DosPackets

Bruce M. <bsimpson@toucdwn.demon.co.uk>

COLLABORATORS

	TITLE : DosPackets		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Bruce M. <bsimpson@toucdwn.demon.co.uk>	February 12, 2023	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

DosPackets

1.1 DosPackets.doc

```
--unknown--

ACTION_CREATE_DIR

ACTION_CURRENT_VOLUME

ACTION_DELETE_OBJECT

ACTION_DIE

ACTION_DISK_INFO

ACTION_END

ACTION_EXAMINE_FH

ACTION_EXAMINE_OBJECT

ACTION_EXAMINE_NEXT

ACTION_FH_FROM_LOCK

ACTION_FINDINPUT

ACTION_FINDOUTPUT

ACTION_FINDUPDATE

ACTION_FLUSH

ACTION_FORMAT

ACTION_FREE_LOCK

ACTION_INFO

ACTION_INHIBIT
```

```
ACTION_IS_FILESYSTEM

ACTION_LOCATE_OBJECT

ACTION_MORE_CACHE

ACTION_PARENT

ACTION_PARENT_FH

ACTION_READ

ACTION_READ_LINK

ACTION_RENAME_DISK

ACTION_RENAME_OBJECT

ACTION_SAME_LOCK

ACTION_SCREEN_MODE

ACTION_SEEK

ACTION_SERIALIZE_DISK

ACTION_SET_COMMENT

ACTION_SET_DATE

ACTION_SET_OWNER

ACTION_SET_PROTECT

ACTION_WRITE

ACTION_WRITE_PROTECT
```

1.2 DosPackets/--unknown--

```
Unknown Packets

These are packets I don’t have any information on at this time.

ACTION_ADD_NOTIFY
ACTION_CHANGE_MODE
ACTION_CHANGE_SIGNAL
ACTION_COPY_DIR
ACTION_COPY_DIR_FH
ACTION_DEBUG
ACTION_DISK_CHANGE
ACTION_DISK_TYPE
ACTION_EVENT
ACTION_EXAMINE_ALL
```

```
ACTION_EXAMINE_ALL_END
ACTION_FREE_DISK_FSSM
ACTION_FREE_RECORD
ACTION_GET_DISK_FSSM
ACTION_LOCK_RECORD
ACTION_MAKE_LINK
ACTION_NETWORK_HELLO
ACTION_QUEUE
ACTION_READ_RETURN
ACTION_REMOVE_NOTIFY
ACTION_SETTRANS
ACTION_SET_FILE_SIZE
ACTION_SET_MAP
ACTION_STACK
ACTION_STARTUP
ACTION_TIMER
ACTION_WAIT_CHAR
ACTION_WRITE_RETURN
```

1.3 DosPackets/ACTION_CREATE_DIR

```
 NAME
ACTION_CREATE_DIR -- create a directory

 SYNOPSIS
ACTION_CREATE_DIR ( lock, name )
                    Arg1  Arg2

ACTION_CREATE_DIR ( BPTR, BSTR );

 FUNCTION
Create a subdirectory of name 'name' in directory specified by 'lock'.

 INPUTS
lock - lock specifying directory in which to create new subdir.
name - name of new subdir.

 RESULTS
res1 - DOSTRUE if successful, else DOSFALSE.
res2 - If res1 was DOSFALSE, this tells you why the error occured.

 SEE ALSO
dos.library/CreateDir()
```

1.4 DosPackets/ACTION_CURRENT_VOLUME

```
 NAME
ACTION_CURRENT_VOLUME -- inquire about a file's volume and unit number

 SYNOPSIS
ACTION_CURRENT_VOLUME ( fh_Args )
                        Arg1
```

```
ACTION_CURRENT_VOLUME ( BPTR );

 FUNCTION
Inquire about a file's volume and unit number

 INPUTS
fh_Args - the fh_Args field of the active filehandle, or pointer to a
          filehandle.  If NULL, return current volume and unit number.

 RESULTS
res1 - BPTR to current volume node where file is located
res2 - unit number where file is located
```

