

## Visual Basic For Applications POWER DLL

1

### Information Procedures

1

Function (vba)WinDir() as String

1

Function (vba)SysDir as String

1

Function (vba)TmpDrv() as string

1

Function (vba)TmpFile() as string

1

Function (vba)DrvType(Drive Letter) as Integer

1

Function (vba)DosVersion() as String

2

Function (vba)WinVersion() as String

2

Function (vba)DosEnv(byval lineno%) as String

2

Function (vba)CPUMode () As String

2

Function (vba)MathCoprocessor () As Integer

2

Function (vba)CPU () As String

2

### Ini Functions

2

Function (vba)WinIniGet(byval Section\$,byval Entry\$) as String

2

Function (vba)WinIniGetAll(byval Section\$) as String

2

Function (vba)IniGet(byval Section\$,byval Entry\$,byval FileName\$) as String

3

Function (vba)IniGetAll(byval Section\$,byval FileName\$) as String

3

Sub (vba)WinIniFlush()

```

3
Sub (vba)WinIniDel(byval Section$,byval Entry$)

3
Sub (vba)WinIniDelSect(byval Section)

3
Sub (vba)IniDel(byval Section$,byval Entry$, byval FileName$)

3
    Sub (vba)IniDelSect(byval Section$,byval Entry$, byval FileName$)

3
File Version

3
    Function vbaFileVersion (ByVal FileName$, ByVal info%) As String

3
    Function (vba)Memory (ByVal info%) As Integer

3
    Function (vba)PageMemory (ByVal info%) As Integer

3
    Function (vba)ModuleLocation (ByVal fn$) As String

4
STOPWATCH

4
    Sub vbaResetStopWatch ()

4
    Function (vba)StopWatch () As Long

4
STRING PROCEDURES

4
    Function (vba)ParseSz (ByVal txt, ByVal lineno%) As String

4
    Function (vba)Empty (ByVal txt$) As Integer

4
    Function (vba)CountStr (ByVal txt$, ByVal CountChar$) As Integer

4
    Function (vba)NthRow (ByVal txt$, ByVal nth%) As String

4
    Function (vba)NthCol (ByVal txt$, ByVal nth%) As String

5
    Function (vba)Alltrim (ByVal txt$) As String

```

5  
Function (vba)About (byval key as long) as string

5  
© COPYRIGHT 1993,1994

5  
LICENSE AND REGISTRATION

6  
Payment

6  
Update Services:

6  
Ghost Coders

6  
**Visual Basic For Applications POWER DLL**  
**Bainbridge Knowledge Management Arts, Inc.**

This DLL uses certain unique characteristics of some VBA to allow the following function to appear as *Native* functions. Normally there is considerable overhead in doing API calls, for example:

The usual way to get a Win.ini value is:

```
A$=GetWinIni("Fonts","Arial")
....
Function GetWinIni(byval Section$,Byval Entry$)
Dim Buffer$
Dim rc%
Buffer$=String(2048," ")
rc%=GetProfileString(Section$,Entry$,"",Buffer$,len(Buffer$))
GetWinIni=Left(Buffer$,rc%)
End Sub
```

with this DLL it is just:

***A\$=WinIniGet("Fonts","Arial")***

The difference in performance and code simplicity is dramatic. The functions in this DLL are most commonly used -- or needed, Including a Millisecond Accurate Clock Watch to time code performance, system information and Version Information for exec's, DLL etc.

No warranty is made whether this product will work or continue working with different VBA applications from Microsoft, i.e. Excel, WinWord, Access, VB. See Copyright Notice. This code will not work with WinWord 2.

Files:

VBA_BKMA.DLL	VBA Power DLL
VBA_INI.FRM	VB Code for Demo
VBA_STR1.FRM	VB Code for Demo
VBA_BKMA.FRM	VB Code for Demo
VBA_BKMA.FRX	VB Code for Demo
TESTDLL.MAK	VB Code for Demo
BKMA_VBA.EXE	VB Demo Program
VBA_BKMA.TXT	Functions Declaration

VBA_BKMA.HLP	Help File
VBA_BKMA.WRI	HardCopy of WinHelp
VBA_BKMA.MDB	Access 2.0 Example

### **Information Procedures**

These functions allow Critical Information to be easily obtained.

