() "" '' [ ] { } < >

A statement that assigns a value to a variable, array, data field, or control.

Example: intResult = intValue1 + intValue2
Example: txtFirstName.Text = "Rush"

Bracket Example

In this example, we want to use the RTrim$() function on all text to the right side of the '=' sign. The following text block has been hilighted in a VB code window, or appears in the Handyman Edit Window:

```
txtFirstNam.Text = strFName
txtLastName.Text = strLName
txtAddress1.Text = strAddr1
txtAddress2.Text = strAddr2
txtCityName.Text = strCity
txtState_Cd.Text = strState
txtZip_Code.Text = strZip
txtPhoneNum.Text = strPhone
```

Bring up the Bracket Window and select the option button for Parentheses (the default). Enter "rtrim$" (without the quotes) in the Prefix Text Box. Select the option button for Right Side. Click on "Go". The text now looks like this:

```
txtFirstNam.Text = RTrim$(strFName)
txtLastName.Text = RTrim$(strLName)
txtAddress1.Text = RTrim$(strAddr1)
txtAddress2.Text = RTrim$(strAddr2)
txtCityName.Text = RTrim$(strCity)
txtState_Cd.Text = RTrim$(strState)
txtZip_Code.Text = RTrim$(strZip)
txtPhoneNum.Text = RTrim$(strPhone)
```

Contents

Overview

Handyman is a utility for working in the Microsoft Visual Basic development environment. Its purpose is to save keystrokes by automating many of the tasks that VB programmers perform while writing code.

Most of Handyman's functions operate on Selected Text - sections or lines of VB statements in .FRM or .BAS modules that have been hilighted by the user, usually with the mouse (highly recommended for working in Handyman).

Summary of Features

Comment / Uncomment

Similar to VB's Indent feature (Alt-Tab), this will allow you to comment or un-comment several lines of hilighted text with a click of the mouse.

Swap Sides

'Sides' refers to. This function will swap the text on the left side of the '=' sign in an assignment statement with the text on the right side. This is handy in situations such as assigning variables to screen objects, then assigning the values of the screen objects back to the variables.

Example

Bracket

This is a separate dialog box that lets you enclose selected text or sides of an assignment statement in Bracket Characters with an optional prefix. For example, you can hilight several statements that assign string variables, and wrap each statement on the right side in a function such as RTrim$().

Example

Access-SQL to VB Text String

If you use the Microsoft Access query builder to create queries and copy the SQL statement into your VB code modules, you know how much fun it is to manually parse that text into individual strings and concatenate them together. Handyman does this for you!

Example

Sub Grabber

This is a dialog box that lets you select a .BAS, .FRM, or ASCII text file from disk, locate the name of a specific Sub or Function, then load that Sub or Function onto the Clipboard or into Handyman's Edit Window. Note: the .BAS and .FRM files must be saved as text.

Function Stripper

Each mouse click removes the outermost function from selected text, or from the Left, Right, or Both sides of a block of assignment statements. This is often used before performing Handyman's "Swap Sides" function, to strip the functions from the right side of an assigment statement before swapping.

Example

Assign

The functions in this dialog box are useful for initializing groups of variables. It will take the text from the Left or Right side of an assignment statement and assign an empty

string, a number, a text string, an array, or several other possibilities.
Example

Print Procedure/Print Selected Text

A "quick and dirty" way to send your hilighted text (or entire procedure) to the current Windows printer. There is no formatting or printer setup.

Edit Window

A "VB-and-Handyman-aware notepad", Edit Window lets you perform Handyman functions on a block of text separate from your VB code window. When you have it the way you want it, just paste it back into VB. Edit Window has most of the same functions as the Main Screen, and some additional ones including retrieving/pasting selected VB text, entire VB procedures, or clipboard text, indenting text, and extracting a column of text. Most Handyman functions operate on the entire contents of the Edit Window.

Clip List

This lets you store several lines or sections of code in a simple VB list box, and paste them back into VB as needed. Very useful in situations where you find yourself typing the same line or lines into several modules in a large project.

See also:
Getting Started

Swap Sides - Examples

The following lines of text would be hilighted in a VB code window, or appear in the Handyman Edit Window. Bring up a Handyman menu and select 'Swap Sides'; the text will be changed as shown below.