1.5 DosPackets/ACTION_DELETE_OBJECT

```
 NAME
ACTION_DELETE_OBJECT -- delete a file or an empty directory

 SYNOPSIS
ACTION_DELETE_OBJECT ( lock, name )
                       Arg1  Arg2

ACTION_DELETE_OBJECT ( BPTR, BSTR );

 FUNCTION
Delete an object.

 INPUTS
lock - lock on directory containing object or NULL for root of device
name - name of object to delete.

 RESULTS
res1 - DOSTRUE if successful, else DOSFALSE.
res2 - If res1 was DOSFALSE, this tells you why the error occured.

 NOTES
If the object to be deleted is a directory, and it still contains
files, an error will be returned.

 SEE ALSO
dos.library/DeleteFile()
```

1.6 DosPackets/ACTION_DIE

```
 NAME
ACTION_DIE -- unload handler code from memory and free buffers

 SYNOPSIS
ACTION_DIE ()

ACTION_DIE ( void );
```

```
 FUNCTION
Tell a handler to unload and flush.

 INPUTS
none

 RESULTS
res1 - DOSTRUE if successful, else DOSFALSE.
res2 - If res1 was DOSFALSE, this tells you why the error occured.
```

1.7 DosPackets/ACTION_DISK_INFO

```
 NAME
ACTION_DISK_INFO -- obtain information about a disk

 SYNOPSIS
ACTION_DISK_INFO(  id  )
                  Arg1

ACTION_DISK_INFO( BPTR );

 FUNCTION
Fill the specified InfoData structure with information about the
device.

 INPUTS
id - BPTR to an InfoData structure.

 RESULT
res1 - DOSTRUE if successful, else DOSFALSE.
res2 - If res1 was DOSFALSE, this tells you why the error occured.

 NOTES
When sent to a console handler, this packet now returns not only the
window pointer in the id_VolumeNode field, but also a pointer to the
console handler's console IO block in the id_InUse field.  (These
fields are part of the InfoData structure initialized by
ACTION_DISK_INFO).  Remember that you must AllocMem your InfoData
structure to assure longword alignment since a BPTR to this structure
is arg[0] for the packet.

A pointer to the ConUnit structure (see devices/conunit.h, .i) can be
found from the returned console IO block pointer:

conUnit = (struct ConUnit *)
      ((struct IOStdReq *)infoData->id_InUse)->io_Unit;

There is a lot of useful information in the ConUnit structure such as
text cursor position and limits.  If you are using the exec
console.device directly, you should be able to get the ConUnit pointer
from yourIoRequest->io_Unit.

 SEE ALSO
```

```
dos.library/Info()
```

1.8 DosPackets/ACTION_END

```
 NAME
ACTION_END -- terminate access to a file via a handle.

 SYNOPSIS
ACTION_END ( fh_Args )
              Arg1

ACTION_END ( LONG );

 FUNCTION
Terminate access to a filehandle. Send this packet when done reading or
writing from or to a file. Don't reuse the filehandle after sending
ACTION_END, and don't send it more than once - doing so has been known
to trash entire hard disks...

 INPUTS
fh_Args - the fh_Args field of the FileHandle.

 RESULTS
res1 - DOSTRUE if successful, else DOSFALSE.
res2 - If res1 was DOSFALSE, this tells you why the error occured.

 NOTES
I don't think this flushes the device buffers.

 SEE ALSO
dos.library/Close()
```

1.9 DosPackets/ACTION_EXAMINE_FH

```
 NAME
ACTION_EXAMINE_FH -- examine an already open file

 SYNOPSIS
ACTION_EXAMINE_FH ( fh_Args,  fib  )
                      Arg1   Arg2

ACTION_EXAMINE_FH ( LONG, BPTR );

 FUNCTION
Examine an already opened file.

 INPUTS
fh_Args - the fh_Args field of the file's filehandle.
fib     - BPTR to a FileInfoBlock to recieve info.

 RESULTS
res1 - DOSTRUE if successful, else DOSFALSE.
```

res2 - If res1 was DOSFALSE, this tells you why the error occured.

