Global Variables

The following variables MUST be set as global at the beginning of the Coach.

ScrnMode Retains a value to resize screens according to text screen mode. Must also include the following statement at the beginning of the Coach:

```
ScrnMode=?ScreenHigh-24
```

MsgAdj Used to adjust posted messages properly. Must also include the following statement at the beginning of the Coach:

```
MsgAdj=DlgMessage()
```

DefTitle Used as a Main Title for dialog boxes globally.

HotElement Used for Edit Windows in Dialog Boxes. You must set a value to this variable before using WinEdControl. This tells the Coach which, if any, of the hotspots define the current edit window. If the edit window is not set as a hotspot the value is set to 0. If the edit window is the third defined hotspot, the value is 3.

HotElement is useful to stop accelerator characters from being typed into a window when you click on it.

SaveScrn Allows the Coach to determine when to restore screens. You must set this variable to 0 before a WHILE (?Dialog=###).

Variables that Should Be Set

<u>At top of Coach</u>

```
ScrnMode=?ScreenHigh-24
MsgAdj=DlgMessage()
DefTitle="Name of Coach"
CancelMsg=0
```

<u>Within Coach</u>

HotElement=# (when using WinEdControl; See WinEdControl below)
SaveScrn=0 (When using a While to control a Dialog)
Others any variables needed to be passed as parameters to a PROCEDURE or FUNCTION (see documentation below)

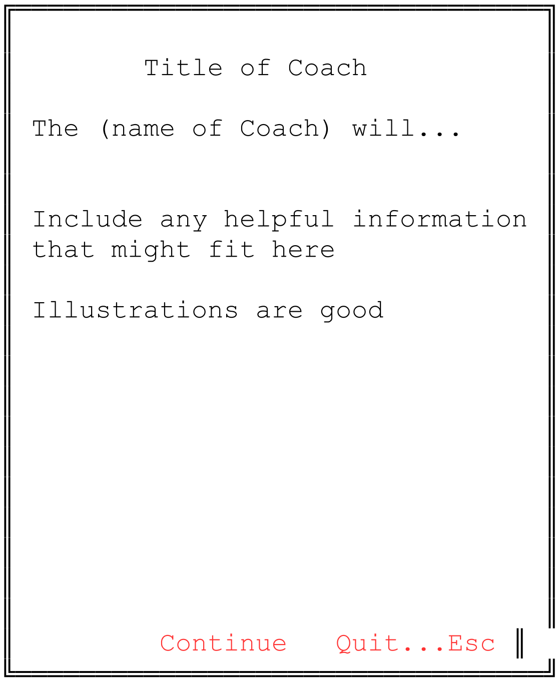
Standard Coach Header Sample

```
//*********************************************
//
//        NAME:           Name of Coach
//        PURPOSE:        Purpose of Coach
//        AUTHOR:         Author of Coach
//
//*********************************************

// Revision History
//
//

Error(Off!)
Global(ScrnMode;MsgAdj;DefTitle;CancelMsg)
Global(FakeBtns;HotElement;SaveScrn)

DefTitle="Main Title of Coach"                  // Main title name
Title="Coach"
ScrnMode=?ScreenHigh-24
CancelMsg=0

Use("wpchlibr.wpm")                             // Shared library file
ColorScheme()                                   //Sets Coach Colors
ScreenSetup()                                   //Sets button bar, ribbon, etc
MsgAdj=DlgMessage()
//*******************************************************************
```

Coach Style Specifications

Welcome Screen

Welcome Screens will have a Continue button and a Quit Button. It may also have a Hint button as needed. Use the Welcome(Title;&tbox) Procedure to create the Welcome Screen. In some cases, customized Welcome screens may need to be built to accomodate a better user interface.



```
      Title of Coach

The (name of Coach) will...


Include any helpful information
that might fit here

Illustrations are good
```

Helpful text on this screen will be discretionary, depending on the function of the Coach.

Instruction Box

The Instruction Box should be positioned and sized so that it is never covered up when the appropriate pull-down menus are accessed. The Instruction box should be titled, simply, Coach.

There should be one blank line at the top before the text begins, and one blank line at the bottom.

