



VerbalBasic™ User Manual & Training Guide

Table of Contents (use hyperlinks)

[Preliminary](#)

[Hierarchy](#)

[Simple Global Text Macro Example \(w/Bookmarks\)](#)

[Simple Global Text Macro Example \(w/Formatting\)](#)

[Macro Tools](#)

[Beginning](#)

[Steps 2 & 3](#)

[Step 4](#)

[Keystroke Commands](#)

[Repeating Keystrokes](#)

[Scripting Commands](#)

[Repeating Scripts](#)

[Wait Commands](#)

[Mouse Commands](#)

[Heard Word Commands](#)

[Looping Commands](#)

[Argument \(Listing\) Commands](#)

[Example Commands](#)

[Example 1](#)

[Example 2](#)

[Example 3](#)

[How to Edit a Listing Command](#) (such as [**Open <File>**])

[Creating Commands in Programs that KnowBrainer Doesn't Currently Support](#)

Preliminary

VerbalBasic is a macro *toolbox*. These assorted tools will assist the user to *verbally* create commands as quickly and easily as possible. We have endeavored to expedite this process by *eliminating Steps in the Command Wizard* wherever practical.

The following sections are a brief overview on using VerbalBasic:

Hierarchy

Before you begin creating commands, please familiarize yourself with NatSpeak's command hierarchy.

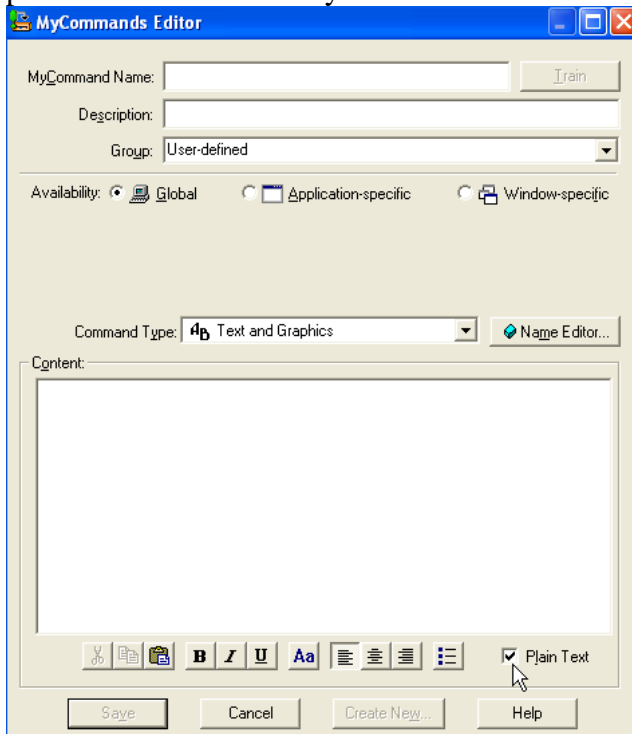
1. Global Commands - *function at all times*. When you say “**New Global Command**” KnowBrainer will open its Command Browser to the global command (system.dvc) and click the **Add** button
2. Application Specific Commands - function only when a *specific application* is

open, but are supposed to take precedence over Global commands (when both commands are the same). Please note that this is not always the case but we are working on a solution.

3. Boilerplate text commands can be global, application-specific or even Windows specific. We use Dragon NaturallySpeaking's Command Browser for creating boilerplate text commands but use the KnowBrainer Command Browser for all other types of commands. Additional information on this procedure is included in the NaturallySpeaking manual and KnowBrainer training guide.

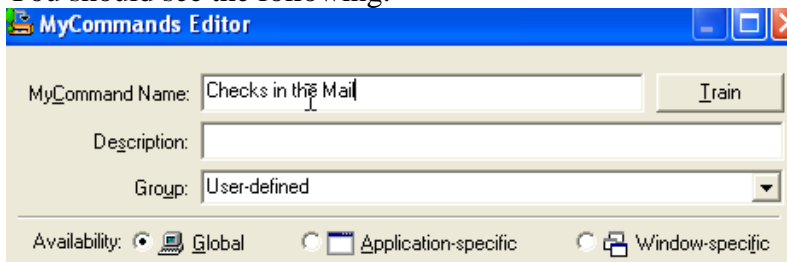
Global Text Command Example w/Bookmarks (utilizing NaturallySpeaking's Command Browser) Note That This Capability Is Not Available in DNS Standard

1. Say “**New Text Command**” to start NaturallySpeaking's Command Wizard, place a check mark in the Plain Text box (to force the text to take on the formatting of whatever document you deploy it into), place the cursor in the MyCommand Name field and prompt you to name the command.



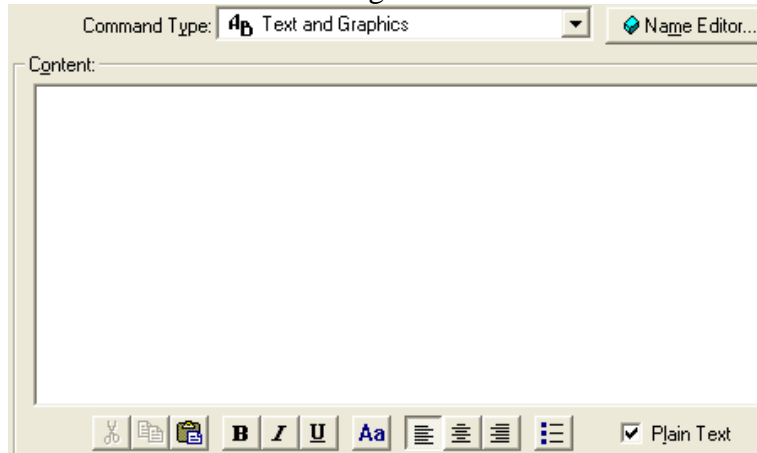
2. Say “**Checks in the Mail**” to name your command. Note that the text automatically appears in Title Case to make it stand out.

You should see the following:



3. Say “**Content**” to move the cursor to the Content field.

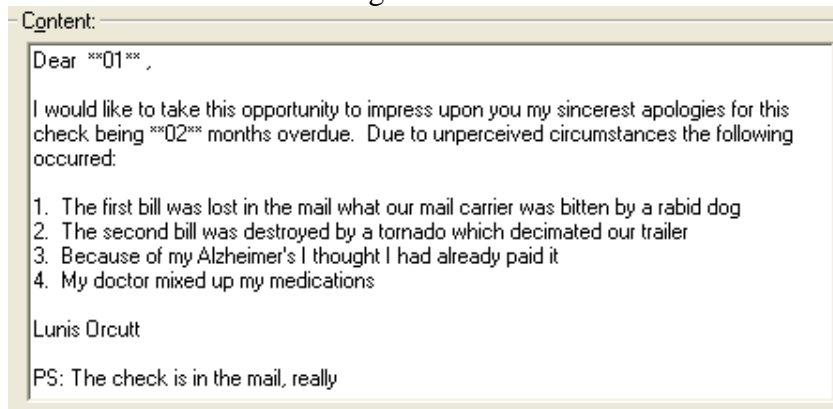
You should see the following:



4. Dictate the following: **Mr.** {Pause} **Bookmark 1** {Pause}, {Pause} **Select Mr.** {Pause} **Dear** {Pause} **Add Paragraph** {Pause} **I would like to take this opportunity to impress upon you my sincerest apologies for this check being** {Pause} **Bookmark 2** {Pause} **months overdue. Due to unforeseen circumstances the following occurred:** {Pause} **Double List 1** {Pause} **The first bill was lost in the mail when our mail carrier was bitten by a rabid dog** {Pause} **List 2** {Pause} **The second bill was destroyed by a tornado which decimated our trailer** {Pause} **List 3** {Pause} **Because of my Alzheimer's I thought I had already paid it** {Pause} **List 4** {Pause} **My doctor mixed up my medications** {Pause} **Add Paragraph** {Pause} **(Your name here)** {Pause} **PS** {Pause} **The check is in the mail, really**

Note: Insert a brief Pause in your speech wherever you see this: {Pause}

You should see the following:



5. Say “**Save**” to compile your new global text command.

To try your new command, open any word processor and say “**Checks in the Mail**”.