SEE ALSO
dos.library/ExamineFH()

1.10 DosPackets/ACTION_EXAMINE_OBJECT

NAME
ACTION_EXAMINE_OBJECT -- obtain information about an object

SYNOPSIS

```
ACTION_EXAMINE_OBJECT ( lock, fib )
                        Arg1  Arg2

ACTION_EXAMINE_OBJECT ( BPTR, BPTR );
```

FUNCTION
Obtain information about the specified object.

INPUTS
lock - a lock on the object you wish to examine.
fib - BPTR to a FileInfoBlock to hold the information.

RESULTS
res1 - DOSTRUE if successful, else DOSFALSE.
res2 - If res1 was DOSFALSE, this tells you why the error occured.

SEE ALSO
dos.library/Examine()

1.11 DosPackets/ACTION_EXAMINE_NEXT

NAME
ACTION_EXAMINE_NEXT -- examine next entry in a directory

SYNOPSIS

```
ACTION_EXAMINE_NEXT ( lock, fib )
                      Arg1  Arg2

ACTION_EXAMINE_NEXT ( BPTR, BPTR );
```

FUNCTION
Obtain information on the next entry in a directory. When you send this packet initially, the lock must be on the directory you wish to examine, and the FileInfoBlock must have been initialised from a previous invoation of
ACTION_EXAMINE_OBJECT
on the directory.

For more detailed information, see dos.library/ExNext().

INPUTS

```
lock - a lock on the directory you wish to examine.
fib  - BPTR to a previously initialised FileInfoBlock.

 RESULTS
res1 - DOSTRUE if successful, else DOSFALSE.
res2 - If res1 was DOSFALSE, this tells you why the error occured.
       The usual cause of failure is ERROR_NO_MORE_ENTRIES.

 SEE ALSO
dos.library/ExNext()
```

1.12 DosPackets/ACTION_FH_FROM_LOCK

```
 NAME
ACTION_FH_FROM_LOCK -- open a file from a lock

 SYNOPSIS
ACTION_FH_FROM_LOCK (   fh, lock )
                      Arg1  Arg2

ACTION_FH_FROM_LOCK ( BPTR, BPTR )

 FUNCTION
Open a file from a previously existing lock. This effectively
relinquishes the lock.  The access type is determined by the type of
the lock you pass in - SHARED_LOCK is similar to MODE_OLDFILE, whereas
EXCLUSIVE_LOCK is similar to MODE_NEWFILE.

 INPUTS
fh   - pointer to previously allocated filehandle
lock - lock on the file to open

 RESULTS
res1 - DOSTRUE if successful, else DOSFALSE.
res2 - If res1 was DOSFALSE, this tells you why the error occured.

 NOTES
If the open failed, the lock is still usable.

 SEE ALSO
dos.library/OpenFromLock()
```

1.13 DosPackets/ACTION_FINDINPUT

```
 NAME
ACTION_FINDINPUT -- open an existing file for input

 SYNOPSIS
ACTION_FINDINPUT (  fh,  lock, name )
                   Arg1  Arg2  Arg3

ACTION_FINDINPUT ( BPTR, BPTR, BSTR );
```

```
 FUNCTION
Open a previously existing file for reading.

 INPUTS
fh   - BPTR to a previously allocated FileHandle.
lock - lock on the directory holding file 'name'.
name - BPTR to BCPL string specifying name of the file to open.

 RESULTS
res1 - DOSTRUE if successful, else DOSFALSE.
res2 - If res1 was DOSFALSE, this tells you why the error occured.

 SEE ALSO
dos.library/Open()
```

