Zoomer Programmer's Guide

Z

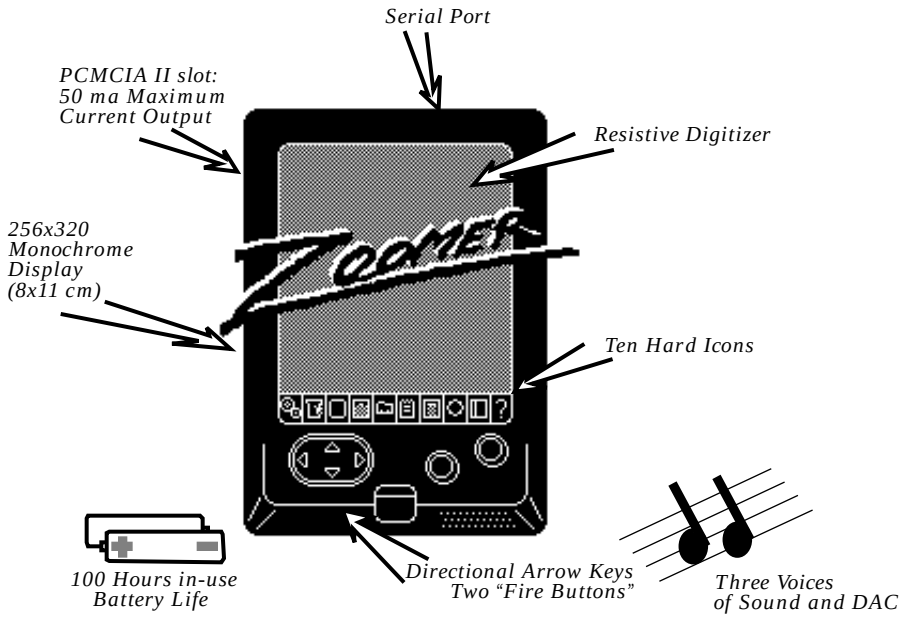
This booklet is a storehouse of advice for Zoomer programmers. We assume you know a reasonable amount about GEOS®—as much as you might pick up from the System Software Overview. We also assume you have some knowledge of Zoomer.

Z

Z

z.1 First Look

Hardware



NEC V20-based "PC on a Chip"	1 Meg of RAM
8 MHz Clock Speed	640K Heap
4 Meg of ROM	352K RAM Disk
	32K Video Memory

System Software:

GeoDOS, modified for use with Zoomer.
GEOS 2.0 with additional File Transfer Driver.

Included Applications:

Datebook, Address Book, Notebook, Calculator, World Clock, and Dictionary*
Setup, Preferences, File Manager, Uki, Solitaire, and Pyramid.

*For information about applications listed on this line, contact Palm (see back page for contact information).

z.2 Programming Issues

This page summarizes the main issues Zoomer programmers must address.

Heap Space

640K

Zoomer's total heap space is 640K, equal to the lowest of desktop PCs. This space must be shared by the application, active desk accessories, and GEOS libraries used by these programs. Applications must be highly sensitive to their memory usage; organize your resources efficiently and don't lock large blocks. Use fixed memory only when absolutely necessary. When optimizing, almost always save bytes of code in favor of cycles of CPU time.

Processor Speed

Zoomer's processor runs at 8MHz, slower than most desktop machines. The GEOS system software is highly optimized assembly code; whenever possible, you should rely on the system services. Budget time for your own code optimization. Swat provides many advanced optimization tools; use them.

Rare Shutdowns

Zoomer never shuts down. It has no machine-level resets. If your program has memory "leaks," lost memory stays lost until the batteries need changing, and Zoomer is designed to make that a long time. Swat has tools to aid in tracking down memory leaks; use them. To conserve batteries, don't force idle program threads to be active; on Zoomer, a wait loop uses more power than an inactive thread.

Screen Size

Zoomer's screen is tiny compared to desktop displays. Its 256x320 screen is about 8x11 cm (about 3.2x4.3 in). To make the most out of this area, applications must choose their on-screen gadgetry carefully. An application with buttons for each function may find itself with no space to display a document. Dialog boxes must be laid out carefully to restrain them from spilling beyond the screen's boundaries. At the same time, you should remember that applications must share the screen with Desk Accessories and dialogs.

z.3 Application Style

Zoomer style is rooted in simplicity. Zoomer users won't be familiar with other computer programs; given a choice between a simpler UI or an extra feature, you should probably choose the simpler UI. The following style guidelines may prove useful.

