in

COLLABORATORS

	TITLE : in	

ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		October 9, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

in

1.1 RIAmosFuncLib

```
                .-------------------------------------------------------------------------
|                        RI AMOS Func Library V2.5                           |
`----------------------------------------------------------------------------'

                        ©1996 Red When Excited Ltd

       Undocumented commands included by Toby Zuijdveld 28/02/1999.
                      mailto: hotcakes@abacus.net.au


                     Overview
                                                         Command Index


                 BankLimit

                 BlitterDone

                 BLoad

                 BSave

                 CopyByte

                 CopyLong

                 CopyWord

                 DeviceName$

                 Erase

                 EraseAll

                 FileSize

                 FillMem
```

```
            CachesOn     bitmask              CachesOff     bitmask
GTTagValue   tagvalue             SwapBank     sourcebank,destbank
ChangeDir    dir$                 SaveExec     filename$,list()
```

1.2 Whats it all about?

Overview

This library was originally designed to emulate useful commands present in AMOS. However, it has now grown to an all-purpose library containing miscellaneous commands which did not warrant a separate library.

You will find commands for easy memory bank management, file loading/saving commands, timing commands, memory manipulation and Amiga hardware commands.

1.3 RIAmosFuncLib

```
Function: BankLimit
-----------------------------------------------------------------------------
Modes  : Amiga/Blitz
Syntax : [numbanks=]BankLimit[(maxbanks)]

  By default, the library lets you use banks 0-49. If this is not enough,
then call this command using :   BankLimit numbanks

This will currently erase all banks so be warned!

To find out the current bank limit use :   numbanks=BankLimit
```

1.4 RIAmosFuncLib

```
Function: Reserve
-----------------------------------------------------------------------------
Modes  : Amiga/Blitz
Syntax : success=Reserve(length) | (banknumber,length[,requirements])

  If only length is specified, then this function returns the number of the
bank allocated or -1 for failure.

  This will attempt to reserve <length> bytes of memory. If succesfull,
it will return -1. If unsuccessfull, 0 is returned.

The optional <requirements> parameter specifies which type of memory you
want:

    %1           =PUBLIC
    %10          =CHIP
    %100           =FAST
    %100000000       =LOCAL
    %1000000000      =24BITDMA
    %10000000000       =KICK
    %10000000000000000      =CLEAR
    %100000000000000000      =REVERSE
    %10000000000000000000000000000000 =NO_EXPUNGE

OR the values together for different combinations.

e.g.

  suc=Reserve(0,1024,%10)   ; Reserve 1k of Chip Mem returns -1
  suc=Reserve(1024)   ; Reserve 1k of Any Mem returns 1
```

1.5 RIAmosFuncLib

```
Statement: Erase
-----------------------------------------------------------------------------
Modes  : Amiga/Blitz
```

```
Syntax : Erase banknumber
```

The Erase command will erase the specified memory bank.

e.g.

```
  suc=Reserve(0,1024,%10)                     ; Reserve 1k of Chip Mem
  Erase 0
```

1.6 RIAmosFuncLib

```
                Statement: EraseAll
--------------------------------------------------------------------------------
Modes  : Amiga/Blitz
Syntax : EraseAll
```

This command will erase ALL allocated memory banks.

e.g.

```
  suc=
                Reserve
                (0,1024,%10)    ; Reserve 1k of Chip Mem
  suc=Reserve(1,2048,0)            ; Reserve 1k of ANY Mem
  EraseAll
```

1.7 RIAmosFuncLib

```
                Function: BLoad
--------------------------------------------------------------------------------
Modes  : Amiga
Syntax : success=BLoad(filename$) | (filename$,bank/addr[,length,offset,memtype])
```

If only filename$ is specified, then the next available bank is allocated, and the command returns the number of the bank for success or -1 for failure.

If bank is specified, then the file is loaded into that bank. If address is specified then it is loaded into the address. Valid banks are 0-49.
If the bank does not exist, Blitz will reserve a bank for you.
If the bank does exist, Blitz will erase the bank from memory, and allocate a new one.
The return result is -1 for success, or 0 for failure (not enough RAM, file does not exist). If offset is specified, then <length> bytes will be read from the specified offset position in the file.
If memtype is specified, then the file is loaded into a memory block of that particular memtype (see
Reserve
)
If you wish to leave either length/offset unspecified, simply use the value 0

```
e.g.

  suc=BLoad("s:startup-sequence",0) ; returns -1
  suc=BLoad("c:dir",0,0,0,%10)    ; Loads into CHIP
  suc=BLoad("c:list")     ; returns 1
```