1.14 DosPackets/ACTION_FINDOUTPUT

```
 NAME
ACTION_FINDOUTPUT -- open a new file for output

 SYNOPSIS
ACTION_FINDOUTPUT (  fh,  lock, name )
                    Arg1  Arg2  Arg3

ACTION_FINDOUTPUT ( BPTR, BPTR, BSTR );

 FUNCTION
Open a new file for writing, overwriting any file of the same name
which already exists.

 INPUTS
fh   - BPTR to a previously allocated FileHandle.
lock - lock on the directory holding file 'name'.
name - BPTR to BCPL string specifying name of the file to open.

 RESULTS
res1 - DOSTRUE if successful, else DOSFALSE.
res2 - If res1 was DOSFALSE, this tells you why the error occured.

 SEE ALSO
dos.library/Open()
```

1.15 DosPackets/ACTION_FINDUPDATE

```
 NAME
ACTION_FINDUPDATE -- open a file for updating

 SYNOPSIS
ACTION_FINDUPDATE (  fh,  lock, name )
                    Arg1  Arg2  Arg3
```

```
ACTION_FINDUPDATE ( BPTR, BPTR, BSTR );
```

FUNCTION

Open a file for reading and writing, creating it if it didn’t exist.

INPUTS

```
fh   - BPTR to a previously allocated FileHandle.
lock - lock on the directory holding file ’name’.
name - name of the file to open.
```

RESULTS

res1 - DOSTRUE if successful, else DOSFALSE.
res2 - If res1 was DOSFALSE, this tells you why the error occured.

SEE ALSO

dos.library/Open()

1.16 DosPackets/ACTION_FLUSH

NAME

ACTION_FLUSH -- flush contents of device’s I/O buffers

SYNOPSIS

```
ACTION_FLUSH ()

ACTION_FLUSH ( void );
```

FUNCTION

Cause pending blocks to be written out and motor turned off. This is expensive, so should not be done after every write. It is used by the system before putting up a requester saying "Change Disk" and the packet is only returned when the job is done. This action would be useful in a database when it wished to commit.

RESULT

res1 - DOSTRUE if successful, else DOSFALSE.
res2 - If res1 was DOSFALSE, this tells you why the error occured.

SEE ALSO

dos.library/Flush()

1.17 DosPackets/ACTION_FORMAT

NAME

ACTION_FORMAT -- initialise a filesystem for use

SYNOPSIS

```
ACTION_FORMAT ( newname, dostype )
                  Arg1     Arg2

ACTION_FORMAT ( BSTR, LONG );
```

```
 FUNCTION
Initialise a filesystem.  This function performs no kind of low level
formatting of media whatsoever.

 INPUTS
newname - name of new volume to create
dostype - DosType to use, if filesystem supports multiple types.

 RESULTS
res1 - DOSTRUE if successful, else DOSFALSE.
res2 - If res1 was DOSFALSE, this tells you why the error occured.

 SEE ALSO
dos.library/Format()
```

1.18 DosPackets/ACTION_FREE_LOCK

```
                 NAME
ACTION_FREE_LOCK -- free a lock on an object

 SYNOPSIS
ACTION_FREE_LOCK ( lock )
                   Arg1

ACTION_FREE_LOCK ( BPTR );

 FUNCTION
Free a lock that was previously obtained on an object. THE OBJECT MUST
RESIDE ON THE FILESYSTEM YOU ARE SENDING THE PACKET TO!

 INPUTS
lock - the lock to free

 RESULTS
res1 - DOSTRUE if successful, else DOSFALSE.
res2 - If res1 was DOSFALSE, this tells you why the error occured.

 SEE ALSO

              ACTION_LOCATE_OBJECT
              , dos.library/UnLock()
```

1.19 DosPackets/ACTION_INFO

```
 NAME
ACTION_INFO -- obtain information about a handler

 SYNOPSIS
ACTION_INFO ( lock,  id  )
              Arg1  Arg2

ACTION_INFO ( BPTR, BPTR )
```

FUNCTION
Obtain information about the disk on which an object resides.

INPUTS
lock - lock on an object which is on the device to be examined
id   - BPTR to an InfoData structure to hold the information.

RESULTS
res1 - DOSTRUE if successful, else DOSFALSE.
res2 - If res1 was DOSFALSE, this tells you why the error occured.

