ELECTRIFIED VISUAL BASIC

ABOUT ELECTRIFIED VISUAL BASIC MAGAZINE

FROM THE PUBLISHERS DESK

CONTENTS

ABOUT ELECTRIFIED VISUAL BASIC MAGAZINE

ELECTRIFIED VISUAL BASIC MAGAZINE is an "electronic" magazine. It is a Shareware product. It is not free. If you enjoyed or found this issue of value, the Shareware price for this single issue is only $3.50. From all of the letters and E-Mail we received about the first issue it was a great success. However, we received only a few Shareware payments. The two suggestions we received the most were the magazine price is too high and not enough information in the magazine. Due to these suggestions we dropped the price and have increased the size of the magazine. Your support is necessary if you want us to continue to produce this magazine. As an incentive to get more people to pay for the magazine we are offering exclusive Visual Basic source code to a program called LOANCALC. To get this source code you must send in the SHAREWARE PAYMENT FORM along with your $3.50 payment and check the box stating you wish to receive LOANCALC (there is also a shipping & handling charge for the code).

SHAREWARE PAYMENT FORM

EVB MAGAZINE SHAREWARE PAYMENT FORM

NAME: _____

COMPANY NAME: _____

ADDRESS: _____

CITY: _____

STATE: _____

ZIP: _____

COUNTRY: _____

PHONE: _____

Where did you get issue #2? _____

What you liked about issue #2? _____
_____
_____
_____

What you did not like about issue #2? _____
_____
_____
_____

Single issue Price is $3.50 per issue.

Do you want LOANCALC source code?   YES [   ]   NO [   ]
S/H is $3.50   - Foreign S/H is $8.50   - (TOTAL $7.00 US or $12.00 Foreign)

California residents add 7-3/4 % sales tax.

Send check or money order drawn on a US Bank made payable to PRO-DATA and send to the following address:

PRO-DATA
7151 EL CAJON BLVD. STE. L
SAN DIEGO, CA   92115

FROM THE PUBLISHERS DESK

By Don Welsh

I would like to take this opportunity to welcome all of the new readers of ELECTRIFIED VISUAL BASIC MAGAZINE. It is a Bi-Monthly electronic magazine devoted to covering the exploding Visual Basic market. We have chosen to publish the magazine in Windows 3.1 Help format. Using this method we can provide a colorful magazine with hypertext, pop up graphics, searching and printing capabilities. The magazine features Reviews of commercial and shareware Visual Basic Add-ons, Visual Basic programming articles (with source code), Tips and Tricks, New Visual Basic Products, Contests and more.

Are you tired of searching through many different magazines hoping to find an article or review relating to Visual Basic? If you answered yes to this question then this magazine is for you!

We welcome your suggestions and comments on this magazine. We will be striving to improve the magazine with every issue. We can only accomplish that with your input.

Software Companies may send press releases and/or review copies of Visual Basic related programs to the below address:

PRO-DATA
7151 EL CAJON BLVD. STE. L
SAN DIEGO, CA   92115

To advertise in the PRODUCT SHOWCASE section of this magazine call (619) 464-0767 for information.

Publisher
Don Welsh

Associate Editor
Pam Johns

Associate Editor
Jon West

Design
Bill Hamilton

Programming
John Tredwell

Authors
Mike Able
Dave Johnston
Richard Prince
Robert Thomas
Jack Thomas
Thomas Wagner

Publisher

This column is written by Don Welsh. If you have any questions or ideas for this "electronic" magazine please contact me through ELECTRIFIED VISUAL BASIC MAGAZINE:

Don Welsh
7151 El Cajon Blvd. STE. L
San Diego, Ca   92115

CONTENTS

SUCCESSFUL SHAREWARE
PART 2
By Robert Thomas

This column is going to look at the different aspects of becoming a successful Shareware author. For this second installment I will be writing about how and where to distribute your program and more!

First off I would like to mention writing help files for your programs. As you have probably noticed there are articles and reviews on help files in this issue of the magazine. If you remember in my last installment I talked about first impressions towards your shareware program can be very important. A well written .HLP file gives your application a more polished and professional look that your users will appreciate. Which in turn should help you receive more registrations. So be sure and read the help related information throughout the magazine.

Now on to the main focus of this article - Distributing your Shareware program. Marketing your programs as Shareware requires a new way of distribution. You need to get as many copies of your Shareware program out to the public as possible. You do this by giving away as many copies as possible for free. This might sound funny when your trying to make money from your program. However, this is the only way you will receive registrations. If you give out free disks of your $20.00 program to 100 people and 10 register you have made $200.00. Not very much if you have spent months working on a complex program. Now lets say you give out 1000 free disks and 100 people register it you have made $2000.00. Since disks are cheap it makes sense to give away as many as you can.

You will do this by sending copies to shareware vendors and user groups. Computer Swapmeets and Shows are other good places to give out disks. Another good way to spread your program around to as many people as possible is to put it on lots of BBS systems and also the major on-line systems. To do this you will need a modem and communications software. Get the fastest modem you can afford because it will save money in the long run due to smaller phone bills. You will want to upload your program to Compuserve, Genie, America Online and Delphi. These are all huge systems and your program will spread fast on these systems. A way to get your program spread even faster is to ask in your README.TXT file that you would appreciate it if anyone who has your shareware program would upload it to their local BBS systems.

You will also want to send a copy of your program to large User groups. These groups have shareware libraries that are available to their members. To find out the user group

addresses, pick up a copy of *COMPUTER SHOPPER*. It also has a list of BBS system phone numbers.

You will also want to send copies of your disk to Magazines. I have noticed that the Windows Magazines have started to review more and more Shareware programs. Even general computer magazines are also including Shareware reviews. If you are sending disks to magazines you must provide them with a very professional looking package. This is not to say you need to spend lots of money on color boxes. Just remember what I said in the previous issue. Make sure you have README.TXT files on the disk and printed disk labels stating the name of the program and installation instructions. It is also a good idea to send printed documentation along with your disk. Editors of these magazine receive lots of programs to review. You need to grab their attention and let them know what your program can do.

This brings this installment to a close. In the next issue of ELECTRIFIED VISUAL BASIC MAGAZINE I will discuss business aspects of running your shareware business!

Author

This column was written by Robert Thomas, a shareware author and shareware disk distributor.
If you have any questions or ideas for topics for this column please contact me through
ELECTRIFIED VISUAL BASIC MAGAZINE:

Robert Thomas
7151 El Cajon Blvd. STE. L
San Diego, Ca   92115

VISUAL DATA-BASICS

By <u>Thomas Wagner</u>

VB/ISAM
manufactured by Software Source

42808 Christy Street, Suite 222   Tel: (510) 623-7854   Single User Version: $99.95

Fremont, CA 94538   Fax: (510) 651-6039   Multi User Version: $169.95

<u>Short Summary</u>
<u>Detailed Review Comments</u>
<u>Code Sample</u>

SHORT SUMMARY

VB/ISAM
by Software Source

42808 Christy Street, Suite 222   Tel: (510) 623-7854   Single User Version: $99.95
Fremont, CA 94538   Fax: (510) 651-6039   Multi User Version: $169.95

VB/ISAM is a tightly written VB library consisting of 18 functions. Data management can be accomplished with as little as 8 of these functions. **The entire DLL takes up less than 50kb.** The program is known for its speed and efficient use of memory. Received the 1992 Readers Choice Award by the readership of BasicPro Magazine.

Features include:

> File size up to 512 MB.
> Implemented in commercial applications since 1991.
> Up to 80 indices per file, all information stored in one DOS file.
> Information is immediately updated in sort order when performing updates or   modify /   delete records.
> File / Index space is maintained automatically, no periodic compaction required.
> Variable length records and fields; any record can be a variable length string   field.
> Multiple files can be open at the same time.
> Both Random and Sequential Data Access - You can read records randomly (by   index lookup) and then move forward or backward through the file with   Next and Previous functions.
> Range Positioning / Nearest Match / Partial Key Search - If no exact match is   found the pointer is positioned in the appropriate index.
> Arrays, Multiple Record Formats, Substructures - You can have any number of   different record types in the same file plus a unique technology allows   you to insert different substructure into records.

DETAILED REVIEW COMMENTS

I am a firm believer in the idea of "every job calls for a specific tool". Of course, thinking along those lines has advantages and drawbacks. And while I know many of you like working with specific programming tools, I also know that you are interested in adding to that tool chest. Why else would you read this magazine ?

By now you may have found, as I did, that while there are numerous DLL's and VBX's out there, simply because one is not as flashy as another does not mean it will have a lower performance. The reverse of this statement is equally true, just because one company has the funds to promote their product lavishly doesn't mean it is a better product. I like nothing better than finding a little gem that is not well known but does a great job.

Which brings me to the subject of this   product review - VB/ISAM by Software Source. I will leave it up to you to decide whether this one is a gem, but it may interest you to know that it has been given the Readers Choice Award by the readership of BasicPro magazine (our esteemed competitor). In addition, it has been utilized in the creation of VB Assistant by Sheridan Software. What better compliment to a product, and indicator of its quality,   than the usage by another member of the same industry for programming purposes. The library itself is available for VB/Win, Dos and C. In addition you can purchase either single or multi-user copies, costing $99.95 and $169.95. respectively.

Oh, by the way this is the company I was talking about at the close of our last issue. Anytime I run into people who stand by their word - in this or any other business -   I like to make sure everyone knows about it. The good people at Software Source,   especially Sam Cohen, fall into that category. On my scale for customer service and helpful attitude this company ranks highly. Such a characteristic may not mean much to you until you are working on a job, utilizing a third party library, and find out the manufacturer has an attitude problem - some of them do you know. You quickly learn to appreciate things like good customer service.

Before I get too deeply into the details of the product, I want to reiterate that there are several different approaches to managing data in the VB environment. The manufacturers of add on products by their very marketing efforts are adding new ways all the time. Some of the better known efforts include methods that are very compatible with Dbase, FoxPro and Clipper programming, down to the function names. CodeBasic comes to mind in this category. Others approach the subject from a more visually oriented standpoint, among them Agility and QE/VB are prominent.
Furthermore, there are considerations of what are commonly called "low level" and "high level" approaches. Each has it's advantages and disadvantages. High level will not require an individual to write as much code (if any), at a cost of size and speed. Low level requires more coding, however, the libraries are smaller and oftentimes the resulting applications are faster.

VB/ISAM takes an approach that falls somewhere in the middle of the "visual scale", meaning that it requires coding and does not have the "click and run" interface of other libraries. The name stands for Indexed Sequential Access Method. As you know, native VB itself has

several ways of managing and retrieving data. Two of those ways are the Random and the Sequential Access Methods. For illustration purposes it may help to think of VB/ISAM as being the Random Access Method "*on serious steroids*". And no, it is not a library using straight dBase, FoxPro or Clipper file formats.

Comments made by programmers who have tried VB/ISAM oftentimes allude to the idea that this is the way Microsoft should have handled data I/O to begin with. (But where would all the third party add-on developers be if MS did everything right?)

Gregg Irwin, an independent developer based in Claremont, California reports that he was amazed at the speed of VB/ISAM.   "I've only had VB/ISAM for a week and put together a test because I couldn't believe how fast the thing was. The test routine generated records and put them into a dataset, which has seven fields and five of them are indexed. It took VB/ISAM only twelve minutes to crate a dataset of 10,000 records. Our next step was to perform some retrieval tests.   The first test was reading all records in the set, it took 25 seconds with full data retrieval, 12 seconds without. another test was retrieving a group of 100 records, time - 2 seconds. This test included overhead for concatenating 5 strings per record and putting them into a list box for display.

Mr. Irwin described two projects, one current the other a future possibility, for which he thinks VB/ISAM lends itself. The first involves transferring images and data among a network of 1000 Information "Kiosks", the other entailed the creation of language / country localized applications that can be re-configured through remote access. The transfer of graphic images in the first project is handled in a particularly interesting manner. The image is compressed to approximately 10k, then via code assigned to a specific database field. When the record is recalled at another location, the image is decompressed into a temp file and displayed. As Mr. Irwin puts it :"Very fast, very easy, and I feel comfortable undertaking these projects with VB/ISAM because the program, among other things,   can   handle file sizes up to 512 MB, and creates extremely fast indexes". I'm sure that there are a number of other users who will agree with Mr. Irwin's assessment.

My research indicates that the people who are responsible for the development of VB/ISAM have a good deal of experience in the industry. This is born out in the name itself. ISAM is an "ancient" acronym. However, once you get past the name you'll find a tightly written library consisting of 18 functions to aid and assist you in the management of your data. **The entire overhead created by this library is less than 50KB**.

Some of the other features include:

Variable Length Records - within VB ISAM you can mix records of any length up to 64K and index keys of any size up to 253 bytes.
Large Files - As mentioned before up to 512 megabytes. The program dynamically allocates disk space.
Both Random and Sequential Data Access - You can read records randomly (by index lookup) and then move forward or backward through the file with Next and Previous functions.

Range Positioning / Nearest Match / Partial Key Search - If no exact match is found the pointer is positioned in the appropriate index.

Arrays, Multiple Record Formats, Substructures - You can have any number of different record types in the same file plus a unique technology allows you to insert different substructure into records.

No Compacting or Reorganization - VB/ISAM keeps its file space organized at all times, hence no compacting is required.

Sequence Protection - To minimize keyboard errors VB/ISAM treats upper-case letters as lower case when doing lookups, ordering or comparisons.

