Sourcecode: Examine.c

COLLABORATORS

	TITLE : Sourcecode: Examine.c	

ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		February 12, 2023	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

Sourcecode: Examine.c

1.1 Examine.c

```
/* Examine.c   V1.0   93-09-27                          */
/* ROM library: "dos.library/Examine", (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: Examine 1.0";

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


  /* 1. Lock the object we want to examine: */
  my_lock = Lock( "C:Dir", 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: */
  ok = Examine( my_lock, my_fib );
  if( !ok )
  {
    printf( "Could not examine the object!\n" );
    FreeMem( my_fib, sizeof( struct FileInfoBlock ) );
    UnLock( my_lock );
    exit( 22 );
  }

  /* 4. Print some info about the object: */
  printf( "Name:      %s\n", my_fib->fib_FileName );
  printf( "Size:      %d\n", my_fib->fib_Size );

  /* 5. Clean up: */
  FreeMem( my_fib, sizeof( struct FileInfoBlock ) );
  UnLock( my_lock );
  exit( 0 );
}
```