

---

**Subject: Loading Files from Disk in Assembly Language using GS/OS**Posted by [Oz](#) on Sat, 11 Oct 2014 10:37:15 GMT[View Forum Message](#) <> [Reply to Message](#)

---

Most of the game resources (graphic, sound, music, score table, animation...) are stored outside of the code, in dedicated files. So we need to get these files and to load them in memory, when required.

The first step is of course to have allocated the right number of memory Banks to store the files. On the Apple IIgs, most of the files are 64 KB long, or less. Because of the memory organization and its 64 KB bank boundary, it is always better to cut a large file into sub-files, each of them to be 64 KB or less.

The next code can be used to load, in memory, a <=64 KB file located under a file system (Prodos, HFS...) that GS/OS can access (using the system FSTs). The advantage of using GS/OS here is its simplicity and its speed. We don't care anymore about what kind of file system we have (ram disk, Prodos 800 KB floppy disk, HFS hard drive...), GS/OS handle that for us.

As for any operating system or language, we chain the following commands : Open File, Get File Size, Read File Data and the Close File. We don't have to care about file size, the LoadFile code do it for us :

```
GSOS      =    $E100A8
```

```
*----- Load one file in memory -----*
```

```
LoadFile  STX  gsosOPEN+4    ; X = File Path,  
          STA  gsosREAD+5    ; A = Bank XX/00
```

```
*--
```

```
LF_Open   JSL  GSOS          ; Open File
```

```
    dw    $2010  
    adrl  gsosOPEN  
    BCS   LF_ErrorEnd  
    LDA   gsosOPEN+2  
    STA   gsosGETEOF+2  
    STA   gsosREAD+2  
    STA   gsosCLOSE+2
```

```
*--
```

```
LF_GetSize JSL  GSOS          ; Get File Size
```

```
    dw    $2019  
    adrl  gsosGETEOF  
    LDA   gsosGETEOF+4  
    STA   gsosREAD+8  
    LDA   gsosGETEOF+6  
    STA   gsosREAD+10
```

```
*--
```

```
LF_Read   JSL  GSOS          ; Read File Content
```

```
    dw    $2012
```

```

        adrl gsosREAD
        BCS LF_Error
*--
LF_Close  JSL  GSOS          ; Close File
          dw  $2014
          adrl gsosCLOSE
*--
LF_End    CLC                ; No Error
          LDA  gsosGETEOF+4  ; A = File Size
          RTS
*-----
LF_Error  JSL  GSOS          ; Close File
          dw  $2014
          adrl gsosCLOSE
*--
LF_ErrorEnd SEC              ; Error
          RTS
*-----
gsosOPEN  dw  2              ; pCount
          ds  2              ; refNum
          adrl File1_Path    ; pathname (init with an existing file path, so we only have to change
2 bytes)

gsosGETEOF dw  2            ; pCount
          ds  2            ; refNum
          ds  4            ; eof

gsosREAD  dw  4            ; pCount
          ds  2            ; refNum
          ds  4            ; dataBuffer
          ds  4            ; requestCount
          ds  4            ; transferCount

gsosCLOSE dw  1            ; pCount
          ds  2            ; refNum
*-----

```

We call this code by giving as parameter the Path of the file and the Bank where the file must be loaded :

```

*----- Allocate Memory Banks -----
        JSR  AllocOneBank  ; Memory Allocation for File
        STA  BankFile      ; A = XX/00
        ...

*----- Loading Files -----

```

```
LDX #File1_Path ; Load File1.bin file
LDA BankFile ; in BankFile at address $0000
JSR LoadFile
BCS ErrorQuit
...
```

\*-----

```
BankFile HEX 0000
```

```
File1_Path strl '1/Data/File1.bin'
```

If we want to load the file at address \$8000 in the Bank (instead of \$0000), you simply have to modify the loading address :

```
LDA BankFile ; in BankFile
ORA #$0080 ; at address $8000
JSR LoadFile
```

Olivier

---