Say “**Number 1**” to highlight your first bookmark. Then dictate the proper company name.

Say “**Number 2**” to highlight your second bookmark. Then dictate the proper number of months.

Note: For more information regarding Bookmarks refer to the NaturallySpeaking manual or KnowBrainer Training Guide

MACRO TOOLS for KnowBrainer Advanced-Scripting

Note: The following VerbalBasic commands are designed to work in the KnowBrainer Command Browser but we also support the NaturallySpeaking Command Browser (Professional and higher). Some command names have been altered slightly to work in the NaturallySpeaking Command Browser.

Keystroke Examples (must be prefaced with the word **SendKeys**):

SendKeys "{Backspace}{Enter}{Del 3}{Tab 18}{Shift+Tab 20}"
{Alt+F4}{Alt+p}{Ctrl+r}{Alt+Ctrl+z}{Alt+Ctrl+Shift+x}{Alt+Ctrl+7}

Script Examples:

SendKeys "{Enter 5}" - presses the Enter Key 5 times

SendKeys "{Alt+Ctrl+Shift+F2}" - presses Alt+Ctrl+Shift+F2

MenuPick "Save" - selects the "Save" menu item

ControlPick "OK" - presses the "OK" button

AppBringUp "WAOL" - launches the "AOL" application

Note: As long as the application has been previously entered into the registry, the precise application address is unnecessary.

AppSwapWith "Winword" - switches (brings forward) the "Microsoft Word" application

Wait 3000 - equals a 3 second pause

MsgBoxConfirm "Are You Sure?",52, "Warning!" - The message confirmation box visually asks "are you sure" (simply in order to confirm the accuracy of the new instruction). You must say "Yes" before the remainder of the command will execute. This is used for mission critical commands.

TTSPPlayString "my girdle is killing me" - This command prompts the text-to-speech reader to *verbally* read the message (when verbal confirmation is useful). Generally, you would use this command for:

1. Lengthy commands, such as "Check My Email"
2. Informative commands
3. Invisible commands, such as "Copy"

Example: Sleep Mode verbally confirms that the microphone has been placed in sleep mode.

PlaySound "C:\Windows\sounds\explosion.wav" - This command plays an *explosion wav file* located at the specified address.

GoToSleep - places NatSpeak in sleep mode

WakeUp - wakes up the microphone

SetMicrophone 1 - turns the microphone on

SetMicrophone 0 - turns the microphone off

ClearDesktop - minimizes all open Windows

MouseGrid 1, 1 - This command selects the #1 block (of the MouseGrid), its range encompassing #<1-9>.

SetMousePosition 1, x, y - This command is used to create an exact mouse position. Say **Find Mouse Position** to locate the xy coordinates.

Note #1: "x" represents the "left-to-right" control. "y" represents the "up-and-down" control.

Note #2: "x" If you want to find the exact xy coordinates place your mouse over the correct position and say "Find Mouse Position"

Example: When you say "**Mouse Position**" NaturallySpeaking will type the following into the Advanced Scripting box and highlight the x, y coordinates:

```
SetMousePosition 1, x, y
ButtonClick 1,1
```

ButtonClick 1,1 - left mouse button click

ButtonClick 2,1 - right mouse button click

ButtonClick 1,2 - double left button mouse click

HeardWord "print", "document" - executes the **Print Document** command

SendSystemKeys "{Alt+Space}x" - maximizes the window with a *system keys* command

SendSystemKeys "{Alt+Space}n" - minimizes the window

SendSystemKeys "{Alt+Esc}" - selects next window

SendSystemKeys "{Shift+Alt+Esc}" - selects previous window

SendSystemKeys "{Alt+Tab}" This command is used to toggle back and forth between two windows.

Simple Loop:

Delete <1To20>

```
SendKeys "{Del "+_arg1+"}"
```

(the previous loop command allows you to press the delete key up to 20 times)

Complex Loops:

Back <1To20>

```
loop& = Val(_arg1)
```

```
while ( loop& )
```

```
SendKeys "{Alt+Left}"
```

```
Wait .5
```

```
loop& = loop& - 1
```

```
wend
```

Line <1To100> Times

```
loop& = Val(_arg1)
```

```
while ( loop& )
```

```
SendKeys "_"
```

```
Wait .01
```

```
loop& = loop& - 1
```

```
wend
```

(the previous loop command allows you to press the “_” key up to 100 times)

Interior Loop:

```
I = 22
DO UNTIL I = 0
SendKeys “My Name Is Norman Bates {Enter}”
I = I -1
Loop
```

(the previous loop command prints Norman Bates on 22 separate lines by pressing the “Enter” key after each line)

Argumentative Statements:

```
Start <AppList>
if _arg1 = "Access" then AppBringUp "MSAccess"
if _arg1 = "Act" then AppBringUp "ACT"
```

```
Email <Name>
AppBringUp "MSIMN"
Wait 2000
SendKeys "{Alt+f}nm"
if _arg1 = "Lunis" then SendKeys "lunis@home.com"
if _arg1 = "Bob Johnson" then SendKeys "notarealaddress@aol.com"
Wait 500
SendKeys "{Tab 2}"
TTSPlayString "subject"
```

Simple Argument:

```
<NameList> Field
SendKeys "{Ctrl+f}"
Wait 250
SendKeys ""+_arg1+":{Alt+f}"
Wait 250
SendKeys "{Alt+F4}"
SendKeys "{Right}{Space}"
```

Note: The commands in this section should ONLY be used to edit a .dvc file which you have imported from an earlier version of NaturallySpeaking or copied from a KnowBrainer sample command.

Beginning

*Note: #1: When creating commands verbally KnowBrainer works best if you run it with the **KB 2008** bar displayed in Windows XP rather than minimized. Minimized is fine in Windows Vista*

Note: #2: New KnowBrainer commands will not work until you close and re-launch KB 2008 which can be accomplished by saying [Refresh KnowBrainer] but if you are just adding or editing a single command you can substitute [Refresh Command] which will save the current command, close and re-launch KnowBrainer. Note that this command will not be available until REV D

[<Edit or New> <Application> Command] - will open the Command Browser to a specific application-specific program. The choices are ~ Microsoft Word, Word, Microsoft Excel, Excel, Microsoft Access, Access, Microsoft Paint, Paint, Microsoft Outlook, Outlook, Microsoft PowerPoint, PowerPoint, Microsoft FrontPage, FrontPage, OmniPage, DragonPad, NaturallySpeaking (DragonPad), Windows Explorer, Explorer, Internet Explorer, Word Perfect, WordPad and VerbalBasic (our KB Command Browser command building macros)

Example # 1: [New Word Command] will open the Command Browser to MS Word application-specific commands only.

Example # 2: [Edit Microsoft FrontPage Command] will open the Command Browser to MS FrontPage application-specific commands only.

If you wish to create an application-specific command for a program that KnowBrainer doesn't currently support you should place your cursor inside of the application you wish to create your first application-specific command for and say [Add Application] – which will open the KnowBrainer Command Browser, click *Add Application*, click the *Find Application* button, place the cursor in the previous application (the one you wish to create a command for), add the application to the *Add a New Application* window, place the cursor in the *Name* field and prompt you to dictate your new command.

[New Text Command] - launches NaturallySpeaking's Add New Command, places the cursor in the Name field and verbally prompts you to name the new command

Keystroke Commands

Alternate < a - z, 0 - 9, **or** F1 - F12 >

Note: If, for example, you say, "Alternate B" or "Alternate Bravo" (ICD alphabet), {Alt+b} will appear in the command box.