Before:

```
txtFirstNam.Text = strFName
txtLastName.Text = strLName
txtAddress1.Text = strAddr1
txtAddress2.Text = strAddr2
txtCityName.Text = strCity
txtState_Cd.Text = strState
txtZip_Code.Text = strZip
txtPhoneNum.Text = strPhone
```

After:

```
strFName = txtFirstNam.Text
strLName = txtLastName.Text
strAddr1 = txtAddress1.Text
strAddr2 = txtAddress2.Text
strCity  = txtCityName.Text
strState = txtState_Cd.Text
strZip   = txtZip_Code.Text
strPhone = txtPhoneNum.Text
```

Note: The spacing of some of the fields in this example has been adjusted slightly for clarity of illustration.

Swapping part of a line - Before:

After:

If strValue1 = strSaveValue then Exit Sub

Function Stripper Example

The following lines of text would be hilighted in a VB code window, or appear in the Handyman Edit Window. Bring up a Handyman menu and select 'Strip Functions (Right Side)''; the text will be changed as shown below.

Before:

```
txtFirstNam.Text = RTrim$(strFName)
txtLastName.Text = RTrim$(strLName)
txtAddress1.Text = RTrim$(strAddr1)
txtAddress2.Text = RTrim$(strAddr2)
txtCityName.Text = RTrim$(strCity)
txtState_Cd.Text = RTrim$(strState)
txtZip_Code.Text = RTrim$(strZip)
txtPhoneNum.Text = RTrim$(strPhone)
```

After:

```
txtFirstNam.Text = strFName
txtLastName.Text = strLName
txtAddress1.Text = strAddr1
txtAddress2.Text = strAddr2
txtCityName.Text = strCity
txtState_Cd.Text = strState
txtZip_Code.Text = strZip
txtPhoneNum.Text = strPhone
```

Assign Examples

The following lines of text would be hilighted in a VB code window, or appear in the Handyman Edit Window. These examples all use the left side of an assignment statement as their source.

Before:

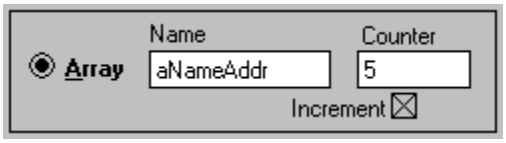
```
strFName = txtFirstNam.Text
strLName = txtLastName.Text
strPhone = txtPhoneNum.Text
```

After:

strFName = ""
strLName = ""
strPhone = ""

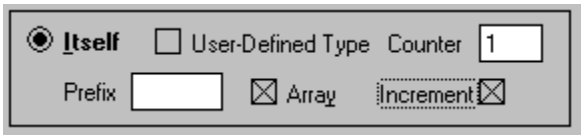
strFName = 0
strLName = 1
strPhone = 2

strFName = "default"
strLName = "default"
strPhone = "default"

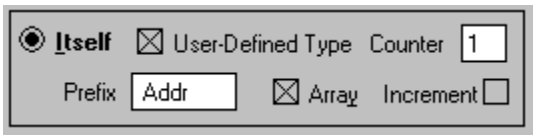


strFName = aNameAddr(5)
strLName = aNameAddr(6)
strPhone = aNameAddr(7)
Note: Counter does not have to be numeric.

strFName = ss!strFName
strLName = ss!strLName
strPhone = ss!strPhone



strFName = strFName(1)
strLName = strLName(2)
strPhone = strPhone(3)



strFName = Addr(1).strFName
strLName = Addr(1).strLName
strPhone = Addr(1).strPhone

Getting Started

Handyman can save you a lot of typing, time, and aggravation, and is easy to use. The latter, of course, is a claim that all programmers make about their products!

However, it will be worth your while to take a half-hour or so, to do two things:

1. Read this section.
2. Practice on a scrap project.

Suggestions for learning Handyman

It would be a good idea work with a copy of the source code from one of your programs or one of the VB sample programs. Practice on the copy until you are confident with how Handyman works; then, if you want, delete the copy.

Basic Points:

- **Handyman is a text formatting tool.**
  Most functions do NOT check syntax. The author is not responsible for any errors that may occur as a result of using this program. Always check the results of a Handyman operation to make sure your code has been formatted the way that you intended.

- **Handyman operates only on Visual Basic code windows of .frm and .bas files.**
  It operates on the last code window to have the focus, and that code window must not be minimized.

- **Most Handyman functions operate on Selected Text within the code window.**
  The user selects (hilights) some text and selects a Handyman function. Internally, the program makes a copy of the text and applies the requested formatting. It then replaces the selected text with the modified version.