A small statement should appear at the top to remind users what they are doing.
Example: To mark the selected text:

Any instructions should be prefaced with the ASCII 16 character (▸).

The names of menu options should be in Bold. No brackets should be placed around the words.

One ▸ should precede the first menu choice, and another should precede the keystroke equivalent.

The instruction box should remain on the screen until the desired Dialog Box has appeared.

```
╔════════════Coach═════════════╗
║ To mark a phrase for TOC,    ║
║                              ║
║  ▸ Choose Tools,             ║
║    then Table of Contents,   ║
║    then Mark.                ║
║      or                      ║
║  ▸ Press Alt+F5              ║
║                              ║
║      Show Me   Quit...Esc ║  ║
╚══════════════════════════════╝
```

This instruction box is created with a Postit command, with buttons set to 2.

When the Show Me button is used, the Instruction Box should remain on the screen. The pulldown menu should not cover the Instruction Box. When the last item has been reached, a smaller, untitled dialog should appear near that item, and instruct the user to choose that item. (HBOX3)
Example: Choose *Mark*.

A message should pop up if the user is choosing the wrong items that prompts the user to try again, then reiterates the steps they need to perform. If they are not on the right main menu, a message appears that tells them which menu item to choose. (HBOX1)
Example: Choose *Tools*

Once they are in the right main menu, but are having trouble finding the next item, a second box will pop up with further instructions. (hbox2)
Example: Choose *Table of Contents*

The Fakeit Instruction Box

The ▸ character should precede any instructions. No bullets should precede informational text.

There should be Continue...F7 and Quit...Esc buttons. Optionally, when needed include a Hint or Show Me button.

Wording Conventions

Use **Choose** to instruct users to choose an item from a menu.
Example: ► Choose *Tools*

Use **Choose** the *** Button when instructing the user to choose a specific button in a dialog box.
Example: ► Choose *Continue*

Use Select when instructing the user to select an item from a list or popup box.
Example: Select the labels definition that you need.

Use Press when giving instruction for the user to press a key on the keyboard.
Example: ► Press *Alt+F7*

When referring to function keys, use a + to designate keys that should be pressed at the same time.
Example: *Alt+F7*

Instructions should be prefaced with ASCII character 16 (►).
Information that is not an instruction should have no bullets.

In dialog boxes displaying any text, there should be one blank line between the title and the first line of text, and one blank line between the last line of text and the bottom of the dialog box, or the buttons, if any.

There should be one space between the left and right borders of the dialog box and the text inside.

Text should not be purposely centered in the boxes.

The size of the box will generally be dependent on the amount of text inside. If there are size constraints, the blank lines at the top and bottom can be used for text.

Do not instruct the user to choose a button when that button is on the Coach Dialog box, and is obvious. For Example, *Continue*.

When instructing what the user should choose, place the name of the menu item or button in bold.

When you need to emphasize text, place that text between Underline codes.

When instructing the user to type information, place that information between Underline codes.

Message Naming Conventions

Hints	5000's

List of Commands in WPCHLIBR.WPM
Coach Library File

Procedures

Procedures do not return a value.

Functions

Functions return a value

Procedure: BlockEdit(Title;EnableBlk;EnableMouse;&DefMsg;Helps)

Purpose: BlockEdit is used to allow user to block text in the document. Users can use the search feature. User can choose block from the menu as well as using the mouse to block and drag. User can press Alt+F4 to turn block on and use the arrow keys to find the end of the selection.

Parameters:

Title	Title to appear--Should always be Coach	
EnableBlk	0	Select mode not allowed
	1	Select mode allowed
EnableMouse	0	Mouse clicks not allowed
	1	Mouse clicks allowed
&DefMsg		Message to appear

Helps Help prompts can appear that give the user guidance in blocking the text. It gives appropriate info whether the user is using the mouse or keystrokes to block.