Update Protection - You can protect the integrity of the data by (optionally) specifying "add - only" or "replace - only" during writes

Data Validation - VB/ISAM aggressively checks file and record validity, returning error codes if your data has been corrupted.

Additional information indicates that VB/ISAM is comparatively "easy" on your hard disk, showing less disk access than other software.

One of the more unusual features of this software is the user manual, which was written with a   sense of humor. For example, in an introductory paragraph the discussion centers around the name of the program and the specific job of a record manager as opposed to a database manager.

".....Database management software is scary and cumbersome, sort of like Boris Karloff with an ax. Record managers are more like Erroll Flynn with a saber - most of the power, but with much more flexibility and control....."

There are other passages. This kind of writing really makes it entertaining to crack a manual open. I found myself reading the manual just to see what the next joke will be. Granted, it's not Eddie Murphy or Robin Williams, but it's not supposed to be. What it does - namely getting a person to read the manual - it does pretty well.

The actual data management, or in this case record management. is accomplished via the use of 16 functions (18 in the multi user version), and up to 80 automatically maintained indices. My own experience with the program shows that you can accomplish a good deal of data management with only ten functions. They are VmxCreate, VmxOpen, VmxClose, VmxPut, VmxDelete, VmxLookup, VmxGet/Next, VmxPrevious, VmxBOF, and VmxEOF. Does some of this sound familiar to you? Open? Put? When working in a multi user environment you will also need to implement VmxLock and VmxUnlock to manage the data access by several users.

The program's high speed is attained through the use of memory pointers. For example the VmxGet/Lookup function sets a pointer for a specified index to the insertion point of your lookup value. If no exact match was found the pointer is set to the closest matching position.   If you are looking for Mr. Baldwin, and you have no entry beginning with the letter "B", the pointer will end up on the last entry for A (the closest entry above B) or in-between A and C, in which case the result field will be empty. From there you can step through the data one entry at a time. This method requires a little getting used to, especially in the design phase of a project, because it is easy to search for a string or value and end up with an empty results field. Of course that is

not an error, it just means there is no exact match. Using Next and Previous will show where in the file your pointer came to "rest".

Did I thoroughly confuse you?
Just remember, it's wickedly fast and has up to 80 auto -matically maintained indices.

Rather than including screen shots of my test application, I want to refer you to a shareware utility that is in large parts based on VB/ISAM. The program is called Time & Chaos by iSibister Software. It was featured as a "Windows Utility of the Month" by one of the leading PC magazines. What better way to inspect the creative use of VB/ISAM?

I'd like to close with a preview of what's in the works for future issues. Having become more familiar with data management under VB you may find that once you have a database established you are lacking a way of getting the information out of the computer and onto paper in a meaningful manner. This function is being solved by report generating libraries, such as Crystal Reports by Crystal Services and The Report Generator by MicroHelp. I am hoping to bring you information regarding these tools. In addition, I will tell you about a new release by Pioneer Software, makers of Q+E/VB. Pioneer is truly living up to it's motto of being "the connectivity company". Presently their new product called Multilink/VB is undergoing beta testing. Multilink/VB enables you to produce VB data entry and manipulation (query) front-end's for up to 16 different databases, including dBase IV, Paradox, Informix, DBII, Oracle, Progress, Netware SQL and even Excel files.

And again, if you have seen an innovative program, or a new way of doing the same old thing, or just want to share your ideas, please drop us a line.

CODE SAMPLE

The following code shows the global declarations for an example dataset that will keep track of the information described in the Type....End Type section. This is the basic setup of an application. From here the data is managed via the use of function calls, such as VmxOpen, Vmx Lookup, etc.

```
-- Sample.bas --

Global rc As Integer        '"rc" stands for "return code"
Global Handle As Integer

Type RecordStructureType
     LastName            As String    ' <-- secondary index key...field 1
     Nickname            As String
     JobTitle            As String    ' <-- secondary index key...field 3
     Department          As String    ' <-- secondary index key...field 4
     Salary              As Integer
     GolfScore           As Integer
End Type

Global PersonRecord As RecordStructureType

Global SocSecNo As String     'This is the primary key.
```

(The code immediatly to the right of RECORDFORMAT describes to VB/ISAM which fields are indexed and what those fields are i.e. integers or strings)

```
Global Const RECORDFORMAT = "X20$, $, X20$, X20$, %, %"
Global Const XSOCSECNO = 0
Global Const XLASTNAME = 1
Global Const XJOBTITLE = 3
Global Const XDEPARTMENT = 4
Global SelectedIndex As Integer

Global GroupKey As String
Global SaveKey As String
Global DisplayString As String
Global SaveDisplayString As String

Global IndexChoice As Integer
Global FindTypeChoice As Integer
```

****

The following code shows a sub procedure that enables the user to find a record based on the SocSecNum entered by him or her. This short sample utilizes the VmxGet / Lookup function, and again it makes mention of the fact that the record pointer will be positioned in a way that indicates where the record is - whether it's there or not. If it's there it will show, if not you will see no result, however the pointer will be in the middle of your data (if the example is located somewhere in the middle). It also shows the implementation of error trapping and error return codes that are supplied with each function:

```
Sub FindKeyMatchOne ()

    rc = VmxGet(Handle, SelectedIndex, 0, (MatchKey.Text), Throwaway$, SocSecNo,
PersonRecord)

        'The "0," by the way, specifies the default options for VmxGet:   Lookup access
        'mode (as opposed to Next, Previous, or Current step-wise access), and
        'full data retrieval (as opposed to retrieving only the primary key.)

    If rc = VIS_NOT_FOUND Then
        FindList.AddItem "<<No match>> .. note that the pointer for this index has been"
        FindList.AddItem "repositioned to where your match key WOULD BE; if you move"
        FindList.AddItem "forward (Next) or backward (Previous) in this index, the step will"
        FindList.AddItem "begin FROM THAT 'INSERTION POINT.'"
    ElseIf rc <> VIS_OK Then
        ErrorExit (11)
    Else
        DisplayTheRecord
    End If

End Sub
```

Author

Thomas Wagner
1201 South Ogden Drive
Los Angeles, CA 90019

(Mr. Wagner is an independent database programmer. He has been involved with the creation of   applications on a number of platforms and is a member of several software quality assurance programs.   Most recently he has been working on database solutions utilizing Microsoft Access.)

CODE FRAGMENTS

By <u>Bob Dombroski</u>

Welcome to "Code Fragments". This feature is dedicated to the Visual Basic developer who wants to get a little more out of his code...or into it. The articles will vary in depth from basic, to intermediate and advanced programming topics. The goal will be to provide practical examples and techniques described in plain and simple language accompanied by useful Code Fragments.

This month I want to take about using the Windows API. It didn't take long for me to realize that there were things I wanted to do in my VB applications that I just couldn't accomplish from the confines of the Visual Basic language.   The obvious place to look was the Windows API. I've coded in C for years in UNIX and DOS but never in Windows but I knew their must be some use for the 500 or so functions in Windows SDK. This month I'll show you a few that I now use regularly.

From programming applications in the DOS world I'd found that having configuration files to keep local customizations was extremely useful. Window applications uses this idea extensively and the files are called initialization files, `.INI` files. Windows initialization files are broken into logical groups called sections. Each section follows this format:

[section]
keyname=value

An example is the extensions section of the `WIN.INI` file where file extensions are tied to executable programs.

```
[Extensions]
MAK=c:\vb\vb.exe ^.mak
```

Well then, how to we read and write these values? First, there are four functions useful for this purpose; GetProfileString, WriteProfileString, GetPrivateProfileString and WritePrivateProfileString. They work in pairs. GetProfileString and WriteProfileString read and write to the `WIN.INI` file while GetPrivateProfileString and WritePrivateProfileString read and write to `YOURAPP.INI`. I recommend   that you only ever read from the WIN.INI file. Although the 32K size limitation on this file was removed with Windows 3.1 adding more "clutter" to WIN.INI is generally view as unacceptable these days. It is also easier to un-install you application if it uses its own INI file.

Let's get started. First we need to include the declarations of the functions we are going to use. They are:

```
Declare Function GetProfileString Lib "Kernel" (ByVal lpAppName
As String, ByVal lpKeyName As String, ByVal lpDefault As String,
```

```
ByVal lpReturnedString As String, ByVal nSize As Integer) As
Integer
Declare Function WriteProfileString Lib "Kernel" (ByVal
lpApplicationName As String, ByVal lpKeyName As String, ByVal
lpString As String) As Integer

Declare Function GetPrivateProfileString Lib "Kernel" (ByVal
lpApplicationName As String, ByVal lpKeyName As String, ByVal
lpDefault As String, ByVal lpReturnedString As String, ByVal
nSize As Integer, ByVal lpFileName As String) As Integer
Declare Function WritePrivateProfileString Lib "Kernel" (ByVal
lpApplicationName As String, ByVal lpKeyName As String, ByVal
lpString As String, ByVal lpFileName As String) As Integer
```

Now   we'll put a little code together.

```
    Dim szTemp As String * 65
    Dim szPath As String

    ' The Get-functions are particular about how the target
variable is declared.
    ' Declare it as I have above. If you care why you can look it
up in the
    ' MS VB Knowledgebase.
    szTemp = ""
    ' GetProfileString requires 5 arguments
     arg 1: the INI file section name
     arg 2: the INI file keyname
     arg 3: a default result, which can be a NULL string
     arg 4: the string that will contain the result
     arg 5: the maximum number of characters to store in the
string
    ok% = GetProfileString("Extensions", "mak", "\vb\vb", szTemp,
64)
    ' Trim off the stuff (" ^.mak") in the return string that I
don't want & save the result.
    pos% = InStr(szTemp, " ") - 1
    szVB2path = Mid$(szTemp, 1, pos%)

    szTemp = ""
    ' GetPrivateProfileString requires 6 arguments, the same 1-5
as above plus
     arg 6: the name of the INI file to read
    ok% = GetPrivateProfileString("VBPT2", "Viewer", "notepad",
szTemp, 64, "VBPT2.ini")
    ' Notice this trim procedure is different than the one above.
SDK functions returning strings
```

```
    ' will return NULL characters to VB and if you don't get rid
of them you are in danger of
    ' truncating data.
    pos% = InStr(szTemp, Chr$(0)) - 1
    szViewer = Mid$(szTemp, 1, pos%)

    szPath = "c:\dos\edit.com"
    ' This is all it takes to write to an INI file. If the
Section and Keyname don't exist
    ' they will be created
    ok% = WritePrivateProfileString("VBPT2", "Editor", szPath,
"VBPT2.ini")
```

I'll conclude with one final example. Ever wish your list box could have a horizontal scroll bar?
Well it can! Windows is a message based system, that is applications sit and wait for messages
and then react to them. Therefore we have the multi-purpose function. "SendMessage". One use
of "Send Message" is to create a horizontal scroll bar.

I have constructed a function to create horizontal scroll bars for a given list box control name. Its
second argument is the new width of the list box as a factor of its current size. That is the
increment would be 1.5 or 2 (or whatever) which represents one and a half times the size or
twice the size of the list box's current size.

```
'  The function requires three SDK functions to be called, so
they must be declared
Declare Function SendMessage Lib "User" (ByVal hWnd As Integer,
ByVal wMsg As Integer, ByVal  wParam As Integer, lParam As Any)
As Long
Declare Function GetFocus Lib "User" () As Integer
Declare Function APISetFocus Lib "User" Alias "SetFocus" (ByVal
hWnd As Integer) As Integer

' SendMessage uses a global constant that I dug out of the VB 2.0
help files
Global Const WM_USER = &H400
Global Const LB_SETHORIZONTALEXTENT = (WM_USER + 21)

' The function declaration
Function LB_SetHzScrl (Ctrl As Control, Increment#) As Integer
'
'
'
    ' Get the focus of the current control so we know who to
return focus to
    OriginalHandle% = GetFocus()
    ' Set the focus to the control and obtain the handle of the
```

```
control
   Ctrl.SetFocus
   ListhWnd& = Ctrl.hWnd
   ' Add the scroll bar to the control via the handle
   LB_SetHzScrl = SendMessage(ListhWnd&, LB_SETHORIZONTALEXTENT,
Ctrl.Width / Screen.TwipsPerPixelX + (Ctrl.Width /
Screen.TwipsPerPixelX) * Increment, 0)
   ' Return focus to the previous control
   iResult% = APISetFocus(OriginalHandle%)

End Function
```

Put this reusable function into a .BAS file and you're ready to use it. Suppose your form has a list box named, lstCompanyNames. To create the horizontal scroll bar

```
result% = LB_SetHzScrl(lstCompanyNames, 2)
```

Getting to know the Windows API provides rewarding results in you application programs.

Author

This column was written by BOB DOMBROSKI. If you have any questions or ideas for topics for this column please contact me through ELECTRIFIED VISUAL BASIC MAGAZINE:

BOB DOMBROSKI
7151 El CAJON BLVD. STE. L
SAN DIEGO, CA   92115

ROBOHELP REVIEW
By <u>Mike Able</u>

If you have already read the article Writing Help Files with Word for Windows you can begin to imagine how long and tedious a job this will become if your help file is of any great size. If you are going to have more than a couple of pages you need some help to write your help files. <u>RoboHelp</u> fits the bill nicely.