  Other functions (such as Edit Window or Clip List operations) insert text into your code; it is helpful to get into the habit of setting your insertion point before executing a Handyman function, so that when you set up and execute the function, you know that the results will be pasted into the expected location.

- **You can use Control-Z (or 'Undo' from VB's Edit Menu) to undo the result of a Handyman operation, just as if you had typed the changes in manually.**

The Handyman Main Window

This is a floating toolbar which contains the most often-used Handyman functions. Here is how to comment a block of text:

-

Bring up a code window, preferably one with a lot of text in it.

- Hilight several lines of text (do not hilight the Sub or Function header).
- Click on the Handyman Comment Button.
  Each line of your hilighted text has been made into a comment (prefixed with the single quote character). Note: if a line was already a comment, it now begins with two single-quote characters. This is intentional; it lets you see at a glance that these lines were already commented.
- Press Control-Z (or select menu options Edit/Undo) to undo your comments, just to see how you can recover when Handyman does something you didn't really want.

Now let's try the Uncomment function...

- Find some text that has been commented, or use Handyman to comment several lines.
- Click on the Uncomment Button.
  The single quotes have been removed. Note: if a line begins with more than one single quote character, only the first one will be removed.

Additional Functions: the Handman Main Menu

Click on the button marked "Mnu" to bring up the Main Menu. This provides several more functions; since you now have a feel for how the program works, please continue with the help topic "Main Screen Buttons & Menus".

Main Screen Buttons & Menus

Main Screen Buttons

Comment a block of text

Un-Comment a block of text

Show Main Menu

Help - also available by pressing F1 when in a Handyman form or on a menu item.

Main Screen Menu Functions

Click on the Menu Button to bring up the Main Screen Menu. Items with numbers to the left can be invoked by pressing that number. Items followed by an elipsis (...) will bring up a dialog box.

0 Edit Window...

Brings up a "Handyman-aware" notepad screen for working with small sections of text.

1 Bracket ...

Brings up a dialog box to enclose text in parentheses and other bracket characters; useful for wrapping several lines of text in functions such as "Ltrim$( )".

2 Assign ...

Brings up a dialog box to create assignment statements.

3 Swap Sides

Exchanges text on right and left sides of one or more assignment statements.

4 Strip Functions (Selected Text)

5 Strip Functions (Right Side)

6 Strip Functions (Left Side)

The above three features remove the outermost function from one or more lines of text.

7 Clone Selected Text

Creates a commented copy of the selected lines.

8 Make SQL String

Converts a copy of an Access SQL statement into a text string for creating recordsets in Visual Basic.

9 Comment Procedure Body

Comments all statements in a procedure except for the first and last lines (Function / End Function, Sub / End Sub, etc.)

Clip List ...

Brings up a dialog box for storing blocks of text in a list for fast pasting into VB code windows.

Sub Grabber ...

Brings up a diaog box similar to a "file" common dialog, for locating a Sub or Function and copying it to the clipboard or to the Edit Window.

Print Procedure

Copies current procedure to the printer, without formatting.

Print Selected Text

Sends the current selection to the printer, without formatting.

Always On Top

If checked, the Handyman main window will float on top of all other windows on the screen.

About

Displays information about Handyman. Instructions for registering are also available from this screen.

Exit Handyman

Exit the program. As there is no control box on the form (to save space), this is the only way to exit the program. Note: if your copy of Handyman is unregistered, you will get a lovely yellow* screen reminding you to register and providing access to registration instructions. This will remain on-screen for 10 seconds; click on "Continue" to exit or better still, click on "Registration Info"!

* color monitors only <g>

Comment Button (Edit Window)
Cmnt Button (Main Screen)

This button will convert one or more lines of text into Comments, by placing a single quotation mark at the beginning of each line.

In the Edit Window, if any lines are selected, only those lines will be commented; otherwise, every line in the window will be commented.

Instructions - Main Screen

- In a VB code window, hilight text to be commented
- Click on the 'Cmnt' Button on the Handyman Main Screen
- The text will be commented and you will be returned to the VB code window.
- Press Control-Z (or menu options Edit/Undo) to undo the effects of this operation.

Instructions - Edit Window

- To comment specific lines of text, select the text, then click on the 'Comment' button.
- To comment all text in the Edit Window, make sure no text is hilighted, then click on the 'Comment' button.
- The text will be commented. Click on the 'Undo' button to undo the effects of this operation.

Note: Handyman does not try to figure out if your selected text is already commented - it merely adds a single quote to the beginning of a line, regardless of what the line contains.

