A: VDIBIND.H is not necessary because all functions return an integer value--if ever. All these functions are in the VDIBIND

object library.

Q: I was using vg_text to output a string. When I switched to line-a to output the characters of the string one character at a time it was slower -- why?

A: When you pass a string to the VDI you are performing only one Trap or software interrupt. If you use line-a, you do a software interrupt--and all of the overhead--on each character you output. In the end, the same code outputs the pixels to the video RAM. In some cases, like this, VDI can exceed line-a.

Q: What is the standard address of screen memory?

A: Unlike older systems, the ST can have video display memory located anywhere as long as it is on 1k boundaries. Various device drivers may be loaded in at boot time so it is impossible to say that video memory has a fixed address. The only thing that can be said for sure about the location is that it is usually the high end of RAM.

---

5. AES

Q: Is it possible to track the mouse through the process of menu item selection? I need to change the appearance of items on the fly.

A: No, this is not an option of the AES. The usual scenario is to stay in the evnt_multi until a menu_event message is returned.

Q: How many parameters does the objc_edit command have?

A: The correct number is five. The AES manual incorrectly states six.

---

6. Desktop

Q: I would like to know if information from the desktop.inf is available to normal applications. I have noticed that the control panel and VT 52 emulator accessories have access to this information.

A: It is the responsibility of every application in the system to save and restore changes to system variables. We do not recommend an application going in to directly change the desktop.inf. There is a constant danger of user modifications causing damage to the system that cannot be supported. The best way to change colors is through the vdi. Modifying the hardware

registers directly is also discouraged since you are locking your application to one version of ST hardware. There are also xbios entries for changing and inquiring printer and rs-232 data.

---

7. BASIC

Q: How can I access the RS-232 from BASIC?

A: Where X is a character: For output use OUT 1,X and for input use X = INP(1).

---

8. Development Tools

Q: The function itoa is not in the C-runtime libraries. What can I use for this function?

A: If you have a special case where you need a small number of digits in a hurry, you can write a quick C function to convert an integer to ASCII characters. Another way is if you have included stdio, you can use sprintf to output an integer to a string.

Q: We are having serious troubles with C. The functions getchar and stream reading does not work properly. In fact a German version has modified "stdio.h" definitions, (sic) though it works better has still problems.

A: There is another version of stdio.h floating around Europe which uses a #define of getchar to be a BIOS Bconin(CON). This does not help the scanf function however. There is also a danger in using this version because when you mix Gem DOS calls (which the C run-time uses), and direct BIOS calls to get characters from the keyboard. A symptom of this has been noted as "10 characters disappearing" as the DOS bufferes them up whenever a printf is called. The safest replacements for getchar and scanf are using Cconin and Cconrs+sscanf respectively.

---

9. New On Compuserve

In data library 7 (for registered Atari Developers only) in the Atari Developers SIG on Compuserve, the following files are new this month:

gemlib	Works with both Alcyon versions.
test.c	Example using stdio to lst: and con:.
qa4.doc	May Q/A newsletter.

Atari Corp.
1196 Borregas Avenue
Sunnyvale, California 94088-3427
(408) 745-2000

CONFIDENTIAL DISCLOSURE AGREEMENT (BILATERAL)

Atari Corp., Sunnyvale, California and _____ _____ _____ agree that certain information exchanged between the parties relating to _____ _____ which is submitted in writing and marked as "Confidential Information" and is reduced to writing (30) days of the original oral disclosure shall be considered Confidential Information.

The parties agree to use the same degree of care to avoid unauthorized dissemination of Confidential Information as they employ with respect to their own information of a similar nature which they do not desire to have disseminated; provided, however, that this agreement shall impose no obligation upon either party with respect to any Confidential Information which; (i) is now or which subsequently becomes generally known or available by publication, commercial use or otherwise; (ii) is known by the receiving party at the time of receiving such information; (iii) is furnished by the disclosing party to third parties without restriction on disclosure; (iv) is subsequently rightfully furnished to the receiving party by a third party without a restriction on disclosure; or (v) is independently developed by the receiving party, provided that the person or persons developing same have not had access to the Confidential Information.

The parties shall be relieved of all obligations under this agreement with regards to Confidential Information _____ months after _____, 198 _.

Understood and agreed:

_____	ATARI CORP.
_____	_____
BY:	BY:
TITLE: _____	TITLE: _____
DATE: _____	DATE: _____

1055L