To create a ".HLP" file without RoboHelp you have to use an .RTF format capable word processor to enter codes into footnotes. These footnotes will specify the topic names, any search keys, jumps, browse sequences, and any bitmap files that will be used in your help file. Then you use the Windows Help Compiler to create the ".HLP" file. Hopefully you did not make any mistakes or its back to the word processor and trying to sort out the cryptic codes in the footnotes.

With RoboHelp you begin a new help file by simply creating a new document   with the RoboHelp template in Word for Windows. You must already own Word For Windows. You will perform all of the steps of creating, compiling and testing your help file right from within the WinWord document.

After launching WinWord with the RoboHelp template you will save your document and then a floating button bar will appear on your screen. This ToolBar gives quick access to the functions used in creating help files.


The saved document will have the first topic already created for you. To create additional help topics just click on the TOPIC button. RoboHelp puts up the following dialog box so you can set up your next topic.
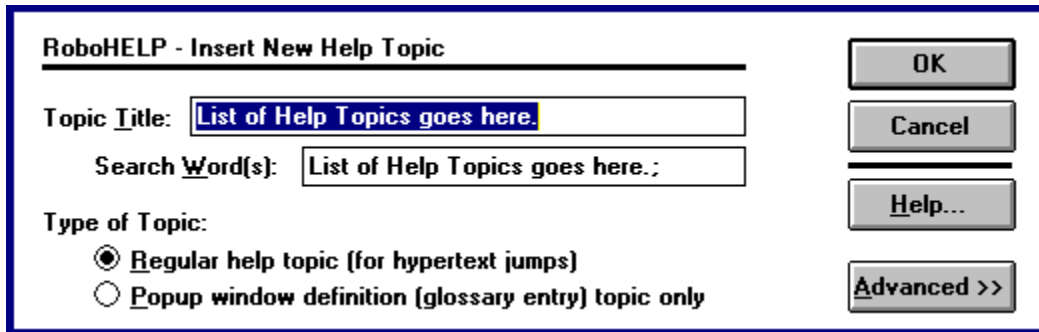
This is where you will enter the topic title, search keywords and more. When you click on OK RoboHelp places the correct codes into WinWord footnotes for you.

Once you have several topics defined you are ready to make jumps from topics to topics. To create a jump select the word or words you want to jump from and click on the JUMP button. RoboHelp puts up the following dialog box listing all of the topics you have created.
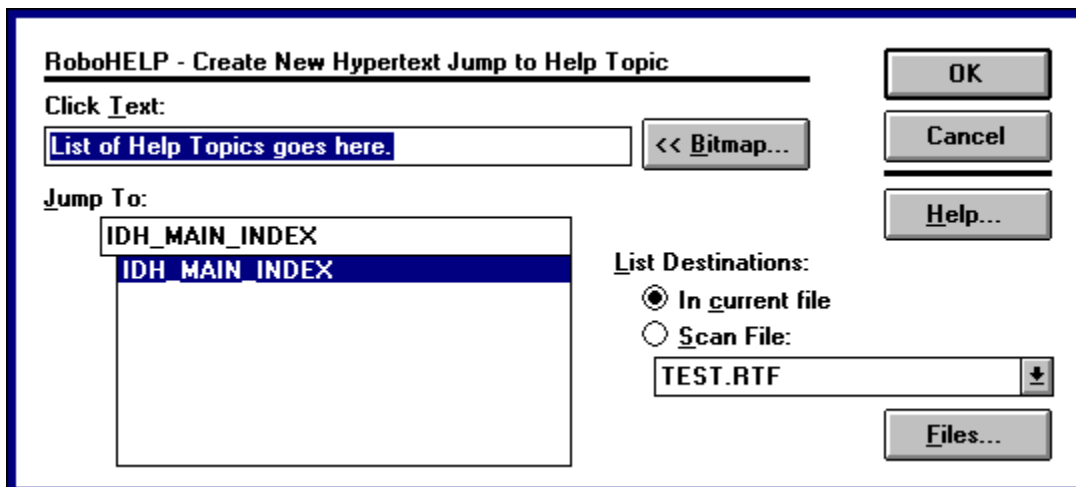
Select the topic you want to jump to and RoboHelp creates the jump by underlining your chosen word or words and placing hidden text after them that specifies the destination topic.

Next you can create popups. To create a popup you follow the same basic procedure as creating jumps except after selecting a word or word you click on the POPUP button. RoboHelp the displays the following dialog box listing all of the topics you have created.
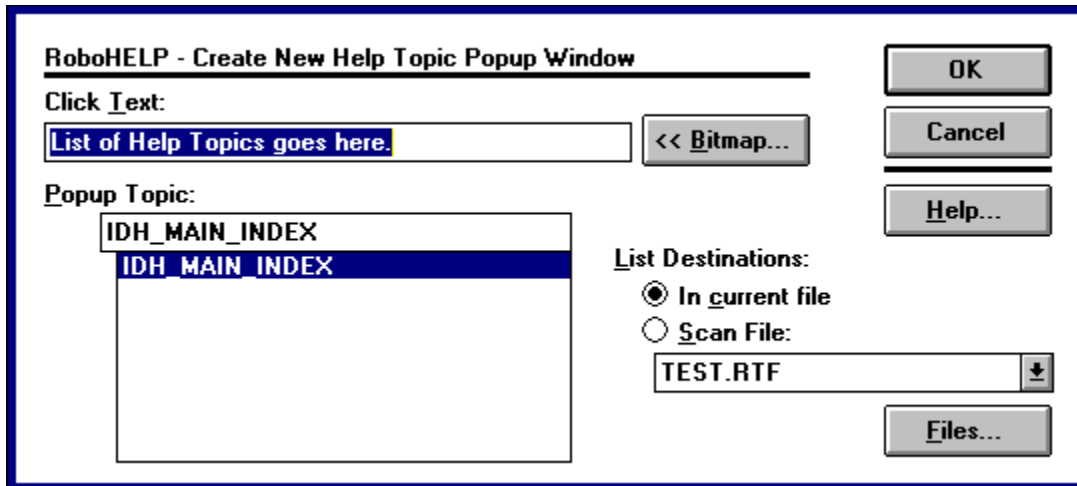
Select the topic you want to popup to and RoboHelp creates the popup by underlining your chosen word or words and placing hidden text after them that specifies the destination topic.

You can also place .BMP pictures in your help files. Keep in mind that the pictures can greatly increase the size of your help file. You can place the same picture in the same document as many times as you want without increasing the size. For example you could have either 10 small .BMP bullets or 100 and the file size will be the same.

To place a bitmap click on BITMAP BY REFERENCE from the INSERT menu (or by pressing CTRL-B). All of the .BMP files you want to use must be in the current work directory. RoboHelp will then display the following dialog box.

Select the bitmap files you want and the position you want them in and click OK. RoboHelp will the automatically place the necessary code to display the bitmap in your help file.

Once you have all of your jumps, popups and bitmaps the way you want it's time to save and compile. RoboHelp makes this an easy process. First click on the Save and Generate button.

This causes RoboHelp to save the file in .RTF format and creates the necessary .HPJ file. Next you will click on Make Help system button.

This causes RoboHelp to call the Help Compiler and make the .HLP file. Finally you will click on the Run The Help System button.

The only thing I would like to see added to RoboHelp is some way to track the jumps automatically. I had to make up an outline to keep track of all my links in the large test file I did for this review.

Overall , RoboHelp is great help to help file writers. It is very simple and easy to use. After completing my test help file using RoboHelp I did it all over again using just WinWord. Using RoboHelp cut the time in half that it took to develop a large .HLP file.
The RoboHelp manual contains information that will be very helpful to people that have never created a help file before. A good help file gives your program a professional appearance and RoboHelp makes it quick and easy.

ROBOHELP

REQUIREMENTS:

                    WINDOWS 3.X
                    MICROSOFT WORD FOR WINDOWS
                    MICROSOFT HELP COMPILER
                    4MB RAM

FOR MORE INFO CONTACT:

                    BLUE SKY SOFTWARE
                    7486 LA JOLLA BLVD., STE. 3
                    LA JOLLA,   CA   92037

                    PHONE: (619) 459-6365
                    FAX: (619) 459-6366

Author

This review was written by Mike Able, a shareware author writing programs in Visual Basic for Windows.

HELP MAGICIAN

By <u>Mike Able</u>

       Help Magician is the only .HLP file creator that does not require a Word Processor. It is a completely stand alone help file creator (the only other item needed id the Help Compiler itself - it is not included in the package). It is a standalone Windows program that lets you edit your help file from within Help Magician


       The editor lets you define font attributes for your help file including font name, size and color. It also supports cut, copy, paste, find and replace. Text can be pasted from the Windows Clipboard into the editor. You can also import and export ASCII and RTF files directly into the editor.

       Each page in the editor is a help Topic, with a name you provide. On each page you will define text to be jumps or pop-ups. This is a simple process done by marking the desired text and clicking on the Jump or Popup button and selecting the destination. You edit the help file by paging through your help file by clicking on the Pg Up and Pg Dn buttons. You can also pick a .BMP graphic to display in your help file.

       While working on your help file you can go into a Test Mode, that simulates Windows help system. This lets you instantly test your help system without having to go through the compile process every time you want to see if a portion of your help file is working correctly.

You can test your jumps, pop-ups and view any bitmaps included in your help file.

After you have completed and tested your help system the next step is to create a .HLP file. Select Write RTF from the build menu and Help Magician will create the proper .RTF file for you. Next select Run Compiler from the build menu and your help system will be compiled into a .HLP file. By clicking on Call Winhelp.exe you can view your completed help file right from within Help Magician.
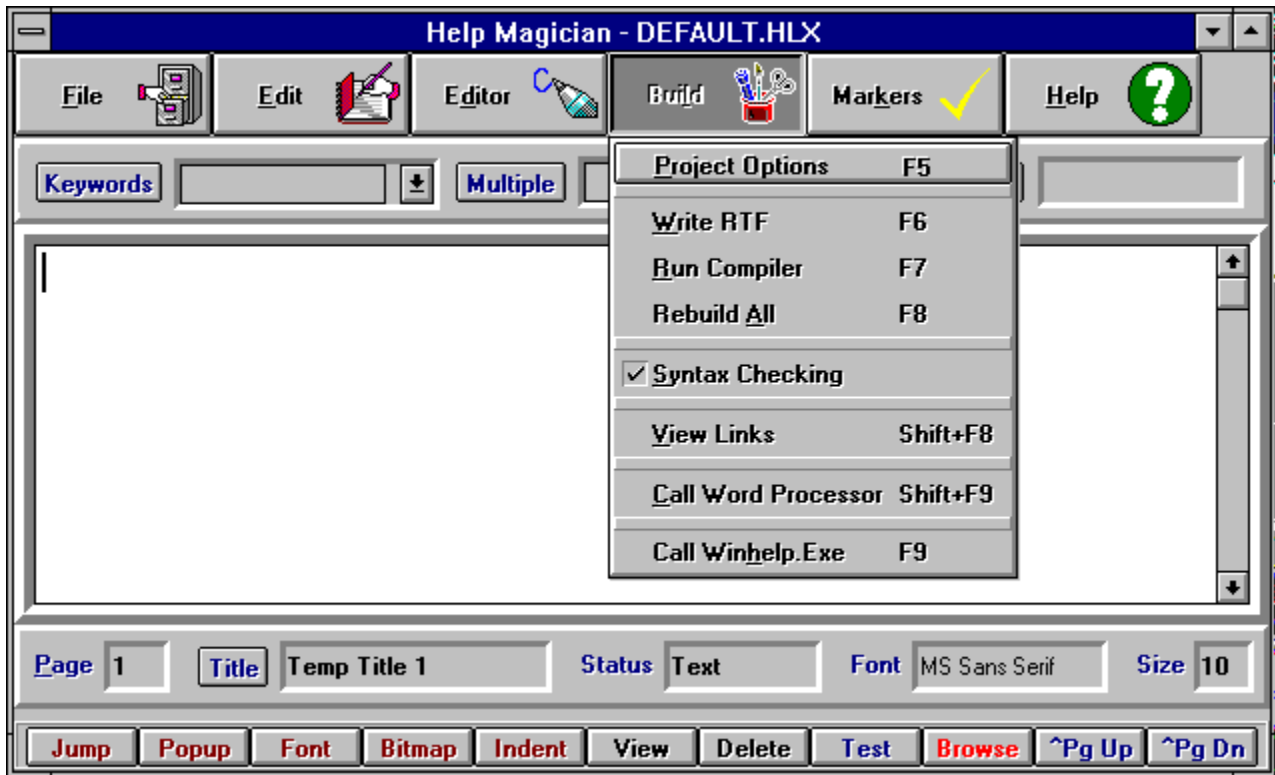
One feature I found very useful is a command called View Links that is available from the build menu. This allows you to view the list of links for your jumps and pop-ups. You can also print the lists out. This feature is a big help in keeping your help file organized.

My only complaint with Help Magician is that the help file is limited to 1 megabyte. When putting together a large help file for a complex program it may be possible to exceed this limit.

If you don't have an RTF compatible word processor this is an excellent choice. Even if you do this may be a good addition to your toolkit due to its Test Mode and other handy features. With Help Magician you will be cranking out help files in no time and the list price is a reasonable $199.00.