#### **Function (vba)WinDir() as String**

Returns the Windows Directory.

#### **Function (vba)SysDir as String**

Returns the Windows System Directory.

#### **Function (vba)TmpDrv() as string**

Returns the a drive that is available for temporary files. This returns the letter of the first available hard drive. This is on your TEMP, if it is specified.

#### **Function (vba)TmpFile() as string**

Returns the name of a temporary file on the current default drive.

#### **Function (vba)DrvType(Drive Letter) as Integer**

Determine the type of drive, i.e. i%=DrvType("G:")

Returns

- 0-Error
- 1-Removable
- 2-Fixed
- 3-Remote

*Example*

if 2<>vbaDrvType("C") Then msgbox "C is not a fixed drive. Workstation?"

#### **Function (vba)DosVersion() as String**

Returns the DOS version as a string.

#### **Function (vba)WinVersion() as String**

Returns the Windows Version in use as a string.

#### **Function (vba)DosEnv(byval lineno%) as String**

This returns the Dos Environment, The line number indicates which line of the Dos Environment is returned. When no more lines are available an empty string is returned.

#### **Function (vba)CPUMode () As String**

Returns the Mode windows is operating in:

- "Enhanced(386)"
- "Paging"
- "Standard(286)"
- "Protected Mode"
- "Windows Emulation"
- "Unknown";

#### **Function (vba)MathCoproprocessor () As Integer**

Returns True is a x87 is installed.

### **Function (vba)CPU () As String**

Returns the type of CPU running:

"8086"  
"80186"  
"80286"  
"80386"  
"80486"  
"Nox86"

### **Ini Functions**

These functions allow all the Ini manipulations to be done easily.

### **Function (vba)WinIniGet(byval Section\$,byval Entry\$) as String**

Return the value from *Win.ini* of *entry\$* in *Section\$*.

### **Function (vba)WinIniGetAll(byval Section\$) as String**

Return the entire contents from *Win.ini* of *Section\$*. Each line is delimited by chr\$(0). See ParseSz to extract individual components

### **Function (vba)IniGet(byval Section\$,byval Entry\$,byval FileName\$) as String**

Return the value from *Filename\$* of *entry\$* in *Section\$*.

### **Function (vba)IniGetAll(byval Section\$,byval FileName\$) as String**

Return the entire contents from *Filename\$* of *Section\$*. Each line is delimited by chr\$(0). See ParseSz to extract individual components

### **Sub (vba)WinIniFlush()**

Win.ini is cached by windows. If you intend to read the file sequentially or need the contents written to disk, this will flush the cache.

### **Sub (vba)WinIniDel(byval Section\$,byval Entry\$)**

Deletes an entry from Win.ini

### **Sub (vba)WinIniDelSect(byval Section\$)**

Deletes an entire section from Win.Ini -- Be careful!

### **Sub (vba)IniDel(byval Section\$,byval Entry\$, byval FileName\$)**

Deletes an entry from *FileName\$*.

### **Sub (vba)IniDelSect(byval Section\$,byval Entry\$, byval FileName\$)**

Deletes an entire section from *FileName\$*.

### **Sub (vba)About\_vba\_BKMA(ByVal Key As Long) As String**

### **File Version**

Versions of DLL's, VBX's OCX's are a joy the programmer would prefer to avoid. The following functions can simplify some problems.

### **Function vbaFileVersion (ByVal FileName\$, ByVal info%) As String**

Returns the internal information (not the String information--this is usually more reliable).

Global Const FV\_Version = 1  
Global Const FV\_ProductVersion = 2

### **Function (vba)Memory (ByVal info%) As Integer**

Returns the Percentage of Memory Available. These are rough measures.

Global Const M\_UserFreePercent = 1

Global Const M\_GDIFreePercent = 2

### **Function (vba)PageMemory (ByVal info%) As Integer**

Returns the pages of different memory available:

- 1 LargestFreeBlock
- 2 MaxPagesAvailable
- 3 MaxPagesLockable
- 4 TotalLinearSpace
- 5 TotalUnlockedPages
- 6 FreePages
- 7 TotalPages
- 8 FreeLinearSpace
- 9 SwapFilePages
- 10 PageSize

### **Function (vba)ModuleLocation (ByVal fn\$) As String**

Returns where a Module in Memory was loaded from. This is very helpful for problems with different versions of a DLL -- you can identify where the 'bad' version is located and take appropriate action.