0	Do not allow help prompts
1	Allow help prompts

Procedure: CancelCoach()

Purpose: Used when user presses Escape to Cancel the Coach

Parameters:

no parameters

Procedure: PROCEDURE CancelMenus()

Purpose: CancelMenus will cancel all active menus

Parameters:

no parameters

Procedure: CancelDialogs()

Purpose: CancelDialogs will cancel all active dialogs

Parameters:

No parameters

Procedure: CheckColorSet()

Purpose:

Parameters:

no parameters

Procedure: ColorScheme()

Purpose: Sets the WPCoach Color Scheme

Parameters:

no parameters

Procedure: CommentsBox(&cmnt;xpos;ypos;width;height;Style)

Purpose: CommentsBox is used for posting messages anywhere on the document screen. No editing control is provided. This proc will post the message and return control back to the macro.

Parameters:

&cmnt		Array message
Xpos		Horizontal position
Ypos		Vertical position
Width		Width of box
Height		height of box
Style	0	Use xpos and ypos parms
	1	Centered

Procedure: EditMenu(&valid)

Purpose: Sets the valid menu IDs for the Edit menu.

Parameters:

valid	Array of all Dialog Control IDs found on the Edit Menu

Procedure: ExitDlgProc()

Purpose: ExitDlgProc() helps the user with the Exit Dialog Box

Parameters:

no parameters

Procedure: ExitDocProc()

Purpose: ExitDocProc() Helps the user exit a document

Parameters:

no parameters

Procedure: ExitWPProc()

Purpose: ExitWPProc() helps the user exit WordPerfect

Parameters:

no parameters

Procedure: FakeIt(Title;&tbox;Buttons;Frame;Steps)

Purpose: FakeIt is used for posting messages at bottom of the document screen. The document screen is framed and sized according to the size of the FAKEIT message. No editing control is provided. This proc will post the message and return control back to the macro. Follow this command with another such as DoEdit to allow the user to perform a task.

Parameters:

Title		The title at the top of the FAKEIT message box	
&tbox		Message array	
Buttons	0	No Buttons	
		1	Continue (w/ default)
		2	Continue (w/ StyGray!)
		3	Continue...F7 (w/o default) Quit...Esc
		4	Show Me...Ctrl+S Continue...F7 Quit...Esc
		5	Hint...F1 Continue...F7 Quit...Esc
		6	Hint...F1 Quit...Esc
		7	Show Me...Ctrl+S Quit...Esc
		8	Quit...Esc
Frame		0	No DisplayRewrite and WindowSize
		1	Cause a DisplayRewrite and WindowSize to occur
Steps		0	Do not show steps
		1	Show Steps

Return Values: RETURN: NONE Values will be returned by DoEdit

-1 = Quit
1 = Continue
2 = Show Me or Hint

Procedure: FileMenu(&valid)

Purpose: Sets the valid menu IDs for the File menu.

Parameters:

valid	Array of all Dialog Control IDs found on the File Menu

Procedure: FileSaveDlgProc()

Purpose: FileSaveDlgProc() helps the user save a file

Procedure: FontMenu(&valid)

Purpose: Sets the valid menu IDs for the Font menu.

Parameters:

valid	Array of all Dialog Control IDs found on the Font Menu

Procedure: GraphicsMenu(&valid)

Purpose: Sets the valid menu IDs for the Graphics menu.

Parameters:

valid	Array of all Dialog Control IDs found on the Graphics Menu

Procedure: HelpMenu(&valid)

Purpose: Sets the valid menu IDs for the Help menu.

Parameters:

valid	Array of all Dialog Control IDs found on the Help Menu

Procedure: LayoutMenu(&valid)

Purpose: Sets the valid menu IDs for the Layout menu.

Parameters:

valid	Array of all Dialog Control IDs found on the Layout Menu

Procedure: MenuBar(&bar)

Purpose:

Parameters:

bar	Array of all Dialog Control IDs on the menu bar

Procedure: MenuPop(&pop)

Purpose:

Parameters:

pop	Array of all Dialog Control IDs that belong to popouts on the menus

Procedure: NoSubMenu(&subval;&subacl;&exsubacl;&subs)

Purpose:	This is a quick way to set submenu values to -1. Use to set subaccel, subval, subs, and exsubacl for MenuControl. Also used by BlockFromMenu Procedure.

