



Contents

Introduction - Why VB Servants?

Starting VB Servants

Language Menu

Objects Menu

Dialogs Menu

Reference Menu

Launch Menu

How to contact us



Introduction-Why VB Servants?

VB Servants is designed to make the life of the Visual Basic programmer easy. Most people find it a chore to remember syntax details, but computers don't. The answer: automate the creation of the building blocks of your program!

VB Servants provides assistance on over 140 elements of syntax - 140 details that you don't have to remember. Now you can get on with the task of doing what you do well - designing & testing the flow of your application.



Starting VB Servants

- 1. Install VB Servants onto your hard drive by inserting the setup disk and running setup.exe.**
- 2. When you run VB Servants it will ask you if you want to run Visual Basic itself. If VB is not at the default path ("c:\vb\vb.exe"), VB Servants will ask you to locate vb.exe. The path of vb.exe is then saved in a private "ini" file in the VB Servants directory so that VB will load automatically next time.**



Language Menu

Arrays

Conversions

Date/Time

DDE

Environment

Error Handling

File Input/Output

File Management

Graphics

Math

Strings

Variables, constants & user-defined types



Objects Menu

Clipboard

Controls

Cursor

Forms

List & Combo boxes

Menus

Printer

Text boxes



Dialogs Menu

Message Box

Input Box

Open

Save As

Print

Color

Font



Reference Menu

Code Base

Constants

Picture Browser

Color Browser

Default properties/ naming conventions

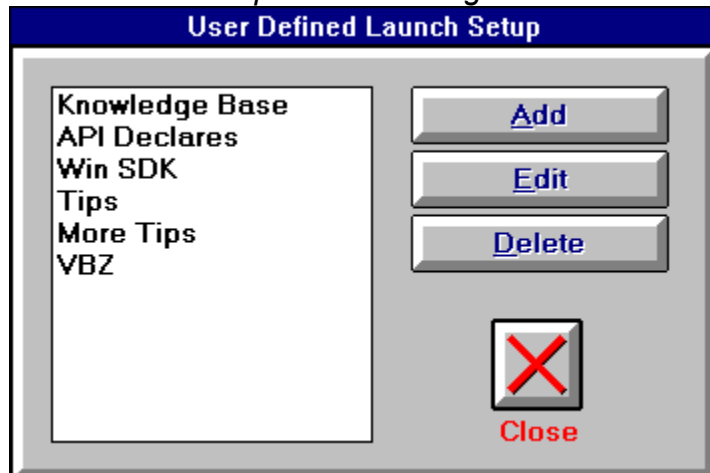


Launch Menu

Here you can launch commonly used applications & utilities. The path of VB is saved in a private "ini" file. The other preset applications are assumed to be in the Dos path.

User-define

Click on the hot spots of the image below for instructions.

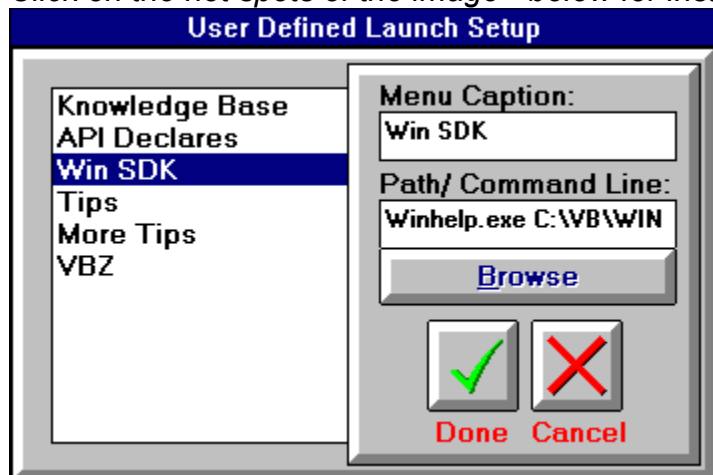


Click this button to add a new program to the user-defined launch menu. A dialog will appear that is identical to the "Edit" dialog.



Edit user-defined launch

Click on the hot spots of the image below for instructions.



Delete a user-defined launch

First select an application to delete from the list box. Then click on "Delete"



How to contact us

Send your comments, questions, suggestions, and constructive criticisms to Ian Barnes at:

Learning Freedom
10 Birdwood Ave,
Springwood 2777
NSW Australia
Phone: +61-47-54-3266
E Mail: ianbarns@zeta.org.au

For information on ordering see the included text files: "usorder.txt", "ausorder.txt" and "register.txt".

VB Servants is dedicated to the Suffering Servant.

[Menu Caption](#)

Edit this text box to change the text that appears on the Launch menu.

[Path/ Command Line](#)

Enter the full path or command line of the application you want to launch.

Browse

If you use "Browse" to find the application or file you want to launch, a default application name will be added before files with the following extensions: *.txt, *.bas, *.ini, & *.frm files will be preceded by notepad.exe; *.hlp files will be preceded by winhelp.exe. These defaults can be manually modified after you return from the "Browse" dialog box.

Message Box

Message: Simply type in your message. To add a new line press enter.

Title: Type in what you want to appear in the title bar of the message box.

Modality: Choose "Application modal" if you want the message box to suspend your application only. Choose "System modal" if you want it to suspend all applications.

Buttons: Select the combination of buttons that you want to appear on the message box.

Response variable: If you have more than one button, you will need to nominate the name of the variable that will receive the user's choice returned from the message box.

Default button: If you have more than one button, you can select which button will get the focus.

Icons: Select an icon appropriate for your message.

Beep: If you select "Yes" the message box will be preceded by a Windows system sound appropriate to the icon chosen. You will also have to paste a Windows API declaration into the declarations section of a form or code module.

