

**Sourcecode: ExNext.c**

**COLLABORATORS**

	<i>TITLE :</i> Sourcecode: ExNext.c		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		February 12, 2023	

**REVISION HISTORY**

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>Sourcecode: ExNext.c</b>	<b>1</b>
1.1	ExNext.c . . . . .	1

## Chapter 1

# Sourcecode: ExNext.c

### 1.1 ExNext.c

```
/* ExNext.c   V1.0   93-09-27                               */
/* ROM library: "dos.library/ExNext", (All versions) */
/* Copyright 1993, Anders Bjerin, Amiga C Club           */

#include <dos/dos.h>
#include <exec/memory.h>

#include <clib/dos_protos.h>
#include <clib/exec_protos.h>
#include <stdio.h>
#include <stdlib.h>

UBYTE *version = "$VER: ExNext 1.0";

int main( int argc, char *argv[] );
int main( int argc, char *argv[] )
{
    BPTR my_lock;
    struct FileInfoBlock *my_fib;

    /* 1. Lock the directory: */
    my_lock = Lock( "RAM:", SHARED_LOCK );
    if( !my_lock )
    {
        printf( "Could not lock the object!\n" );
        exit( 20 );
    }

    /* 2. Allocate a FileInfoBlock structure: */
    my_fib = (struct FileInfoBlock *)
        AllocMem( sizeof( struct FileInfoBlock ), MEMF_ANY | MEMF_CLEAR );
    if( !my_fib )
    {
        printf( "Not enough memory!\n" );
        UnLock( my_lock );
        exit( 21 );
    }
};
```

```
/* 3. Examine the locked object: */
if( Examine( my_lock, my_fib ) )
{
    /* 4. Check if it is a directory or volume: */
    if( my_fib->fib_DirEntryType > 0 )
    {
        printf( "%s\n", my_fib->fib_FileName );

        /* 5. As long as we find objects we stay in the loop: */
        while( ExNext( my_lock, my_fib ) )
            printf( " %s\n", my_fib->fib_FileName );

        /* Were all objects listed? */
        if( IoErr() == ERROR_NO_MORE_ENTRIES )
            printf( "No more files!\n" );
        else
            printf("Error while reading!\n");
    }
    else
        printf( "This program needs a directory or volume name!\n" );
}
else
    printf( "Could not examine the object!\n" );

FreeMem( my_fib, sizeof( struct FileInfoBlock ) );
UnLock( my_lock );
exit( 0 );
}
```