Alternate < a - z, 0 - 9, **or** F1 - F12 > < 1 - 20 >

Control < a - z, 0 - 9, **or** F1 - F12 >

Control < a - z, 0 - 9, **or** F1 - F12 > < 1 - 20 >

Shift < a - z, 0 - 9, **or** F1 - F12 >

Alternate Control < a - z, 0 - 9, **or** F1 - F12 >

Control Alternate < a - z, 0 - 9, **or** F1 - F12 >

Alternate Shift < a - z, 0 - 9, **or** F1 - F12 >

Control Shift < a - z, 0 - 9, **or** F1 - F12 >

Alternate Control Shift < a - z, 0 - 9, **or** F1 - F12 >

Backspace 1 - types {Backspace}

Delete 1 - types {Del}

Down 1 - types {Down}

End 1 - types {End}

Enter 1 - types {Enter}

Escape 1 - types {Esc}

Home 1 - types {Home}
Left 1 - types {Left}
Page Down 1 - types {PgDn}
Page Up 1 - types {PgUp}
Right 1 - types {Right}
Space 1 - types {Space}
Shift Tab 1 - types {Shift+Tab}
Tab 1 - types {Tab}
Up 1 - types {Up}
Print Screen - types {Prtsc}

Repeating Keystrokes

Backspace < 1 - 20 >

Delete < 1 - 20 >

Note: If, for example, you say “Delete 3” {Del 3} will appear in the Advanced Scripting box.

Down < 1 - 20 >

Enter < 1 - 20 >

Left < 1 - 20 >

Right < 1 - 20 >

Space < 1 - 20 >

Shift Tab < 1 - 20 >

Tab < 1 - 20 >

Up < 1 - 20 >

Scripting Commands

SendKeys Alternate < a - z, 0 - 9, **or** F1 - F12 >

Note: If, for example, you say “SendKeys Alternate C” or “SendKeys Alternate Charlie” (military alphabet), SendKeys “{Alt+c}” will appear in the Advanced Scripting box.

SendKeys Alternate < a - z, 0 - 9, **or** F1 - F12 > < 1 - 20 >

SendKeys Control < a - z, 0 - 9, **or** F1 - F12 >

SendKeys Control < a - z, 0 - 9, **or** F1 - F12 > < 1 - 20 >

SendKeys Shift < a - z, 0 - 9, **or** F1 - F12 >

SendKeys Alternate Control < a - z, 0 - 9, **or** F1 - F12 >

SendKeys Control Alternate < a - z, 0 - 9, **or** F1 - F12 >

SendKeys Alternate Shift < a - z, 0 - 9, **or** F1 - F12 >

SendKeys Control Shift < a - z, 0 - 9, **or** F1 - F12 >

SendKeys Alternate Control Shift < a - z, 0 - 9, **or** F1 - F12 >

SendKeys Backspace - types SendKeys “{Backspace}”

SendKeys Delete - types SendKeys “{Del}”

SendKeys Down - types SendKeys “{Down}”

SendKeys End - types SendKeys “{End}”

SendKeys Enter - types SendKeys “{Enter}”

SendKeys Escape - types SendKeys “{Esc}”

SendKeys Home - types SendKeys “{Home}”

SendKeys Left - types SendKeys “{Left}”

SendKeys Page Down - types `SendKeys "{PgDn}"`
SendKeys Page Up - types `SendKeys "{PgUp}"`
SendKeys Right - types `SendKeys "{Right}"`
SendKeys Space - types `SendKeys "{Space}"`
SendKeys Shift Tab - types `SendKeys "{Shift+Tab}"`
SendKeys Tab - types `SendKeys "{Tab}"`
SendKeys Up - types `SendKeys "{Up}"`

Send System Keys - types `SendSystemKeys "|"`
Example: `SendSystemKeys "{Ctrl+Esc}"` triggers the *Start Menu* command.

SendKeys - types `SendKeys "|"`
Example: `SendKeys "{Ctrl+p}"` triggers the global *Print* command.

MessageBox - types `MsgBoxConfirm "|",52,"Warning!"`
This command offers a "Yes" or "No" choice, before executing the command.
The upright line "|" represents the cursor which shows *where to place the message* that you want NaturallySpeaking to visually display before executing the command.

Read Text - types `TTSPlayString "|"`
This command allows you to insert a message into the *highlighted* text field. The upright line "|" represents the cursor which shows *where to place the message* that you want NaturallySpeaking to read. NatSpeak will then read the message.

Play Sound - types `PlaySound "|.wav"`
This command plays a .wav file (digital sound recording).
Example: `PlaySound "C:\My Documents\Thunder.wav"` will play a "Thunder" .wav file in your "My Documents" folder.

Control Pick - types `ControlPick "|"`
This command presses the "Control" key. "|" shows *where to type the name of the Control button* you want pressed.
Example: `ControlPick "OK"` will click on the "OK" button in the control window.

The following Control Pick commands incorporate some of the more common "Control Pick" functions:

Control Pick Cancel - types `ControlPick "Cancel"`
Control Pick Done - types `ControlPick "Done"`
Control Pick Finish - types `ControlPick "Finish"`
Control Pick No - types `ControlPick "No"`
Control Pick No To All - types `ControlPick "No To All"`
Control Pick OK - types `ControlPick "OK"`
Control Pick Yes - types `ControlPick "Yes"`
Control Pick Yes To All - types `ControlPick "Yes To All"`

Menu Pick - types `MenuPick "|"`
This command selects a menu item. The highlighted word "|" shows *where to type the name of the Menu item* you want selected.
Example: `MenuPick "Save"` will click on the "Save" menu item under the "File" menu in NaturallySpeaking.

App Bring Up - types **AppBringUp** “|”

This command launches (or brings forward) the specified application.

Example: **AppBringUp** “Winword” will launch (or bring forward) Microsoft Word.

Application Bring Up - same as App Bring Up

App Swap With - types **AppSwapWith** “|”

This command launches (or brings forward) the specified application.

Example: **AppBringUp** “IExplorer” will launch (or bring forward) Internet Explorer.

Application Swap With - same as App Swap With

Note: This command is similar to the “Application Start” command with one notable exception: It will not trigger an additional instance of a program. AppSwapWith prevents multiple copies from occurring. Internet Explorer is one such example.

Go To Sleep State - types **GoToSleep**

Places NatSpeak in sleep mode

Wake Up State - types **WakeUp**

Wakes up the microphone

Microphone Off State - types **SetMicrophone 0**

Turns the microphone off

Microphone On State - types **SetMicrophone 1**

Turns the microphone on

Clear Desktop State - types **ClearDesktop**

Minimizes all open windows

Application Argument <1To3> - types **if _arg(1, 2 or 3) = "" then AppBringUp ""**

SendKeys Argument <1To3> - types **if _arg(1, 2 or 3) = "" then SendKeys ""**

Argument <1To3> same as SendKeys Argument <1To3>

Next Word – Moves the cursor to the next pair of quotes.

Example: If you say **Next Word** when the cursor is here:

HeardWord “Change|”, “”

It will move to here:

HeardWord “Change”, “|”

Finish Command - saves the command, closes out of the KB Command Browser and minimizes KB 2008

Save Command – copies the entire macro to the clipboard (for possible future use), saves the command, leaves the Add Command Window open and moves the cursor to the **Name** field

Paste Command - when deployed from within the Name field (see previous [Save Command](#) macro) will move the cursor to the script box and paste a copy of the previous script.