Input Box

Title: Type in what you want to appear in the title bar of the input box.

Prompt: This is the question you ask or the directions you give to the user. To add a new line press enter.

Default answer: If you want to display a common answer to the question you are asking, type it here.

Answer variable: Nominate the name of the variable that will receive the user's answer returned from the input box.

Position: If you choose the default, the input box will be horizontally centered and vertically positioned approximately one third of the way down the screen. Otherwise enter the distance of the top-left corner of the input box from the left(x) and top(y) of the screen. The units of measurement are twips.

Open

Filters: Select the type of file you want the user to be able to open from the combo box and click on "Add". Multiple filters are allowed. Alternatively, type your own filter into the text box portion of the combo box using the same format as the pre-defined filters. "Clear" will delete all filters. "Filter index number" determines the default filter when the "Open" dialog is displayed. Enter the number of the filter that you want to be the default.

Dialog title: Choose the default("Open") or customize your own title.

Initial directory: The default is the current operating system directory. Customize this instead if you want the dialog box to open at a particular directory.

Initial file name: You can set the initial file name which will appear in the file name text box of the dialog box.

Dialog box name: The name of the common dialog control on your form.

Save As

Filters: Select the type of file you want the user to be able to save from the combo box and click on "Add". Multiple filters are allowed. Alternatively, type your own filter into the text box portion of the combo box using the same format as the pre-defined filters. "Clear" will delete all filters. "Filter index number" determines the default filter when the "Save As" dialog is displayed. Enter the number of the filter that you want to be the default. "Default extension number" determines the extension that will automatically be appended to the file name if the user saves a file without an extension.

Dialog title: Choose the default("Save As") or customize your own title.

Initial directory: The default is the current operating system directory. Customize this instead if you want the dialog box to open at a particular directory.

Initial file name: You can set the initial file name which will appear in the file name text box of the dialog box.

Dialog box name: The name of the common dialog control on your form.

Color

Control and property to change: For example: Form1.BackColor; Picture1.ForeColor.

Dialog box name: The name of the common dialog control on your form.

Print

Printer: "Display print setup dialog box" disables "Print range", "Print to file" and "Copies". You can also prevent a warning message from being displayed when there is no default printer. If you want to get the device context or an information context for the selected printer choose either option. The chosen context will be returned in the dialog box's hDC property.

Print range: You must enter the minimum and maximum allowed pages. If you set the "Pages" option button, you can set default "From page" and "To page" values.

Copies: The "Disable copies..." check box has the following function: if a printer does not support multiple copies, setting this flag will disable the copies text box; if a printer does support multiple copies, setting this flag indicates that the common dialog control should store the requested number of copies in its "Copies" property. The "Copies" text box allows you to set a default value for the number of copies to be printed.

Dialog box name: The name of the common dialog control on your form.

Font

Effects: Do you want to give the user the options of Underline, ~~Strikeout~~, and **Color**?

Limit size: Specify maximum and minimum font sizes.

No GDI font simulations: Printer fonts will not be simulated for screen display.

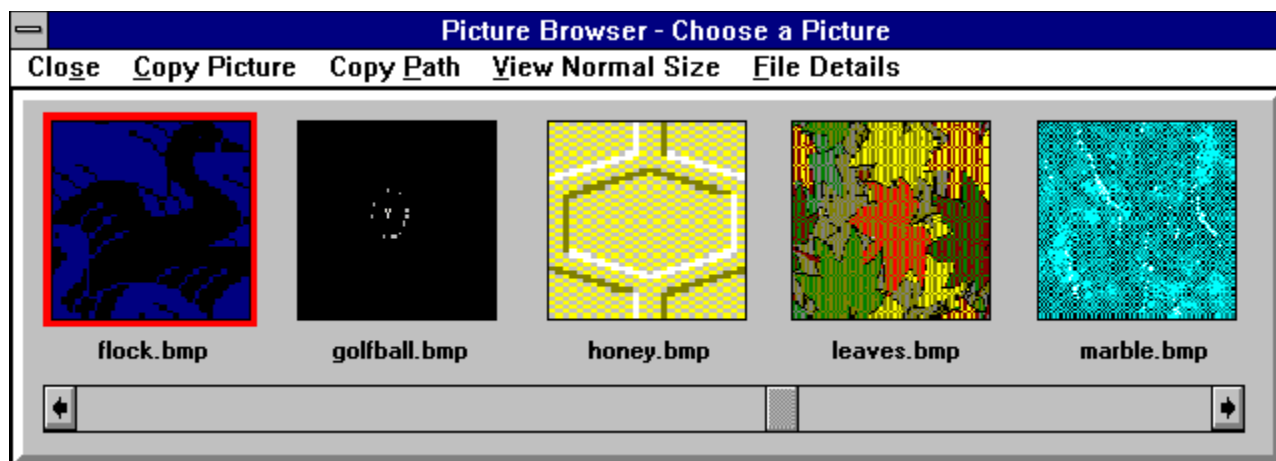
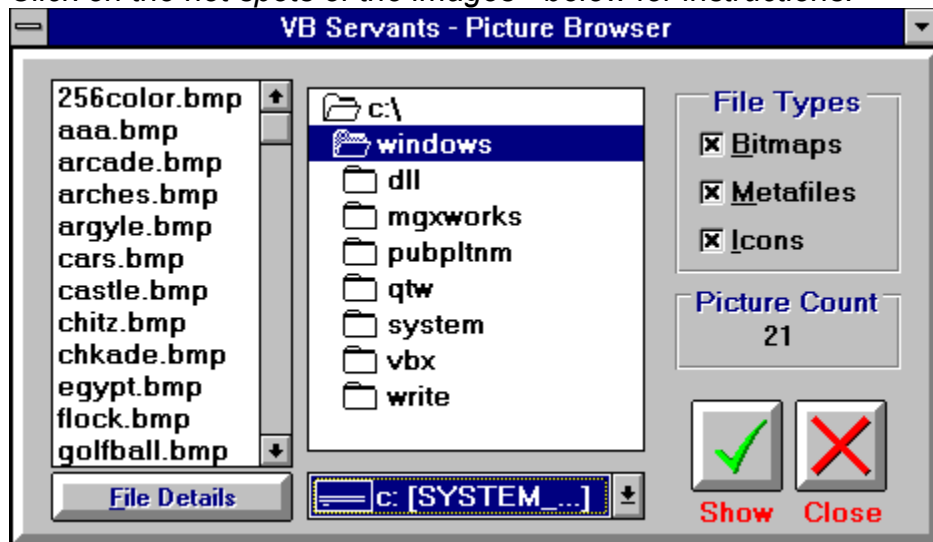
ANSI fonts only: Fonts such as Windings that show symbols will be excluded.