Full Screen Applications

Applications normally come up full-screen. It would not be practical to allow more than one application on the screen at a time, since neither would have enough room to do anything. If you are making a small application, you should probably make it a Desk Accessory, which may share the screen with other applications.

Simple Navigation

Zoomer applications must be easy to navigate. It is common to break up the gadgetry of many-featured applications over several "screens." If not all UI gadgetry will appear at once, the user should have an easy way to reach hidden gadgetry.

Optional Menus

The Zoomer UI tends to avoid menus. Menus take up valuable screen real estate. The Zoomer allows for a floating menu bar, but most users will choose not to use it (again, to save screen space). Thus, any commonly used UI gadgetry should appear within the body of your primary window, and not hidden in a menu. In Zoomer, menus are meant to hold the UI for the more obscure pieces of functionality.

Integrated Applications

Zoomer is an integrated device—the built-in applications will communicate with each other a great deal. Your applications should communicate with the built-in applications where appropriate, and with themselves certainly.

Pen-Centric

Zoomer is meant to be controlled by a pen. Most applications should be oriented toward pen input.

The GEOS system software includes a full-featured Inter-Application Communication Protocol (IACP), and your programs should make full use of this protocol to transmit information and requests between programs.

On Zoomer, only one main application will run at a time. If two main applications need to communicate, one will have to contact the other though the other is not active. GEOS provides the ability for applications to run in "engine" mode, where they have full functionality but do not appear on screen. Thus, it is simple to communicate with other programs even when those programs are not active.

z.4 Memory Usage

Zoomer has a 640K general heap. All application resources which will be active at one time must fit within this 640K. They must share this space with any loaded GEOS library resources and desk accessories. The GEOS kernel has been set up to execute in place to give you more heap space to work with. If all programs are to co-exist, you must organize your code sensibly.

The system can discard unused code resources. You should group routines which will be used together into the same resource—if the user isn't using some piece of functionality, then the system may discard that resource.

Program Resources

Since many users won't use the menu bar, it's a good idea to put menu gadgetry in a separate resource. If you are splitting your application's gadgetry into several screens, each screen's gadgetry should probably be in a separate resource so that the gadgetry associated with absent screens won't take up valuable heap space. Verify that your resources are properly constructed by using Swat. Unintended messages can negate careful resource planning.

Zoomer does not swap memory. Normally, if you wanted to set up a large block of data to be accessed quickly, you would mark it swapable so that the system could swap it out if memory got tight. However, when programming for the Zoomer, you should instead mark such blocks of data as discardable and be prepared to reload them if the system has been forced to discard them.

Mobile Memory

You should not lock data blocks for any longer than necessary, and you should never mark any blocks of data as fixed. With memory so precious, the memory manager must be able to reposition blocks so it can allow programs to allocate contiguous blocks. If you are programming in C, avoid the temptation to use malloc() or calloc(), each of which allocates a block of fixed memory. Instead you should use the GEOS memory allocation routines.

z.5 Specifying Memory Usage

To optimize Zoomer memory management, your application must include an extra line in its Glue Parameters (.gp) file. This line, with the keyword heapspace, specifies the memory that the application needs. Zoomer will use this to determine how many applications may be kept running at a time.

```
heapspace 3409
```

To find out how much heapspace your application needs (and thus find out what value to enter in the .gp file's *heapspace* line), run the application under Swat and use the **heapspace** Swat command. Note that you should try using heapspace in different situations to find out the maximum heapspace needed. Remember that if the user has menus and/or dialogs showing, more heap space will be used—try to have as many open as possible when testing for needed heap space.

To find out what applications are loaded into the application cache, use the **pappcache** Swat command.

z.6 Disk Use

Desk Accessories

Applications may either come up full screen or share the screen with other applications by operating as desk accessories. DAs float above all other applications and therefore should be as small as practical.

The Zoomer RAM Disk holds 352K. This is a fair amount of room, but applications which create large data files should probably compress their data. Remember that even though your application may not create especially large files, it is polite to compress them since that 352K is all the user will have access to. Remember: the Zoomer doesn't have a floppy drive, and if the RAM disk fills up users can find themselves in real trouble. Unless they have another computer that they can connect to and upload their files, they will have to discard some data (or buy another PCMCIA card). Most users would of course prefer to put off that eventuality as long as possible.