1.8 RIAmosFuncLib

```
                Function: BSave
------------------------------------------------------------------------------
Modes  : Amiga
Syntax : success=BSave(filename$,bank/address,length)

  This will save <length> bytes at bank/address to the file. Return result
is -1 for success, 0 for failure. If length > bank length, then the length
of the bank is saved instead. If 0 is specified, the entire bank is saved.

e.g.

  suc=
                BLoad
                ("c:dir",0,0,0,%10) ; Loads into CHIP
  suc=BSave("ram:temp",0)
```

1.9 RIAmosFuncLib

```
                Function: Start
------------------------------------------------------------------------------
Modes  : Amiga/Blitz
Syntax : start_address.l=Start(banknumber.b)

  This will return the start address of the specified bank. (0=no bank)

e.g.
        suc=
                Reserve
                (0,1024,%10)
  NPrint Start(0)
  MouseWait
  End
```

1.10 RIAmosFuncLib

```
                Function: Length
------------------------------------------------------------------------------
Modes  : Amiga/Blitz
Syntax : length_of_bank.l=Length(banknumber.b)
```

```
  This will return the length of the specified bank in bytes. (0=No bank)

e.g.

  suc=
                Reserve
                (0,1024,%10)
  NPrint Length(0)
  MouseWait
  End
```

1.11 RIAmosFuncLib

```
Function: MemFree
-----------------------------------------------------------------------------
Modes  : Amiga/Blitz
Syntax : bytes.l=MemFree

  This will return the total amount of Public Free RAM available to the
system.

e.g.

  NPrint "Total bytes free = ",MemFree
  MouseWait
  End
```

1.12 RIAmosFuncLib

```
                Function: NextBank
-----------------------------------------------------------------------------
Modes  : Amiga/Blitz
Syntax : bank.b=NextBank

  This will return the number of the first available bank (-1 if none free).

e.g.

  suc=
                Reserve
                (0,1024)
  suc=Reserve(0,2048)
  NPrint NextBank
  MouseWait
  End
```

1.13 RIAmosFuncLib

```
                Statement: FillMem
-----------------------------------------------------------------------------
Modes  : Amiga/Blitz
Syntax : FillMem address.l,length.l[,value.b]

   This will fill 'length' bytes starting from the specified address with
'value'. If 'value' is ommitted, 0 is filled.

e.g.

  suc=
                Reserve
                (0,1024)       ; Allocate some memory
  FillMem Start(0),Length(0)                   ; Clear it
  MouseWait
  End
```

1.14 RIAmosFuncLib

```
                Statement: CopyByte
-----------------------------------------------------------------------------
Modes  : Amiga/Blitz
Syntax : CopyByte source.l,dest.l,num.l

  This will copy <num> bytes from <source> to <dest>

e.g.

  CopyByte
                Start
                (0),Start(1),
                Length
                (0)
```

1.15 RIAmosFuncLib

```
                Statement: CopyWord
-----------------------------------------------------------------------------
Modes  : Amiga/Blitz
Syntax : CopyByte source.l,dest.l,num.l

  This will copy <num> words from <source> to <dest>

e.g.

  CopyWord
                Start
                (0),Start(1),
                Length
                (0)/2
```

1.16 RIAmosFuncLib

```
Statement: CopyLong
-----------------------------------------------------------------------------
Modes  : Amiga/Blitz
Syntax : CopyByte source.l,dest.l,num.l

  This will copy <num> longwords from <source> to <dest>

e.g.

  CopyLong Start(0),Start(1),Length(0)/4
```

1.17 RIAmosFuncLib

```
Function: MakeDir
-----------------------------------------------------------------------------
Modes  : Amiga
Syntax : success=MakeDir(name$)

   This function attempts to create a directory called <name$>. If it is
unsuccessfull, 0 is returned else -1 is returned.

e.g.

  suc=MakeDir("RAM:MYDIR")
```

1.18 RIAmosFuncLib

```
Function: Rename
-----------------------------------------------------------------------------
Modes  : Amiga
Syntax : success=Rename(source$,dest$)

   This attempts to rename the file <source$> to <dest$>. -1 is returned if
successfull, else 0.

   NOTE: It is not possible to rename across devices.

e.g.

  suc=Rename("S:Startup-Sequence","S:Startup2") ; Do not run this!
```

1.19 RIAmosFuncLib

```
Function: Timer
-----------------------------------------------------------------------------
Modes  : Amiga/Blitz
Syntax : t.l=Timer

   This will return the number of 50ths of a second since startup or the
last call to ResetTimer.

e.g.

  NPrint Timer
  VWait
  NPrint Timer
  MouseWait
  End
```