Control Click – Presses {Ctrl+Left Mouse Click}

ShiftKey 2,1
ButtonClick 1,1
ShiftKey 2,0

Shift Click – Presses {Shift+Left Mouse Click}

ShiftKey 2,2
ButtonClick 1,1
ShiftKey 2,0

Repeating Scripts

SendKeys Backspace < 1 - 20 >

Example: If you say, “SendKeys Backspace 3”, NatSpeak will type:

SendKeys “{Backspace 3}”

The following are other “Repeating Scripts” commands:

SendKeys Delete < 1 - 20 >
SendKeys Down < 1 - 20 >
SendKeys Enter < 1 - 20 >
SendKeys Left < 1 - 20 >
SendKeys Right < 1 - 20 >
SendKeys Space < 1 - 20 >
SendKeys Shift Tab < 1 - 20 >
SendKeys Tab < 1 - 20 >
SendKeys Up < 1 - 20 >

Wait Commands

Note: Wait Commands insert pauses into the script (in order to slow down the execution of the command). To see an example of this, see the [“Check My Email”](#) global command.

Wait < Seconds > - This command tells NaturallySpeaking *how long to wait* before executing the next part of the script.

Example #1: If you say, “Wait 200” KnowBRAINER will type **Wait 200** (which is 200 milliseconds or 1/5 of a second pause).

Example #2: If you say, “Wait 500” KnowBRAINER will type **Wait 500** (1/2 second pause)

Example #3: If you say, “Wait 3000” KnowBRAINER will type **Wait 3000** (3 seconds pause)

The built-in Wait commands range from <10, 20, 30...90>, <100, 200, 250, 300, 400,

500, 600, 700, 750, 800, 900> and <1000, 2000, 3000...32,000>

Mouse Commands

Mouse Area < 1 - 9 >

This command is used to insert the “Mouse Grid” command of your choice, from 1 to 9.

Example: To type **MouseGrid 1, 5** (which selects the #5 field in the MouseGrid) say “**Mouse Area 5**”.

Mouse Position

This command is used to select precise “X” and “Y” positions.

Example: When you say “**Mouse Position**”, **SetMousePositon 1, 400, 300** is typed. The second number (400) represents the left-to-right control. The third number (300) represents the up-and-down control.

Note: Place your cursor (mouse) exactly where you want to click and say “Start Paint”. MS Paint features a Mouse position pointer in the lower right hand corner of the screen

Note: If you need to locate the exact x & y coordinates of the mouse position say **Find Mouse Position**

Left Click State - types **ButtonClick 1,1**, a left mouse button click

Right Click State - types **ButtonClick 2,1**, a right mouse button click

Double Click State - types **ButtonClick 1,2**, a double mouse left button click

Heard Word Commands

Heard 2 Words - types **HeardWord “|”, “”**

Example #1: If you say “Heard 2 Words”, NaturallySpeaking will type: **HeardWord “|”, “”** placing the cursor between the first and second quotes.

Example #2: To create a new global print command, try the following:

1. Say, “**New Global Command**”
2. Dictate a name for the command, such as: “print this now”
3. Say, “**Script Heard 2 Words**” (which will place the cursor in the Advanced Script box and type **HeardWord “|”, “”**)
4. Type the Word “print” within the first set of quotes, say ‘Next Word’ and then type “document” within the last set of quotes. It should look like this:

HeardWord “Print”, “Document”

Note: HeardWord commands eliminate the need to retype previously created commands. Rather than having to create a “Print This Now” command from scratch, the “HeardWord” command will simply execute the script you have written into the “Print Document” command. This is a major time saver when creating several commands that all perform the same function. Example: “Print Document”, which is the same as “Print File”.

This command is also commonly used to combine previously written commands.

Example: “Surf the Net”.

Heard 3 Words - types **HeardWord “|”, “”, “”**

Heard 4 Words - types **HeardWord “|”, “”, “”, “”**

Heard 5 Words - types **HeardWord “|”, “”, “”, “”, “”**

Heard Word All Caps - types **HeardWord “\All-Caps”**, the “All Caps” command, which will completely capitalize your dictated text.

Heard Word All Caps Off - types **HeardWord “\All-Caps-Off”**, the “All Caps Off”

command, which will turn the “All-Caps” command off.

Heard Word All Caps On - types **HeardWord** “\All-Caps-On”, the “All Caps On” command, which will turn the “All-Caps” command on.

Heard Word Cap - types **HeardWord** “\Cap”, the “Cap” command, which will capitalize your next word.

Heard Word Caps Off - types **HeardWord** “\Caps-Off”, the “Caps Off” command, which will restore future text to regular case.

Heard Word Caps On - types **HeardWord** “\Caps-On”, the “Caps On” command, which will capitalize the first letter of each word of your dictated text.

Example: **HeardWord** “\Caps-On” An example of this command can be found in the “Rename Folder” global command. When you say, “Rename Folder” it creates Initial Caps in your dictation.

Heard Word No Caps - types **HeardWord** “\No-Caps”

Heard Word No Caps On - types **HeardWord** “\No-Caps-On”

Heard Word No Space Off - types **HeardWord** “\No-Space-Off”

Heard Word No Space On - types **HeardWord** “\No-Space-On”

Heard Word Start - types **HeardWord** “start”, “|” which will launch whatever program that you insert between the last pair of quotes.

Example: **HeardWord** “start”, “outlook” will launch Outlook

Heard Word Select - types **HeardWord** “select”, “|” which will select the first word or phrase, you insert between the last pair of quotes.

Heard Word Insert Before - types **HeardWord** “insert before”, “|” which will insert the cursor before the first word or phrase, you insert between the last pair of quotes.

Heard Word Insert After - types **HeardWord** “insert after”, “|” which will insert the cursor after the first word or phrase, you insert between the last pair of quotes.

Looping Commands

Loop - This command, used for looping a part of a command, types the following:

```
I = 1  
DO UNTIL I = 0  
I = I - 1  
Loop
```

Note #1: The cursor is placed to the right of the ‘0’ until you dictate the 1st line of code. The next command line dictated will insert itself into the middle of the Loop command.

*Note #2: After you have finished writing the looping script, say “End Of Document” to move the cursor to the end of the loop, where you may continue dictating lines of code. For additional information see **Example 3** on page 16*

Loop < 1 - 50 > - This command allows you to loop a command (or partial command) up to 50 times.

Note: There is really no upper limit imposed upon this command. Should you need to loop something more than 50 times, simply say “Loop 50”, then change the “50” to the required number.

Example: If you wanted the phrase, “I love Spam” to be typed 49 times, one under the other, you would create a looping command that looks like this:

```
I = 49
```

```
DO UNTIL I = 0
SendKeys "I Love Spam {Enter}"
I = I - 1
Loop
```

Argument (Listing) Commands
(Used to create or edit Listing Commands)

Argument (Listing) Commands
(Used to create Listing Commands)

Application Argument <1To3> types `If _arg<1-3> = "" then AppBringUp ""`

An example of this command is the "Start Up <AppList>" global command.

Note: Argument commands consist of 2 parts. The argument and the List. For more information see [How to Edit a Listing Command](#).