Control to change: For example: Text1; Label1.

Dialog box name: The name of the common dialog control on your form.

Picture Browser

Click on the hot spots of the images below for instructions.



Color Browser

This menu item opens a color dialog box. After you choose the desired color, the value of that color will be placed on the clipboard ready to paste. Color values can be used like so:

```
Text1.BackColor = 255    ' red was chosen
```


Constants

This menu item loads constants.txt (the text file that lists Visual Basic constants) into Notepad, and shows the "Find" dialog. When VB Servants is installed, constants.txt is placed in the VB Servants directory and is assumed to be there for the purposes of this menu item.

Default properties/naming conventions

The default property of a control: The default property of a control is the property you can refer to by simply referring to the control. For example: Text1.Text can be simplified to Text1; Picture1 is a shorter way of saying Picture1.Picture. Every control has one property that can be used this way. When the default property of a control is used in code rather than explicitly naming the property, your program executes faster.

Naming conventions: To make code more readable and to group objects together in the Object list box of the properties window, use naming conventions. That is, precede each control name with a prefix such as "txt" for text boxes or "pic" for picture boxes. For example: txtName; picHouse.

Choose the file types you want to search for and display.

Picture Count

This frame shows the number of picture files in the selected directory.

[Show](#)

Click here to view the picture files in the chosen directory.

Arrange MDI windows or icons

You can choose to either cascade or tile the child windows OR to arrange the icons.

Menu

Close: go back to picture browser's first window.

Copy picture: copy the picture highlighted with a red border to the clipboard.

Copy path: copy the path of the highlighted picture to the clipboard.

View normal size: view the highlighted picture at normal size.

File details: see the size and date/time of the highlighted picture.

Double-click on any picture to see it at normal size. To close the normally-sized picture, simply click on it.



Code Base

Menus

Fields

Finding a record

Search



Arrays

Change default lower limit

Declare an array

Erase an array

Find the upper or lower limits of a dynamic array

Reinitialize a dynamic array

Code Base Menus

File: Choose "Print" from this menu to print the current record.

Data: This menu gives alternative access to the "New", "Delete" and "Search" commands.

Text: From here you can load or save text into/from the code text box, change the font of the code text box, and change the background color of all fields.

Code Base Fields

Category: For example: Graphics; Text Boxes; Strings; or anything else you like!

Purpose: A more detailed definition of what this code fragment does.

Code: The code itself. You can import or export text from the "Text" menu by using "Load text" or "Save text".

Code Base Finding a Record

There are two ways to find a record: Use the drop-down list box **OR** use the "Search" facility.



Code Base Search

CB_Done

Click on the hot spots of the image below for instructions.

Search

Search Conditions

Text to search for:

☒ Category includes search text
☐ Category begins with search text
☐ Category must match search text exactly

Search By?

☒ Category
☐ Purpose
☐ Code

Done

Close

Show All Records

Code Base Search By

Specify if you want to search for something in the category field, the purpose field, or in the code field.

Code Base Search Conditions

Type the text that you are looking for. You can specify that the field you are searching (category, purpose, or code) includes the search text, begins with the search text, or matches the search text exactly.

[Code Base Show All](#)

Click here to access all the records in the database again.

[Code Base Done](#)

Click here to begin the search.

Change the default lower limit for arrays

The default lower limit (or bound) for array subscripts is, by default, 0. You can change this to 1 by choosing 1 from this window, clicking on "Done" and pasting the resulting code into the declarations section of a form or code module. You can only do this once for each form or code module. This will have no effect on arrays within user-defined types in VB3.

Declare an array

Fixed/dynamic: If you choose "Dynamic Array", the "Dimensions" and "Limits" frames will be disabled.

Scope: Specify if you want your array to be used throughout the whole application, throughout a form or code module, or within a procedure only.

Data type: You must choose a data type for your array. Fundamental data types and object variables are provided in the drop-down combo box.

Limits: If your array is a fixed-size array, you must specify the upper and lower limits (or bounds) for every dimension in the array.



Conversions

ANSI value to a string

Decimal number to hexadecimal or octal

Number to a string

One data type to another

Return the integer portion of a number

String to an ANSI value

String to a number

Reinitialize a dynamic array

You can use this window to declare procedure-level dynamic-array variables **OR** to size or resize a dynamic array that has already been declared at module or application level. If you are sizing or resizing an array that has already been formally declared, you cannot change the number of dimensions or the data type. Furthermore, when you choose to preserve the existing contents of an array, you can only change the upper limit of the last dimension in the array.