Parameters:

&subval	Submenu item to allow to execute
&subacl	Submenu accelerators to allow to execute
&exsubacl	Execute bits for Submenu accelerators
&subs	Main menu IDs which pop out a sub menu

Procedure: NotFoundCoach()

Purpose: Used when a Not Found condition occurs

Parameters:

no parameters

Procedure: PleaseWait()

Purpose:

Parameters:

no parameters

Procedure: QuitCoach()

Purpose: Quits the Coach without asking permission. Also resets screen setup from variables assigned with the SetupScreen() procedure.

Parameters:

no parameters

Procedure: ReplaceProc()

Purpose: ReplaceProc() Helps the user save and replace a document

Parameters:

no parameters

Procedure: RestoreStates()

Purpose: Restores Button Bar, Ribbon, or Outline Bar if they were on when Coach began.

Parameters:

no parameters

Procedure: SavingDoc()

Purpose: SavingDoc() helps the user save a document

Parameters:

no parameters

Procedure: ScreenSetup()

Purpose: Turns off Button Bar, ruler bar, outline bar, and sets a bit if they were on. Use RestoreStates to turn them back on at end of Coach. Also checks for screen mode, and if they are in an unsuported mode, a message tells then that, and ends the Coach.

Parameters:

no parameters

Procedure: SetBoxSize()

Purpose: Determines the X and Y values for a dialog box, given an array message, tbox. These values, W and H, can be used for the width and height parameters for dialog boxes.

Parameters:

no parameters

Procedure: ShowMenu(&sbox;xpos;ypos;Mnu;Opt;Pop;Execute)

Purpose: ShowMenu is a proc that automates the menus to find a given menu option

Parameters:

&sbox	Array message to appear	
		-1 in no message is required
xpos	Horizontal Position of message	
		-1 if no message
ypos	Vertical Position of message	
		-1 if no message
Mnu	Menu number from 1 to 9	
		File = 1
		Edit = 2
		View = 3
		Layout = 4
		Tools = 5
		Font = 6
		Graphics = 7
		Windows =8
		Help = 9
Opt	Option number from the top	
		The first item on the menu is a 1. Count down to the item you need and specify its number.
Pop	0	Item is not located on a pop out menu
	1	Item located on a pop out menu
Execute	0	Do not allow execute
	1	Allow execute menus of menu items
	2	Executes item automatically

Procedure: ToolsMenu(&valid)

Purpose: Sets the valid menu IDs for the Tools menu.

Parameters:

valid	Array of all Dialog Control IDs found on the Tools Menu

Procedure: ViewMenu(&valid)

Purpose: Sets the valid menu IDs for the View menu.

Parameters:

valid	Array of all Dialog Control IDs found on the View Menu

Procedure: WindowMenu(&valid)

Purpose: Sets the valid menu IDs for the Window menu.

Parameters:

valid	Array of all Dialog Control IDs found on the Window Menu

Function: ButtonControl (&info; &x1; &x2; &y1; &y2; &hotkey; &hotpnt; &accel; &aclpnt; &tbox; &helpmsg; &allow)

Purpose: Used to control buttons and line items in a dialog box.

Parameters:

info contains 10 parameters:

1 Control ID-The ID of the current control
2 Dialog ID (can use ?Dialog)
  Always 0
3 Allow Up/Down Arrows
  1 = yes 0 = no
4 Title for Postit Message
5 xpos for Postit
  -99 = Center
6 ypos for Postit
  -99 = Center
7 Postit Message Type
  1=Continue, Quit w/ default
  2=Show Me w/ default
  3=Show Me w/ gray
  4=Hint
  5=Show Me Hint Quit
  6=Quit

8 Title of HelpMsg (Usually "Coach")
9 Restore Screen -- Uses ShowCode(SaveScreen!) and ShowCode(RestoreScreen!)
  1 = yes 0 = no
10 Reserved
  1 = positions cursor to fix a bug where cursor jumps to upper left corner.

x1 arrays for hotspots
x2
y1
y2

hotkey key equivalent for each hotspot item defined with the x1, x2, y1, and x2 parameters

hotpnt restore screen for current hotspot to clear the Current Coach dialog.
  0=do not repaint
  1=repaint
  2=waits 2 seconds for doubleclick
This parameter must have one element for each defined hotspot element.