THE VISUAL GUIDE TO VISUAL BASIC FOR WINDOWS BOOK REVIEW

By Dave Johnston

If you read the book review I did on Visual Basic Super Bible you will remember that I think the manuals that come with Visual Basic are good. However, I still collect and read all of the 3rd party books I can lay my hands on. Well, this month I am going to cover another good reference book. THE VISUAL GUIDE TO VISUAL BASIC FOR WINDOWS. This is also definitely not a book that you read from cover to cover. It is a huge book at almost 1400 pages. This book covers Visual Basic version 2.

The book contains three sections. Section I is a step by step guide to programming Visual Basic. Section II is the Reference A-Z. Section III is the four appendices.

The first section covers the elements of Visual basic - explaining forms, events, properties, subroutines, and modules. Then continues on with explaing the various menus of Visual Basic - file, edit, view, run, debug etc. The last part of Section I shows the creation of a disk view program and then goes on to expand it into a graphics viewer. While I feel this section is very well written it is only 44 pages long. I don't think I would have called it "A step by step guide to programming Visual Basic". That section title seems a little misleading. I let my wife (who is just getting started with VB) read the section and she felt the same way.

Section II is the meat and potatoes of this book. This huge section covers from page 47 to page 1321. It is a thorough examination of Visual Basic 2.0 commands. The list is in alphabetical order and starts with ABS and ends with ZORDER. Each entry in this section includes uses for the command, cautions to keep in mind when using the command and an example of the command in use. The writing style is plain English and very easy to read. The book is also filled with hundreds of screen shots of the examples and makes there understanding easier. This book has replaced my copy of The Super Bible as the book I turn to when I need information on a command.

Section III has four appendices covering the following:

☞ Windows API

☞ Getting Rid of Bugs

☞ Operators

☞ Add-on Products for Visual Basic & Windows

These are small sections that just briefly cover there topics. The subject of the Windows API can fill a book this size all of it's own.

There is also a companion disk that includes a plain ASCII text of all of the examples from the book. Since I did not have the Disk I really can't say much about it. It does cost an extra $29.95 plus $3.90 for shipping.

OVERALL I HIGHLY RECOMMEND THIS BOOK FOR VISUAL BASIC FOR WINDOWS VER.2 PROGRAMMERS!

This book is published by Ventana Press. It is written by Richard Mansfield.
List Price is $29.95

Ventana Press
P.O. Box 2468
Chapel Hill, NC 27515
(919) 942-0220

Author

This column was written by Dave Johnston. If you have any questions or ideas for topics for this column please contact me through ELECTRIFIED VISUAL BASIC MAGAZINE:

Dave Johnston
7151 El Cajon Blvd. STE. L
San Diego, Ca   92115

MINI-REVIEWS

Mini-Reviews are small   reviews of shareware and public domain utilities, source code, reference files etc. These are great in the fact that they are good learning tools and you can stop re-inventing the wheel. They are usually smaller and do not warrant a full review in another section of this magazine.

VBBOOK.ZIP
Bytes: 31949
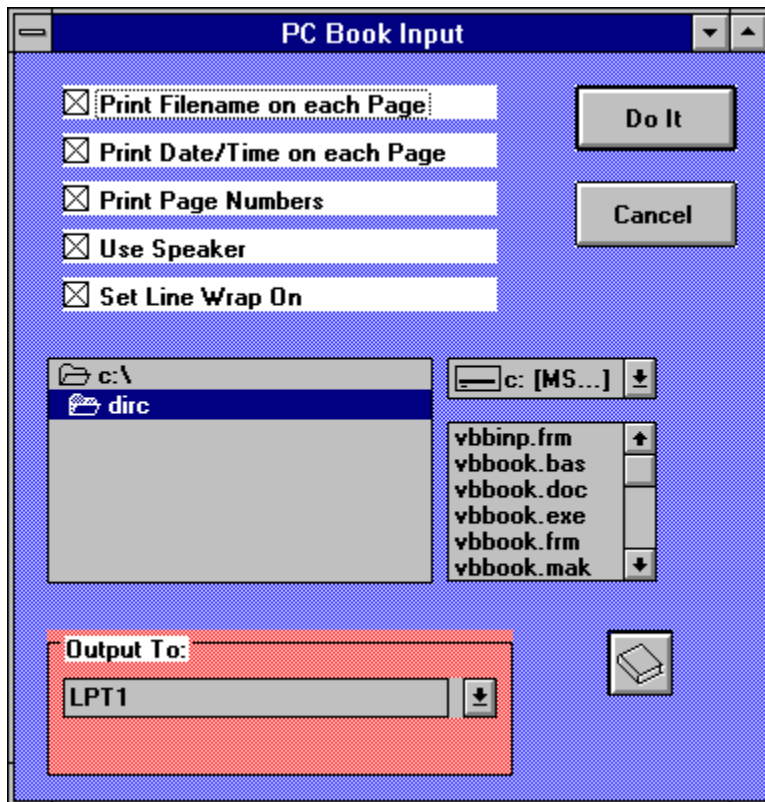
This is a conversion of PC Book from QuickBASIC to Visual Basic. VB BOOK prints ASCII files in booklet format on an HP LaserJet II, IIp, or III printer.   Uses two-sided printing in landscape mode, with Line Printer font so that four standard pages of 80 characters per line, 66 lines per page are printed on each sheet without formatting changes. Includes executable and full source code.

SBAR2.ZIP
Bytes: 15104
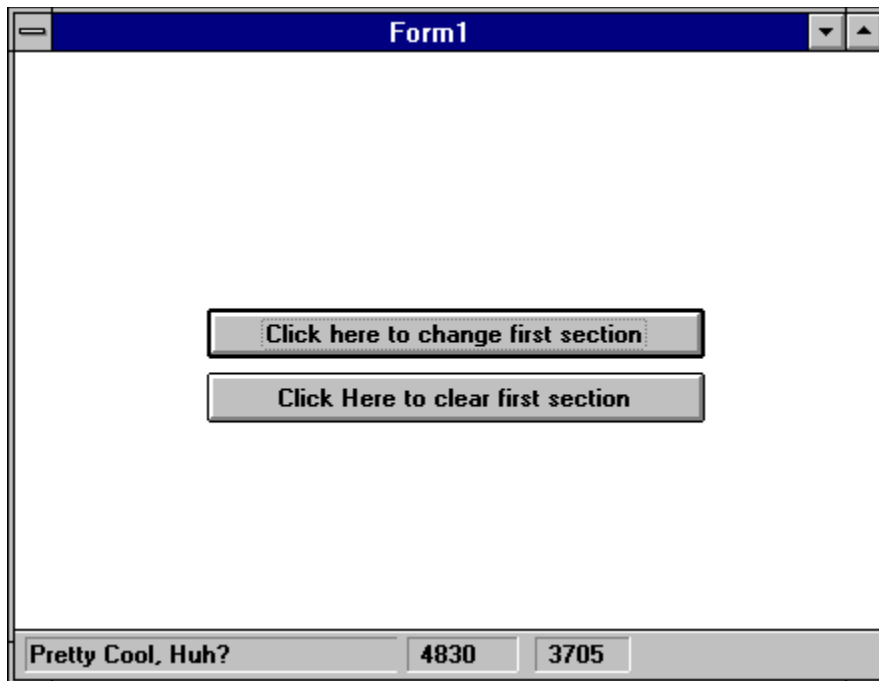
This is a Visual Basic status bar control! To use it, simply click on File - Add File and load status.dll. You will then see a button that looks like a status bar in the toolbox. You can now use the status bar control as if it came with Visual Basic.

The current version will (at run time) automatically force all instances of status bars to the bottom of the screen and will force the first instance to be the width of your form and the all others to be flush right. If the form is resized, the bar will remain "glued" to the bottom. The first section will resize itself to the width of the form and the others will remain glued to the bottom right.

You can change captions, colors and so forth as you would any other control.

SOUNDL.ZIP
Bytes: 6129
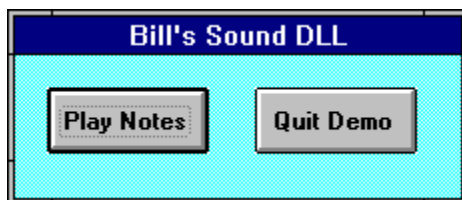
This is a sound DLL for use with Visual BASIC. This little demo and the source code are placed into the public domain for any who would like to use it or are just curious about how to call these sound functions. Included is the source code for the DLL in Turbo Pascal for Windows, the compiled DLL that can be called from any language, the declaration for the call

in Visual BASIC (the syntax should apply to most language calls), the compiled Visual BASIC demo program, and the source code for the Visual BASIC demo program.

ADIALR.ZIP
Bytes: 21138
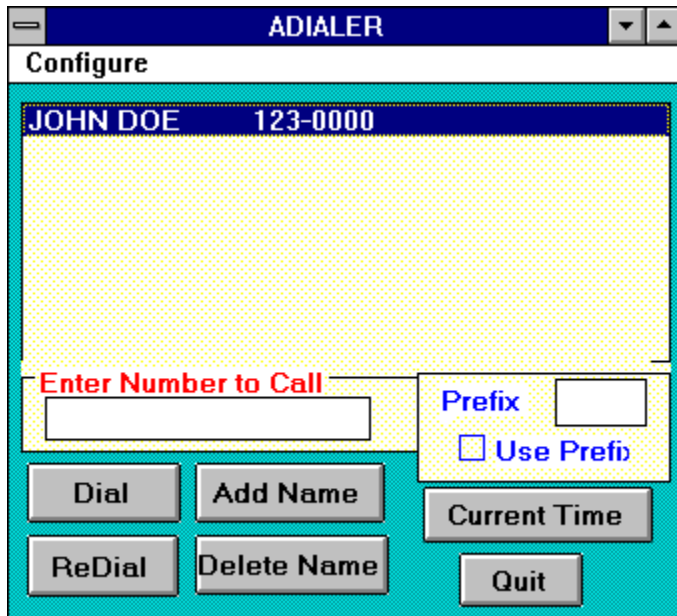
This is a simple phone dialing program   Names can be entered in and stored in the list box.   A number can also be entered directly into the entry text box.   When a number has been selected or entered it will be dialed when the Dial button is pressed.   When the dialog box appears, the phone should be picked up and the enter key pressed to disconnect the modem and engage the phone line. The ReDial button will redial the last number called. If a prefix for long distance or and outside line is needed, it can be entered in the Prefix text box and toggled using the check box.   The current time switch will give the time from the system clock for those occasions when an answering machine is engaged and it's necessary to leave the time called.

This program is freeware and includes the source code to show the "simple" API calls necessary for comm port access and dialing with a modem.

FREE2V.ZIP
Bytes: 30759

This is a FreeWare version of two Visual Basic controls.   The about property of these controls contains all of the information needed to program with them.

TimeQueue is an event string sequencer with a resolution of 1/10 second. All controls of this type share a common time.   Strings-time pairs are added to the queues of the controls and when the time matches the control time, an event is generated with the string as a parameter.

TextKey is an interactive text display control.   Pages of text can be displayed with word wrap and keywords.   There is a separate cursor for keywords and when the left mouse button is clicked over a keyword an event is generated with the keyword as a parameter.

MENUON.ZIP
Bytes: 6176
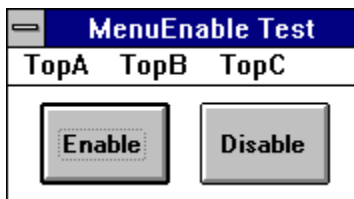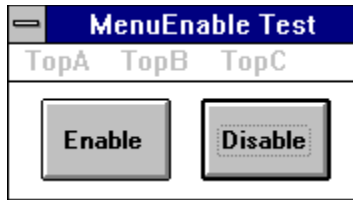
MENUON is a sample program and VB source code to demonstrate enabling and disabling menus.

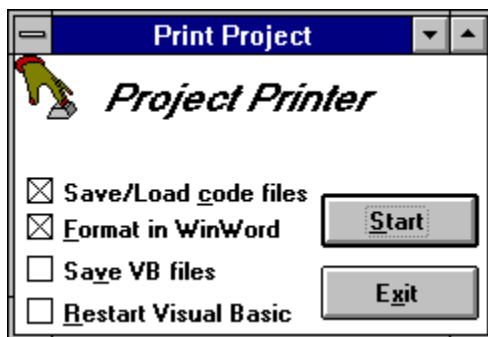## PROJPR.ZIP
Bytes: 24076

Project Printer allows you to get a nicely formatted listing of your Visual Basic file in Word for Windows, while also saving text/loading text for you. Includes VB source code.

This program has the following features:

1.   It save/loads Visual Basic code to get rid of all the junk from previous versions of your program that would otherwise increase the size of your EXE file.

2.   It formats your code nicely in Word for Windows - if you've ever wanted hard copy of your program, you know that getting a nice printout is a problem. Project Printer formats your code nicely in WinWord, complete with outlining features, and you can print from there.

3.   The code shows you how to do neat things, such as find out from design time Visual Basic what path your project is in, and what files are in the project. It does that by using SendKeys, GetActiveWindow, GetWindowText, and list box messages such as LB_GETTEXT and LB_GETCOUNT.