SendKeys Argument <1To3> - types `If _arg<1-3> = "" then SendKeys ""`

Argument <1To3> - types `If _arg<1-3> = "" then SendKeys ""`

Explanation of Listing Commands

Listing commands are created in 2 steps:

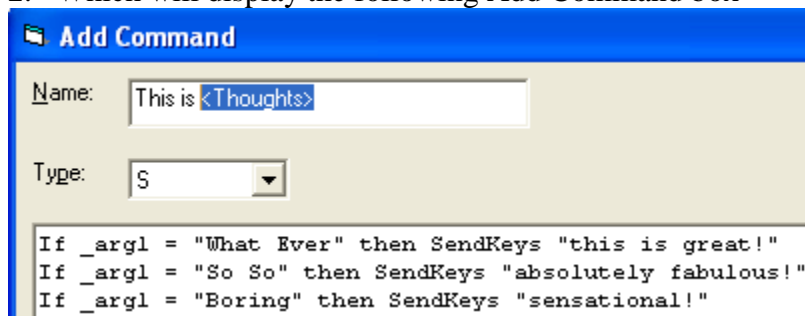
Step#1

1. The first step of the command is created by clicking the Add button



which will display the following Add Command box

2. Which will display the following Add Command box

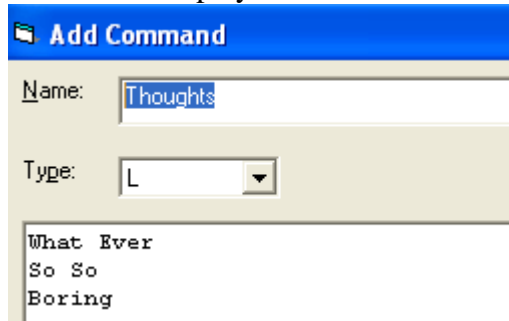


3. The name of the command goes in the Name field, the type of command defaults to "S", for script, and the code goes in the lower box. The <Thoughts> part of the name field will have to be created as the second part of command in Step #2

Step#2

1. After the command script is finished you have to make an additional listing command that includes the words (lists) that you typed in between the first set of quotes.
2. After clicking the add button you will need to type the word

“Thoughts” in the name field, select “L” to denote that this is a list to support the previous command and type the name lists on individual lines as displayed below



3. For additional information on editing and creating listing commands see [How to Edit a Listing Command](#)

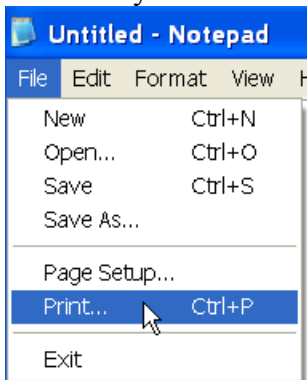
Example Commands

The following examples should have you up and running in record time. You may want to print this section of the VerbalBasic manual. Begin with Example 1; then take them in order. If taken out of order, some of these commands may not work properly.

Create the following commands in Windows Notepad, beginning with a simple print command:

Example 1

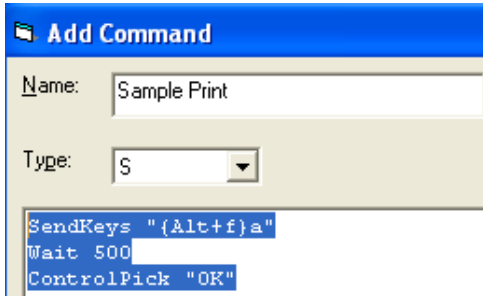
1. Begin by saying “[Start Notepad](#) *or* [Switch to Notepad](#)”
2. Say “**File**”. You should see the following:



3. Say “[New Notepad Command](#)” to open the KB Command Browser, select Notepad_kb (Notepad application-specific commands) and click the **Add** button
4. Say “[Sample Print Command](#)” to name your command and fill in the Name field. *Note: Initial capitalization is automatic.*



5. Say “**Send Keys Alternate F**” ” (or “**SendKeys Alternate Foxtrot**”, if you know your ICD alphabet) to move the cursor from the Name field and type **SendKeys “{Alt+f}”** in the Scripting field.
6. Say “**P**” (or “**Papa**”, if you know your ICD alphabet).
7. Say “**Wait 500**” to move the cursor to the next line (even when it is in the middle of the line) and type **Wait 500** (which is a half second pause)
8. Say “**Control Pick OK**” to type **ControlPick “OK”** on the next line (which mouse clicks the **OK** button. The command should look like the following:



9. Say “**Save Command**” or press the **Save** button to save your new command. Don't close the window because you will need it for the next Example Command.

This command will print the current document.

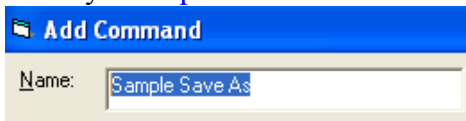
When you say “**Save Command**”, the following should occur:

- (1) The entire command will be copied to the clipboard, for possible future use.
- (2) The **Save** button will be pressed, to compile the command.
- (3) The Add Command window will remain open and the cursor will be placed in the **Name** field so that you can quickly dictate your next Notepad application-specific command.

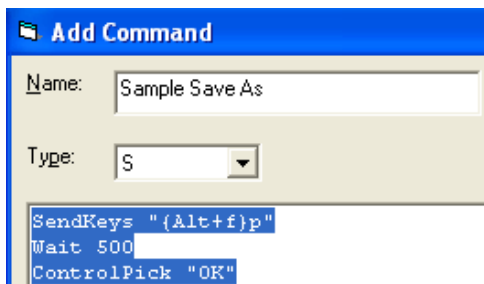
Note: Remember that the previous command script is still copied to the clipboard and you may want to paste it into the script box if your next command is similar to your previous one. This can be a major timesaver.

Example 2

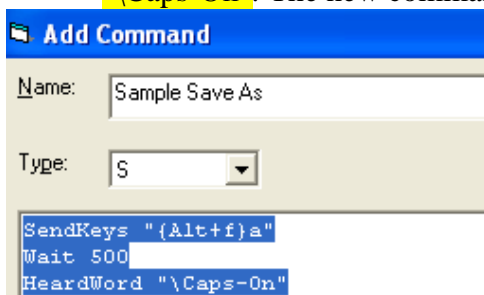
1. Say “**Sample Save As**” to name the next application-specific Notepad command.



2. Say “**Paste Command**” to move the cursor to the script field and paste a copy of the script from the previous command and highlight the macro (in case you need to rewrite the command)



3. Change the “p” in the first line to an “a”, delete the last line, move the cursor the end of the Wait 500 line and say “**Heard Word Caps On**” which will type **HeardWord** “\Caps-On”. The new command will look like the following:

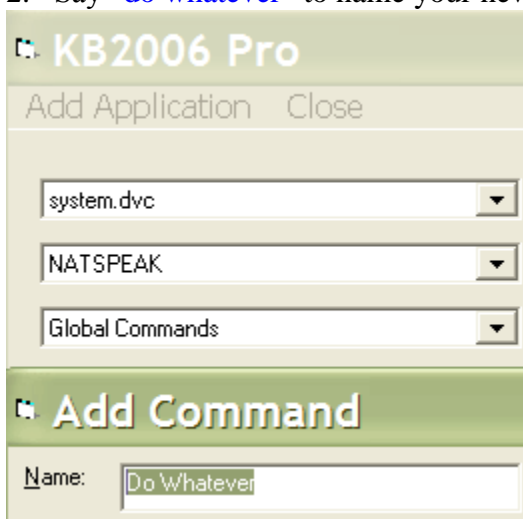


This command will open the Save As dialog and force each dictated word to appear in Title Case.

4. Say “**Finish Command**” to Save the command, close out of the KB Command Browser and minimize KB 2008

Example 3

1. Say “**New Global Command**” to open the KB Command Browser, select system_kb (Global commands) and click the **Add** button
2. Say “do whatever” to name your new command. You should see the following:



3. Say “**Read Text**” to move the cursor to the script field, type **TTSPPlayString "|"** and place the cursor in between the 2 quotes so that you can dictate or type a message for NaturallySpeaking to verbalize

- Dictate “are you sure” between the quotes
- Say “Message Box” to type `MsgBoxConfirm "|? ",52, "Warning!"` on following line, move the cursor between the first quote and question mark and turn on initial capitalization so that your dictation will appear in Title Case (for appearances).

NaturallySpeaking will visually display what you have typed into the `Text` field.