SEE ALSO
dos.library/Info()

1.20 DosPackets/ACTION_INHIBIT

NAME
ACTION_INHIBIT -- prevent user from accessing a filesystem

SYNOPSIS

```
ACTION_INHIBIT ( flag )
                 Arg1

ACTION_INHIBIT ( LONG );
```

FUNCTION
Inhibit a filesystem, i.e. prevent operations from taking place on it until it is un-inhibited. Commonly used to disable file operations before formatting or carrying out similar maintenance on a device.

The volume appears to the user as 'Not a DOS Disk'.

When called via dos.library, an
ACTION_FLUSH
is usually sent to the
handler first.

INPUTS
flag - 0 to uninhibit, anything other than 0 to inhibit.

RESULTS
res1 - DOSTRUE if successful, else DOSFALSE.
res2 - If res1 was DOSFALSE, this tells you why the error occured.

SEE ALSO
dos.library/Inhibit()

1.21 DosPackets/ACTION_IS_FILESYSTEM

NAME
ACTION_IS_FILESYSTEM -- determine if handler is a filesystem

```
 SYNOPSIS
ACTION_IS_FILESYSTEM ( )

ACTION_IS_FILESYSTEM ( void );

 FUNCTION
Determine if handler is a filesystem.

 RESULTS
res1 - DOSTRUE if handler is a filesystem, else DOSFALSE.
res2 - If res1 was DOSFALSE, this tells you why the error occured.

 SEE ALSO
dos.library/IsFileSystem()
```

1.22 DosPackets/ACTION_LOCATE_OBJECT

```
                 NAME
ACTION_LOCATE_OBJECT -- locate an object ('lock' it)

 SYNOPSIS
ACTION_LOCATE_OBJECT ( lock, name, type )
                       Arg1  Arg2  Arg3

ACTION_LOCATE_OBJECT ( BPTR, BSTR, LONG );

 FUNCTION
Locate an object and return a FileLock on it.

 INPUTS
lock - lock on a directory. NULL for root directory.
name - A pathname relative to 'lock'.
type - Type of lock to obtain. Can be SHARED_LOCK or EXCLUSIVE_LOCK.

 RESULTS
res1 - Lock on the specified object or DOSFALSE.
res2 - If res1 was DOSFALSE, this tells you why the error occured.

 SEE ALSO

              ACTION_FREE_LOCK
              , dos.library/Lock()
```

1.23 DosPackets/ACTION_MORE_CACHE

```
 NAME
ACTION_MORE_CACHE -- modify amount of cache buffers

 SYNOPSIS
ACTION_MORE_CACHE( buffers )
                     Arg1
```

```
ACTION_MORE_CACHE( LONG );

 FUNCTION
Change the amount of cache buffers allocated by a handler.

 INPUTS
buffers - signed longword integer specifying number of cache buffers to
          create or delete. 0 returns the current number.

 RESULT
res1 - the new amount of cache buffers held by the handler.
res2 - Secondary return code. You get this even if packet was successful

 NOTES
Older handlers may only support the creation of new buffers and not the
deallocation of them. Examples include the release 1.3 FileSystem.

 SEE ALSO
dos.library/AddBuffers()
```

1.24 DosPackets/ACTION_PARENT

```
 NAME
ACTION_PARENT -- obtain lock on parent directory of object

 SYNOPSIS
ACTION_PARENT ( lock )
                Arg1

ACTION_PARENT ( BPTR );

 FUNCTION
Obtains a lock on the parent directory of 'lock', which can either be a
file or a directory.

For more detailed information see dos.library/ParentDir().

 INPUTS
lock - lock to find parent of.

 RESULTS
res1 - Lock on parent directory if successful, else DOSFALSE.
res2 - If res1 was DOSFALSE, this tells you why the error occured.

 SEE ALSO
dos.library/ParentDir()
```