Find the upper or lower limits of a dynamic array

"Upper or lower limits" are otherwise known as "upper or lower bounds".

ANSI to string

Select an ANSI value from the list box, nominate the name of a variable, choose the return variable's data type, and click on "Done". For example: if you selected the ANSI value of 9 (Tab), you could then insert the return variable into a string expression to generate a tab.

Decimal to hexadecimal or octal

Enter the decimal numeric expression, specify a conversion to hexadecimal or octal, nominate a return variable to hold the result, opt for a string or variant result, and click on done.

Number to string

Enter the numeric expression, nominate a return variable, opt for a string or variant result, and click on done.

Convert one data type to another

Enter the expression you want to convert (the expression must be identified as either a "Numeric or string expression" OR a "String literal"), nominate a return variable, choose the data type to convert to, and click on "Done".

String literal

For example: ***"This is a string literal" & Chr(13) & Chr(10) & "This is the second line of my string literal."*** By way of contrast, this is not a string literal: ***Text1.Text & "Hello" & LastName.*** The second example includes a property and a variable, whereas the first example has neither.

Numeric expression

For example: **256894** OR $(i+j)*k$ where i,j and k are variables of numeric data types. Numeric data types include integer, single, long, double.

Return the integer portion of a number

Enter the numeric expression, the return variable name, and specify the option to take if the number is negative. If the number is positive, only the integer portion of the number will be returned. For example: **11.6** would be returned as **11**.

Return variable

Throughout VB servants, the return variable is the variable that will hold the result of the given function.

Convert a string character to its ANSI value

Enter the string to convert, identify it as either a "Variable or property" OR as a "String literal", nominate a return variable name, and click on "Done". Only the first character of the variable, property or string literal will be converted to an ANSI value.

String expression

For example, this is a string expression: ***Text1.Text & "Hello" & LastName.*** By way of contrast, this is not a string expression: ***This is a string literal" & Chr(13) & Chr(10) & "This is the second line of my string literal . "***: The first example includes a property and a variable, whereas the second example has neither.



Date/Time

Can an expression be converted to a date?

Convert a date/time serial to its components

Convert a string to a date or time serial number

Convert hour/minute/seconds to a time serial number

Convert year/month/day to a date serial number

Format a date/time expression

Get the current date and/or time

Time a process

Convert a string to a number

Enter the string to convert, identify it as either a "Variable or property" OR as a "String literal", nominate a return variable name, and click on "Done". The numeric value of the string will be returned. For example: "**Hello55**" would return **55**. A string expression can also be converted this way (treat it as a "Variable or property").

Date?

Use this function to see if a variant expression can be converted to a valid date.

Convert a serial to its components

The expression to convert can be a date and/or time expression. "Now", the default expression, gives the current system date and time.

Convert a string to a date or time serial

Enter the string you want to convert. The string can either be a string expression or a string literal.

Enter the name of a return variable, specify a conversion to either a "Date serial" OR a "Time serial", and click on "Done".

Convert hour/min/sec to a time serial

Use this function to get a time serial number from values for hours, minutes, and seconds. The valid ranges are: hours(0-23); minutes(0-59); seconds(0-59). Numeric expression. Return variable.

Convert yrs/mths/days to a date serial

Use this function to get a date serial number from values for year, month, and day. The valid ranges are: year(100-9999); month(1-12); day(1-31). Numeric expression. Return variable.

Format a date/time expression

Enter the date and/or time expression to format (e.g. "Now", which returns the current system date/time).
Nominate the format you would like to get by either clicking in the format list box OR entering the format name in the format text box. Enter a return variable name and then click on "Done".

Get the current date and/or time

You can opt to get the current date/time OR date OR time. Return variable.

Time a process

You will be automatically be given four lines of code. Place two of the lines (marked) at the start of the process and the other two at the end of the process. This code is designed for use within a procedure, To use it at module or application level, declare the variables in the declarations section of a form or code module respectively.



Close a DDE conversation

Get data from a source

Handle link errors

Open a DDE conversation

Send a command to a source

Send a picture to a destination

Send data to a source

Close a DDE conversation

You can only close a DDE conversation if your application is acting as the destination.

Get data from a source

In this case, your application will be acting as a DDE destination. The data can be received by a text box, label or picture box that has established a DDE conversation (see "[Open a DDE conversation](#)") in manual or notify link mode (not automatic mode).

Open a DDE conversation

First, nominate the name of the text box, label or picture box on your form that you want to receive data. To define the "Link topic" and "Link item", consult the documentation for the source application. The "Link mode" may be "Automatic" (the source will send data whenever the data changes), "Manual" (you have to make a request for the data; see ["Get data from a source"](#)), or "Notify" (you have to request the data but you will be notified by the source when the data has changed in your control's Link_Notify event).

Handle DDE errors

Here you will find handy code to handle link errors. Link errors are unlike normal run-time errors because they occur even when your code is not executing. The Link Error event is provided in VB to handle these errors. Specify whether you want to handle errors for a DDE destination (a text box, label or picture box) or DDE source (a form). Click on "Done" and paste the code into the relevant Link Error event procedure.

Send a command to a source

Applies when your control is acting as a DDE destination. Specify the control name and the command string. Valid command strings to send will be found by consulting the source application's documentation. String literal. Treat a string expression as a variable or property.

Send a picture to a destination

When your form is acting as a DDE source you will need to send picture data to any destination applications whenever your application updates a picture.

Send data to a source

When your application is acting as a DDE destination you can send data to the DDE source (from a text box, label, or picture box).