1.20 RIAmosFuncLib

```
Statement: ResetTimer
-----------------------------------------------------------------------------
Modes  : Amiga/Blitz
Syntax : ResetTimer

   This will reset the CIA timer to 0.

e.g.

  NPrint Timer
  VWait
  ResetTimer
  NPrint Timer
  MouseWait
  End
```

1.21 RIAmosFuncLib

```
Function: Lisa
-----------------------------------------------------------------------------
Modes  : Amiga/Blitz
Syntax : chipver=Lisa

   This will return the current Lisa chip version :

    $00 for OCS Denise
    $F7 for ECS Denise
    $F8 for AGA Lisa

e.g.

  Select Lisa
```

```
    Case 0
      NPrint "You have an OCS Machine!"
    Case $F7
      NPrint "You have an ECS Machine!"
    Case $F8
      NPrint "You have an AGA Machine!"
    Case $F9
      NPrint "You have a AAA Machine?!" ; Maybe... :)
  End Select
  MouseWait
  End
```

1.22 RIAmosFuncLib

```
Statement: Reboot
------------------------------------------------------------------------------
Modes  : Amiga/Blitz
Syntax : Reboot

   This will perform a cold reboot

e.g.

  NPrint "Press mousebutton to reset.."
  MouseWait
  Reboot
```

1.23 RIAmosFuncLib

```
Function: FileSize
------------------------------------------------------------------------------
Modes  : Amiga
Syntax : size.l=FileSize(filename$)

  This returns the length (in bytes) of the file.

e.g.

  NPrint "Startup is ",FileSize("S:startup-sequence")," bytes long!"
  MouseWait
  End
```

1.24 RIAmosFuncLib

```
Function: XOR
------------------------------------------------------------------------------
Modes  : Amiga/Blitz
Syntax : x.l=XOR(x.l,y.l)

  This will perform an Exclusive-Or operation between X and Y and put the
```

```
result back into X

e.g.

    x=XOR(%101,%100)

Will place %001 into X (%101 XOR %100 = %001)
```

1.25 RIAmosFuncLib

```
Function: Max/Min
-----------------------------------------------------------------------------
Modes  : Amiga/Blitz
Syntax : value=Max(first_var,second_var)
         value=Min(first_var,second_var)

  This will compare both values and return either the Higher of the values
(Max) or the Lower (Min). This currently supports INTEGERs only.

e.g.

  NPrint Max(30,50)
  NPrint Min(30,50)
  MouseWait
  End
```

1.26 RIAmosFuncLib

```
Function: KeyCode
-----------------------------------------------------------------------------
Modes  : Amiga/Blitz
Syntax : keycode=KeyCode

    This will return the status of the keyboard in the form of a keycode.
You will need to experiment to find out the desired keycode for a particular
key. This merely peeks address $bfec01 and returns the value found.

e.g.

  NPrint KeyCode
  MouseWait
  End
```

1.27 RIAmosFuncLib

```
Function : FindVolume
-----------------------------------------------------------------------------
Modes  : Amiga
Syntax : success=FindVolume(volumename$)
```

```
  This will look to see if the specified volume is present, and returns
0 if it is not, or -1 if it is. If the volume is not present, this function
will NOT bring up a Requester ("Please insert Volume...")
The ":" should not be included in the volumename.

This is useful when waiting for diskswaps if you have a BlitzMode display.

e.g.

    <BlitzModesStatements>
    QAMIGA
    Repeat
      VWait
    Until FindVolume("DISK2")
    BLITZ
    <More statements>
```

1.28 RIAmosFuncLib

```
Function : DeviceName$
-----------------------------------------------------------------------------
Modes  : Amiga
Syntax : devname$=DeviceName$(volumename$)

  This will return the device name of the specified volume or "" if the
volume was not found. The ":" may or may not be included.

e.g.

    NPrint DeviceName$("WORK:")
```

1.29 RIAmosFuncLib

```
Function : BlitterDone
-----------------------------------------------------------------------------
Modes  : Amiga/Blitz
Syntax : status=BlitterDone

  This checks to see if the Blitter has finished BLITting. -1=Yes, 0=No

e.g.

  Repeat
  Unti BlitterDone
```

1.30 RIAmosFuncLib

```
Statement : WaitBlitter
-----------------------------------------------------------------------------
Modes  : Amiga/Blitz
```

```
Syntax : WaitBlitter

  This will halt program execution until the Blitter is ready for use.

e.g.

  Blit 0,0,0
  WaitBlitter
     ..
     ..
```

1.31 Example Programs

```
                             Example Programs

EXAMPLE 1 - Example usage of most RIAMOSFuncLib commands:

   Load Example 1
   Compile It!
```