



# Drive Utilities DLL

## Quick Start

[Overview](#)  
[Installation](#)  
[A Brief Tour](#)  
[Benefits of Registration](#)

## How to Use the Drive Utilities DLL

[General Information](#)  
[The DrvDemo Program](#)  
[Calling Conventions](#)

## Everything Else

[Support & Registration](#)  
[The Product Line](#)  
[Order Form \(Print Me!\)](#)  
[Legalese](#)  
[Revision History](#)

# Overview

**Drive Utilities DLL** is a collection of disk and file routines for the Visual Basic programmer. Disk routines will provide disk size, cluster size, sector size, and bytes used and free, which are not available through Visual Basic. File routines will provide file size, date, time and attributes, and allow date, time and attributes to be changed. The **Drive Utilities DLL** File ZAPP routine can ZAPP a file by overwriting its clusters with a pattern, then deleting it.

# Installation

The Drive Utilities package contains the following files:

DRVUTILS.DLL	dynamic link library
DRVUTILS.HLP	documentation in Help file
DRVUTILS.BAS	declarations and constants
DRVUTILS.TXT	installation instructions
DRVDEMO.MAK	Visual Basic demo make file
DRVDEMO.FRM	Visual Basic demo main form, in ASCII
DRVDEMO.EXE	Visual Basic demo executable program

Include the .BAS file in your Visual Basic project. Place the .DLL file somewhere on the path. The Visual Basic runtime library VBRUN300.DLL is required to run the demo executable program, and is not included in this archive.

# A Brief Tour

The **Drive Utilities DLL** consists of the following routines:

DiskBytesTotal	Number of bytes on drive
DiskBytesUsed	Number of bytes used on drive
DiskBytesFree	Number of bytes free on drive
DiskBytesSector	Number of bytes per sector
DiskBytesCluster	Number of bytes per cluster
FileGetBytes	Get size of file
FileGetAttributes	Get file attributes
FileSetAttributes	Change file attributes
FileGetDateTime	Get file date/time
FileSetDateTime	Change file date/time
FileZAPP	Overwrite file and delete

# Benefits of Registration

When you register **Drive Utilities DLL**, you get the following...

**Full source code to the .DLL.** The source is standard 'C', can be compiled using Microsoft Visual C++ 1.0 and 1.5, and could probably be ported to Borland with little trouble. And it's fairly well commented.

**Distribution rights.** You'll need this to send **Drive Utilities DLL** out with your code that uses it. Perfect for VB shareware authors.

**Possibly a later version, with more features and/or fewer bugs.** There are a few more routines that could be added, such as disk serial numbers and such. And, despite all testing, there are always bugs.

**A clear conscience.** Bound to be worth something.

## General Information

The essential part of the **Drive Utilities DLL** consists of the .DLL file, containing the executable routines, and the .BAS file containing the declarations, constants and date/time structure.

Add the .BAS file to your Visual Basic project. Put the .DLL file anywhere on the path (generally, in \WINDOWS or \WINDOWS\SYSTEM).

# The DrvDemo Program

The DrvDemo program contains code that calls all of the **Drive Utilities DLL** routines. The .FRM file is ASCII, so you don't need Visual Basic to view it. The demo program operates as follows:

Select a disk drive from the drop-down drive box. The program will then get information about the selected drive, and update the directory and file list boxes.

Select a file, changing the directory as necessary. Double-clicks are required to change the directory and file. When you select a file, the program will get information about that file, and update the file information area.

To change file attributes, select a file, select the new attributes using the check boxes, then press the **Set Attributes** button.

To change the date and time, select a file, enter the new values in the text boxes, then press the **Set Time/Date** button.

To delete the file by overwriting it, select the file and press the **ZAPP This File** button.

# Calling Conventions

DiskBytesTotal

DiskBytesUsed

DiskBytesFree

DiskBytesSector

DiskBytesCluster

FileGetBytes

FileGetAttributes

FileSetAttributes

FileGetDateTime

FileSetDateTime

FileZAPP



## DiskBytesTotal

DiskBytesTotal (ByVal Drv As String) As Long

Calling parameter      String containing drive designator

Return value            -1 if error  
                             otherwise total drive size

This routine accepts a string containing a drive designation or full path name, and returns the total number of bytes on the specified drive. The drive string must contain at least a letter and a colon ("A:"). If the string does not have the specified format, the space on the current drive is returned.

*NOTE: Since the size is returned in a Visual Basic long signed integer, the largest drive that will be accurate is 2 gigabytes (2,147,483,647 bytes).*

## DiskBytesUsed

DiskBytesUsed (ByVal Drv As String) As Long

Calling parameter      String containing drive designator

Return value            -1 if error  
                             otherwise number of bytes used

This routine accepts a string containing a drive designation or full path name, and returns the number of bytes in use on the specified drive. The drive string must contain at least a letter and a colon ("A:"). If the string does not have the specified format, the space used on the current drive is returned.