1.25 DosPackets/ACTION_PARENT_FH

```
 NAME
ACTION_PARENT_FH -- obtain parent directory of an open file
```

```
 SYNOPSIS
ACTION_PARENT_FH ( fh_Args )
                    Arg1

ACTION_PARENT_FH ( BPTR );

 FUNCTION
duplicate the lock on the parent dir of the specified file

 INPUTS
fh_Args - the fh_Args field of the file's filehandle structure.

 RESULTS
res1 - Lock on parent directory if successful, else DOSFALSE.
res2 - If res1 was DOSFALSE, this tells you why the error occured.

 SEE ALSO
dos.library/ParentOfFH()
```

1.26 DosPackets/ACTION_READ

```
                 NAME
ACTION_READ -- read bytes of data from specified stream

 SYNOPSIS
ACTION_READ ( fh_Args,  buf, numbytes )
                Arg1   Arg2    Arg3

ACTION_READ ( LONG, APTR , ULONG );

 FUNCTION
Read data from specified filehandle, obtained with
              ACTION_FINDINPUT
               or

              ACTION_FINDUPDATE
              .

 INPUTS
fh_Args  - the fh_Args field of the filehandle obtained with

              ACTION_FINDINPUT
               or
              ACTION_FINDUPDATE
              .
buf      - pointer to the buffer which is to receive the data.
numbytes - number of bytes to read from file.

 RESULTS
res1 - actual number of bytes read, or DOSFALSE if failed.
res2 - If res1 was DOSFALSE, this tells you why the error occured.

 SEE ALSO
```

```
dos.library/Read()
```

1.27 DosPackets/ACTION_READ_LINK

```
 NAME
ACTION_READ_LINK -- resolve a soft link

 SYNOPSIS
ACTION_READ_LINK ( lock, path, buf, size )
                   Arg1  Arg2  Arg3 Arg4

ACTION_READ_LINK ( BPTR, STRPTR, STRPTR, ULONG );

 FUNCTION
Resolve a soft link.

 INPUTS
lock - lock this path is relative to filesystem
path - path that caused an ERROR_IS_SOFT_LINK
buf  - pointer to buffer for new path from handler
size - size of buffer.

 RESULTS
res1 - DOSTRUE if successful, else DOSFALSE.
res2 - If res1 was DOSFALSE, this tells you why the error occured.

 SEE ALSO
dos.library/ReadLink()
```

1.28 DosPackets/ACTION_RENAME_DISK

```
 NAME
ACTION_RENAME_DISK -- rename a volume.

 SYNOPSIS
ACTION_RENAME_DISK( name )
                    Arg1

ACTION_RENAME_DISK( BSTR );

 FUNCTION
Set the volume name of a device.

 INPUTS
name - pointer to a string containing the new name of the device.

 RESULTS
res1 - DOSTRUE if successful, else DOSFALSE.
res2 - If res1 was DOSFALSE, this tells you why the error occured.

 SEE ALSO
dos.library/Relabel()
```

1.29 DosPackets/ACTION_RENAME_OBJECT

```
 NAME
ACTION_RENAME_OBJECT -- rename an object

 SYNOPSIS
ACTION_RENAME_OBJECT ( lock, name, newlock, newname )
                       Arg1  Arg2    Arg3     Arg4

ACTION_RENAME_OBJECT ( BPTR, BSTR, BPTR, BSTR );

 FUNCTION
Rename an object.

 INPUTS
lock    - Lock on directory holding object (NULL for root).
name    - Name of the object to be renamed / moved.
newlock - Lock on directory to move object to (NULL for root).
newname - New name for the object.

 RESULTS
res1 - DOSTRUE if successful, else DOSFALSE.
res2 - If res1 was DOSFALSE, this tells you why the error occured.

 SEE ALSO
dos.library/Rename()
```

1.30 DosPackets/ACTION_SAME_LOCK

```
 NAME
ACTION_SAME_LOCK -- determine if two locks refer to same object (V36)

 SYNOPSIS
ACTION_SAME_LOCK ( lock1, lock2 )
                   Arg1   Arg2

ACTION_SAME_LOCK ( BPTR, BPTR );

 FUNCTION
Test if two locks are on the same object.

 RESULTS
res1 - DOSTRUE if both locks refer to same object, else DOSFALSE.
res2 - If res1 was DOSFALSE, this tells you why the error occured.

 SEE ALSO
dos.library/SameLock()
```