Uncomment Button (Edit Window)
Uncomnt Button (Main Screen)

These buttons produce the opposite effect of Comment/Cmnt: Any lines that begin with a single quote will be un-commented.

In the Edit Window, if any lines are selected, only those lines will be un-commented; otherwise, every line in the window will be un-commented.

For instructions, please see instructions for Commenting, above.

Note: Handyman looks at the first non-space character of each line and removes it if it is a single quotation character. It ignores the rest of the line.

Clone Selected Text
(Main Screen Only)

This replaces a block of hilighted text with:

- A commented version (duplicate) of the selected text, followed by...
- An un-commented copy of the selected text.

This function can be used to make a "backup" of a section of code that you need to modify. If your modifications don't work out, delete the unwanted lines, then uncomment the original. If the modifications are OK,delete the original section.

Instructions:

- Hilight a block of text in a VB code window.
- Select "Clone Selected Text" from the Main Menu..
- Your text will appear as described above, and the cursor will be in the VB code window below the last line that you selected.
- Press Control-Z (or select VB menu options Edit/Undo) to undo the effects of this operation.

Example:
Before:

```
txtFirstNam.Text = RTrim$(strFName)
txtLastName.Text = RTrim$(strLName)
txtAddress1.Text = RTrim$(strAddr1)
txtAddress2.Text = RTrim$(strAddr2)
txtCityName.Text = RTrim$(strCity)
txtState_Cd.Text = RTrim$(strState)
txtZip_Code.Text = RTrim$(strZip)
txtPhoneNum.Text = RTrim$(strPhone)
```

After:

```
'txtFirstNam.Text = RTrim$(strFName)
'txtLastName.Text = RTrim$(strLName)
'txtAddress1.Text = RTrim$(strAddr1)
'txtAddress2.Text = RTrim$(strAddr2)
'txtCityName.Text = RTrim$(strCity)
'txtState_Cd.Text = RTrim$(strState)
'txtZip_Code.Text = RTrim$(strZip)
'txtPhoneNum.Text = RTrim$(strPhone)

txtFirstNam.Text = RTrim$(strFName)
txtLastName.Text = RTrim$(strLName)
txtAddress1.Text = RTrim$(strAddr1)
txtAddress2.Text = RTrim$(strAddr2)
txtCityName.Text = RTrim$(strCity)
txtState_Cd.Text = RTrim$(strState)
```

```
txtZip_Code.Text = RTrim$(strZip)
txtPhoneNum.Text = RTrim$(strPhone)
```

Handyman Edit Window

Overview

Edit Window is like a "VB and Handyman-aware notepad". It was designed to be used like this:

- Copy in a block of text from a VB code window.
- Perform one or more Handyman functions and/or minor editing operations on the entire block.
- Paste the modified block back into the VB code window.

The Edit Window can be resized or maximized. If closed or minimized, it is hidden, and will retain its contents, size, and location the next time it is called from the Main Menu.

Bringing up the Edit Window

- Click on the button marked "Mnu" on the Handyman Main Screen to bring up the Main Menu. Select the first option, "0 Edit Window ..."

Displaying the Edit Window Menu

- Click on the Menu Button to bring up the Edit Window Menu

What can you can do in the Edit Window:

- Most of the Main Menu Functions, including Bracket, Assign, Swap Sides, Function Stripper, Comment, and Uncomment.
- Indent or Unindent (hold down the left mouse button for continuous indent or unindent)
- Extract a column of text (hilight a section of any line and click on the "Col" Button).
- Print the contents of the window (output is unformatted)
- Merge several blocks of text from VB or the Clipboard.
- Perform several Handyman functions on all of the text in the Edit Window before pasting it back into VB or copying to the Clipboard.

Loading code or text into the Edit Window

There are several ways to do this, which are summarized below. For more detail, please see Help Topic "Edit Window Buttons and Menus".

Importing Code from VB:

- Select a block of code in a VB code window.
- Bring up the Edit Window, then click on the Menu Button.
-

Select the first item, "Get Selected VB Text". Your VB Code will copied into the Edit Window.

Appending Code to the Edit Window:

- Select a block of code and bring up the Edit Window and Edit Window menu as shown above.
- Select the second item, "Append Selected VB Text"; this will add the selected block of code to any text that is already in the Edit Window.