Environment

Activate another program

Find out if another program is running

Get command-line arguments

Get environment variables

Send keystrokes to the active window

Start another program

Yield execution to Windows

Activate another program

Simply supply the name that appears in the title bar of the window you want to activate. This command applies only to programs that are already running.

Find out if another program is running

This window introduces a function that is external to VB from the Windows API.. You only need to specify the class name OR the title bar caption of the application you want to test for.

Windows API

The Windows **A**pplications **P**rogramming **I**nterface. A collection of DLLs that provide extra functionality for Windows programmers.

Get command-line arguments

A command-line argument is the string the user places after your application's ExeName when running your application from Program Manager's or File Manager's File/Run menu.

For example: **Notepad.exe Readme.txt** would start Notepad with Readme.txt loaded. The return variable you nominate will get these command-line argument(s), if any.

Get environment variables

VB allows you to get certain information about the operating system including the Dos path, the Temp directory, the Windows directory, and the path of command.com.

Send keystrokes to the active window

Simply enter the keystrokes you want to send in the order you want to send them in the keystrokes text box. All normal (ASCII) characters can be entered directly. Special keys, action keys and Shift, Alt and Control should be selected from the list boxes. To specify that Shift, Alt or Control should be held down while other keys are pressed, enclose the other keys in parentheses. For example: **+(EC)** signifies Shift with E and C; **+({F4}))** signifies Shift, Alt and F4 together. By default, the keystrokes will be processed before returning control to your application.

Start another program

Specify the command string (e.g. ***c:\vblvb.exe***; OR ***notepad.exe readme.txt***), the window style, the return variable and click on "Done".

Yield execution to Windows

Use this command when you are executing a long loop for example, to give Windows a chance to process things like mouse-clicks and keystrokes .



Error Handling

Delay error handling

Get error information

"Multi-purpose" error handler

Resume execution after handling an error

Simulate an error

Turn off error handling

Turn on error handling

Delay error handling

Normally, when an error occurs it will interrupt the execution of your application. If you don't want this to happen, choose to delay error handling. The following code and comments will be automatically placed on the clipboard ready for you to past into a procedure:

***On Error Resume Next 'Delays error handling so that errors don't interrupt execution.
'With delayed error handling, check for errors by testing the value of Err.
'If Err <> 0, then an error has occurred.***

As the comments above suggest, after you delay error handling by using the ***On Error Resume Next*** statement, you can check if any errors have occurred by testing the value of ***Err***. The advantage of delayed error handling is that you can test the value of ***Err*** and handle errors at the point in the procedure that you choose.

Get error information

When an error occurs, you can get information about the error to display in a message box for example. This item will get the error number and the error message for you.



File Input/Output

Close a file

Close all files

Control multi-user access to an open file

Get the current position within an open file

Get the file mode or DOS handle of an open file

Get the next unused file number

Get the size of an open file

Load a text file into a text box

Open or create a file

Read from a file into a variable

Save the contents of a text box to file

Set the current position within an open file

Test for the end of a file during input

Write from a variable to a file

"Multi-purpose" error handler

To use this multi-purpose error handler you will first have to turn on error handling. This versatile error handler will handle just about any error you throw at it. If an error occurs it will bring up a message box that shows the error number and the error message and then gives the user the opportunity to **Abort** or **Retry** the procedure or to **Ignore** the error.

Turn on error handling

Normally a run-time error will halt the execution of your program. To remedy this you must either turn on error handling OR delay error handling. To turn on error handling, enter the name that you want to give your error handler, remembering that each error handler must have a unique name. The relevant code will be placed onto the clipboard. You will need to put part of the code generated at the start of your procedure and the rest at the end of the procedure.

Resume execution after handling an error

To use this item you will first have to turn on error handling or delay error handling. You have three choices when resuming execution after an error: (1) Resume at the line that caused the error; i.e. retry (2) Resume execution at the line following the line where the error occurred; i.e. ignore OR (3) Resume at a specific line. Alternatively you could choose to exit the procedure; i.e. abort. To do this use ***Exit Sub*** or ***Exit Function***.

Simulate an error

It is sometimes useful to simulate an error to test your error handling routines. Choose the category of error you would like to simulate, the error in particular, and click on "Done".

Turn off error handling

Once you have turned on error handling in a procedure you can turn it off again at any time, even within an error handling routine.

Close a file

Use this item to close a file that has been opened for I/O. See [Open or create a file](#).

Close all files

Use this item to close all files currently opened for I/O.

Open or create a file

This item will open a file for I/O OR if the file doesn't exist, create a file and open it for I/O. Specify the file's name and path (this can be a string literal OR a variable or property; treat a string expression as a variable or property). Nominate a file number; accept the default of **FreeFile** to use the next available file number. Select a file mode. If you select **Random** mode you will need to specify a **Record Length**. If you select one of the **Sequential** modes you will need to specify a **buffer size**. You can also choose to set multi-user options.

Record length

The record length you specify should match the size of the user-defined type (in bytes) that you are accessing.

Buffer size

The larger the buffer size the quicker the access time. On the other hand, more memory will be used. The maximum buffer size is 32,767 bytes.

Set multi-user options

To use multi-user options you must run **Share.exe**.

Sharing: The following options refer to the processes that are enabled while you have the file open for I/O and apply only if the file has not already been opened by another process: **Shared** (any process on any machine can read from or write to this file); **Lock Read** (no other process is granted read access); **Lock Write** (no other process is granted write access); **Lock Read Write** (no other process is granted read or write access). If you don't set multi-user options, other processes are denied access to the file while it is open.

Access: **Read/write** (opens the file for both reading and writing; this mode is valid only for Random and Binary files and files opened for Sequential Append); **Read** (opens the file for reading only); **Write** (opens the file for writing only).