accel the keys you want to allow the user to press

aclpnt restore screen for current accelerator to clear the Current Coach dialog. 0=do not repaint 1=repaint. This parameter must have one element for each defined accel element.

tbox message to appear in main message box

Helpmsg help message--appears if user flounders

hbox Hint message if needed-- requires the following placed just after the WHILE command, or if you need a different hint for each control, within the CaseOf for the controls requiring the hint.

```
If (Result="Hint")
        hbox={"Hint Message"}
        Hint=(MessageBox("HINT";&hbox;1)
Endif
```

allow Array address of all dialog control Id's that have a case represented. This option will allow focus to stay in invalid until tab or an arrow key causes focus to land on a dialog control for which a case or control has been defined in the Coach.

Function: CodeFilter(&array;Code)

Purpose:

Parameters:

no parameters

Function: DlgFilter(&array;&brk)

Purpose: DlgFilter is a function that will compare Dialog Control IDs against allowable Dialog Control IDs and executable bits for each Control ID.

Parameters:

&array	Array address of items to check
&brk	Array address of execution bits

Return Values:

0	No valid item found
1	A valid item was found
2	A valid item was found plus an execution bit was also found

Function: DlgMessage()

Purpose: This function will determine the offset for Coach dialogs that are used in conjuction with the system dialogs to assure proper screen registration.

Parameters:

no parameters

Return Values:

0	24 line screen mode is currently set
12	49 line screen mode is currently set

Function: DoEdit(EnableMouse;&Chkcode;&KeyPress)

Purpose: DoEdit will allow editing capability in the document and break out when the hotspot region was clicked on by the mouse or the key passed in KeyPress has been entered. This can be used with FAKEIT.

Parameters:

EnableMouse Allow the user to use the mouse

0	Mouse clicks not allowed
1	Mouse clicks allowed

Chkcode Array address of codes that, when the user attempts to delete, will delete the code and return control back to the macro, passing the deleted code back. Set ChkCode to -1 if no deleted codes need to be tracked.

KeyPress Keys to allow user to press

Array of values of additional keys to allow
Example: -8154 = F7

Return Values:

-1	Quit
1	Continue
2	Show Me or Hint from a Fakeit
Key	Any other value will equal key number

Function: Filter(&array;Key;Accel)

Purpose: Filter is a func that will compare key values or Dialog Control IDs against allowable key IDs or allowable Dialog Control IDs.

Parameters:

&array	Array address of items to check	
Key		Item to check against those items found in &array
Accel	0	Checking accelerators
	1	Checking menu values

Return Values:	0	No valid item found
	1	A valid item was found

Function: Hint(&tbox;xpos;ypos;Buttons)

Purpose:

Parameters:

tbox	The message to appear on the screen			
xpos	horizontal position			
ypos	vertical position			
Button	0	Title=Quick Tip	~Continue	
	1	Title=Hint Tip	~Show Me	~Return to Coach
	2	Title=Quick Tip	~Show Me	~Continue
	3	Title=Hint	~Return to Coach	

Function: HotSpot(xpos1;ypos1;xpos2;ypos2;AltKey;&funkey;Execute)

Purpose: HotSpot will set up one hotspot region. It allows the user to choose the hotspot, or press an Alt key or multiple Funtion keys.

Parameters:

xpos1	left horizontal position
ypos1	top vertical position
xpos2	right horizontal position
ypos2	bottom vertical position
AltKey	key value of allowable Alt Key Example: -529 = Alt-T
&funkey	Array address of valid function keys to allow Example -8137 = Alt-F7
Execute	Execute the user's choice 0 Do not execute 1 Allow execution of the chosen key

Return Values:

1	Show button pressed or Enter key pressed
3	Menu accelerator key was pressed

Function: InvalidControl (&info; &x1; &x2; &y1; &y2; &hotkey; &hotpnt; &accel; &aclpnt; &tbox; &helpmsg)

Purpose: Used to control any other item. This command is normally used with the Dafault parameter of a CASE statement.