Importing an entire VB Sub or Function:

- Make sure the Sub or Function you want to import was the last VB window to have the focus. It is not necessary to select any text.
- Select "Get Entire Procedure" from the Edit Window menu. The entire procedure, including the Sub or Function Header and the End Sub or End Function statement, will be copied into the Edit Window.

Hint: this is useful for copying a procedure. After bringing the sub or function into the Edit Window, change the sub or function name, then click on the "Put VB" button. Two copies of your procedure now exist in the same module, one with the old name, another with the new. You may need to change the names of statements such as those that assign function return values to avoid "duplicate definition" errors.

Importing from the Clipboard:

- Text already on the clipboard can be pasted into the Edit Window using Control-V, or selecting "Get From Clipboard" from the Edit Window Menu.

Pasting the Edit Window contents back into VB:

- Make sure your insertion point (location of the cursor) is where you want the Edit Window text to be pasted. As with any cut & paste operation, if lines are hilighted in the code window, they will be replaced by the new text from the Edit Window.
- Click on Edit Window button "Put VB"

For more details on the Edit Window, please see topics "Edit Window Buttons" and "Edit Window Menus"

Edit Window Buttons
Edit Window Menus

Note: If you bring up one of the dialog boxes, such as Bracket, after hilighting text in the Edit Window, the text will no longer appear hilighted; however, functions that perform on selected text will work correctly.

The Clip List

Overview

Clip List is a listbox where you can temporarily store several lines or sections of code for pasting into a VB code window.

This can save a lot of typing in situations such as large projects where you find yourself having to place the same line or lines of code into many procedures or modules.

Getting text into the Clip List

- Hilight one or more lines in a VB code window and click on the "Get-VB" Button; your selection will be loaded into the next available position in the list box OR ...
- Place some text onto the Clipboard, then click on the "Get-Clip" button; your selection will be loaded into the next available position in the list box.

Pasting Clip List Items into VB

- Set your insertion point in a VB code window, or hilight some text to be overwritten.
- Double-click on the line in the list box that contains the code that you want pasted, or click on the "Put-VB" button; your code will be pasted into the VB code window.

Copying Clip List Items to the Clipboard

- The process is similar to Pasting Clip List Items into VB, except that you click on the "Put-Clip" button, and the text is placed onto the Clipboard instead of pasted into a VB code window.

Other Clip List Buttons

- DELETE - Select an item in the Clip List and click on Delete to remove it from the list. You can also press the Delete key to remove the selected item.
- CLOSE - Closes the Clip List window

Clip List Checkboxes

- On Top - causes Clip List to float
- Close After Paste - hides the Clip List form

Swap Sides x=y -> y=x

This function will swap the text on the left and right sides of an assignment statement. This is useful when you have to assign several variables to other variables, then re-assign them later on.

Example

Instructions:

- In a VB code window or in the Handyman Edit Window, hilight one or more lines of text (or part of a line) containing assignment statements.
- Select "Swap Sides x=y -> y=x" from the Main or Edit Window menu.
- The left and right sides of the hilighed assignment statements (or selected text in a single line) will be reversed.

Please Note:

- Handyman will obediently swap the left and right sides of any lines or sections of text that contain an "=" sign, whether swapping sides makes any sense or not. It does not check for syntax errors.
- You can include lines that are not assignment statements, they will just be ignored.
- If you are working in the Edit Window and do not have any text selected, Handyman will perform the operation on every assignment statement in the window.
- The text "x=y -> y=x" in the title of this function is included just to provide a visual reminder as to what the function does.

The Bracket Dialog Box

This function will bracket (enclose) text in your choice of bracket characters, with an optional text prefix. It works on a selected text string, on the left or right sides of an assignment statement, or on complete lines of text. It is useful for tasks such as:

- Enclosing a long string of text in a VB or user-defined function.
- Enclosing the right side of several assignment statements in a function.

Example

To bracket a text string in a VB code window:

- Hilight the text string or lines that you want bracketed.
- Click on the 'Mnu' button to bring up the Main Menu.
- Select the radio button for the type of Bracket Character that is required
- Enter a Prefix (optional - usually used when creating a call to a Function)
- Select the radio button that describes the text you want bracketed: Left or Right Side of an assignment statement, Entire Line, or Selected Text.
- Click on the 'Go' button.

    The text that you selected in the VB code window will be bracketed.

To bracket text in the Edit Window:

- Hilight the text that you want bracketed. If no text is selected, Handyman will bracket the left or right sides or entire lines in the Edit Window, depending on which option button you select.
- Fill in the dialog box as described above.