BMPKIT.ZIP
Bytes: 22566
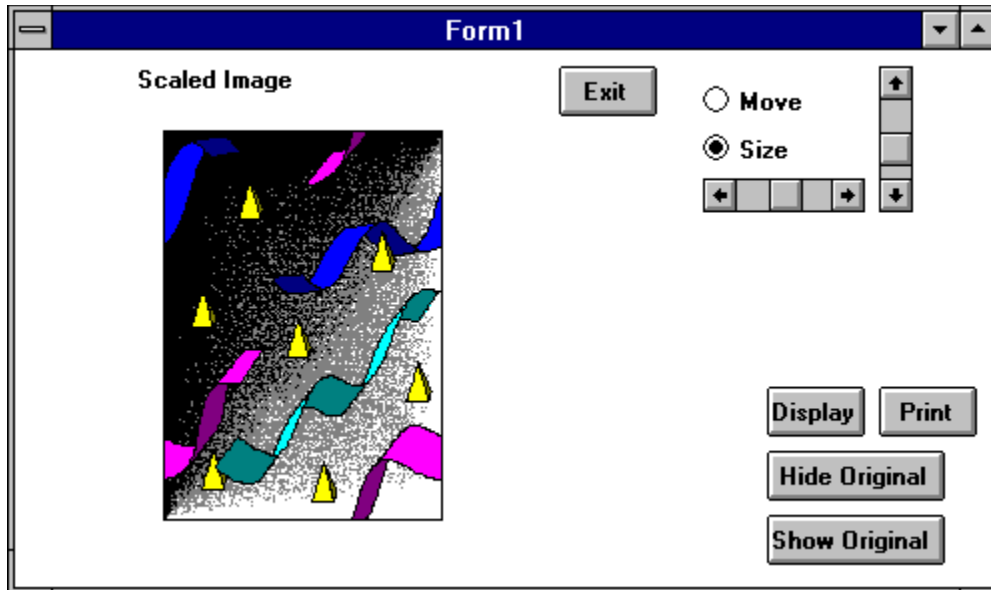
This a set of routines for dealing with .BMP files from within Visual Basic. Using these routines, you can scale, print, and display bitmaps in VB applications, something not included in the internal commands in the language. Includes a demo application with VB source code which deals with scaling and printing of   images.

These files are available in the Shareware libraries in the PRODUCT SHOWCASE section of this magazine.

-OR-

Members of the BBS can obtain the above files form the Electrified Visual Basic BBS.

ELECTRIFIED VISUAL BASIC BBS
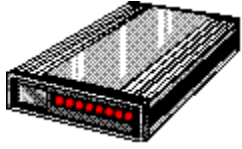
This is a BBS for Visual Basic programmers. It is running high speed BOCA 14.4 V32bis modems. It is a fully graphical, mouse driven BBS. Use your mouse to click on icons for all BBS operations. It has many Visual Basic file sections for VB source code, Shareware and Freeware utilities / custom controls etc. There are also many general Windows file sections on-line. These include Windows games, utilities, icons, fonts, graphics, WinWord files, Word Perfect files, programming etc. There are also message sections dedicated to Visual Basic programming. It also has sections for Electrified Visual Basic Staff and Freelance writers.

CALL (619) 464-0188 with your modem today!

Using Microsoft Word for Windows to Create Help Files

By <u>Richard Prince</u>

<u>Please Read Me First</u>

<u>What you need to get started</u>
<u>Overview of Help File construction</u>
<u>Word for Windows and Help Files</u>
<u>Constructing Help .RTF Files</u>

Syntax References:
<u>Footnote Syntax</u>
<u>Creating Jumps</u>
<u>Creating Pop-ups</u>
<u>Bitmap Displays</u>
<u>Bitmap Jumps</u>
<u>Bitmap Pop-ups</u>
<u>.HPJ file</u>

<u>notes</u>

Despite what you may have heard, it is possible to create decent help files by using only Word for Windows and the SDK help compiler (hc31.exe or hc30.exe).   There are at least two Microsoft SDK books describing Help Files and their construction -- but, neither gives the information about how all these cryptic commands can be entered using just Word for Windows.

   This help file was constructed using: **1**) **Word for Windows 2.0a   2**) **hc31.exe** and **3**) Windows' **Paintbrush** (for icons).   If you want absolutely first rate Help Files, that are relatively easy to construct -- check out the RoboHelp and Windows Help Magician reviews elsewhere in this issue.

Requirements for Creating Help Files

For Basic Help Files -- You'll need the following items:
   Microsoft Windows 3.1
   Microsoft Help Compiler (hc31.exe)
   Microsoft Word for Windows 2.0a

It's Sure Nice to Have --
   Microsoft SDK "Programming Tools" booklet
   Microsoft SDK Hotspot Editor (SHED.EXE)
   Microsoft SDK Image Editor (IMAGEDIT.EXE)
   Patience and time to experiment

Overview of Help File construction

When you create a C program, many people use a **makefile** and run **nmake makefile** when creating the executable file.   The **makefile** is just a text file that gives the names of various files and instructions to the compiler/linker to create the executable program.   In the same fashion, the Help Compiler (**hc31.exe**) requires a text file with instructions on how to build the help file. For a C program the makefile will usually have .c, .h, .def, etc. files referenced.   The Help Compiler usually only needs to know **.rtf** (rich text format) file(s).   The Help project file that contains the instructions and file names is a **.hpj** file instead of the C **makefile**.   A basic Help file project can consist of just two files: **somename.rtf** and **somename.hpj**.   To make the help file you would enter the command **hc31 somename.hpj** from the DOS command line.   When creating the files using Word for Windows -- be sure to choose **Save As** from the **File** menu and then select **text only** for the **.hpj** file and **rtf** for your **.rtf** file.

Just like the C **makefile** has syntax rules , the **.hpj** file also has syntax rules.   View the topic for .HPJ syntax for more information.

A popular thing to do is to add Bitmaps (pictures) to the Help file.   Bitmaps can be used as just pictures, as jumps to other topics, and pop-ups.   You can use Windows' Paintbrush to create the Bitmaps (.bmp files).

Microsoft Word for Windows is used to create the **.rtf** and the **.hpj** files.   Creating the **.rtf** file is easy because you should already be familiar with the basics of Word for Windows.   In addition, it's easy because the text you see in Word is what you will see in the Help File.   View the topic Constructing Help Files for more information about using Word to create the **.rtf** files.

Word for Windows <--> Help Files

Here are some suggestions to try when you are just starting to make help files.   As you gain more experience and feel more comfortable with it -- you'll develop your own "routine".

Set Word for Windows
   Before you start anything -- you need to know that much of the Help Files process makes use of "hidden" text and "footnotes".   You should already have an general idea about what footnotes are.   You will need to set Word for Windows to show hidden text.   Start Word for Windows, but before you go too far, choose **Tools** then **Options** from the menu. In the **View** Category -- make sure the **hidden text** box is checked in the Non-Printing Characters section.
   It's a pretty good idea to use Helv 10pt font for much of your text to start.   It doesn't look bad, and most systems have this font on them.

First Time suggestions
   1.   Try making a first "test" Help file.   This is just to see how things work.   Make up a short file just to experiment with.
   2.   Taking a little time and maybe a few notes before you start your text file usually pays off in less confusion when it comes time to do jumps, etc.   In other words, try as best you can to "rough out" how the Help File looks and operates before doing a lot of typing.
   3.   After reading through these notes.....Open a .rtf file (like wfwhlp.rtf) and take a look at some of the things you have been reading about.

Constructing .RTF files

Be aware of what Windows takes care of and what you must take care of when it comes to Help Files.   Windows will handle the help window size and wrapping text in the window.   You have to take care of fonts, sizes, colors, and bitmaps/graphics in your .rtf and .hpj files.   Windows takes care of the Contents, Back, and History buttons.   You have to tell windows to use the Search and/or Browse (<<   >>) buttons in your .rtf file.

IN A NUTSHELL --

Help files are very similar to a card file.   Each "screen" in the Help file is like a single card.   You get to different screens by selecting one of the "jump" words.   Jump words are green with a solid underline.   When using Word for Windows to create the .rtf text file -- each screen is a separate page.   Each of the pages is referred to as a "topic".   Each topic has a unique tag(character string) associated with that page.   Each topic **must** have a string.   Each topic **can** also have a title, keyword, and identifier.   If you use a keyword, you must also use a title.   Since each topic(i.e. page) is identified with a string, jumps are possible by using the string.

The string, title, keyword, and identifier are stored with the .rtf text file in the form of footnotes.   For each topic -- you enter the string(and the optional title, keyword, and identifier) as a footnote using "special" characters for the footnote mark.   Instead of the footnote belonging to a particular word -- as you might in a report -- these footnotes belong to the topic(i.e. page) and are usually found at the start of the page.

Just so you won't get confused when reading Microsoft documentation; the string I mentioned above is called a "context string".   The items and their associated footnote marks are as follows:

Item	Footnote mark
context string	#
title	$
keyword	K
identifier	+

Simple Help File listing --

....

   #   **First page title**   ("#" footnote mark for pg 1)

   press <u>jumppage_two</u> to jump to the next page.

....

\# main_index   (footnote for page 1)

....

   #   ("#" footnote mark for pg 2)
   came here from page one.

   press <u>jumpmain_index</u> to return to page 1.

....

 \# page_two   (footnote for page 2)

....

First Time Tips -- Take a look at the wfwhlp.rtf file provided to see the footnotes, etc.

Footnote Syntax

footnote mark	meaning
#	context string
$	title
K	keyword
+	browse sequence

Footnotes are associated with a topic (page) as opposed to an individual word or phrase. Generally, footnote marks appear at the beginning of the page.

Jumps Syntax

Jumps are created by identifying a word or phrase to be clicked on by the user to jump to another context string (page).   In Word for Windows the jump word is double underlined by selecting Format then Character then double underline from the menu.   Immediately following the double underline jump word is the context string name in hidden text.   In Word for Windows hidden text is done by selecting Format then Character then hidden from the menu.   Hidden text is displayed with an underline of dots.   Windows takes care of making the word green colored in the help file.   Sample of a jump with hidden context topic:

jumppage two

Pop-up Syntax

   Pop-ups are created by identifying a word or phrase to be clicked on by the user to pop-up a dialog box containing context string info.   In Word for Windows the pop-up word is single underlined by selecting Format then Character then single underline from the menu. Immediately following the single underline pop-up word is the context string name in hidden text.   In Word for Windows hidden text is done by selecting Format then Character then hidden from the menu.   Hidden text is displayed with an underline of dots.   Windows takes care of making the word green colored in the help file.   Sample of a pop-up with hidden context topic:

popuppopup_string

Bitmap Displays

Only 16 color bitmap displays are supported.

Graphics can be incorporated by using the normal procedures in Word for Windows.

Bitmaps can also be used for jumps or pop-ups.   Bitmaps can also be displayed in either the right or left margin with text wrapping around them as shown by the bitmaps at left and right. Bitmaps can also be displayed as a "character" where the bitmap will be displayed with the bottom and lower left starting at a character location as shown below:

some text some more text.

Bitmap Jumps

   Bitmap jumps are created by identifying the bitmap to be clicked on by the user to jump to another context string (page).   In Word for Windows the bitmap style and filename is double underlined by selecting Format then Character then double underline from the menu. Immediately following the double underline bitmap info is the context string name in hidden text.   In Word for Windows hidden text is done by selecting Format then Character then hidden from the menu.   Hidden text is displayed with an underline of dots.   Sample of a bitmap jump with hidden context topic:

{bmc filename.bmp}jumpto contextstring

Bitmap Pop-ups

Bitmap pop-ups are created by identifying the bitmap to be clicked on by the user to pop-up another context string (page).   In Word for Windows the bitmap style and filename is single underlined by selecting Format then Character then single underline from the menu. Immediately following the single underline bitmap info is the context string name in hidden text. In Word for Windows hidden text is done by selecting Format then Character then hidden from the menu.   Hidden text is displayed with an underline of dots.   Sample of a bitmap pop-up with hidden context topic:

{bmc filename.bmp}popup contextstring

.HPJ Syntax

   There are lots of styles/options to the .hpj file.   Best to read the documentation if you have it.
The following is a sample .hpj text file with comments (comments begin with a ";"):

```
; sample .hpj file
[OPTIONS]
CONTENTS=main_index     ;usually context of 1st pg
TITLE=Help Window Title string
COMPRESS=TRUE       ;compress hlp file
WARNING=3                ;higher more strict
   << optional >>
BMROOT=path              ;path to .bmp files
ROOT=path                ;path to project
ICON=filename.ico        ;minimize icon to use

[FILES]
filename.rtf             ;must include all .rtf's
   << and/or >>
#include <filenamelist.h>

[CONFIG]
BrowseButtons()          ;to use browser

[BITMAPS]
first.bmp                ;section can be skipped
second.bmp               ;if BMROOT is used
```

NOTES

The usual copyright info is in effect.   Mostly everything here is by Microsoft including:
hc31.exe -- Word for Windows -- Windows -- Paintbrush -- etc.

After going to all the trouble to add browser identifiers in your .rtf file -- don't forget to enable the browse buttons by adding **BrowseButtons() macro in the **[CONFIG]** section of the .hpj file.

**Multiple keywords ("K" footnote mark) can be assigned by separating with semicolons (i.e. K First Keyword;Second Keyword;etc.)

You **must provide a title ("$" footnote mark) if you assign keywords.

Author

This column was written by Richard Prince and donated by him for use in the magazine.