Parameters:

info contains 10 parameters:

1 Control ID

2 Dialog

3 Allow Up/Down Arrows

1 = yes 0 = no

4 Title for Postit Message

5 xpos for Postit Take care that the Postit Box does not overlay any hotspots.

-99 = Center

6 ypos for Postit Take care that the Postit Box does not overlay any hotspots.

-99 = Center

7 Postit Message Type

1=Continue, Quit w/ default
2=Show Me w/ default
3=Show Me w/ gray
4=Hint
5=Show Me Hint Quit
6=Quit

8 Title of HelpMsg--(Usually "Coach")

9 Restore Screen -- Uses ShowCode(SaveScreen!) and ShowCode(RestoreScreen!)

1 = yes 0 = no

10 Reserved

x1 arrays for hotspots
x2
y1
y2

hotkey key equivalent for each hotspot item defined with the x1, x2, y1, and x2 parameters

hotpnt restore screen for current hotspot to clear the Current Coach dialog.

0=do not repaint
1=repaint
2=waits 2 seconds for a doubleclick

This parameter must have one element for each defined hotspot element.

accel the keys you want to allow the user to press

aclpnt restore screen for current accelerator to clear the Current Coach dialog. 0=do not repaint 1=repaint. This parameter must have one element for each defined accel element.

tbox message

helpmsg	help message--appears if user flounders

hbox	Hint message if needed-- requires the following placed just after the WHILE command, or if you need a different hint for each control, within the CaseOf for the controls requiring the hint.

```
If (Result="Hint")
        hbox={"Hint Message"}
        Hint=(MessageBox("HINT";&hbox;1)
Endif
```

Function: KeyFilter(&array;Key;&brk)

Purpose: KeyFilter is a function that will compare key values against allowable Key IDs and executable bits for each key.

Parameters:

&array	Array address of items to check
Key	Item to check against those items found in &array
&brk	Array address of execution bits

Return Values:

0	No valid item found
1	A valid item was found
2	A valid item was found plus an execution bit was also found

Function: LstBxControl (&info; &x1; &x2; &y1; &y2; &hotkey; &hotpnt; &accel; &aclpnt; &tbox; &helpmsg)

Purpose: Used to control List Boxes in Dialog Boxes

Parameters:

info contains 10 parameters:

1 Control ID
2 Dialog ID
3 Allow Up/Down Arrows
    1 = yes 0 = no
4 Title for Postit Message
5 xpos for Postit
    -99 = Center
6 ypos for Postit
    -99 = Center
7 Postit Message Type
    1=Continue, Quit w/ default
    2=Show Me w/ default
    3=Show Me w/ gray
    4=Hint
    5=Show Me Hint Quit

8 Title of HelpMsg--(Usually "Coach")
9 Restore Screen -- Uses ShowCode(SaveScreen!) and ShowCode(RestoreScreen!)
    1 = yes 0 = no
10 Reserved

x1 arrays for hotspots
x2
y1
y2

hotkey key equivalent for each hotspot item defined with the x1, x2, y1, and x2 parameters
hotpnt restore screen for current hotspot to clear the Current Coach dialog.
    0=do not repaint
    1=repaint.
    2=waits 2 seconds for a doubleclick
This parameter must have one element for each defined hotspot element.
accel the keys you want to allow the user to press
aclpnt restore screen for current accelerator to clear the Current Coach dialog. 0=do not repaint 1=repaint. This parameter must have one element for each defined accel element.
tbox message
helpmsg help message--appears if user flounders
hbox Hint message if needed-- requires the following placed just after the WHILE command, or if you need a different hint for each control, within the CaseOf for the controls requiring the hint.

```
If (Result="Hint")
```

```
          hbox={"Hint Message"}
          Hint=(MessageBox("HINT";&hbox;1)
Endif
```

Function: MenuBarControl(Title;&hbox1;&sbox;xpos;ypos;MnuLtr;&funkey)

Purpose: MenuBarControl is a proc that controls the menu bar until a desired menu has been located before returning control to the macro. Use within a Repeat/Until (MenuActive<>-1) loop to allow only the correct item to be selected from the Menu Bar.