    The text that you selected in the Edit Window (or all lines in the window) will be bracketed.

- If you need to undo any individual Bracket operation in the Edit Window, click on the "Undo" button.

Additional Options

- 'Prefix Only' Radio Button: Select this option button to insert a prefix without bracket characters.
- 'On Top' checkbox: If checked, the Bracket Dialog Box will float over other windows on the screen.
- 'Close After Paste' checkbox: If checked, the Bracket Dialog Box will be hidden after the 'Go' button is clicked. Uncheck this box for the dialog box to remain on the screen.

Make SQL String

This function is intended for programmers who:

- Develop database applications.
- Use a query-building tool such as the one provided with Microsoft Access to create SQL statements.
- Copy these SQL statements from Access and paste them into their VB code.

Unfortunately, the programmer must then convert the raw SQL statement, which may consist of several lines of text, into a string that can be used to open a recordset or perform an action query:

Set dynAddress = dbCustomers.CreateDynaset (SQL)

Here's where Handyman comes in:

- Create a query in Access or other application.
- Copy the SQL text to the Clipboard.
- Bring up a VB code window and set your insertion point.
- Select "Make SQL String" from the Handyman main menu.

Handyman converts the SQL text into one or more strings assigned to a variable named "SQL" and pastes it into your VB code window. (Note: If you are using Option Explicit in your module you will need to include the statement "Dim SQL as String").

Using 'Make SQL String' in the Handyman Edit Window

- Create a query in Access or other application.
- Copy the SQL text to the Clipboard.
- Bring up the Edit Window.
- Select "Get From Clipboard" from the Edit Window Menu, or press Control-V. This will paste the SQL text into the Edit Window.
- Select "Make SQL String" from the Edit Window menu. Handyman converts all of the text in the window to one or more strings assigned to a variable named "SQL".
- Set the insertion point in a VB code window and select "Paste to VB" from the Edit Window menu.

Note: At this time, Handyman does not attempt to parse a long string into shorter ones to make it readable; you may need to do a lot of horizontal scrolling to view a complete line.

Getting the SQL text from Access - a brief tutorial

This is intended for those who havn't tried this technique before; unless you are a real SQL

guru, you will probably find that it saves you a lot of the time and aggravation of creating SQL statements manually.

- In Microsoft Access, open the database you want to work with.
- Select the "Query" tab.
- Select the table or tables you want to use in the query.
- Create your query by joining tables, choosing field names, and selecting options. (See the Access documentation for detailed instructions).
- If desired, run the query to make sure it does what it is supposed to do.
- From the toolbar just beneath the menus, select the button that says "SQL". You will see the SQL statement that Access created based on your selections in the design window.
- Hilight the text and press Control-C to copy it to the Clipboard.

You can now convert the query text using "Make SQL String" as shown above, and paste it into your application.

Sub Grabber

This function will let you load a VB Sub, Function, Declarations Section, or small text file onto the Clipboard or into the Handyman Edit Window. It is intended to let you work with files that are not a part of the current project, and for cutting and pasting sections of code or entire procedures.

Using a form similar to the "File/Open" common dialog, you can specify a .bas or .frm file and get a list of all of the Subs or Functions found in that file. Clicking on the "Get Proc" button will cause the Sub or Function to be read in and sent to the selected destination, the Clipboard or the Edit Window.

Note: When reading a Sub or Function from a VB .frm or .bas file, that file must be saved as text.

Instructions:

Before starting, it is helpful to know (or at least have a good idea) which Sub or Function you want to load, what file it is in, and the directory on which the file resides.

- From the Handyman main screen, right click to bring up the Main Menu. From the Edit Window, right click in the text area to bring up the Edit Window menu.
- Select "Sub Grabber...". The Sub Grabber dialog box appears.
- Select a drive letter from the "Drives" combo box.
- Select the path from the "Directories" combo box.
- Select the file type from the "List Files of Type" combo box. The options are "Code (.bas)", "Form (.frm)", and "All files (*.*)". As stated above, .bas and .frm files must be in text format; if reading in a file of a different type, that file must be an ASCII text file.
- Select a file from the File List, or type the name into the "File Name" text box. If you type the name in, please press Tab so Handyman knows it should proceed to the next step.
- Select the "Subs" or "Functions" option button in the Source frame. Based on this setting, the "Procedures" combo box will contain the default [declarations], plus the names of all Subs or Functions found in the selected file.
- Select the Sub or Function that you want to load.
- Select the Clipboard or Edit Window option button in the "Destination" frame.
- Click on the "Get Proc" button. If your destination is the Clipboard, you will get a message informing you of the success of the operation. If your destination is the Edit Window, the Edit Window appears with the selected procedure loaded.