Control multi-user access to an open file

Use this item to stop other processes from accessing the file while you have it open. Every file that you lock must be unlocked with an exactly matching statement before you close the file. Specify the file number and the range of records (for a file opened in Random mode) or bytes (for a file opened in Binary mode) you want to lock. For files opened in Sequential mode, the entire file will be locked or unlocked regardless of the range that you specify.

Clear data from a form when it is unloaded

Use this item to clear the contents of any form-level variables or arrays when the form unloads, otherwise they will retain their values. Paste the code into the Form_Unload event.

Get the current position within an open file

Use this item to find out where you are at in the specified file. For Binary and Sequential files the current position is given in bytes. For Random files the current record number will be returned.

Get the file mode or DOS handle of an open file

Use this item to find out whether you are in Random, Binary or Sequential mode OR to get the DOS handle of the file.

Get the next unused file number

Use this item to make sure the file number you use when you open or create a file is valid.

Change drive or directory

First specify whether you want to change drive or directory. Then enter the drive or directory name.

This can be a string literal OR a variable or property. Treat a string expression as a variable or property.

Get the size of an open file

Use this item to find the size of an open file in bytes.

Load a text file into a text box

Simply follow the four steps. Make sure that the text box has its **MultiLine** property set to **True** and its **ScrollBars** property set to **Both**.

Read from a file into a variable

Use this item to get information from an open file. **File number:** specify the number of the open file that you want to read from. **Variable:** Nominate the name of the variable and its data type that you want to put the data into. **Read Position:** Choose the next record (Random mode) or byte (Binary mode) OR specify a position.

Save the contents of a text box to a file

Simply follow the four steps outlined. If you wish, let the user choose the file/path to save to by using a Save As common dialog.

Set the current position within an open file

For files in Random mode the position is a record number. For other file modes the position is the byte position relative to the start of the file.



File Management

Change drive or directory

Copy a file

Delete a file

Get a file or directory's attributes

Get a file's date/time

Get a file's size

Get the current directory

Load or unload a picture

Make or remove a directory

Move a file

Rename a file or directory

Save a picture

Set a file's attributes

Test for the end of a file during input

Sometimes when you are reading from a file you might need to see if you have reached its end.

Write from a variable to a file

Use this item to write to an open file. **File number:** specify the number of the open file that you want to write to. **Variable name:** Nominate the name of the variable that you want to write from. **Write Position:** Choose the next record (Random mode) or byte (Binary mode) OR specify a position.

Copy a file

Enter the source file name/path and the name/path of the destination file. String literal. String expression.

Delete a file

Enter the name/path of the file you want to delete. This can be a string literal OR a variable or property.
Treat a string expression as a variable or property.

Get a file or directory's attributes

First select the attributes you want to look for. Enter the file or directory name, and nominate the name of the return variable.

Get a file's date/time

Enter the name/path of the file whose date/time you want to get. Nominate a return variable.

Get a file's size

Enter the name/path of the file whose size you want to get. Nominate a return variable. This item is for when the file is not open. String literal.

Get the current directory

Enter the name of the drive whose current directory you want to get. Nominate a return variable. You can have either a ***String*** OR ***Variant*** return variable.

Load or unload a picture

Load: Enter the name/path of the picture file you want to load. Then decide if you want to load the picture onto the clipboard or into a control. If you are loading the picture into a control, specify the control's name and, if necessary, its property. For example: ***Picture1***; OR ***Form1.Picture***.

Unload: Specify the control's name and, if necessary, its property.



Graphics

Clear run-time graphics

Color browser

Define an object's co-ordinate system

Force the repaint of a form or control

Load or unload a picture

Save a picture

Make or remove a directory

Enter the directory name/path and choose either ***Make a directory*** OR ***Remove a directory***.

Move a file

Enter the old file name/path and the new file name/path. These can either be string literals OR variables or properties. String expressions can also be used (treat as ***Variable or Property***).

Rename a file or directory

Enter the old file or directory name/path and the new file or directory name/path. These can either be string literals OR variables or properties. String expressions can also be used (treat as ***Variable or Property***).

Set a file's attributes

Enter the name/path of the file whose attributes you want to set. Then select the attributes to set. The **Normal** attribute cannot be set with any of the other attributes.

Save a picture

Enter the name/path for where you want to save the picture file. Specify the name of the control to save from and, if necessary, its property. For example: **Picture1**; OR **Form1.Picture** OR **Form1.Image**.

Erase an array

Enter the name of the array. For fixed arrays no memory will be recovered but the value of each element will be cleared. For dynamic arrays, memory will be freed.

Clear run-time graphics

This action will clear graphics created at run-time by graphics methods such as ***Line***, ***Circle*** and ***Print***.
Accept the default of Me if you want to clear the current form.

Define an object's co-ordinate system

Co-ordinates: By entering X and Y values for the upper-left and lower-right of the object you define a custom co-ordinate system. The scale mode of the object will be automatically set to ***UserDefined***.

Object: Enter an object name or accept the default of Me to act on the current form.

Force the repaint of a form or control

Accept the default of Me to force the repaint of the current form.

Me

A short way of referring to the current form.

Add items to a list or combo box

First enter the item to add and then click the Add button. To delete an item from the list, select it and click the Delete button. Finally, enter the name of the list or combo box you want to add the items to.



Convert decimal to hexadecimal or octal

Format a numeric expression

General calculations

Generate random numbers

Return the integer portion of a number

Trigonometry

Format a numeric expression

Enter the numeric expression you want to format, nominate a name for the return variable, then select a format. If you choose to define your own format, combine the available characters for the desired result.

