

geoProgrammer Errata

We urge you to read these notes in their entirety. They describe late improvements, corrections, and problems with versions 1.0 and 1.1 of geoProgrammer.

Problems Common to geoAssembler and geoLinker

Intermediate Files After a Disk Error

If geoAssembler or geoLinker terminates after a disk error, it does not attempt to close or remove any newly-created files (.rel, .err, etc.), which means you cannot, for example, view the .err in geoWrite. It is best to delete these intermediate files and validate the disk.

Colors

geoAssembler and geoLinker do not properly handle Preference Manager color settings. If you use colors other than the default dark grey on white, you will be able to see the symbol table insertion sort in the screen area (fun to watch, but not harmful).

geoAssembler

Pass1 Flag

Misuse of the **Pass1** flag is the single-most problematic aspect of geoAssembler. If you get errors you don't understand or you get the mysterious "Hidden error found," remove the **Pass1** conditionals from your program. By eliminating equates and macros from the second pass of the assembly you are bypassing much of the sophisticated error-checking offered by geoAssembler in exchange for a faster assembly. Use this capability carefully and judiciously.

.header

Pages 5-50 through 5-53: the .header directive does syntax checking on 12 fields as opposed to the 11 mentioned in the manual. The 12th field in the *authorname* field and must generate exactly 20 bytes. The example on page 53 has this field correctly shown.

geoAssembler Addressing Mode Substitution

geoAssembler will not attempt to substitute absolute mode addressing when a zero-page operand is used with an instruction that does not support zero-page addressing. For example, geoAssembler would flag an addressing mode error on **eor \$30,y** because the eor instruction does not have a zero-page indexed with Y addressing mode even though there is an absolute indexed with Y version of the instruction which could use \$30 as a two-byte absolute address (\$0030). Future versions of geoAssembler will attempt to substitute absolute mode addressing when no zero-page equivalent exists.

Using Relocatable Symbols in Immediate Mode and .byte Expressions

When using a relocatable symbol (a psect label, for example) in the context of an immediate mode address or a .byte directive, the symbol must be preceded by a low- or high-byte operator ([or]). Otherwise, incorrect object code will be generated.

Incorrect:

```
.psect
label:          ;psect labels are relocatable
    lda    #label    ;this will generate bad object code
    sta    lowByte
    rts
    .byte label    ;this will also
```

Correct:

```
.psect
label:          ;psect labels are relocatable
    lda    #[label   ;the will generate the correct obj code
    sta    lowByte
    rts
    .byte [label    ;here, too
```

Incorrect Line Numbers in Error Messages

geoAssembler does not always print the correct line and page numbers in its error messages. The line number reflects the number of lines from the top of the geoWrite page after the macro expansion. Also, when geoAssembler detects a branch out of range error, it often writes out the incorrect page number (although the line number is usually correct). Check the specified line number on the pages preceding the page specified in the error message.

Local Regions Larger Than 250 bytes

If any local region (the area between two successive global labels) exceeds 250 bytes, there is a slight chance that code buffer will overwrite the beginning of the external reference buffer. This bug can be avoided in most cases by inserting global labels every 70 or so instructions. Also, avoid using large (>250 bytes) .block directives in a psect section.

The * Symbol

The * symbol used in an expression will evaluate to the current value of the active section counter. At this time, it always evaluates to the value of the current psect counter, which means it does not work correctly in zsect or ramsect sections.

Example usage:

```
.psect
String:    .byte    "This is a string...", NULL
StringLen  == (* - String)    ;length of string
```

Bitmap Decompaaction

geoAssembler will not correctly decompact bitmaps that are shorter than 18 pixels. Also, bitmaps that use the bigcount format will not be handled correctly (no Berkeley Softworks application or desk accessory uses the bigcount format, so this is not much of a problem).

geoLinker

With One Drive Systems

geoLinker cannot generate a viewable symbol table on one drive systems (a RAM-expansion unit configured as a RAMdisk counts as a second drive); the dialog box at the bottom of page 4-19 will never be shown.

Module Symbol Table Overflow

If more than one `.rel` file is being linked into a single module and each file `.include's` the `geoSym` file, the linker's symbol table buffer will soon overflow because the entire repertoire of symbols will be sent to the linker with each `.rel` file. Add `.noeqin / .eqin` around all but one of the `.include geoSym's`.

Maximum of Ten .rel Files in a Single Module

A single module cannot have more than ten `.rel` files. However, geoLinker will not correctly flag this error. Be sure to have no more than ten `.rel` files in a single module.

.rel Filenames

geoLinker does not correctly flag errors when the `.rel` file names are mistyped in the linker command file. Check these carefully.

GEOS 1.2 and CBM Applications

geoLinker does not correctly delete header blocks from CBM files under GEOS 1.2. Upgrade to GEOS 1.3 if using geoProgrammer to generate CBM applications.

.dbg Zero-page Symbols

geoLinker sometimes leaves the `zsect` bit set in zero page symbols it sends to the debugger. This results in a graphic character printed in the third character position of the symbol in the debugger. Use the debugger `?` wildcard to circumvent this problem.

geoDebugger

Write-protected Files

The geoDebugger file-selection dialog box will not display write-protected files.

Tutorial

The sample application cannot be rerun with a `go $400` or a `go StartApp`. Also, page 7-11 is incorrect in stating that the `runto` command will briefly flash to the application screen.

Super-Debugger Keystroke Correction

page 8-1: the `£` character is created by typing the `[£]` key (without the `[C=]` key) on the Commodore keyboard.

Super-debugger Runto Command

page 8-43: the `runto` command is based on `p` (proceed), not `go` as stated in the manual. The GEOS screen will only be shown if `opt 5` is enabled.

Mini-debugger Rt Command

Page 9-23: the `rt` command will only display the GEOS screen if this option has been enabled with the `g1` command.

Appendix C: Debugger Isolation and off-limits area

In addition to the GEOS routines mentioned on page A-15, the super-debugger configuration uses the **StashRAM**, **FetchRAM**, and **SwapRAM** RAM-expansion unit routines and expects GEOS's interrupt code to be intact. Also, the first off-limits area \$350 to \$3ff has been extended to the range \$340 to \$3ff.

Sample Files

geosMemoryMap File

The `APP_RAM == $0400` equate in the `geosMemoryMap` file listed in the manual has been changed to `APP_RAM = $0400` (double equal-sign changed to a single equal-sign) so that the application's label at this address will be visible in the debugger.

Sample Applications and the Mini-debugger

The sample VLIR and sample sequential application will not work correctly with the mini-debugger because the variable spaces were inadvertently placed in the off-limits RAM at \$5000. Change the `.psect $5000` in the `SamVlir.lnk` and `SamSeq.lnk` file to `.psect $3000`.

Sample Application Menus

The menu data structure in the `SamSeq` file under `MenuTable` has two lines which read

```
.byte VERTICAL          ;type of menu
```

These should be changed to

```
.byte SUB_MENU          ;type of menu
```

The same error exists in the `SamVlirRes` file (under `MenuTable`), except there are three occurrences instead of two.