- Dictate “Are You Sure” which will automatically appear in Title Case
- Say “Loop 5” which will display the following in place the cursor at the end of the `DO UNTIL I = 0` line

```
TTSPlayString "are your sure"
MsgBoxConfirm "Are You Sure? ",52, "Warning!"
I = 5
DO UNTIL I = 0|
I = I -1
Loop
```

- Say “Send Keys” which will insert SendKeys “|”
- Say “yes” to insert the word “yes” between the quotes
- Say “Enter” to add and {Enter} Key to your command. Your command script should look like the following:

```
TTSPlayString "are your sure"
MsgBoxConfirm "Are You Sure? ",52, "Warning!"
I = 5
DO UNTIL I = 0
SendKeys "yes{Enter}"
I = I -1
Loop
```

This command will do the following

`TTSPlayString "are your sure"` verbally asks you if you want to deploy the command `MsgBoxConfirm "Are You Sure? ",52, "Warning!"` displays a visual **Yes** or **No** message asking if you want to deploy the command. The command will not deploy unless you say “Yes”.

`I = 5`
`DO UNTIL I = 0`
`SendKeys "yes{Enter}"`
`I = I -1`

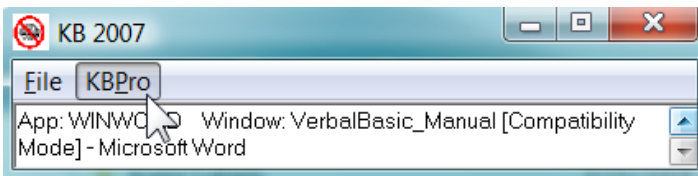
Loop types the word “yes” on 5 consecutive lines

- Say “Finish Command” to compile the script in close out of the KB Command Browser. If you wish to edit or delete this command said “Edit Global Command”, scroll to the “Do Whatever” command in press or say **Delete**

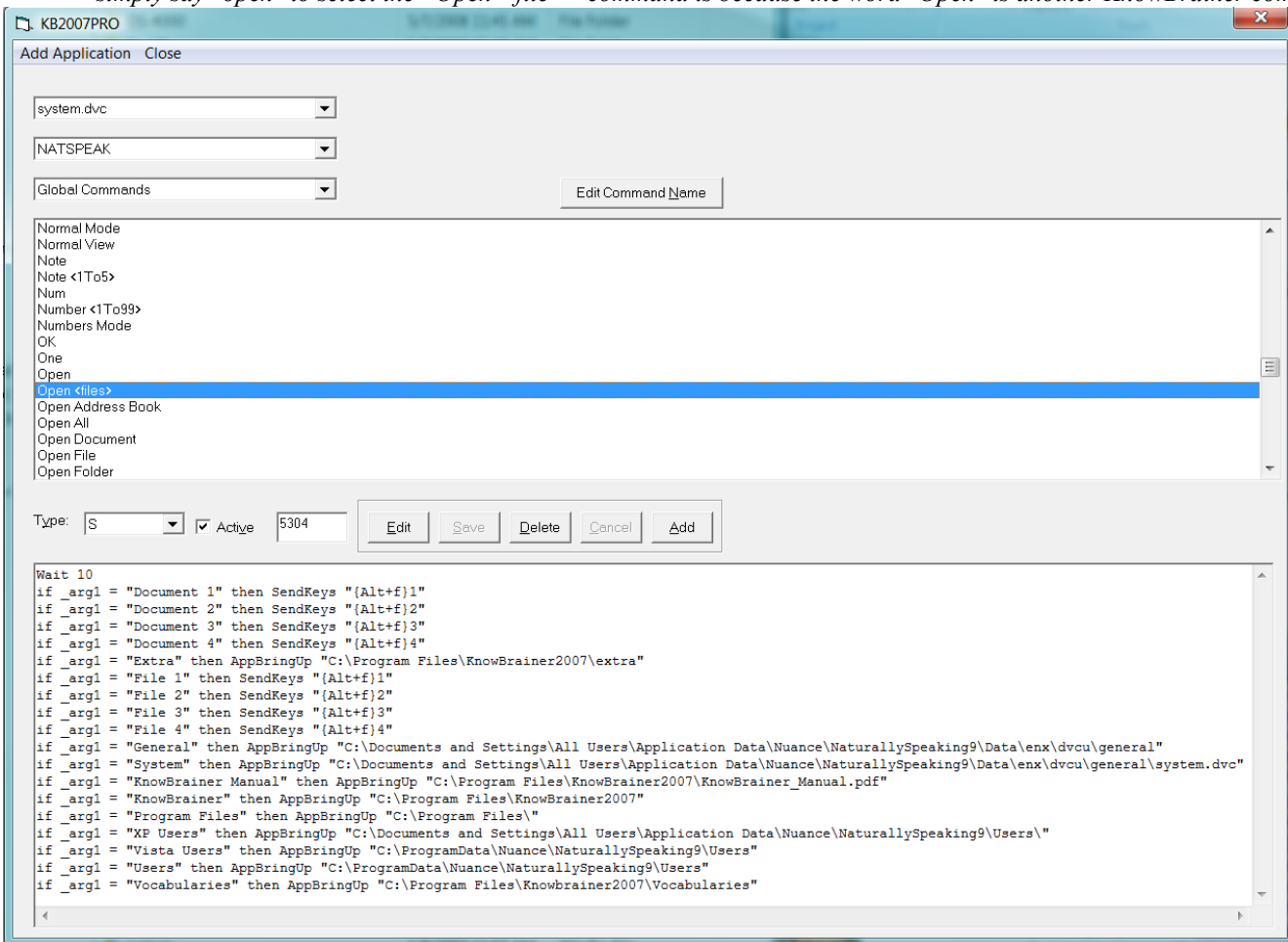
How to Edit a Listing Command

The following 9 steps define the inner workings of the KnowBRAINER “Open <file>” command.

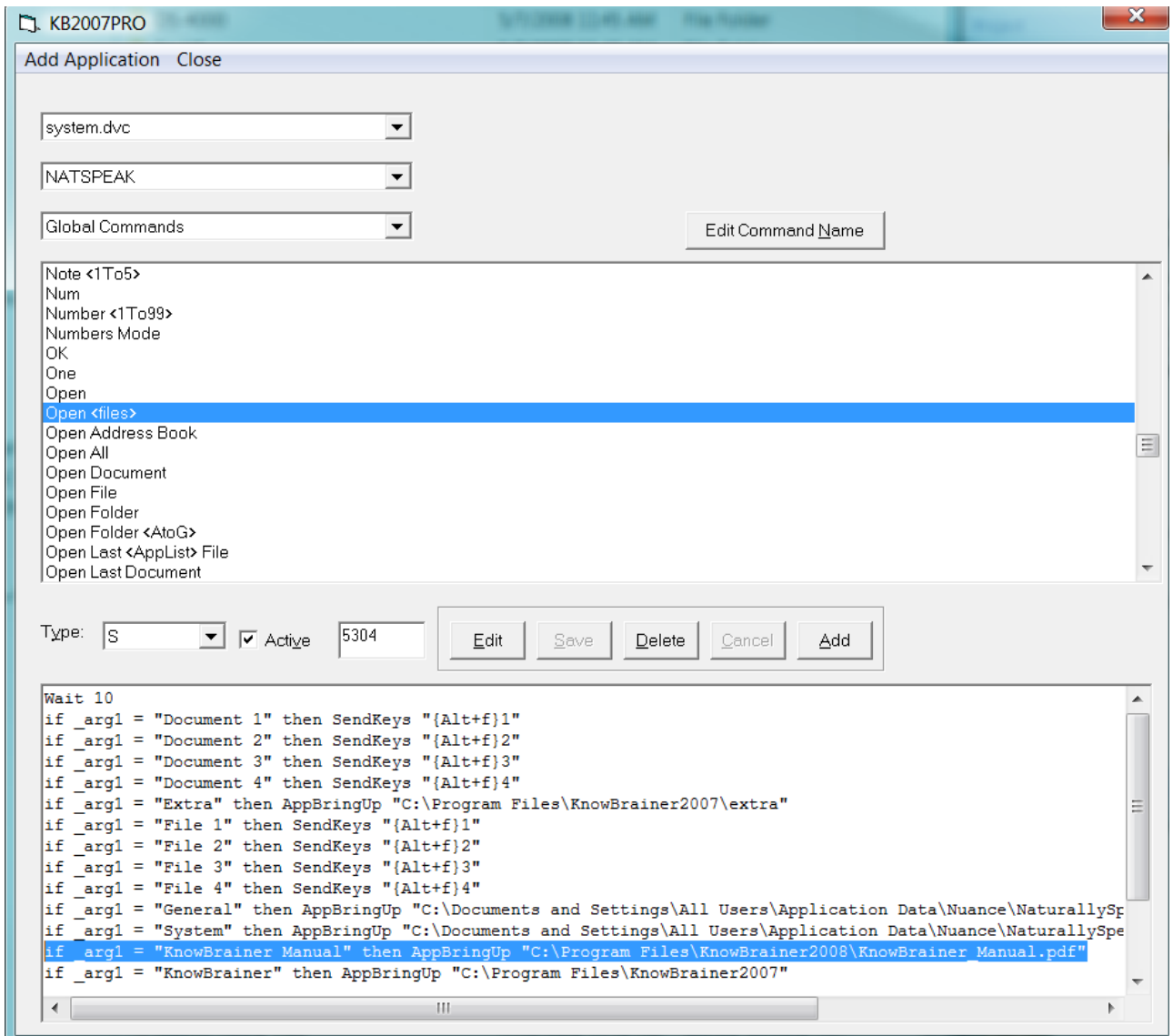
- Say “Edit Global Command” or press {Alt+p} in KB2008.