### **STOPWATCH**

Windows time runs off the DOS interrupt which is accurate to 1/18 second or less. The chips on the motherboard can be programmed to produce more accurate timing -- this uses one of those chips.

#### **Sub vbaResetStopWatch ()**

This initializes or reset the stopwatch to 0.

### **Function (vba)StopWatch () As Long**

Returns the number of milliseconds that has passed since the StopWatch was Reset or Started. The time is real milliseconds not **Time** or **Now**. This is very useful for measuring performance. Do not use DOEVENTS in sections being timed.

### **STRING PROCEDURES**

#### **Function (vba)ParseSz (ByVal txt, ByVal lineno%) As String**

*Warning:* String sent is REQUIRED to end with TWO \0\0 (i.e. chr\$(0)+Chr\$(0)). This is designed to handle API calls problems. For example, returning an array of LPSTR. This is also a dangerous function. It is recommended that you call using:

txt\$=ParseSz(FromAPI\$+CHR\$(0)+CHR\$(0),5) 'Get 6th String in C array

#### **Function (vba)Empty (ByVal txt\$) As Integer**

Often strings are not empty but contains only CR,LF, TAB or Space. This function allows these 'empty' strings to be easily detected.

#### **Function (vba)CountStr (ByVal txt\$, ByVal CountChar\$) As Integer**

Count the number of occurrences of a string on another string. Although this can be done by looping with Instr -- this is much faster.

#### **Function (vba)NthRow (ByVal txt\$, ByVal nth%) As String**

VB uses 'Clips' of strings frequently (i.e. GRID), this allows a row to be manipulated easily by allowing a row to be extracted.

### **Function (vba)NthCol (ByVal txt\$, ByVal nth%) As String**

VB uses 'Clips' of strings frequently (i.e. GRID), this allows a column of the FIRST ROW to be manipulated easily by allowing it to be extracted.

### **Function (vba)Alltrim (ByVal txt\$) As String**

Returns the string stripped of left and right non-printing characters. Removes leading and trailing CR, LF, TABS and spaces (etc).

### **Function (vba)About (byval key as long) as string**

A registered user may receive a Key number by Email which will suppress the warning messages IF vbaAbout is called first with the key. Alternatively, a version may be sent by USMail which does not require the use of a Key.

### **© COPYRIGHT 1993,1994**

This is a copyrighted work by Bainbridge Knowledge Management Arts, Inc. and has all rights reserved. The demo version is made available for evaluation purposes only -- please verify that it works satisfactorily for your needs PRIOR TO PURCHASE. No warranty or remedy is available. This work includes **TRADE SECRETS** and source cannot be made available without appropriate execution of legal agreements.

*legal stuff:*

#### **LIMITED WARRANTY; DISCLAIMER OF DAMAGES**

This work may contain a "Defect" in the form of incomplete, inaccurate or corrupt data, transcription errors, a copyright or other infringement, a defective or damaged disk, computer virus, or codes that damage or cannot be read by your equipment. But for the "Right of Replacement or Refund" described below, BKMA (and any other party you may receive this work from) disclaims all liability to you for damages, costs and expenses, including legal fees, and YOU HAVE NO REMEDIES FOR NEGLIGENCE OR UNDER STRICT LIABILITY, OR FOR BREACH OF WARRANTY OR CONTRACT, INCLUDING BUT NOT LIMITED TO INDIRECT, CONSEQUENTIAL, PUNITIVE OR INCIDENTAL DAMAGES, EVEN IF YOU GIVE NOTICE OF THE POSSIBILITY OF SUCH DAMAGES.

All the functionality is made available prior to purchase for your evaluation - if you discover a Defect in this work after receiving it, you cannot receive a refund of the money (if any) you paid for it, It is supplied AS IS.

THIS WORK IS OTHERWISE PROVIDED TO YOU "AS-IS". NO OTHER WARRANTIES OF ANY KIND, EXPRESS OR IMPLIED, ARE MADE TO YOU AS TO THE WORK OR ANY MEDIUM IT MAY BE ON, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimers of implied warranties or the exclusion or limitation of consequential damages, so the above disclaimers and exclusions may not apply to you, and you may have other legal rights.