The "Set Default Path" Button

This will set the current path to the one you have selected in the dialog box. Each time you bring up Sub Grabber in the current session, or until you change the default path on the selected drive, the same directory will be current. This is useful if you intend to work with several procedures from the same directory.

Function Stripper

Each time it is invoked, Function Stripper will remove the outermost function from a block of text. It is easier and faster than deleting the function name, then locating and deleting the matching parentheses. It works on:

- A selected string of text.
- All strings on the right side of a block of assignment statements.
- All strings on the left side of a block of assignment statements.

Example

Instructions:

- From the Main Menu or Handyman Menu, select the approprate menu item:

    Strip Functions (selected text)
    Strip Functions (right side)
    Strip Functions (left side)

The outermost function of each line in your selection will be removed. To remove additional functions in a VB code window, re-select the text and repeat the process.

Hint:

To strip several nested functions in assignment statements, you may find it easier to work in the Edit Window - this will let you repeatedly strip the functions from the right or left sides without having to re-select the text.

Caution:

Handyman considers a function as any matching pair of parentheses, preceeded by a word. You could accidentally remove pieces of code such as references to arrays:

```
    before:   gArray(iIndex)
after :   iIndex
```

Please be careful when stripping functions!

Text1.Text = ss!FirstName
Text2.Text = ss!LastName

Select 'Left' to perform an operation on text to the left of the "=" sign, as shown by Text1.Text and Text2.Text in the example above.

Select 'Right' to perform an operation on text to the right of the "=" sign, as shown by ss!FirstName and ss!LastName.

The operation will be performed on full lines of text regardless of whether or not the lines contain an "=" sign. Usually used in conjunction with the "Columns" function in the Edit Window.

The operation will be performed only on the hilighted portion of one line. Warning - if more than one line of text is hilighted, the function will not succeed.

The 'Assign' Dialog Box

Overview

This purpose of this function is to build assignment statements from existing ones, or from a column of text in the Edit Window. A few quick examples of what this function can do:

Before:	After #1	After #2
strValue1 = "String Value 1"	strValue1 = ""	strValue1 = aType(1).strValue1
strValue2 = "String Value 2"	strValue2 = ""	strValue2 = aType(1).strValue2
strValue3 = "String Value 3"	strValue3 = ""	strValue3 = aType(1).strValue3

(Please click on "Examples" below to view examples of each type that can be created)

Examples

Using 'Assign'

- Hilight some text in a VB code window or in the Edit Window. If using the Edit Window, all lines will be formatted if no specific text is hilighted.
- From the Main Menu or Edit Window Menu, select "Assign..."
- Select the option button of the text that will be placed to the left of the "=" sign: Left or Right Side, or Entire Line.
- Select the option button that describes the text to be placed on the right side of the "=" sign, and fill in any text fields or check boxes that are applicable. (The examples show each type).
- **For a preview of what your statement will look like, hold down the right mouse button anywhere on the form; a sample using your current settings will appear in the lower-left corner. This is also a good way to become familiar with the different options available.**
- Click on the "Go" button; the formatted lines will be pasted into the VB code window or Edit Window, depending on the menu from which you selected 'Assign...' .

Additional Options

"On Top" checkbox - if checked, the Assign dialog box will float above other windows on the screen.
"Close After Paste" checkbox - uncheck this to have the Assign dialog box stay up after a paste operation.

Edit Window Buttons

Note: Most of these functions operate on the entire contents of the Edit Window, if no text is selected. If text is selected, the functions will operate only on that text. Exceptions are noted.

Comment

Un-Comment

Indent. Single click to indent 1 space, or hold down left mouse button for continuous indent. Does not operate on selected text.

Un-Indent. Single click to unindent 1 space, or hold down left mouse button for continuous un-indent. Does not operate on selected text.

Extract a column of text.

Example

If enabled, this button will undo the last operation to be performed in the Edit Window. (Does not operate on text pasted to a VB code window).

Brings up the Edit Window Menu

Edit Window Menus

Pastes the entire contents of the Edit Window (or selected text, if any lines are selected) into a VB Code Window.

Closes the Edit Window. The contents of the text box are retained.