*NOTE: Since the size is returned in a Visual Basic long signed integer, the largest drive that will be accurate is 2 gigabytes (2,147,483,647 bytes).*

## DiskBytesFree

DiskBytesFree (ByVal Drv As String) As Long

Calling parameter      String containing drive designator

Return value            -1 if error  
                             otherwise number of bytes available

This routine accepts a string containing a drive designation or full path name, and returns the number of bytes free on the specified drive. The drive string must contain at least a letter and a colon ("A:"). If the string does not have the specified format, the free space on the current drive is returned.

*NOTE: Since the size is returned in a Visual Basic long signed integer, the largest free space that will be accurate is 2 gigabytes (2,147,483,647 bytes).*

## DiskBytesSector

DiskBytesSector (ByVal Drv As String) As Long

Calling parameter      String containing drive designator

Return value            -1 if error  
                         otherwise number of bytes per cluster

This routine accepts a string containing a drive designation or full path name, and returns the number of bytes per sector on the specified drive. The drive string must contain at least a letter and a colon ("A:"). If the string does not have the specified format, the sector size for the current drive is returned.

## **DiskBytesCluster**

DiskBytesCluster (ByVal Drv As String) As Long

Calling parameter      String containing drive designator

Return value            -1 if error  
                         otherwise number of bytes per cluster

This routine accepts a string containing a drive designation or full path name, and returns the number of bytes per cluster on the specified drive. The drive string must contain at least a letter and a colon ("A:"). If the string does not have the specified format, the cluster size for the current drive is returned.

## **FileGetBytes**

FileGetBytes(ByVal Fil As String) As Long

Calling parameter      String containing file specification

Return value            -1 if error  
                         otherwise number of bytes per cluster

This routine accepts a string containing a file specification. It returns the size of the file in bytes.

## **FileGetAttributes**

FileGetAttributes (ByVal Fil As String) As Integer

Calling parameter      String containing file specification

Return value            -1 if error  
                         otherwise file attributes

This routine accepts a string containing a file specification. The attributes are packed into the returned integer and are defined in DRVUTILS.BAS. Refer to the demo program for code to unpack the attributes.

## FileSetAttributes

FileSetAttributes (ByVal Fil As String, Attr As Integer) As Integer

Calling parameters	String containing file specification
	Integer containing new file attributes

Return value	0 if no errors
	EACCES          File locked
	ENOENT         File not found

This routine accepts a string containing a file specification and an integer with the new attributes. It returns an error code (defined in DRVUTILS.BAS). Refer to the demo program for code to pack the attributes.



## FileGetDateTime

FileGetDateTime (ByVal Fil As String, Yr As Integer, Mo As Integer, Dy As Integer, Hr As Integer, Mi As Integer, Se As Integer) As Integer

Calling parameters      String containing file specification  
                             Integers to hold year, month, day,  
                             hours, minutes and seconds

Return value              0 if no errors  
                             -1 if errors

This routine accepts a string containing a file specification and six integers to hold the returned time stamp.

## FileSetDateTime

FileSetDateTime (ByVal Fil As String, Yr As Integer, Mo As Integer, Dy As Integer, Hr As Integer, Mi As Integer, Se As Integer) As Integer

Calling parameters      String containing file specification  
Integers containing year, month, day,  
hours, minutes and seconds.

Return value              0 if no errors

BADDATE	Invalid date
BADTIME	Invalid time
EACCES	File locked
EBADF	File is read only
EINVAL	Sharing violation
ENOENT	File not found
EMFILE	Too many open handles

This routine accepts a string containing a file specification and six integers containing the new time stamp. Returned error codes are defined in DRVUTILS.BAS.

A date is considered to be invalid if the day is outside of the range of 1 - 31, the month is outside the range 1 - 12, or the year is before 1980. A time is considered to be invalid if the seconds are outside the range 0 - 59, the minutes are outside the range 0 - 59, or the hours outside the range 0 - 23. Seconds are stored as even numbers, so odd seconds are not available (sorry, that's how DOS works).

## FileZAPP

FileZAPP (ByVal Fil As String, Pattern As Integer) As Integer

Calling parameters      String containing file specification  
                             Integer containing byte pattern to write

Return value            0 if no errors  
                             EACCES            File locked  
                             EBADF            File is read only  
                             EINVAL            Sharing violation  
                             ENOENT           File not found  
                             EMFILE            Too many open handles

This routine accepts a string containing a file specification and an integer with a byte pattern. This routine opens the file, writes the pattern to all of the file's clusters, then closes and deletes it. Returned error codes are defined in DRVUTILS.BAS.

