```
   AAA   TTTTT  AAA   RRRR   III         ****           SSS   TTTTT
  A   A    T   A   A  R   R   I          ****         S     S    T
  A   A    T   A   A  R   R   I          ****         S          T
  AAAAA    T   AAAAA  RRRR    I         ******          SSS      T
  A   A    T   A   A  R  R    I        ** ** **            S     T
  A   A    T   A   A  R   R   I       **  **  **      S     S    T
  A   A    T   A   A  R   R  III     **    **    **     SSS      T

  DDDD  EEEEE V     V EEEEE L      OOO  PPPP  EEEE RRRR   SSS
  D   D E     V     V E     L     O   O P   P E    R   R S   S
  D   D E     V     V E     L     O   O P   P E    R   R S
  D   D EEEE  V     V EEEE  L     O   O PPPP  EEEE RRRR   SSS
  D   D E     V     V E     L     O   O P     E    R   R      S
  D   D E      V   V  E     L     O   O P     E    R   R S   S
  DDDD  EEEEE   V    EEEEE LLLLL  OOO  P     EEEEE R   R  SSS
```

Question and Answer Bulletin   March 1986
Copyright (C) 1986 by Atari Corp.   "all rights reserved"
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 medium 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.