General calculations

Enter a numeric expression and a name for the return variable. Select the operation you want to perform on the numeric expression.

Generate random numbers

Return variable.

Trigonometry

Enter a numeric expression and a name for the return variable. Select the operation you want to perform on the numeric expression and choose between radians or degrees.

The Suffering Servant

Behold, my Servant shall deal prudently,
He shall be exalted and extolled and be very high.
Just as many were appalled at you,
So his visage was marred more than any man,
And his form more than the sons of men;
So shall he sprinkle many nations.
Kings shall shut their mouths at him;
For what had not been told them they shall see,
And what they had not heard they shall consider.

Who has believed our report?
And to whom has the arm of the LORD been revealed?
For he shall grow up before him as a tender plant,
And as a root out of dry ground.
He has no form or comeliness;
And when we see him,
There is no beauty that we should desire him.
He is despised and rejected by men,
A man of sorrows and acquainted with grief.
And we hid, as it were, our faces from him;
He was despised, and we did not esteem him.

Surely he has borne our griefs
And carried our sorrows;
Yet we esteemed him stricken,
Smitten by God, and afflicted.
But he was wounded for our transgressions,
He was bruised for our iniquities;
The chastisement for our peace was upon him,
And by his stripes we are healed.
All we like sheep have gone astray;
We have turned, every one, to his own way;
And the LORD has laid on him the iniquity of us all.

He was oppressed and he was afflicted,
Yet he opened not his mouth.
He was taken from prison and from judgment,
And who will declare his generation?
For he was cut off from the land of the living;
For the transgressions of my people he was stricken.
And they made his grave with the wicked-
But with the rich at his death,
Because he had done no violence,

Nor was any deceit in his mouth.
Yet it pleased the LORD to bruise him;
He has put him to grief.
When you make his soul an offering for sin,
He shall see his seed, he shall prolong his days,
And the pleasure of the LORD shall prosper in his hand.
He shall see the travail of his soul, and be satisfied.
By his knowledge my righteous Servant shall justify many,
For he shall bear their iniquities.
Therefore I will divide him a portion with the great,
And he shall divide the spoil with the strong,
Because he poured out his soul unto death
And he was numbered with the transgressors,
And he bore the sin of many,
And made intercession for the transgressors.

From the prophet Isaiah, circa 700 B.C.



Strings

Convert an ANSI value to a string

Convert a string to an ANSI value

Convert to lowercase or uppercase

Create a string of repeating characters

Extract a portion of a string

Find the length of a string expression

Find the position of one string within another

Make string-comparisons case-insensitive

Remove leading and/or trailing spaces

Replace part of a string

Convert to lowercase or uppercase

Enter the string expression you want to convert, select ***lowercase*** or ***UPPERCASE*** and nominate the name of a return variable.

Create a string of repeating characters

Enter or choose a character, specify the number of repeats and nominate the name of a return variable.

Extract a portion of a string

Enter the string expression you want to extract characters from and nominate a name for the return variable. You can then extract characters from the left, the right, or the middle of the string expression

Find the length of a string expression

Enter the string expression you want to find the length of and nominate a name for the return variable.

Find the position of one string within another

Enter the string expression you want to find, the string expression to search and nominate a name for the return variable. If you want all the expression to be searched accept the default starting position of **1**.
Case-sensitive.

Case-sensitive

If an operation is **case-sensitive** then **SERvants** will not equate to **servants**.

Make string comparisons case-insensitive

You will automatically be given code to paste into the declarations section of a form or code module. By default, VB string comparisons are case-sensitive.

Remove leading and/or trailing spaces

Enter the string expression and nominate a name for the return variable.

Replace part of a string

Enter the string variable you want to modify and the string expression you want to insert. Nominate the character position in the string variable that you want to make the insertion. You can also limit the number of characters from the string expression to be inserted.



Variables, constants & types

Declare a constant

Declare a variable

Define a user-defined type

Find the length of a variable in bytes

Get information about a variant variable

Require explicit variable declaration

Scope

Scope indicates how widely a variable, constant or array will be used in your application. There are three levels of scope: **application level** (make it available to all procedures in every form and code module in the application), **module level** (use the variable, constant or array within one form or code module), and **procedure level** (it is only available to the procedure in which it is declared).

Declare a constant

Constant Name: This is almost entirely arbitrary; call it whatever you like. Use upper-case letters in order to make your constants easy to recognize in your program listings.

Constant Expression: This can be made up of numeric literals (such as **562.31**), other constants, or any of the arithmetic or logical operators except exponentiation (^). You can also use a string literal. You cannot use string concatenation, variables, user-defined functions, or intrinsic VB functions (such as *Chr*).

Declare a variable

Enter the variable name and the data type. Scope.

Define a user-defined type

A user-defined type is a type (or structure or record) that you construct from the fundamental data types and/or other user-defined types. First, nominate a name for your user-defined type. Next, nominate a name for the first element within your user-defined type and enter its data type. Then click on the **Add** button. Enter a name and data type for each additional element in the type. To delete an element, highlight it in the list box and choose **Delete**.



Clipboard

Clear the clipboard

Find out what's on the clipboard

Get something from the clipboard

Place something onto the clipboard

Find the length of a variable in bytes

Enter the variable you want to test and a return variable name.

[Get information about a variant variable](#)

Enter the variant variable you want to test and select the information you want to get.

Require explicit variable declaration

Use this item to force explicit variable declaration for a particular form or module. If you don't do this, all undeclared variables are of **Variant** data type (unless the default type is otherwise specified with a **Deftype** statement). Explicit variable declaration is to be advised to avoid mistyping the name of an existing variable and to avoid confusion in code in which the scope of the variable is not clear.