***WARNING: The ZAPPed file is NOT recoverable!***

*NOTE: If this routine returns an error, the file may be in an indeterminate state (i.e., partially overwritten).*

*NOTE: This routine could take a long time to zapp a large file. While it does yield to Windows after writing every cluster, it will not return to your code until it finishes writing to the file.*

## Support & Registration

Support is available by Email, postal mail, and telephone. Support is limited to the English language. All orders should be mailed to the Post Office box. Please print the help topic **Order Form** for registration. If you call, please call between the hours of 8 AM and 10 PM, Mountain Time. Since I have to work during the day, the best times to call are in the evening.

Dana W. Cline

TANSTAAFL Software  
P.O. Box 260075  
Lakewood, CO 80226  
USA

(303) 989-7389

CompuServe: 73700,3053  
GEnie: tanstaafI  
Internet: 73700.3053@compuserve.com

# The Product Line

2x4 Labeler

3x3 Labeler

DrvUtils.DLL

Farley

CRITTER

## **2x4 Labeler**

**2x4 Labeler** creates labels for 3 1/2" floppy disks using the **Avery L7664** label for the European A4 page size. **2x4 Labeler** is feature-identical with **3x3 Labeler**. Originally released on 03/06/94. Current version 1.51.

## **3x3 Labeler**

**3x3 Labeler** produces labels for 3 ½" floppy disks using the **Avery 5196**, which contains three rows of three labels each on an 8 ½" by 11" page, suitable for a laser or ink jet printer. Each label can contain file names, directory names, or multi-line text. Each label can have different fonts, sizes and edge/banner text, or can use the default global values. Different sets of label defaults can be loaded and saved by name. Originally released on 07/05/93. Current version 1.50.

## DrvUtils.DLL

The **Drive Utilities DLL** is a collection of disk and file routines for Visual Basic programmers. The disk routines allow you to get the total size of any drive, as well as the bytes used and free and sector and cluster sizes. File routines allow you to get or set file dates, times or attributes. A special FileZAPP routine will clear all the clusters of a file, then delete the file. The registered version includes full source code and a license to distribute the .DLL with your software. Originally released on 03/06/94. Current version 1.00.



## **Farley**

Farley is a person-database. Farley allows you to track names, addresses, phone numbers, E-mail addresses, and miscellaneous detailed information about people. Farley uses the Microsoft Access database format, and allows full SQL queries. Includes some novel features. Anticipated release on 04/01/94.

## **CRITTER**

CRITTER is a The TSR Critical Error Handler that DOS forgot. CRITTER replaces the ugly "Abort, Retry, Ignore" message with an attractive pop-up window. Command line options include printer logging, automatic response with time-out, automatic reboot and response patterns (such as Retry 4 times, then Fail). This utility is perfect for unattended systems such as bulletin board systems, packet radio repeaters, robotics assembly lines and print servers. Originally released on 04/26/90. Current version 1.32.

# Order Form

(Print Me!)

NAME: \_\_\_\_\_

COMPANY: \_\_\_\_\_

ADDRESS: \_\_\_\_\_

CITY: \_\_\_\_\_

STATE: \_\_\_\_\_ ZIP CODE: \_\_\_\_\_

PHONE: \_\_\_\_\_

E-MAIL ADDRESS: \_\_\_\_\_

WHERE DID YOU RECEIVE THIS PRODUCT? \_\_\_\_\_

DISK SIZE NEEDED: [ ] 3-1/2 [ ] 5-1/4

COMMENTS: \_\_\_\_\_

2x4 Labeler @ \$ 20.00 \_\_\_\_\_

3x3 Labeler @ \$ 20.00 \_\_\_\_\_

Farley @ \$ 20.00 \_\_\_\_\_

DrvUtils.DLL @ \$ 20.00 \_\_\_\_\_

CRITTER @ \$ 20.00 \_\_\_\_\_

Colorado sales tax (7.1%) \_\_\_\_\_  
(CO residents only)

Non-US Shipping @ \$5.00 \_\_\_\_\_

(Payments must be in US Dollars, US bank)

TOTAL: \_\_\_\_\_

Checks or money orders only. Sorry, no company P.O.'s or credit cards.

(This order form from Drive Utilities DLL 1.00)

Please make checks or money orders payable to:

**Dana W. Cline**

Mail to:

TANSTAAFL Software  
P.O. Box 260075  
Lakewood, CO 80226  
USA

# Legalese

**Drive Utilities DLL** is Copyright 1994 by TANSTAAFL Software. **Drive Utilities DLL** is released as shareware. The unregistered version may be used for evaluation but may not be distributed with your programs. Users are expected to pay for the registered version of this program, which comes with full source code and may be distributed with your software. **Drive Utilities DLL** is not warranted for anything, and TANSTAAFL Software is not responsible for any damages arising from the use of this program.

All product or company names mentioned are probably copyrighted or trademarked by their respective owners.

If you want to give this program away, please distribute the original compressed file. If you want to distribute this program via CD-ROM or software library service, please contact us for the latest shareware version.

# Revision History

**03/06/94 1.00**

Original release. The disk routines in this .DLL are used in **3x3 Labeler** and **2x4 Labeler**. The file routines were added to round out the offering.

