

**Sourcecode: Example6.c**

**COLLABORATORS**

	<i>TITLE :</i> Sourcecode: Example6.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: Example6.c</b>	<b>1</b>
1.1	Example6.c . . . . .	1

## Chapter 1

# Sourcecode: Example6.c

### 1.1 Example6.c

```
/******  
/*  
/* Amiga C Encyclopedia (ACE)           Amiga C Club (ACC) */  
/* -----  
/*  
/* Manual:  AmigaDOS                    Amiga C Club      */  
/* Chapter: Parsing Command Line        Tulevagen 22     */  
/* File:    Example6.c                  181 41  LIDINGO    */  
/* Author:  Anders Bjerin                SWEDEN          */  
/* Date:    93-03-06                     */  
/* Version: 1.0                           */  
/*  
/* Copyright 1993, Anders Bjerin - Amiga C Club (ACC) */  
/*  
/* Registered members may use this program freely in their */  
/* own commercial/noncommercial programs/articles.      */  
/*  
/******  
  
/* This is a simple example on how to use the ReadItem() */  
/* function. It will simply collect all command line     */  
/* arguments (items) and print them each on one line     */  
/* together with some extra information. (If the item was */  
/* inside quotation marks or not, if it is an equal sign, */  
/* if there was an error etc...)                          */  
  
/* Include the dos library definitions: */  
#include <dos/dos.h>  
  
/* Include information about the argument parsing routine: */  
#include <dos/rdargs.h>  
  
/* Now we include the necessary function prototype files: */  
#include <clib/dos_protos.h> /* General dos functions... */  
#include <clib/exec_protos.h> /* System functions... */  
#include <stdio.h> /* Std functions [printf()...] */
```

```
#include <stdlib.h>                                /* Std functions [exit()...] */

/* Store up to 50 characters in our string buffer: */
#define BUFFER_SIZE 50

/* Set name and version number: */
UBYTE *version = "$VER: AmigaDOS/ParsingCommandLine/Example6 1.0";

/* Declare an external global library pointer to the Dos library: */
extern struct DosLibrary *DOSBase;

/* Declared our own functions: */

/* Our main function: */
int main( int argc, char *argv[] );

/* Main function: */

int main( int argc, char *argv[] )
{
    /* Store the item type value here: (Defined */
    /* in header file "dos/dos.h".)          */
    LONG item_type;

    /* Temporary string buffer to store the item name in: */
    UBYTE item_name[ BUFFER_SIZE ];

    /* We need dos library version 37 or higher: */
    if( DOSBase->dl_lib.lib_Version < 37 )
    {
        /* The user's dos library is too old! */
        printf( "This program needs Dos Library V37 or higher!\n" );

        /* Exit with an error code: */
        exit ( 20 );
    }

    /* Collect an item: (Store the item name in the string */
    /* buffer. Up to BUFFER_SIZE characters can be stored, */
    /* and collect the items from the default input handler. */
    item_type = ReadItem( item_name, BUFFER_SIZE, NULL );

    /* As long as we find items we stay in the while loop: */
    while( item_type )
```

---

```
{
/* Print the item type: */
switch( item_type )
{
    case ITEM_EQUAL:      printf( "Equal symbol  " ); break;
    case ITEM_ERROR:     printf( "Item ERROR    " ); break;
    case ITEM_UNQUOTED:  printf( "Unquoted item " ); break;
    case ITEM_QUOTED:    printf( "Quoted item  " ); break;
    default:             printf( "Unknown item! " );
}

/* Print the item string: */
printf( "%s\n", item_name );

/* Collect next item: */
item_type = ReadItem( item_name, BUFFER_SIZE, NULL );
}

/* No more items (item_type is equal to "ITEM_NOTHING"): */
printf( "No more items!\n" );

/* The End! */
exit( 0 );
}
```

---