#### **INDEMNITY**

You will indemnify and hold BKMA, its directors, officers, members and agents harmless from all liability, cost and expense, including legal fees, that arise directly or indirectly from any of the following that you do or cause:

- [1] distribution of this work,
- [2] alteration, modification, or addition to the work, or
- [3] any Defect.

#### **LICENSE AND REGISTRATION**

**BKMA** is a Internet Publisher and will issue Keys for products, or custom builds(no keys) by

Email or US Mail. This product may be registered and a Key obtained for:  
\$20.00 for the VBA Power Dll  
and supplying an Internet Accessible Email Address to send the Key to.  
Add \$5.00 for delivery by Postal Services.

### **Payment**

Payment must be made by  
cheque in US Dollars (drawn on a US Bank) mailed to  
BKMA Inc.,  
POBox 170  
Kingston, WA  
98346-0170

or by

Company Purchase Order or  
letter on Corporate Letterhead, Faxed to  
1-206-779-7747.

Corporate Letterhead letters must include "Acting as an agent of my company I am authorized to order on 15 days net, the following:" and include a Fax and Mail address.

You may not make the key available to anyone but the yourself. You may redistribute the DLL with an application at your own risk. BKMA requires that you accept all the common disclaimers by Borland and Microsoft for their products as a condition of sale.

### **Update Services:**

New versions will be available as requests for additional functions or errors are detected.  
An upgrade subscription service is available for:

\$5.00/Copy ... please specify update frequency:

- ☐ Monthly or each release version -- which ever occurs last
- ☐ Quarterly
- ☐ Yearly.

***This is sent as an Attachment to Internet Mail.***

This may be pre-paid or via a PO or Corporate Letter. Add \$5 per update for postage and handling if the update is to be sent by post.

### **Ghost Coders**

**BKMA** offers ghost coding services. If you need a function -- and are concerned solely with getting the working code (not C source) -- we will code and test the function for you. The flat rate per function is \$100 US -- we retain the rights to the source, and will do free support for 90 days. Our hope is to include this function in this or other DLL's.

BKMA is also available for serious Windows development.

### **Order Form**

Name: \_\_\_\_\_  
Company: \_\_\_\_\_  
Phone No: \_\_\_\_\_ Fax: \_\_\_\_\_  
Email Address: \_\_\_\_\_  
Mail Address: \_\_\_\_\_  
City: \_\_\_\_\_ State/Province: \_\_\_\_\_  
Country: \_\_\_\_\_ Postal Code: \_\_\_\_\_

I have read the License Information and understand that this is an 'AS IS' product with no warranty. I have tried this product sufficiently that I am satisfied to its performance and release BKMA from all responsibility and identifies it from any subsequent use by me.



Signature: \_\_\_\_\_  
Date: \_\_\_\_\_

I wish to receive:

☐ Electronic Key, this permits me to use the version I have. Please Email to me at the above Email Address. I understand that later versions I obtain by BBS/Internet will use the same key.

☐ US Mail. Later version must be ordered at \$5.00 per copy, plus \$5.00 per mailing.

**Registration/License:**

Individual: \_\_\_\_\_ @ \$20 = \_\_\_\_\_.  
Corporate: \$150.00 = \_\_\_\_\_.  
US Mail \$5.00 = \_\_\_\_\_.

Updates:

N/C if you obtain off Internet 0.00 EKey Only!  
\$5.00 per update @ \_\_\_\_\_ updates \_\_\_\_\_ EMail to You!  
\$10.00 per update @ \_\_\_\_\_ updates \_\_\_\_\_ USMail to You  
(\$5.00 per update, \$5.00 Post/Hand)

Updates Interval Requested:

\_\_\_\_\_ Minor Revisions/Monthly  
\_\_\_\_\_ Major Revisions/Quarterly  
\_\_\_\_\_ Yearly

TOTAL \_\_\_\_\_.

Fax to: (PO/Corporate Letter)

1-206-779-7747 or

Mail to: (Check -- US Dollars)

BKMA inc  
POBox 170  
Kingston, WA  
98346-0170