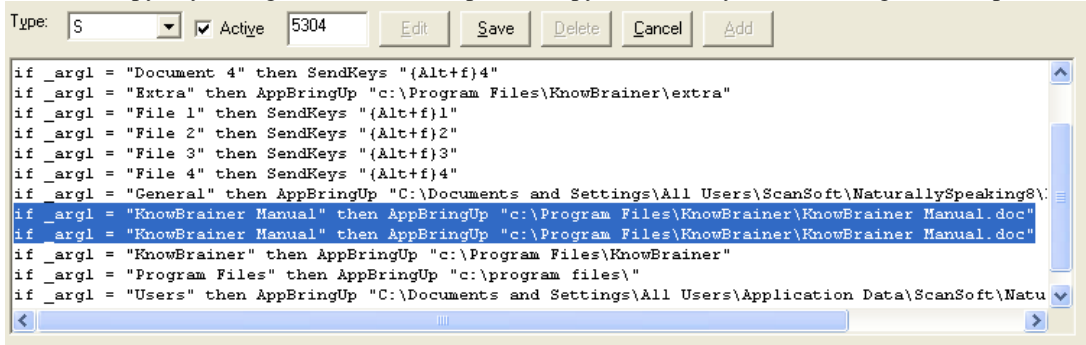
2. Say “**Cap Open**” followed by “**Down 1**” to highlight the “Open <file>” command. *Note: The reason why you cannot simply say "open" to select the “Open <file>” command is because the word "Open" is another KnowBrainer command.*



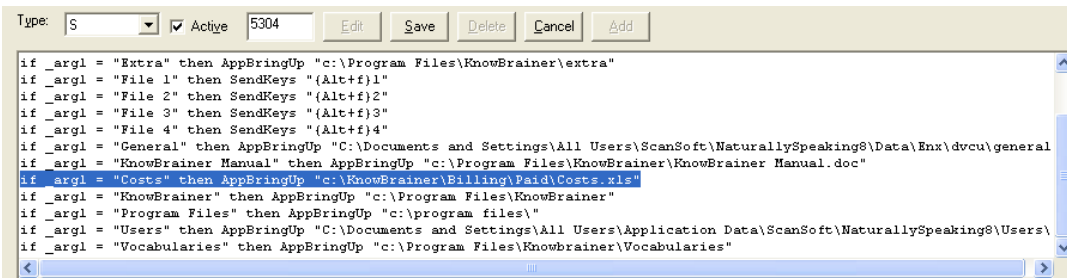
3. Say or Press “**Edit**” to open the command for editing. *Note the highlighted text in the following picture.* If you want to add a specific file or folder address you may do so at this time. Be aware that if you are adding a specific document to the “Open <file>” command, such as Microsoft Word document, you will additionally need to add “.doc” (minus the quotes) so that NaturallySpeaking will understand what type of file it is. This step is only required with files and programs (.exe). Folder addresses do not require this step.



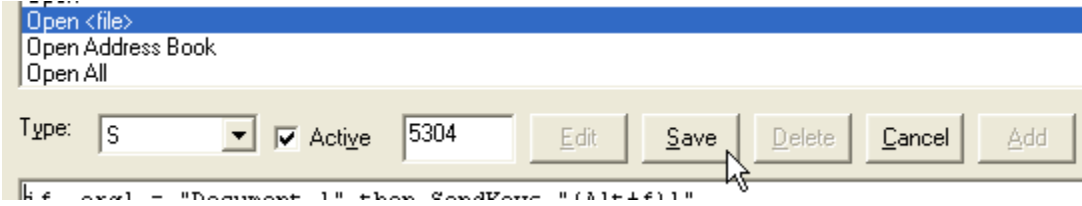
4. Copy any `if _arg1` entire line and paste a copy of it directly under the original as depicted below:



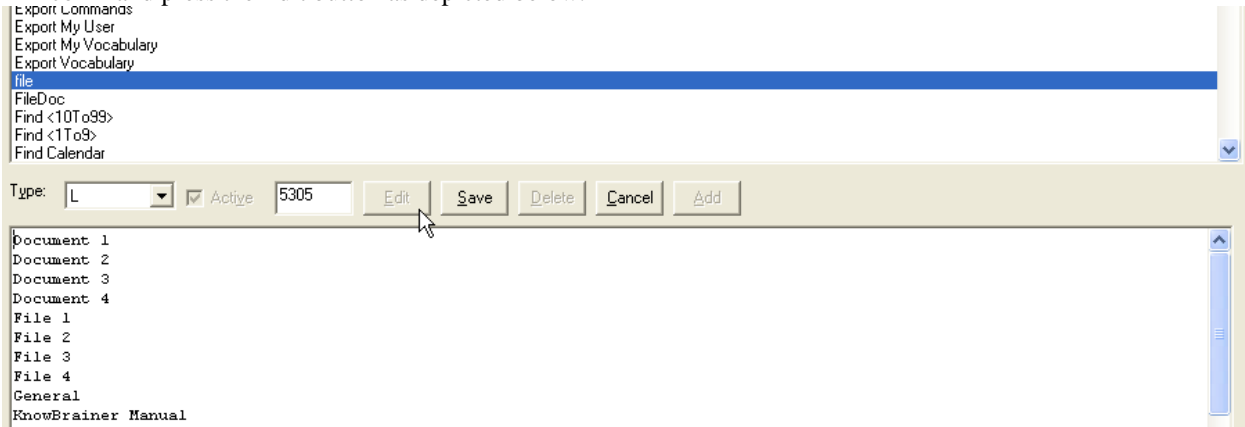
5. The example below contains a Microsoft Excel spreadsheet file by the name of “Costs”. You will need to change both the name, which appears between the quotes immediately following `if _arg1 =` and the address of the file. **Costs** is the file name and the address of the file is `c:\KnowBrainer\BillingPaid\Costs.xls`. Don’t forget to include the extension when typing the addresses for files and programs (.exe). Folders do not require an extension.