Parameters:

&hbox1	Array address of help message This is normally the hbox1 message set for MenuControl
&sbox	Array address of show message
xpos	Horizontal position of the show message
ypos	Vertical position of the show message

MnuLtr	Alpha letter that represents one of the pull down menus
	F = File
	E = Edit
	V = View
	L = Layout
	T = Tools
	o = Font
	G = Graphics
	W = Window
	H = Help

&funkey	allowable function key 1 Continue

Function: MenuControl (&hbox1; &hbox2; &hbox3; &valid; &allow; &exallow; &accel; &exaccel; &subval; &subacl; &exsubacl; &subs; &funkey)

Purpose: MenuControl will control the access to menus. Specify the menu item(s) you want to allow with the &allow parameter. When the user chooses the correct items this command breaks. This command can be used in a Repeat/Until(MenuActive<>-1) loop. It should be the last command in the loop and be followed with a MenuActive=Result to assign the result of MenuControl to Menuactive.

Parameters:

&hbox1	Coach message directing to correct menu bar item
&hbox2	Coach message directing to correct item on menu
&hbox3	Coach message directing to correct item on submenu

&valid Valid options found on the same menu as the allowed items. These are set by using EditMenu(&valid), FileMenu(&valid), etc.

&allow Array address of menu options to allow

Example: Allow={2050;} //Create

&exallow Array address that corresponds with values listed in &allow. A 1 means to allow execute and a 0 means to not allow execution but to break and return either the menu ID or Key value.

Example: exallow={1;} //allow to execute

&accel Valid main menu accelerators

Example: accel={98;} //Ta(b)les

&exaccel Array address that corresponds with values listed in &accel. A 1 means to allow execute and a 0 means to not allow execution but to break and return either the menu ID or Key value.

Example: exaccel={1;} //allow to execute

&subval Options found on the same sub menu as those options allowed

```
Example:  subval={2050;    //Create
                  2051;    //Edit
                  776;     //Insert Row
                  777;     //Delete Row
                  2052;    //Join
                  2053;    //Split
                  2054;    //Calculate
                  2055;    //Create Floating Cell
                  2056;    //Edit Floating Cell
                           }
```

&subacl Submenu accelerators to allow

Example: subacl={99;} //Create

&exsubacl Array address that corresponds with values listed in &subacl. A 1 means to allow execute and a 0 means to not allow execution but to break and return either the menu ID or Key value.

Example: exsubacl={1;} //allow to execute

&subs Valid main menu items that pop out a sub menu
Example: Subs={1537;}//Tables

&funkey allowable function key

Return Values:	Dialog ID	Highlighted menu option or
	Key value	Key value if accelerator key was pressed

Function: MessageBox(Title;&tbox;Buttons)

Purpose: MessageBox will post a centered message that needs to be dismissed by pressing one of its buttons.

Parameters:

Title	The title to appear at the top of the box.
&tbox	The array of text to be displayed in the box
Buttons	The buttons to appear on the box.

0	No buttons
1	OK
2	OK Cancel
3	Yes, No w/ Yes as default
4	Yes, No, Cancel w/ Yes as default
5	Yes, No w/ No as default
6	Quit...Esc
7	Continue
8	Continue Quit...Esc

Return Values:

-1	Quit, Cancel
1	OK, Yes, Continue
2	No

Function: NoBlock()

Purpose: Checks if Block is on. A message appears that says "Nothing is Blocked."

Parameters:

No parameters

Return Values:

1	Yes	
	2	No

Function: NoSelect()

Purpose: Checks if text is selected. A Message appears that says "Block is on although nothing is selected.

Parameters:

no parameters

Return Values:	1	Yes
	2	No

Function: OpenDocument(&tbox)

Purpose:	OpenDocument is a function that helps the user open a document. Returns a 1 or -1.	
Parameters:		
tbox	The instructions for typing the file name.	
	-1	The default text to be displayed will be : The document screen is blank. Would you like to open a file?
Return Values:	-1	User chose to not open a file
	1	User opened a file

Function: Overview(Title;&tbox;Buttons)

Purpose: Creates the Overview dialog box that can be used at the beginning of a Coach.