Default Files

Not all users are comfortable with the idea of data files. For many users, the distinction between application and document is a hazy one—it is difficult to understand how a single address book program might be asked to manage multiple documents. After all, an address book is a single object. For this large group of users, programs which create documents might want to provide a default document for the user to work with and expect this single document to get very large.

Ink Compression

If your application is using an Ink object to collect and display ink, remember that the object has the ability to store its ink to a file in compressed form—you should use this ability instead of saving the object itself with its ink uncompressed.

Read-Only Files

If you're in the habit of opening VM data files in force-read-only mode (i.e. you don't ever write to the things), you should also pass VMAF_FORCE_DENY_WRITE, as this tells the system that not only will you not be writing to the file, but neither will anyone else. If you don't pass this flag and the file is writable, the kernel will read *the entire file* into memory; even if the file's blocks are clean, they can't be discarded; only swapped out. Since swap space is so precious, please pass the deny-write flag.

z.7 Screen Usage

The Zoomer screen is 256x320 pixels, or about 8x11 cm. This puts screen real estate at a premium. The system has been optimized to help you to get the most out of the existing area; you should be thinking along similar lines.

The Zoomer system helps you to get the most out of the existing screen area. The hard icons below the screen allow access to some common UI gadgetry without using the screen. The menu bar was made optional to allow the application maximum screen area to display its own gadgetry. As a programmer, don't cancel the system's effort by including redundant UI. When constructing a suite of related programs, it may be tempting to include launchers so that the user may go between them. Because the Zoomer's launcher already gives them that ability, you should not devote screen space to this redundant gadgetry.

When planning your layout, aim for a compact screen presence. Instead of including separate gadgetry for displaying and editing data, combine them so that the user interacts directly with the display. This feels more natural and saves the space which might go towards the UI for editing the data.

Most users will not use the menu bar. All program functionality should be represented in menus, but the most commonly used functions should be reachable by other means.

If you localize your application for international use, the size of UI components may grow. Allow space for this growth, and read the Localization chapter of your GEOS documentation for more information.

z.8 Sound

Zoomer has great sound, and you should take advantage of it. Zoomer sound hardware supports musical notes and DAC, allowing both music with rich harmonics and sampled sounds. The Zoomer has three tone generators, a noise simulator, and a DAC player, all of which may operate at once. Zoomer allows both musical tones and sampled sound to play at the same time.

GeoWorks has written a new library which allows the Zoomer to play standard .WAV file sampled sounds using the Zoomer. You are free to use this library and to install it on any Zoomer. Note that the library did not go out with the original Zoomer, so you will have to install it on the user's machine.

```
void
PlaySoundFromFile(
  FileHandle file);

void WavPlayFile(
  DiskHandle disk,
  const char *path,
  const char *fname);
```

The library is in a file called WAV.GEO, which should be available in GEOS support libraries on AOL and CompuServe. The header information for the library is in the file WAV.GOH. There are two routines available.

PlaySoundFromFile() is ideal for playing short sounds on threads which are not time-critical (i.e. threads which are not your application's UI thread). It plays the sound in an already opened .WAV file. The calling thread will not continue until the sound finishes playing, and thus it is a bad idea to call this routine to play a long sound.

WavPlayFile() takes the path and file name of a .WAV file. It spawns a background thread to open the file and play the sound. This is the routine to use to play long sounds or to play sounds from a time-critical thread.

Sample Code:

```
{ /* Plays the sound in "TEST.WAV" on a different thread (execution
   * continues on this thread while the sound plays in the background)    */
 WavPlayFile(SP_USER_DATA, "SOUND", "TEST.WAV"); }


{ /* Plays the sound in "TEST.WAV" on the current thread. */
 FileHandle fh;
 (void) FileSetCurrentPath(SP_USER_DATA, "SOUND");
 fh = FileOpen("TEST.WAV", FILE_ACCESS_R | FILE_DENY_W);
 if (fh) {
        PlaySoundFromFile(fh);
        (void) FileClose(fh, TRUE); }; }
```

z.9 Controls

The Zoomer has a simple set of controls: a pen and a set of buttons. There is no keyboard; text input is handled via HWR or an on-screen keyboard. Most applications rely on the pen for all input; Zoomer is a pen-centric machine.

The user will probably use either the pen or the buttons, but not both. Remember that the Zoomer is a palmtop device, meant to be used by people "on the go." It is difficult to physically hold the Zoomer, press the buttons, and move the pen all at the same time. If you can't decide whether pen or button input would be appropriate, conduct user testing to determine which is best.

