

Documentation for  
Rich Levin's Toolkit Objects (TKO)

\*

Dir32 Component  
Version 1.0.0

(Wednesday, January 03, 1996, 15:18:20)

PRELIMINARY RELEASE 1

This document was edited using Microsoft WordPad for Windows 95. Ruler as inches, left margin 0", right margin 6".

The software contained in this archive requires Microsoft Windows 95 or Windows NT, Microsoft Visual Basic v.4.0 (32-bit), or compatible programs.

---

COPYRIGHT NOTICE

Entire contents copyright © 1995 RBLevin, all rights reserved. The following COPYRIGHT RESTRICTIONS apply to this preliminary release.

- YOU MAY SHARE THIS PROGRAM WITH OTHER USERS, PROVIDED NO FEES ARE CHARGED.
- THIS PROGRAM MAY NOT BE UPLOADED TO ANY SERVICE OTHER THAN THE COMPUSERVE INFORMATION SERVICE OR THE MICROSOFT NETWORK.

Direct feature requests and tech support ?'s to 72407,243 via CIS MSBASIC, or RBLevin@msn.com (Internet).

---

This archive contains the following files:

Documentation: Dir32.rtf (this file)  
Demo file: Dir32.exe (double-click for a demo)  
DLL: Dir32.dll (add to VB4-32's Tools\References to use)

---

DISCLAIMER

Dir32 has been thoroughly tested and, as with all my code, appears to be bug-free. There are, however, extensions contained within the compiled DLL which have not, as yet, been enabled.

Future releases will enable these enhancements.

---

---

### Tech Support and Contact Information

Rich Levin  
P.O. Box 14546  
Phila., PA 19115

215-887-8281 (voice)  
215-887-5485 (fax)

RBLevin@msn.com  
72407.243@compuserve.com

Rich Levin is the Associate Editor/Application Development for Information Week magazine (CMP). Levin also produces The PC Report for KYW NewsRadio (Phila.), and is a contributing editor to Windows Magazine (CMP).

---

---

### INTRODUCTION

I've been working with reusable BASIC code since dinosaurs roamed the Earth. Years ago, I started building a library of common functions and subprograms, designing each interface to be as abstract as current technology allowed. Abstraction makes it easier to carry code forward, or to port library routines from one program, or compiler, to another.

The introduction of Class Modules and OLE Server technology in Microsoft Visual Basic 4.0 makes it possible to create precompiled, stand-alone, shared libraries of commonly used procedures. It also marks the return of precompiled libraries, which have been noticeably absent since the decline of DOS BASIC's non-OOP object modules and units.

VB4 OLE Server DLLs have many advantages over .BAS modules. Most important, trivial changes or bug-fixes applied to function or subprogram algorithms do not require recompilation of parent applications. Provided object interfaces haven't changed, only the affected OLE Server DLLs must be recompiled.

When you consider that VB4 OLE Servers can be deployed on remote central file servers, the savings in program and user maintenance times becomes apparent. With UI code centralized in client applications, business logic, file I/O, and shared libraries can be updated without disturbing client applications or the people using them.

Best of all, Class Modules contained in VB OLE Servers represent true reusable components. Programming teams that establish object definition conventions can deploy reusable objects

throughout their VB applications. This reduces development time, compile time, and EXE program size.

Since acquiring VB4, first as a beta tester, later as a user, I've been developing an object collection which I call my "toolkit". The tools I place in my toolkit are fundamental to developing VB4 apps and, as such, would surely be of use to other VB programmers. Noting this, I decided to prepare my toolkit for general release.

Dir32 is the first component of my "Toolkit Objects" (hereinafter referred to as "TKO") to be released. Other individual components will follow, along with a comprehensive DLL that contains all the objects. All of the components have been developed and tested. It's just a matter of finding time to package them for release.

---

---

## INSTALLATION

- Copy **Dir32.dll** to your **\Windows\System** folder.
- Load VB4.
- Open your project.
- Click **Tools**.
- Click **References**.
- Scroll through the list until you find **Rich Levin's TKO**.
- Select the **Dir32 Component**.
- Paste the code sample (below) into your project's main module.
- Compile and run.

---

---

### clsDirectory Object

A reusable, shared object component that collects and contains information about Microsoft Windows 95 VFAT directories. Accepts the Files method for generating arrays that contain filenames meeting the Filespec property.

#### Syntax

```
Dim MyDirectoryObject as New clsDirectory
```

```
MyDirectoryObject[.{property | method}]
```

#### Remarks

Dir32 allows you to create and use objects based on the clsDirectory class. New objects inherit clsDirectory's properties and methods. clsDirectory objects behave similar to MS-DOS' DIR command.

To use the clsDirectory object, you first declare a new instance of the object. Then you set the Filespec property, and use the Files method. The object will return an array of type Variant that contains a list of matching files.

As you will see from the code example, below, clsDirectory objects are similar to VB4 OCX controls.

All errors raise Err.Number 9500. Verbose error messages and suggested solutions are returned in Err.Description.

## Properties

The following property values are required. An error 9500 will be returned if these properties aren't set before using the Files method.

<u>Property</u>	<u>Value</u>
Drive	Disk drive to search. Any valid string expression representing an available disk drive. For example, "C:". Semicolon is optional.
Folder	Starting path. Any valid string expression representing a pathname. For example, "\Windows". Opening and closing slashes are optional.
Filespec	File specification. Any valid string expression representing a valid Windows 95 filespec. For example, "*.*", "*.INI", "*BMP", "F?BA?" are all valid.

The following properties are optional. Their default value is False.

<u>Property</u>	<u>Value</u>
ShowFolder	True or False. Default is False. Setting this property to True instructs the clsDirectory object to add the [d:][path] to the returned [filename][.ext]. For example, Explorer.exe is returned as C:\Windows\Explorer.exe.
ShowSubfolders	True or False. Default is False. Setting this property to True instructs the clsDirectory object to search subfolders, in addition to the parent folder.

The following properties are read-only. Their default value is 0 until after the Files method is used.

<u>Property</u>	<u>Value</u>
FilesCount	Returns a long integer representing number of files found.

## Methods

Files Returns a Variant array of filenames that meet Filespec.

## Example

```
Sub Main()  
'Demonstration of RBL's TKO Dir32 clsDirectory object  
  
Dim Directory as New clsDirectory    'Create new Directory object  
Dim Files As Variant                'DIR array passed here  
Dim I as Long                       'For/Next loop Index  
  
Directory.Drive = "C:"              'Set drive
```

```
Directory.Folder = "\Windows"      'Set starting path
Directory.Filespec = "*.INI"       'Set search filespec
Directory.ShowFolder = True        'Attach folder names
Directory.ShowSubfolders = True    'Search subfolders, too

'Use Files method; pass Variant array of filenames to Files
Files = Directory.Files

'Show the files found using the Directory.FilesCount property
For I = 1 to Directory.FilesCount
    Debug.Print Files(I)
Next I
```

End Sub

---

Entire contents copyright © 1995 RBLevin, all rights reserved

###