Clear the clipboard

Before you place something onto the clipboard it is usually good practice to clear it first.

Get something from the clipboard

Select the kind of data you want to get from the clipboard and then specify the variable OR control and property that will get the data.

Find out what's on the clipboard

Simply specify the format(s) you are looking for.

Place something onto the clipboard

Select the kind of data you want to place onto the clipboard and then specify the variable OR control and property that you will take the data from.



Controls

Iterate through all controls on a form

Manually drag a control

Move a control

Repaint or update a control or object

Send to the front or back

Set the focus to a control

Iterate through all controls on a form

Accept the default of Me if you want to work with the current form.

Manually drag a control

To manually drag a control you will need to set its Drag Mode property to 0 (i.e. manual).

Move a control or form

Only a new value for **Left** is compulsory. However, if you specify a new **Width** then **Top** is no longer optional. Likewise, if you specify a new **Height**, **Width** is no longer optional.

Repaint or update a control or object

This item is for when you may need to, for instance, manually force the repaint of a form or update the display of a list box.

Send a form or control to the front or back

Using this method on a form will send it to the front or back of all other forms in your project. When applied to a control, it will move the control to the front or back of the layer that the control belongs to.

Normally, controls are in one of two layers: **Front** (for non-graphical controls like command buttons, check boxes, and file controls AND containers such as picture boxes and frames that may contain other objects) and **middle** (for graphical controls such as the line and shape controls AND labels).

Furthermore, behind all the controls on a form is the **back** layer which consists of the drawing space of the form where the results of graphical methods appear

Set the focus to a control

Set the focus to a control to guide the user to the next action. Hint: setting the focus to another control in the Got Focus event is a handy way of disabling a control.

Hide, show, load or unload a form

Accept the default of Me to act on the current form.

Hide: the form will be made invisible but be kept in memory.

Show: select this option to both load a form into memory and to make it visible. If you check the modal check box, all other forms in your application will become unavailable to the user until the modal form is closed (useful for dialog boxes).

Load: the form will be loaded into memory but not made visible.

Unload: make the form invisible and clear it from memory.



Forms

Arrange MDI windows or icons

Clear data on Form_Unload

Hide, show, load or unload a form

Iterate through all loaded forms

Make a form always on top

Move a form

Position a form on the screen

Send to the front or back

Set the focus to a form



Cursor

Confine the cursor

Move the cursor to a specific control

Move the cursor to a specific X,Y point

Confine the cursor

The cursor may only be confined to a form or to a control with the hWnd property. Be sure to unconfine the cursor before your application closes. Windows API.

Move the cursor to a specific control

The cursor may only be moved to a control with the hWnd property . Windows API.

Move the cursor to a specific X,Y point

This item may be used to move the cursor to any position on a form. Windows API.



Printer

Print a form 1

Print a form 2

Print a form 1

Accept the default of Me if you want to print an image of the current form. This method will print at the resolution of the screen.

Print a form 2

Accept the default of Me if you want to print the current form. This method has the advantage that it will print at the resolution of the printer. The module PRNTFRM.BAS which this item uses supports the following controls **only**:

- Text boxes (multi-line or single line)

- Labels (single line only)

- Command buttons (not 3D)

- Shape controls & line controls.

- Shapes and lines are printed behind the other controls.

Place all controls directly onto the form. Controls in containers are not supported.

Be sure to use true-type fonts unless you want unpredictable results.



Menus

Cut, copy and paste from an edit menu

Popups

Iterate through all loaded forms

This item is useful when you want to act on all loaded forms with the same method; e.g hide all forms.

Cut, copy and paste from an edit menu

Follow the three easy steps to create an edit menu that allows the user to cut, copy and paste. The ready-made code will enable and disable the cut, copy and paste sub-menus depending on the clipboard format and the type of control that has the focus.

Popup menus

You can make an existing menu popup on a right mouse click using this item. The menu to popup must have at least one sub-menu. **Alignment** refers to the position of the cursor over the popup. **Menu behavior** refers to how the popup responds to the mouse once it is visible. **Position** refers to the position at which the popup will appear. To test the popup properties, right click on the background panel.

Make a form always on top

This code is designed to be used with a menu item named **mnuTop**. First create the menu item and then copy the code into it. Windows API

Position a form on the screen

It is usually best to paste the code from this item into the Load event of the form.



List and Combo boxes

Add items

Allow numbers only (combo box only)

Emulate overstrike mode (combo box only)

Horizontal scroll bar (list box only)

Make all uppercase (combo box only)

Remove items

Set the focus to a form

Set the focus to a form to guide the user to the next action.

Remove items from a list or combo box

You can choose to remove all items in the list or combo box, only the item currently selected by the user or the item with the index number that you specify.

Allow numbers only to be entered in a text or combo box

This item prevents the user from entering anything but numbers in a text or combo box. A minus sign will be allowed if it is at the beginning of the string, and one decimal point will also be permitted.

Emulate overstrike mode in a text or combo box

Paste the code into the Key Press event procedure of the nominated control.

Add a horizontal scroll bar to a list box

Use this item when list box items are too large for the list box. API.

Force all characters in a text or combo box to appear uppercase

Paste this code into the Key Press event procedure of any text or combo box.



Text boxes

Allow numbers only

Emulate overstrike mode

Make all uppercase

Read only

Undo

Make a read-only text box

This item makes a read-only text box without graying the text or background but still enables copying to the clipboard. Windows API.

Undo text box changes

This item allows the user to undo the most recent change to a text box. This would normally operate from a menu item named mnuUndo with a shortcut of CTRL-Z. (mnuUndo would normally be under the Edit menu). Windows API.