PRODUCT SHOWCASE

VISUAL BASIC SHAREWARE LIBRARY

VISUAL BASIC SHAREWARE LIBRARY II

ICON LIBRARY

BUSINESS PLANS
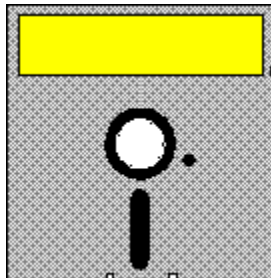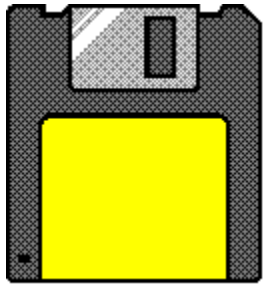
VBMAN

EVB BBS

NEW VISION

VB PROJECT TOOLS

ADVERTISE IN EVB MAGAZINE

VISUAL BASIC SHAREWARE LIBRARY

This library contains Visual Basic Shareware, Freeware and Public Domain files. All of these files are specially picked with the Visual Basic programmer in mind. You get plenty of Custom Controls, DLL's, On-line Help files, Source Code and more. This library contains no filler (some VB file libraries come with regular Shareware programs that are of no real use by VB programmers or come with no availability of Source Code - NOT THIS LIBRARY!) NOTE: SHAREWARE PROGRAMS MAY REQUIRE SEPARATE PAYMENT MADE DIRECTLY TO THE AUTHOR FOR CONTINUED USE.

VB COMMUNICATION FILES

COMMDEMO.ZIP - A limited but functional comm program. Partial Source Incld.
VBCOMM.ZIP - A communications control. Includes sample App.
VBTERM2.ZIP - A basic communications program. Source Code included.
MHCOMM.ZIP - Term program by MicroHelp. Shows what can be done with their VB Addon.
INSTAL.ZIP - Allows easy setup of shared Apps on a Network. Source Available.

VB CUSTOM CONTROLS

CCFACT.ZIP - A Demo of Custom Control Factory. Animated pushbuttons, 3-D effects etc.
CLPSIB.ZIP - Allows overlapping controls. Source code for a demo is included.
DROPDOWN.ZIP - Shows how to simulate a Combo Box with a Text Box & a List Box.
ETDEMO.ZIP - Source Code as a demo of EditTool which creates custom controls.
INPUT.ZIP - A TextBox control that adds 3-d,formatted input, justification.
MLISTBOX.ZIP - A control for extended and multi selection list boxes.
MOVTXT.ZIP - Shows use of a Picture control to simulate dragging text controls.
MSGBOX.ZIP - Easy way to set up MSGBoxes. This utility generates the source.

MULTSEL2.ZIP - A multiple selection list box control.
STSBAR.ZIP - C. Control to place a status bar "glued" to the bottom of a form.
VB4CTRLS.ZIP - Four controls to add 3-d effects, gauges etc.
VBLISTBX.ZIP - Control for multiple selection and multiple column list boxes.
VBMBTTN2.ZIP - A multi-button control that divides itself into sub-buttons.
VBPIC3D.ZIP - Control to provide buttons with pictures and 3-D shading.
FMDROP.ZIP - Control that allows VB to react to files dropped onto VB forms.

VB DATABASE FILES

PDOXDE.ZIP - A demo of using the Paradox engine. Paradox DLL req.
PXCREATE.ZIP - Routine allowing the creation of Paradox tables. Source Incld.
VBDB.ZIP - A database engine that supports dBase format & DDE.
KWI.ZIP - A small Address book with Visual Basic Source Code.

VB DLL'S

ENUMFONT.ZIP - A DLL to call Windows API function ENUMFONT. Source Incld.
HOTKEY.ZIP - A DLL to provide Hotkey support for VB Apps.
HUGEARY.ZIP - DLL to create, maintain & delete arrays greater than 64k.
HUGEGRID.ZIP - Example source to use HUGEARY with the grid control.
MHELP.ZIP - Free DLL from MicroHelp INC. Has VB versions of PEEK, POKE, INP, OUT.
SNDEX.ZIP - A Soundex DLL for Visual Basic.
VBPTR.ZIP - A dll to provide pointer support. Allows Bload/Bsave, MK$, CV?.
VBSW11.ZIP - A DLL to return window handles to active apps. Sample Source Incld.
VBUBBLE.ZIP - A DLL to provide bubble type help windows.
PSETUPCK.ZIP - DLL to access the standard printer setup dialog box. Sample source included.
PRSET.ZIP - A DLL for controlling & configuring printers. Source Included.
WFIND.ZIP - A DLL to supplement the AppActivate command.   Source Incld.
VBBTRV.ZIP - Example code for using Btrieve in VB Applications.

VB DISK ACCESS & FUNCTIONS

DSCAN.ZIP - A .BAS file to scan drives for a certain file.
FBR12C.ZIP - File Manager to Load/Run Apps. Source Included.
FILBX2.ZIP - A Windows type file open dialog box with source code.
FILER1.ZIP - File Finder to view, delete files & print DIR's. Source incld.
FILBX2.ZIP - Windows style file open dialog box. Source included.

VB FONT

FNTCNT.ZIP - Displays and prints fonts. Source code is Available.
FNTVIEW.ZIP - Source Code for an App to find, display & print Windows fonts.
FONTRO.ZIP - Shows how to print text at any angle in V.B. Applications.
VBFON2.ZIP - A simple font viewer. Visual Basic Source Code included.

VB GAMES

ACCRD1.ZIP - Solitaire game done with VB. Source Code Included!

CARD10.ZIP - A card deck DLL for writers of VB card games. Inclds 52 cards, Joker, 7 card backs.
G4K110.ZIP - Four games in one for kids. Source Code is available.
MYMEM1.ZIP - A memory matching game. Source Code is included.
ROLLEM.ZIP - The dice game Craps. Includes Source Code.
RUTHIE.ZIP - A game for young children. Source code Included.
SIMON.ZIP - The challenging memory/color/sound game. VB Source Code incld.
NUMGAME.ZIP - Childrens game for +, -, x, and division. Source Included.
TAB100.ZIP - Card game for two players. Visual Basic Source Code Included.

VB GRAPHICS & VIDEO

256PB2.ZIP - 256 colors in VB picture box controls.
3D4VB.ZIP - Create 3-d effects such as shadowed boxes & text etc.
ANIMAT.ZIP - Source code for creating smooth animation.
BITMAP.ZIP - Source to show how the use of the V.B. Picture Box functions.
BMPKIT.ZIP - Routines to scale, display & print .BMP's. Source incld.
BTDEMO.ZIP - Shows a easy way to create command buttons with pictures.
BUTTON.ZIP - Simulate command buttons with pictures on them. Source Incld.
GRAF20.ZIP - A control to display up to 24 bit color .BMP and .PCX pictures.
ICOBMP.ZIP - Converts .ICO icon files to .BMP format. Source Incld.
ICONDEMO.ZIP - Source to show how to draw graphics on a V.B. App's icon.
ICONDLL2.ZIP - Build a DLL from a collection of icon (.ICO) files.
ICONLOOK.ZIP - An .ICO viewer. Full Source Code is included in the Docs (.WRI).
ICONVIEW.ZIP - An icon viewing utility can save to clipboard. Source incld.
METALIB.ZIP - Routines to create, draw & save metafiles (vector graphics).
MULTIPIC.ZIP - How to apply pictures to buttons without loading from disk.
POLYGO.ZIP - Source for a graphics display. Shows use of iconized graphics.
SSAVER.ZIP - Allows you to create a screen saver (like After Dark) in VB.
VBFLIP.ZIP - Source shows how to store series of images in a single file.
VBMASK.ZIP - How to create a mask from a image & screen object using BitBlt.
VBMENU.ZIP - Shows how to add graphic bitmaps & check marks to a menu.
VBSCRL.ZIP - Demonstrates how to scroll/pan across a bitmap in a picture box.
WALLCH.ZIP - Automatically changes your Windows wallpaper. Source code Incld.
3DDEMO.ZIP - A demo of 3D4VB.ZIP in this file section.
WLPEEP.ZIP - Scroll through BMP, WMF, RLE & ICO files. VB 2.0 Source Included.
WMFBMP.ZIP - Converts WMF files to BMP format. VB 2.0 Source Included.
FADE.ZIP - Routine to make the background of your form fade from blue to back.

VB MISC. FILES

AREACODE.ZIP - Find the name for US & Canadian areacodes. Source Incld.
BASECVRT.ZIP - Converts numbers between bases. Source Code Included.
CM01.ZIP - A phone / address manager. Source available.
EZHELP.ZIP - An easy way to add help to your Apps. Inclds Source Code.
FXLAUNCH.ZIP - Mini launcher for Metz's programs. Source Incld.
LOSE1.ZIP - A program to help learn VB. Inclds Source code.
ORDER.ZIP - Order form for Shareware VB programs. Figures total & Tax etc.
QUITWIN.ZIP - Instant Windows exit with animation & sound. Inclds how.
VBLOTTO.ZIP - A lottery picker. Visual Basic source code included.
VINST4.ZIP - An updated install for VB Pro Toolkit.
WAVPLY.ZIP - A .WAV sound player includes Sourse Code.
WINDCH.ZIP - Windchill computations. Includeds VB Source code.
PSOUND.ZIP - Routine to place sound into VB programs. Source Incld. (Req. SoundBlaster).

LZSSLI.ZIP - Compression library for Windows.

VB PRINTER

HPENV.ZIP - Print addresses on envelopes with an HP laser. Source Code Incld.
HPESC.ZIP - Shows how to send escape sequences from VB to an HPIII printer.
NETPRN.ZIP - Network printer linker for Windows. Source code incld.
PRCLP2.ZIP - Print text in the clipboard by clicking on its icon. Source Incld.
PSETUP.ZIP - A DLL to invoke the standard Printer Setup Dialog. Source incld.
PSOUT.ZIP - Shows how to print bitmap files on HP & Postscript printers.
VBBOOK.ZIP - Source for an App to print text files to HPLJ in booklet form.
VBPSETUP.ZIP - A printer setup dialog box includes Source code.

VB PROGRAMMERS TOOLS

COLORS.ZIP - A tool for calculating Visual Basic color values.
IDVB.ZIP - A utility that finds the ID's of controls on VB forms. Source Incld.
KEYTEST.ZIP - VB App to get the hex key code for any key. Source Code Incld.
PROPVIEW.ZIP - Property viewer control displays/prints properties of current form.
VALET.ZIP - Get to your TXT & HLP files, also logs time spent on project.
TOOLBT.ARJ - Vb controlls tool bars.
VBBAR100.ZIP - Adds a toolbar to access commonly used features of VB.
VBCLN.ZIP - Automates the SAVE/LOAD text procedure to shrink your Apps .EXE size.
VBEHAN.ZIP - Additions to VB for printing selected text & save text as file.
VBTBPB.ZIP - Add on ToolBox that automates many VB tedious tasks. Great!
VBUTIL.ZIP - Two programs to copy or delete VB project files.
SAVEME.ZIP - Form with button to quickly save the current file or project.
CPVB22.ZIP - Pretty code printer utility for VB 2.0
VB2-TB.ZIP - ADD on ToolBox for Vb 2.0, adds many enhancments. Great!

VB REFERENCE / INFO

APIREF.ZIP - Reference to functions of Windows API.
KBWINTXT.ZIP - Text file of help and tips for VB programmers.
VB2NEW.ZIP - A text file describing the new features of VB 2.0.
VBTHLP.ZIP - A large collection of tips & tricks for Visual Basic.
MOREAPI.ZIP - Additional API functions. Sample Source Code Included.

VB SYSTEM & LANGUAGE

ABOUT.ZIP - A bouncing command button in an About box. Source incld.
APPSHELL.ZIP - Source code template for standard Windows file open, save, print etc.
ARRANGE.ZIP - Arrange icons. Source code example on how to use SendKeys.
CFIXVB.ZIP - A routine to round Currency variables at a given # of decimal places.
DOSBUT.ZIP - A button to shell to DOS. Source Code Included.
DSL100.ZIP - DOS shell launcher with Source code.
EDITDEMO.ZIP - Input routines for text controls.   Source Code Included.
FNDWND.ZIP - How to look for a window based on its ClassName or Caption.
FOPEN.ZIP - Source Code for an advanced file open routine.
GRDRTN.ZIP - Examples of grid manipulation. Source Code Included.

GRID.ZIP - Test properties of controls on a VB form. Source Incld.
GRID2.ZIP - A modified version of GRID.ZIP.
LABTAB.ZIP - How to use label controls to display tabular data. Source Incld.
LDEMO.ZIP - Example of linked list/Listbox & Speedfill. Source Code Incld.
LISTDR.ZIP - Routine to drag a text line from one location to another. Source Incld.
METER1.ZIP - Routine to implement scrolling gauges. Sourc Code Included.
MSGDEMO.ZIP - How to send messages to the user. Source Incld.
MULTIPIK.ZIP - Multiple selection list box for VB Apps. Source included.
PROFIL.ZIP - 2 routines to manipulate profil entries in WIN.INI.
RECYCLER.ZIP - A Trashcan for file deletions. Source Code Included.
SYSWITCH - Choose between different Windows .INI files. EXE & Source.
VBFLOAT.ZIP - VB source to show how to keep your apps window always on top.
VBMEM.ZIP - Source for a VB App that displays Windows system info.
WEDI.ZIP - A front end for your favorite text editor. Source Included.
WINDWIP.ZIP - A beginnings to an XTREE GOLD clone. Needs more coding.
VBA102.ZIP - Files to provide support in VB for Huge Arrarys.
INIFIL.ZIP - VB Module for simplifying .INI file operations.
INIEDT.ZIP - INI-EDIT for managing Windows .INI files. VB 2 Source Code Incld.