1.31 DosPackets/ACTION_SCREEN_MODE

```
 NAME
ACTION_SCREEN_MODE -- set the mode of a CON: handler
```

```
 SYNOPSIS
ACTION_SCREEN_MODE ( rawMode )
                       Arg1

ACTION_SCREEN_MODE ( ULONG );

 FUNCTION
Switches CON: into raw mode and back again.

 INPUTS
rawMode - DOSTRUE for RAW: mode, DOSFALSE for CON: mode.

 RESULT
res1 - DOSTRUE if successful, else DOSFALSE.
res2 - If res1 was DOSFALSE, this tells you why the error occured.

 NOTES
The packet should be sent to MsgPort of the console's handler task,
which is usually found in '(struct MsgPort *) process->pr_ConsoleTask'.

Note that in addition to this, an escape sequence may be sent to turn
on or off the automatic translation of LF to CR/LF.  Normally RAW:
does not enable this and CON:  does.  SetRawMode does not affect the
translation.  The escape sequences are CSI 20h to enable, and CSI 20l
to disable this translation.

 SEE ALSO
dos.library/SetMode()
```

1.32 DosPackets/ACTION_SEEK

```
 NAME
ACTION_SEEK -- seek to a new position in a file

 SYNOPSIS
ACTION_SEEK ( fh_Args,  pos, offset )
                Arg1   Arg2   Arg3

ACTION_SEEK ( LONG, LONG, LONG );

 FUNCTION
Change the position of the cursor within a file for reading or writing.

 INPUTS
fh_Args - the fh_Args field of the file to seek within.
pos     - current position in file.
offset  - seek offset relative to 'pos'.

 RESULTS
res1 - previous seek position in file, or DOSTRUE if the seek failed.
res2 - If res1 was DOSTRUE, this tells you why the error occured.

 SEE ALSO
```

```
dos.library/Seek()
```

1.33 DosPackets/ACTION_SERIALIZE_DISK

```
 NAME
ACTION_SERIALIZE_DISK -- make a filesystem unique (V39)

 SYNOPSIS
ACTION_SERIALIZE_DISK ()

ACTION_SERIALIZE_DISK ( void );

 FUNCTION
Takes no parameters, makes the FS change the disk to make it unique
(normally by changing the creation date).  Useful in diskcopy with
non-amiga filesystems (diskcopy knows about amiga filesystems and can
do it without the packet for them).  Normal sorts of return codes.

 INPUTS
none

 RESULT
res1 - DOSTRUE if successful, else DOSFALSE.
res2 - If res1 was DOSFALSE, this tells you why the error occured.
```

1.34 DosPackets/ACTION_SET_COMMENT

```
 NAME
ACTION_SET_COMMENT -- set the comment for an object

 SYNOPSIS
ACTION_SET_COMMENT (   0L, lock, name, comment )
                     Arg1  Arg2  Arg3    Arg4

ACTION_SET_COMMENT ( ULONG, BPTR, BSTR, BSTR );

 FUNCTION
Sets the file note / comment on 'file' in directory specified by 'lock'
to 'comment'.

 INPUTS
lock    - lock on directory containing file
name    - name of file
comment - the comment

 RESULTS
res1 - DOSTRUE if successful, else DOSFALSE.
res2 - If res1 was DOSFALSE, this tells you why the error occured.

 SEE ALSO
dos.library/SetComment()
```

1.35 DosPackets/ACTION_SET_DATE

```
 NAME
ACTION_SET_DATE -- set 'last modified' date of an object

 SYNOPSIS
ACTION_SET_DATE ( 0L, lock, fname, ds )
                  Arg1 Arg2  Arg3  Arg4

ACTION_SET_DATE ( ULONG, BPTR, BSTR, APTR );

 FUNCTION
Sets the date of a file or directory to specified date.

 INPUTS
lock - lock on ParentDir of file
fname - BPTR to BCPL string of filename
ds - APTR to a DateStamp structure

 RESULT
res1 - DOSTRUE if successful, else DOSFALSE.
res2 - If res1 was DOSFALSE, this tells you why the error occured.

 NOTES
Arg1 should always be set to NULL.

 SEE ALSO
dos.library/SetFileDate()
```