Column ('Col' Button)

This button is only available on the Edit Menu. Its purpose is to extract a column of text of a specified width from each line in the Edit Window. It is usually used with the "Assign" function, as shown in the example below.

Instructions:

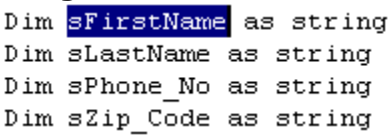
- Hilight a section of text from any line in the Edit Window.
- Click on the "Col" button.
- All text in the Edit Window is discarded except for the column that was extracted.

Example:

- **Bring up the Edit Window and copy in the following:**

```
Dim sFirstName as string
Dim sLastName as string
Dim sPhone_No as string
Dim sZip_Code as string
```

- **Hilight a column as shown:**



- **Click on the "Col" button; the Edit Window now contains only the following:**

```
sFirstName
sLastName
sPhone_No
sZip_Code
```

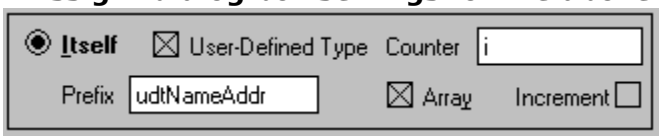
- **A practical use for this function is to use the "Assign..." dialog box to assign some text to each line. (Make sure to select the 'Entire Line' option button). This example assigns an empty string to each line:**

```
sFirstName = ""
sLastName  = ""
sPhone_No  = ""
sZip_Code  = ""
```

- **This more complicated example uses the "Itself" option to assign elements to an array of user-defined types. The dialog box settings are shown below:**

```
sFirstName = udtNameAddr(i).sFirstName
sLastName  = udtNameAddr(i).sLastName
sPhone_No  = udtNameAddr(i).sPhone_No
sZip_Code  = udtNameAddr(i).sZip_Code
```

"Assign" dialog box settings for the above example:



Edit Window Menu Functions

Click on the Menu Button to bring up the Main Screen Menu. Items with numbers to the left can be invoked by pressing that number. Items followed by an elipsis (...) will bring up a dialog box.

Get Selected VB Text

Loads selected text from a VB code window. Overwrites any text already in the Edit Window.

Append Selected VB Text

Loads selected text from a VB code window. Text is pasted into the Edit Window following any existing text.

Get Entire Procedure

Loads entire Sub or Function from a VB code window. Note: This is useful for copying a procedure: Load the procedure, change the name, then click on the "Put VB" button. The new procedure is added to the current VB module.

Get From Clipboard

Copies contents of the clipboard into the Edit Window. Overwrites any text already in the Edit Window.

Append From Clipboard

Copies contents of the clipboard into the Edit Window.. Text is pasted into the Edit Window following any existing text.

Copy to Clipboard

Copies the contents of the Edit Window to the clipboard.

1 Bracket ...

Brings up a dialog box to enclose text in parentheses and other bracket characters; useful for wrapping several lines of text in functions such as "Ltrim$( )".

2 Assign ...

Brings up a dialog box to create assignment statements.

3 Swap Sides

Exchanges text on right and left sides of one or more assignment statements.

4 Strip Functions (Selected Text)

5 Strip Functions (Right Side)

6 Strip Functions (Left Side)

The above three features remove the outermost function from one or more lines of text.

7 (not used)

8 Make SQL String

Converts a copy of an Access SQL statement into a text string for creating recordsets in Visual Basic.

Sub Grabber ...

Brings up a diaog box similar to a "file" common dialog, for locating a Sub or Function and copying it to the clipboard or to the Edit Window.

Clear Window

Clears the Edit Window.

Print Window Contents

Sends the contents of the Edit Window to the current Windows printer. Output is not formatted.

Registration Info

The registration fee for this version of Handyman is $20.00. Please send check (drawn on U.S. banks only) or money order to:

Scott Whittlesey
96 Faneuil St.
Windsor, CT 06095-4522

Purchaser will be sent the latest version of Handyman on diskette or via E-Mail according to their preference. Please refer to README.TXT for more details.

Attn: Compuserve users... Handyman 2.0 can be registered online. GO SWREG and refer to Registration ID# 4986.

Tech Support

Tech support is available from the author via E-Mail only.

Compuserve: 74344,421

America On-Line: vbHandyman

Limitations

Handyman is written entirely in Visual Basic and is subject to the same limitations as any VB program, particularly with regards to the Edit Window text box, Clip List listbox, and the Windows clipboard.