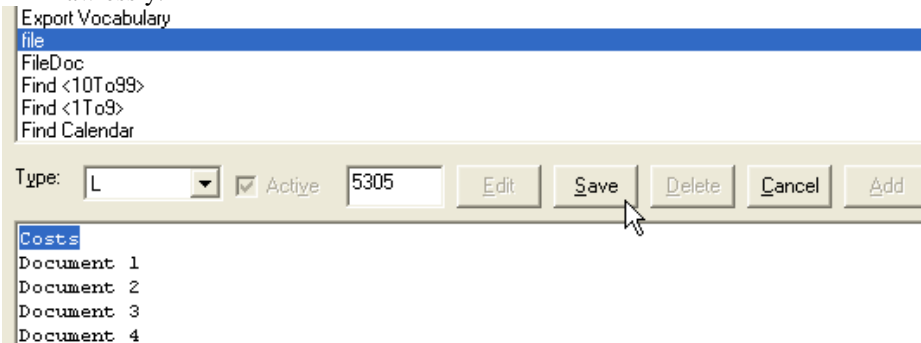
6. Just before you press the **Save** button but note the **<file>** part of the command name (highlighted in blue)



7. Scroll through the command list until you locate the word file which is the listing part of the Open <file> command. Listing commands are broken into 2 parts and are depicted by a name surrounded by <>. After highlighting the file command press the Edit button as depicted below:



8. Type the name of your file, which in this case is **Costs** on a separate line. It doesn't matter what line you put the name **Costs** on but we will choose the first line for this example. Press the **Save** button and you are done. Of course our example won't work in your computer but if you substitute the name and address of 1 of your own files, it should work flawlessly.

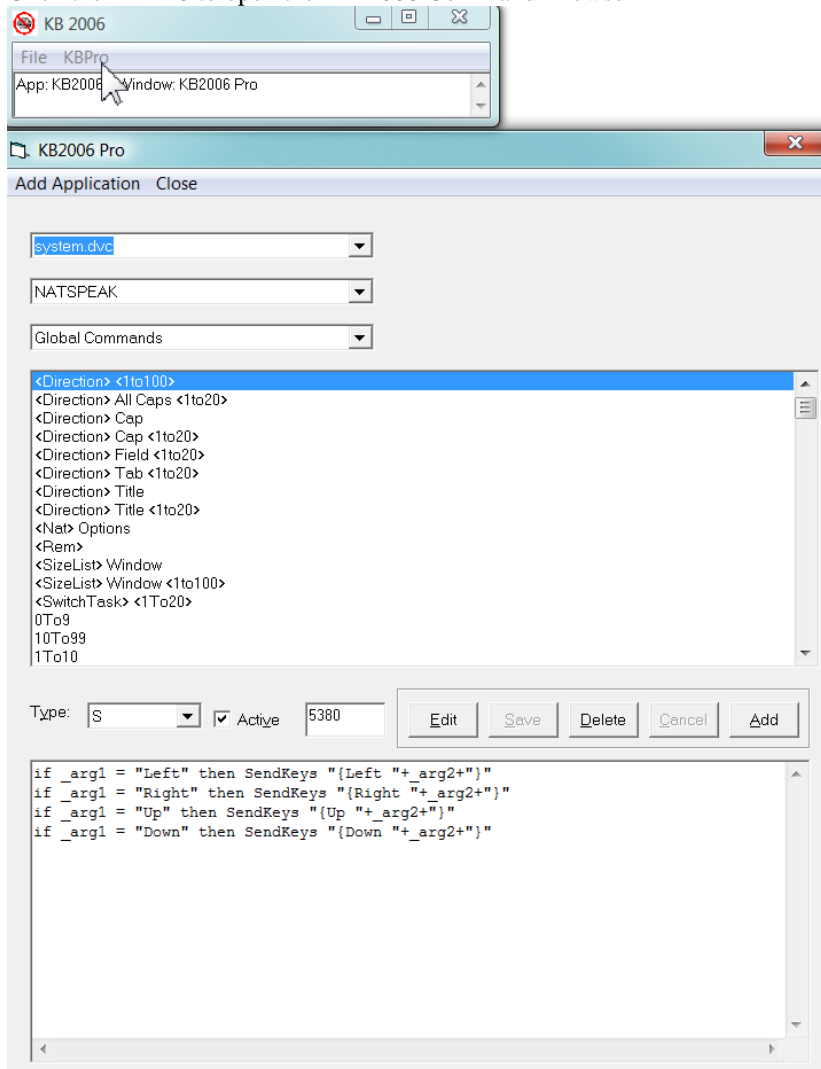


If you require additional assistance with command writing you can freely post your questions on the KnowBrainer Technical Support Phorum at ~ <http://forum.knowbrainer.com> or contact your KnowBrainer vendor who may charge for their services.

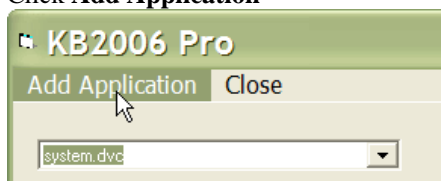
Creating Commands in Programs that KnowBrainer Doesn't Currently Support

KnowBrainer 2008 includes a new feature that will let you create a command for any application that KnowBrainer doesn't already include macros for:

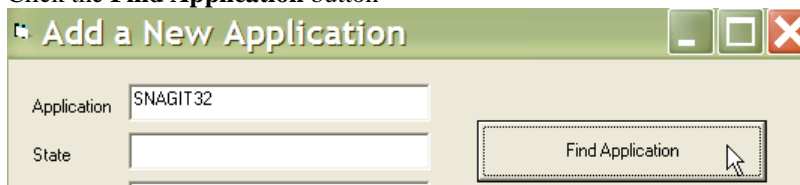
1. Click the **KBPro** to open the KB 2008 Command Browser



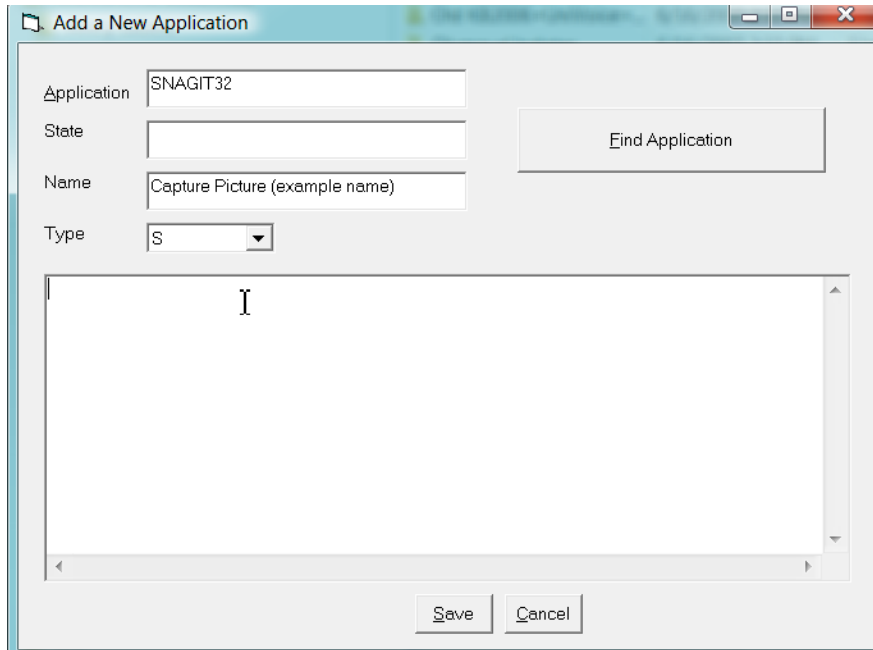
2. Click **Add Application**



3. Click the **Find Application** button



4. Click the **Find Application** button and quickly move the cursor back to the application of your choice and KnowBrainer will add the program to the **Application** field. In the following example we clicked on Snagit



5. Dictate the name of your command and dictate the code in the large box.
6. Save your command by saying Save Command which will save you steps because it will save the command, leave the command box open and copy the command code to the clipboard.
7. To create the next command simply dictate the name of the command (which will automatically appear in Title Case) followed by saying say **Paste Script** which will paste the script of the previous command into the box so that you can make a minor changes rather than being forced to type or dictate the entire next script