1.36 DosPackets/ACTION_SET_OWNER

```
                 NAME
ACTION_SET_OWNER -- set ownership information for an object (V39)

 SYNOPSIS
ACTION_SET_OWNER ( NULL, lock,  name, owner_info )

ACTION_SET_OWNER ( ULONG, BPTR, BSTR, LONG );

 FUNCTION
Sets the ownership information for an object.  Similar arguments to

              ACTION_SET_PROTECT
              , except the data is a longword of owner info:  high
16 bits are GID (group), low 16 are UID (user id).

 INPUTS
lock  - specifies parent directory of the object
name  - name of the object
owner_info  - High 16 bits are GID (group), low 16 are UID (user ID).

 RESULT
res1 - DOSTRUE if successful, else DOSFALSE.
res2 - If res1 was DOSFALSE, this tells you why the error occured.
```

```
   SEE ALSO

                ACTION_SET_PROTECT
                , dos.library/SetOwner()
```

1.37 DosPackets/ACTION_SET_PROTECT

```
                   NAME
  ACTION_SET_PROTECT -- set attributes of an object

   SYNOPSIS
  ACTION_SET_PROTECT ( NULL, lock,  name, attrs )

  ACTION_SET_PROTECT ( ULONG, BPTR, BSTR, LONG );

   FUNCTION
  Sets the attributes or 'protection bits' for an object.

  See <dos/dos.h> for a list of these attributes.

   INPUTS
  lock  - specifies parent directory of the object
  name  - name of the object
  attrs - longword of flags specifying file attributes

   RESULT
  res1 - DOSTRUE if successful, else DOSFALSE.
  res2 - If res1 was DOSFALSE, this tells you why the error occured.

   SEE ALSO
  dos.library/SetProtection(),
                ACTION_SET_OWNER
```

1.38 DosPackets/ACTION_WRITE

```
                   NAME
  ACTION_WRITE -- write bytes of data to specified stream

   SYNOPSIS
  ACTION_WRITE ( fh_Args,  buf, numbytes )
                   Arg1   Arg2    Arg3

  ACTION_WRITE ( LONG, APTR , ULONG );

   FUNCTION
  Write data to specified filehandle, obtained with
                ACTION_FINDOUTPUT
                 or

                ACTION_FINDUPDATE
                .
```

```
 INPUTS
fh_Args  - the fh_Args field of the filehandle obtained with

             ACTION_FINDOUTPUT
              or
             ACTION_FINDUPDATE
             .
buf      - pointer to the buffer which contains the data to be written.
numbytes - number of bytes to write to the file.

 RESULTS
res1 - actual number of bytes written, or DOSFALSE if failed.
res2 - If res1 was DOSFALSE, this tells you why the error occured.

 SEE ALSO
dos.library/Write()
```

1.39 DosPackets/ACTION_WRITE_PROTECT

```
 NAME
ACTION_WRITE_PROTECT -- 'write protect' a filesystem (V34)

 SYNOPSIS
ACTION_WRITE_PROTECT ( flag,  key )
                       Arg1  Arg2

ACTION_WRITE_PROTECT ( LONG, LONG );

 FUNCTION
Prevent write access to a filesystem using a key.

 INPUTS
flag - DOSTRUE to lock device, DOSFALSE to unlock.
key  - Key to lock drive with. Required for unlocking.

 RESULTS
res1 - DOSTRUE if successful, else DOSFALSE.
res2 - If res1 was DOSFALSE, this tells you why the error occured.

 NOTES
The AmigaDOS 'Lock' command uses a 'key' of 0 if no passkey is
specified on the command line, otherwise it uses a simple encoding
algorithm on the key supplied.

This is a method of software write protection, nothing more.
```