VB TIME & DATE

DATETIME.ZIP - Source code to display the date and time in a variety of formats.
SCARY.ZIP - A clock for Halloween. Source Included.
TINYCAL.ZIP - Source code for a very tiny calendar.
VBCLK.ZIP - A clock App that shows time, date & can change colors. Source Incld.
VBCLOCK.ZIP - Source for an App to display the date and time in a small window.

PRICE $49.95 (Plus S/H)

ORDER FORM

VB DISK LIBRARY ORDER FORM

NAME: _____

COMPANY NAME: _____

ADDRESS: _____

CITY: _____

STATE: _____

ZIP: _____

COUNTRY: _____

The library comes on 5-1/4 HD or 3-1/2 HD disks.

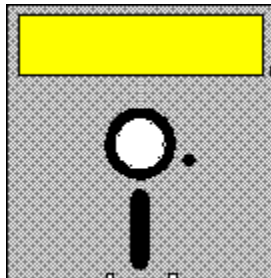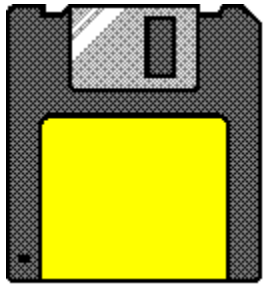DISK PREFERENCE: 5-1/4 [   ]   3-1/2 [   ]

PRICE: $49.95 PLUS $5.00 Shipping ($10.00 Shipping Foreign)

California residents add 7-3/4 % sales tax.

Send check or money order drawn on a US Bank made payable to PRO-DATA and send to the following address:

PRO-DATA / EVB MAG 2

7151 EL CAJON BLVD. STE. L

SAN DIEGO, CA   92115

VISUAL BASIC SHAREWARE LIBRARY II

This library contains more Visual Basic Shareware, Freeware and Public Domain files. All of these files are specially picked with the Visual Basic programmer in mind. You get plenty of Custom Controls, DLL's, On-line Help files, Source Code and more. This library contains no filler (some VB file libraries come with regular Shareware programs that are of no real use by VB programmers or come with no availability of Source Code - NOT THIS LIBRARY!) NOTE: SHAREWARE PROGRAMS MAY REQUIRE SEPARATE PAYMENT MADE DIRECTLY TO THE AUTHOR FOR CONTINUED USE.

Files include:

23PICK.ZIP - Game against the computer in a match of wits. Includes VB Source code.
ADIALR.ZIP - Phone dialing program. Visual Basic Source code included.
AGPR10.ZIP - Agility/VB 1.1 Press Release.
AHELP.ZIP - Code to include a Help Menu in your VB programs.
BAR.ZIP - A gauge type control written in pure Visual Basic source code.
BITSDLL.ZIP - A DLL that supplies bit operations.
CALZIP.ZIP - A small Calendar program. Includes Visual Basic Source Code.
CARD10.ZIP - Card deck DLL for writers of VB card games. Need this for Crispy.ZIP.
CFGTEX.TXT - VB Routine to Configure a Text Box - max length, case conversion.
CLPDEM.ZIP - CLPDEM.ZIP example uses clipboard across forms.
CLPTL.ZIP - Allows you to display, print, save and load text and graphics fro
CLRGAME.ZIP - Children's colorful game. VB Source Code included.
CONTACT.ZIP - Q&E/VB Source links ACT! to Word via DDE.
CRC32.ZIP - A DLL to guarantee the completeness and authenticity of a file.
CRISPY.ZIP - Solitaire variation, with source code. Requires CARD10.ZIP.
CURLIBM.ZIP - Libraries of cursor animations. Include animated cursors in your VB Apps.
CURMAN.ZIP - Control your cursor in VB.
DBHDR.ZIP - How to read DBF file and field structure from the header. Req. QEVB.

DD1A.ZIP - How to implement file manager drag and drop in a VB program. Source Incld.
DEMOBM.ZIP - GUI Custom Control Toolkit demo.
DIALOGS.ZIP - Shows calling the Win 3.1 Common Dialogs of File Open, Save, Print etc.
ENDPRN.ZIP - Source code which makes Print manager the active application. Source Incld.
EXECSC.ZIP - Executive Decision Maker. Includes Visual Basic Source Code.
FAVORITE.ZIP - Program Launcher. Includes Visual Basic Source Code.
FILEBOX.ZIP - How to fake a Windows style file-open box from within VB.   Source Code Incld.
FONTS.ZIP - Lists and displays available fonts. Includes Visual Basic Source Code.
FREE2V.ZIP - Custom Controls for an event string sequencer & interactive text display.
GETFKEY.ZIP   Gets an F key selection. VB Source Included.
GETREG.ZIP - Takes orders for your shareware products and prints them. Source Code Incld.
GETVER.ZIP - GetVer example to call functions in VER.DLL.
HPJ.ZIP - Help Compiler Shell.
HYPER.ZIP - A simple way to add hypertext to your VB programs. Source code Incld.
ICONVI.ZIP - Simple icon viewer. Includes Visual Basic Source Code.
ICONX113.ZIP - Extracts icons from Windows EXE programs, and saves as .ICO file.
ILIST.ZIP - An enhanced listbox custom control features: multi-select and more.
INIMOD.ZIP - A VB module to create, read and update private ini files. Source Incld.
JEOPKE.ZIP - A game for VB. Includes Source Code.
LBSRCH.ZIP - Source code shows how to search a list box for closest match.
LB_FUN.ZIP - Source code shows some SendMessage API commands.
MINCLOCK.ZIP - Mini desktop clock written in Visual Basic. Source Code Included.
NCALC.ZIP - Shows Keyboard and Control handling with a calculator. Source Included.
NOCRUN.ZIP - Number Crunch v. 2.0 - A Number Game. Includes VB Source Code.
PLAYWA.ZIP - Shows how to play, replay, and stop the replay of WAV files in V.B.
PRCL11.ZIP - Click on a icon to print text in the clipboard. Source code included.
PRETTY.ZIP - A source code (txt) printing program for Visual Basic.
PROADD.ZIP - A quick way to add Microsoft Professional Toolkit custom controls to VB.
PROJPR.ZIP - Gives a formatted listing of your V. B. program in Word for Windows.
QDHELP.ZIP - Write help text, to be compiled with the Help Compiler (not included).
QPSUM.ZIP -   Program containing summary of features in QuickPak Professional.
REFRES.ZIP - Shows effect of using Refresh on a dynamically changing control. Source Incld.
RUNFIX.ZIP - Set "Minimize on Use" for each app in the Program Manager. Source Incld.
SHUFL1.ZIP - Designate multiple sound files for each system event. Source Incld.
SOUNDL.ZIP - A sound DLL for use with Visual BASIC. Includes sample app and source code.
SPIN.ZIP - A spin button for Visual Basic. Source Code Included.
STORC1.ZIP - Convert Visual Basic forms into standard Windows 3 resources.
STSBAR2.ZIP   A status bar .dll that can incorporate Status bars in your VB App.
SWITCH20.ZIP - Switch v. 2.0 - A Puzzle Game. Visual Basic Source Code included.
SYSSND.ZIP - Demo of how to access the Windows system sounds.
TAGENV.ZIP - Routines to provide support for tagged string fields in a VB Form.
TIMEFR.ZIP - Display Time in Caption of active Window. VB Source included.
VB2E.ZIP - How to recycle Declare statements for external Subs and Functions.
VBDOCERR.ZIP - List of documentation errors in the first printing of the VB manuals.
VBFAST.ZIP - Speeds up the loading of VB apps by pre-loading vbrun100.dll. Source Incld.
VBFONT30.ZIP - Displays available fonts and prints a list. Visual Basic Source Included.
VBMAC.ZIP - Macros for the Windows Recorder. To shortcut your typing when coding in VB.
VBME31.ZIP - Utility that displays Windows operating mode, and other system details.
VBPATH.ZIP - Gives you the path specs that are stored in the VB make files.
VBPT200S.ZIP - Manages Visual Basic 2.0 projects. SEE AD IN PRODUCT SHOWCASE SECTION!
VBSHEL.ZIP - VBShell can be loaded as the Windows shell. It then loads your VB application.
WHATSM.ZIP - Guess a random number set by the computer. Source code included.
WLZHCXP2.ZIP - A DLL for compression and decompression, copying, appending files.
WQH111.ZIP - Viewer for Windows Help files. Allows printing topics from help files.
XANTIPPE.ZIP - Hypertext authoring. Automates making of Windows help files.

PRICE $39.95 (Plus S/H)

ORDER FORM

VB DISK LIBRARY II ORDER FORM

NAME: _____

COMPANY NAME: _____

ADDRESS: _____

CITY: _____

STATE: _____

ZIP: _____

COUNTRY: _____

The library comes on 5-1/4 HD or 3-1/2 HD disks.

DISK PREFERENCE: 5-1/4 [   ]   3-1/2 [   ]

PRICE: $39.95 PLUS $5.00 Shipping ($10.00 Shipping Foreign)

California residents add 7-3/4 % sales tax.

Send check or money order drawn on a US Bank made payable to PRO-DATA and send to the following address:

PRO-DATA / EVB MAG 2

7151 EL CAJON BLVD. STE. L

SAN DIEGO, CA   92115

BUSINESS PLANS

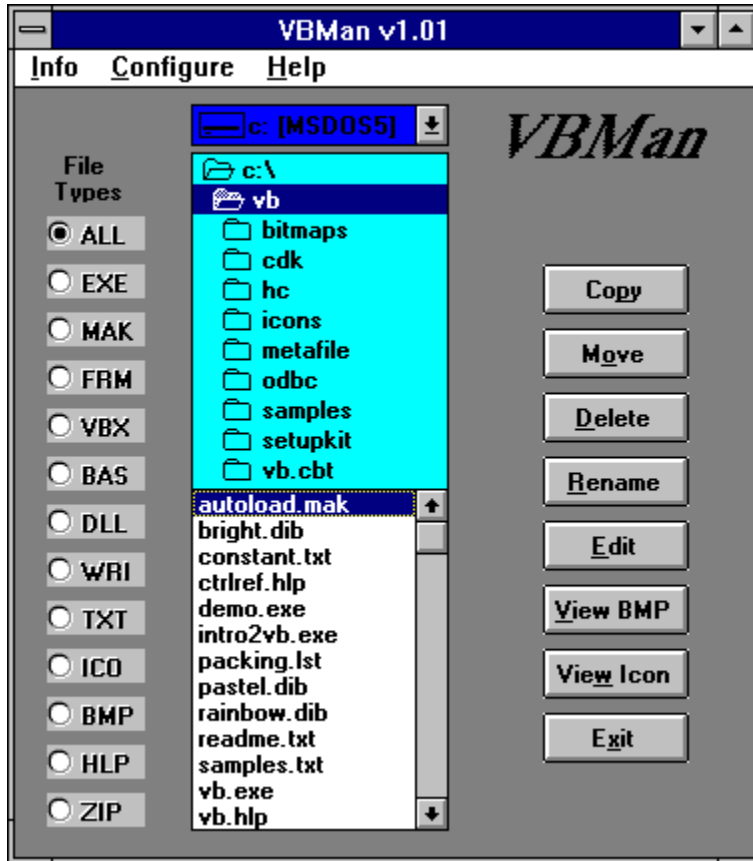### Starting or Expanding a Software Company?
You **NEED** a Business Plan!

Our BPA "mini business plan" is ready in just a few days.
Call now for information on our low prices.

We're the "Don't Do It Yourself" experts.

BUSINESS PLAN ASSOCIATES
Phone/Fax/Modem (714) 683-3914

VBMAN

An easy to use file manager for VB Programmers!

REGULAR PRICE **$29.95** (Plus S/H)

ON SALE NOW FOR ONLY $19.95 (Plus S/H)
EXPIRES 6/1/93

ORDER FORM

VBMAN ORDER FORM

NAME: _____

COMPANY NAME: _____

ADDRESS: _____

CITY: _____

STATE: _____

ZIP: _____

COUNTRY: _____

Any suggestions or improvements for the next version of VBMan? _____

_____

_____

_____

VBMan v1.01 comes on 5-1/4 HD or 3-1/2 HD disk.
DISK PREFERENCE: 5-1/4 [   ]   3-1/2 [   ]

PRICE: $19.95 PLUS $3.50 Shipping ($8.50 Shipping Foreign)
California residents add 7-3/4 % sales tax.

Send check or money order drawn on a US Bank made payable to PRO-DATA and send to the following address:

PRO-DATA / VBMAN
7151 EL CAJON BLVD. STE. L
SAN DIEGO, CA   92115

ADVERTISE IN EVB MAGAZINE

Do you want to reach Visual Basic Programmers?
Do you have a Visual Basic add-on for sale?
Do you do custom Visual Basic programming?
Have you written a Shareware Visual Basic program?

Then you need to advertise in Electrified Visual Basic Magazine!

For exceptional first time advertiser specials.
Call (619) 464-0767

New Vision

Information Systems

Office Automation

**Attn: Contractors &
Small Business Managers.**

If your desk looks like this,   give us a call?

We have a program that **YOU** can use **NOW!**

Print Professional looking Invoices, Work Orders, Purchase Orders etc. on your printer with your LOGO on plain paper. Have COMPLETE access to your data with **NO COMPUTER SKILLS**.
Accounts receivable system, Statements, Late Notices, Form Letters. Customer lists, Account Aging, Mailing Labels, Weekly, Monthly & Annual   Reporting and more...............Call for Details.