Parameters:

Title	Always Coach
&tbox	The text to appear in the box
Buttons	always Prev Next Cancel

Return Values:

1=Prev
2=Next
-1=Cancel

Function: PostIt(Title;&tbox;xpos;ypos;Modal;Buttons;Steps)

Purpose: PostIt will allow an inactive message box to be displayed and set a hotspot location for one or more buttons. The user must press a button to continue. Their choice is assigned to the variable.

Parameters:

Title	The title to be displayed at the top of the box	
tbox	The array essage to be displayed in the box	
xpos	The horizontal position of the box	
	-99	Center Box
ypos	The vertical position of the box	
	-99	Center Box
Modal	Should box wait for user response?	
	0	Do not wait for response
	1	Wait for User Response
Buttons	The buttons to appear on the box	
	0	no buttons
	1	Quit...Esc
	2	Show Me...Ctrl+S (default) Quit...Esc
	3	Show Me...Ctrl+S (StyGray!) Quit...Esc (StyGray!)
	4	Hint...F1 Quit...Esc
	5	Show Me Hint...F1 Quit...Esc
	6	Show Me...Ctrl+S Quit...Esc (Stacked)
	7	Hint...F1 Quit...Esc (stacked)
	Show the current step text.	
Steps	0	Do not show steps
	1	Show Steps

Return Values:

-1	Quit
1	Show Me
2	Hint

Function: Welcome(Title;&tbox)

Purpose:

Parameters:

Title	the title to appear at the top of the Welcome screen
tbox	the Welcome message to appear in the box

Return Values:

1=Continue
-1=Quit

Function: WinEdControl (&info; &x1; &x2; &y1; &y2; &hotkey; &hotpnt; &accel; &aclpnt; &tbox; &helpmsg)

Purpose: Used to control Window Edit boxes in Dialog Boxes

Parameters:

info contains 10 parameters:

1 Control ID
2 Dialog ID
3 Allow Up/Down Arrows
    1 = yes 0 = no
4 Title for Postit Message
5 xpos for Postit
    -99 = Center
6 ypos for Postit
    -99 = Center
7 Postit Message Type
    1=Continue, Quit w/ default
    2=Show Me w/ default
    3=Show Me w/ gray
    4=Hint
    5=Show Me Hint Quit
    6=Quit
8 Title of HelpMsg--(Usually "Coach")
9 Restore Screen -- Uses ShowCode(SaveScreen!) and ShowCode(RestoreScreen!)
    1 = yes 0 = no
10 Reserved

x1 arrays for hotspots
x2
y1
y2

hotkey key equivalent for each hotspot item defined with the x1, x2, y1, and x2 parameters
hotpnt restore screen for current hotspot to clear the Current Coach dialog.
    0=do not repaint
    1=repaint.
    2=waits 2 seconds for doubleclick
  This parameter must have one element for each defined hotspot element.
accel the keys you want to allow the user to press
aclpnt restore screen for current accelerator to clear the Current Coach dialog. 0=do not repaint 1=repaint. This parameter must have one element for each defined accel element.
tbox message
helpmsg help message--appears if user flounders
hbox Hint message if needed-- requires the following placed just after the WHILE command, or if you need a different hint for each control, within the CaseOf for the controls requiring the hint.

```
If (Result="Hint")
```

```
                        hbox={"Hint Message"}
                        Hint=(MessageBox("HINT";&hbox;1)
                    Endif
```

hotElement   number of equivalent hotspot for current Window Edit item, as defined in the Arrays for Hotspots.

Function: XYFilter(&x1;&x2;&y1;&y2)

Purpose: XYFilter is a func that will compare mouse clicks against allowable hotspot regions.

Parameters:

&x1	Left Horizontal position of hotspot rectangle
&x2	Right Horizontal position of hotspot rectangle
&y1	Top Vertical position of hotspot rectangle
&x2	Bottom Vertical position of hotspot rectangle

Return Values:

0	No valid hotspot found
1	A valid hotspot was found

Index of Procedures and Functions