There are six buttons, set up as four "directional" buttons and a pair of "fire" buttons. For applications which are controlled by the buttons, the directional buttons should generally be used for navigation, the fire buttons to activate functionality.

Pen input will be interpreted as mouse or text input depending on what the objects of your application expect to receive. Input from the buttons acts as "keyboard" input, using a custom set of "keys."

These keyboard input events will have a character whose top byte has the value CS_CONTROL and whose bottom byte has one of the following **VChar** ("virtual character") values:

```
VC_JOYSTICK_135   VC_JOYSTICK_90    VC_JOYSTICK_45
VC_JOYSTICK_180                     VC_JOYSTICK_0
VC_JOYSTICK_225   VC_JOYSTICK_270   VC_JOYSTICK_315

VC_FIRE_BUTTON_1  VC_FIRE_BUTTON_2
```

The VC_JOYSTICK values correspond to the Zoomer directional keys. Note that the Zoomer has just four directional buttons—the 45, 135, 225, and 315 degree values will be passed when the user is pressing two buttons at once. The VC_FIRE_BUTTON values correspond to the two zoomer fire buttons.

The buttons may already have some functionality depending on context. The HelpControl responds to the fire buttons by opening the selected topic; to the right/left buttons by going to the next/previous help topic. The Text objects use the right/left buttons to move the cursor forward/back. A scrolling GenView scrolls in response to the directional keys.

z.10 PCMCIA Cards

Zoomer accepts PCMCIA Type II cards. The socket provides a maximum of 50ma of output. This is not enough to power many devices (such as card-based disk drives), but is sufficient for cards containing only ROM or SRAM, as well as some I/O cards.

The data should be set up as a DOS directory and file structure. This file structure will be merged with that already on Zoomer. When setting up your file structure, remember that the WORLD directory is the top-level directory for applications, and that those applications placed in the Desk Accessories directory (DESK_ACC.000) will be recognized as desk accessories.

Do not include Preferences modules on Zoomer cards unless you will copy them onto the Zoomer RAM disk. If the user starts up a Preference module which resides on a PCMCIA card, the Zoomer will crash if the card is removed. As long as the Preferences module is copied to the RAM disk and the user uses that copy, everything should be fine.

z.11 Fonts

By default, Zoomer will use a bitmap-based font rather than an outline-based font. This leads to a faster response time, but bitmap-based fonts are not WYSIWYG. If the user will be printing something from the zoomer, and WYSIWYG output is important for the document, then use an outline-based font. The Zoomer comes with one outline-based font, URW Sans.

If drawing text using kernel routines, use **GrSetFont()**, passing FID_DTC_URW_SANS to access the built-in outline font. To get a list of all installed outline-based fonts at run-time, call the **GrEnumFonts()** routine, being sure to set the FEF_OUTLINES bit in the passed **FontEnumFlags** structure.

If you're working with a text object, remember that there is a **VisTextDefaultFont** value corresponding to URW Sans: VTDF_URW_SANS.

If the user adds a new font, that font will not be recognized until GEOS exits and re-starts.

z.12 Debugging & pccom

For the first stages of debugging, it will be enough to debug applications which are running on a normal PC system which has been configured to act something like a zoomer. However, for true testing, you will want to have your program running on a Zoomer. Attaching the Swat debugger is basically the same as when debugging in any GEOS environment, but with some hardware-specific changes in procedure.

Before debugging can begin, you must download three new or changed files to the Zoomer. Using the Zoomer's File Connect feature, you can download through the serial port these three files, which are stored in the TARGET\ZOOM directory of your SDK CD:

loader.exe A modified loader that will start up with pccom running.

swat.exe The Swat stub to allow Swat to connect.

geos.geo A new kernel that will allow breakpoints to be set (the Zoomer's execute-in-place kernel does not).

The new kernel runs from the RAM disk and takes up quite a bit of space. If you have documents or state files taking up much room, back them up to the PC and then erase the RAM disk by holding the fire buttons while pushing the reset button.

When using pccom to communicate with a Zoomer, it's best to use a relatively low baud rate such as 9600. Note that the Zoomer's communication speed may be set using the Preferences application.

When PCS sends down a file, it assumes that the target machine has been set up with the Desktop's directory structure; you may want to re-create some of WORLD's subdirectories.

Z

Geoworks
960 Atlantic Ave.
Alameda CA 94501
(510) 814-1660 Fax: (510) 814-4250

For information about the Palm applications contact

Palm Computing, Inc.
4410 El Camino Real, Suite 106