(619) 482-2613
CompuServe 76207,2434

ICON LIBRARY

Looking for an icon to use in your program. Currently 50 groups and approx. 1800 icons. Icons have been grouped into logical catagories so that a particular type of icon is much easier to find.

```
ANIMALS.EXE ....... Animals, Fish, Animal Cartoon Characters
ARROWS.EXE ........ Various styles of arrows
BOOKS.EXE ......... Books, Notepads, Bundles of Paper
BUTTONS.EXE ....... Icons on "gray" buttons
CARDFILE.EXE ...... Card index files and cards
CARDS.EXE ......... Playing cards(not complete) and suit symbols
CLOCKCAL.EXE ...... Calendars, Clocks, and Watches
COMMS.EXE ......... Communications, Modems, Port Connectors
CURSICON.EXE ...... Simple icons that can be used as cursors
DICE.EXE .......... Assorted dice and a set of six dice faces
DISKS1.EXE ........ Floppy disks
DISKS2.EXE ........ Hard disks, & virus tool icons
DOS.EXE ........... C: prompts, and "DOS" icons
EYES.EXE .......... Assorted "eye" icons
FILES.EXE ......... Filing Cabinets, Drawers, & Folders
FLAGS.EXE ......... Collection of national flags
GAMES.EXE ......... Icons for various Windows games
GRAPHS.EXE ........ Graphs and Charts
HANDS.EXE ......... Human hands pointing & holding
HOURGLAS.EXE ...... Sequence of hourglasses from full to empty
ICON10.EXE ........ Blank icon file with 10 slots (create your own)
ICON20.EXE ........ Blank icon file with 20 slots
ICON50.EXE ........ Blank icon file with 50 slots
KEYBOARD.EXE ...... Computer keyboards and keys
LOCKS.EXE ......... Padlocks and keys
MAIL.EXE .......... Mailboxes, Letters, Envelopes
MISC.EXE .......... I could not think of where else to put these
MOUSE.EXE ......... Computer Mice, Scanners, Joysticks
OBJECTS1.EXE ...... Everyday objects #1
OBJECTS2.EXE ...... Everyday objects #2
OFFICE.EXE ........ Office related (FILES, PHONES, BOOKS, etc.)
PEOPLE.EXE ........ Humans and Human Cartoon Characters
PHONES.EXE ........ Telephones and Telephone Poles
```

```
PLANES.EXE ........ Airplanes, Balloons, Helicopters, & Rockets
PLANTS.EXE ........ Flowers and Trees
PRINTING.EXE ...... Printers, Paper, Labels, Fonts
PRODUCT1.EXE ...... Commercial Icons #1
PRODUCT2.EXE ...... Commercial Icons #2
PRODUCT3.EXE ...... Commercial Icons #3
PRODUCT4.EXE ...... Commercial Icons #4
SCREENS1.EXE ...... Pictures of computer screens
SCREENS2.EXE ...... Stylized computer screens
SOFTWARE.EXE ...... Generic software icons
SOUNDS.EXE ........ Ears, Music, Instruments, & Loudspeakers
STARS.EXE ......... Astronomy, Binocular, Telescope
SYMBOLS1.EXE ...... Signs and Symbols #1
SYSMOLS2.EXE ...... Signs and Symbols #2
TRAFFIC.EXE ....... Traffic Signs, Traffic Lights
WORLD.EXE ......... Rotating globe sequence, and globes
WRITING.EXE ....... Pens, Pencils, and related materials
```

Also included are a couple of good Shareware utilities for working with Icons.

VB ICON BROWSER & EXTRACTOR
Use VB Icon Browser 2.00 to view, maintain, and print your collection of Windows icons. Copy, rename and delete icons. Print one to thousands of icons per page. VB Icon Extractor 1.11 extracts icons that are imbedded in Windows .EXE and .DLL files.
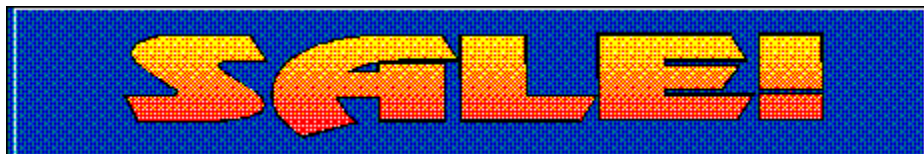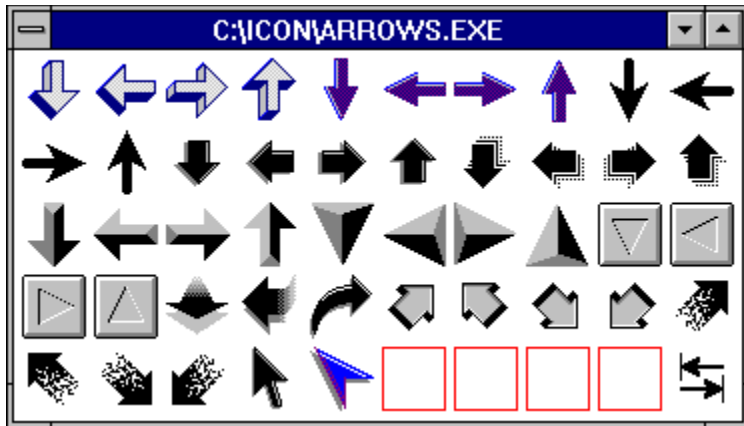
ICONMASTER 1.2C
An Icon editor and management program. Full edit with transparency. Multiple window display with drag and drop, screen capture, automatic installation of icons into Program Manager, handles all the popular icon formats.

Below are just a few of the approx. 1,800 categorized icons.

C:\ICON\SYMBOLS1.EXE

C:\ICON\SCREENS1.EXE

C:\ICON\OBJECTS1.EXE

C:\ICON\ARROWS.EXE

C:\ICON\HANDS.EXE

C:\ICON\HOURGLAS.EXE
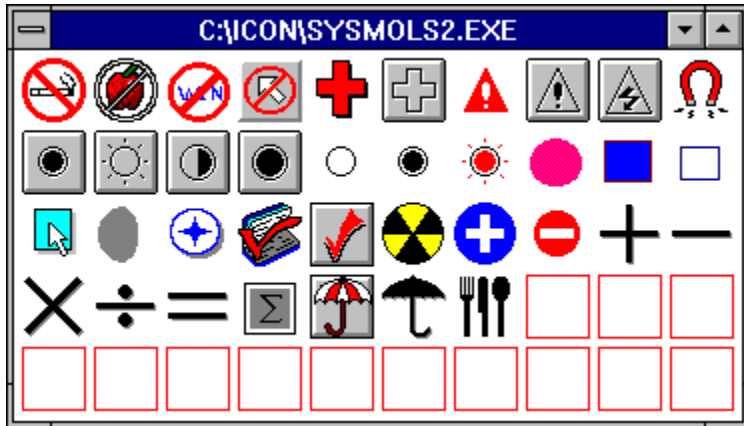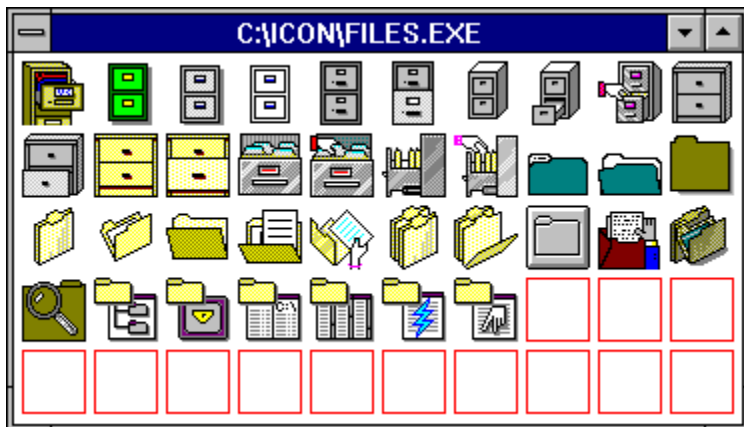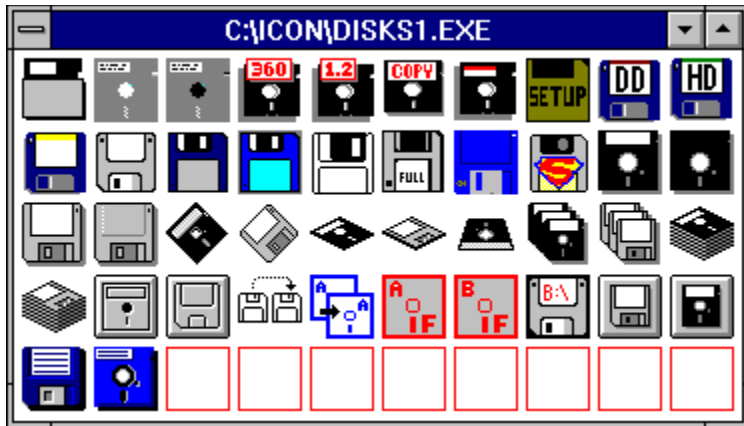
REGULAR PRICE **$29.95** **(Plus S/H)**

ON SALE NOW FOR ONLY $19.95 **(Plus S/H)**
EXPIRES 6/1/93

ORDER FORM

ICON LIBRARY ORDER FORM

NAME: _____

COMPANY NAME: _____

ADDRESS: _____

CITY: _____

STATE: _____

ZIP: _____

COUNTRY: _____

The library comes on 5-1/4 HD or 3-1/2 HD disks.
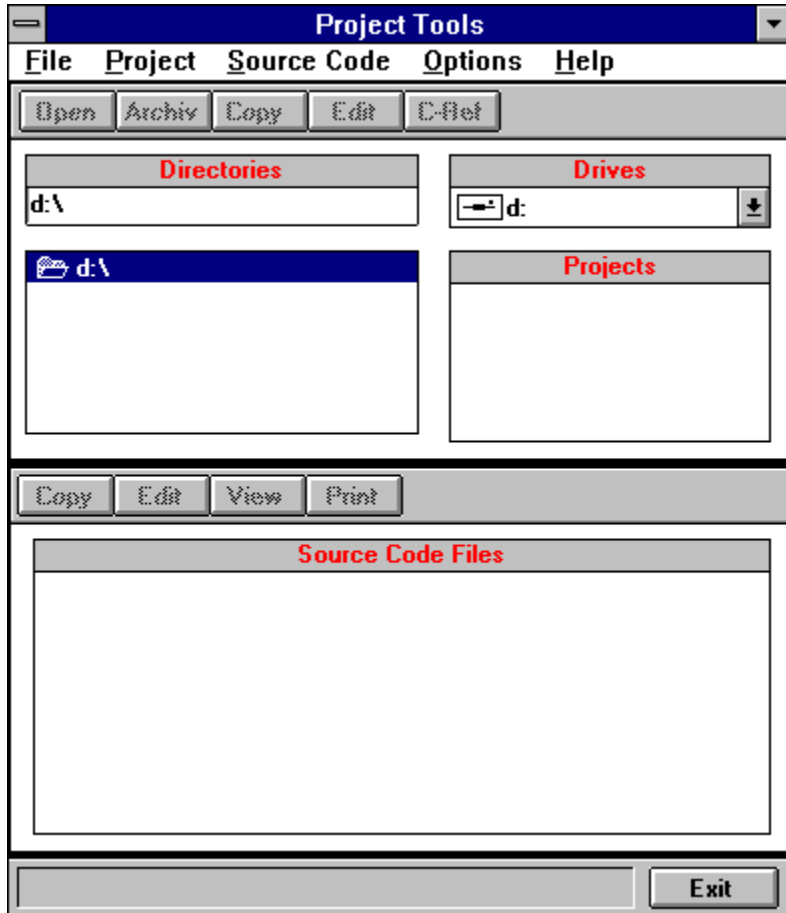DISK PREFERENCE: 5-1/4 [   ]   3-1/2 [   ]

PRICE: $19.95 PLUS $3.50 Shipping ($8.50 Shipping Foreign)
California residents add 7-3/4 % sales tax.

Send check or money order drawn on a US Bank made payable to PRO-DATA and send to the following address:

PRO-DATA / EVB MAG 2
7151 EL CAJON BLVD. STE. L
SAN DIEGO, CA   92115

VB PROJECT TOOLS

Projects Tools is an application designed as a front-end to Microsoft Visual Basic 2.0 for the purpose of managing Visual Basic projects (.MAK files). It is targeted for use mainly by the application developer but it can also be used by MIS departments that need to manage and control source code. Project Tools is completely written in Visual Basic 2.0 and makes occasional use of the Windows 3.1 API. Projects Tools features include the ability to:

Copy project files from one directory to another
Create a new project in its own directory
Archive a project
Launch VB with a particular project
Compile projects
Edit and Print project source code files
Produce a Cross Reference listing for source code files
Edit the project .MAK file

PROJECT TOOLS is shareware.   Users can use an unregistered copy of PROJECT TOOLS for sixty (60) days.   After that time users must either register their version of PROJECT TOOLS or

delete PROJECT TOOLS.

To obtain a registered version send your name, address and $20.00 + $3.50 shipping and handling to the following address.

Bob Dombroski
100 Goshen Road
Schwenksville, PA 19473-2202

Registered users receive one (1) year of free updates ($3.50 shipping and handling) also technical support via phone. Future upgrades will include a project database and a simple version control system.