Windows™ Speech Recognition Toolkit

A Speech Recognition Utility © 2021 MyMSSpeech.com

Introduction

Thank you for your interest in the *Windows™ Speech Recognition Toolkit*. We hope this product will serve you well and that you will have many rewarding hours of hands-free computing as a result of using its features.

The Windows™ Speech Recognition Toolkit was developed by eMicrophones, Inc.
which has 15 years experience in the Speech Recognition industry. Since previewing Windows® Speech Recognition at Redmond in the summer of 2006 we thought very highly of the accuracy of the speech engine, the ability to command and control one's computer and the forethought given to the graphical user interface. However, we realized some important features typical in other Speech Recognition software was missing. The WSR Toolkit is meant to fill in the gaps and make a typical home or business user more efficient.

The Text Macro Builder easily allows you to create re-usable boilerplate text. The text is inserted when you issue the name you called the text block when creating it.

Three other features improve the accuracy of your personal user speech profile. These are:

1. Train From Text - improves the "acoustic model" of your speech profile.
It allows you to train from text of your choosing. We recommend you begin with "The Rainbow Passage" default text as it has an even distribution of the phonemes of the English language.

2. Add To Dictionary - Adds words to your personal speech dictionary (with or without pronunciation) for increased recognition accuracy.

3. Add From File - improves the "language model" of your speech profile by taking the text you pass it (currently Word documents and text files), parses it into words, and then records the relative frequencies of occurrence of each word with respect to the words around it.

Professionals on the run requested a Transcription tool to transcribe .wav files from a high quality digital recorder into text. With our Transcription tool, point to a .wav file (some recorders must convert from .dss files to .wav) for transcribing into text easily copied/pasted into another document.

Table of Contents

Introduction 2
Table of Contents 3
Installing the WSR Toolkit 4
Text Macros 5
Command Macros 8
Macro Editor 16
Train From Text 22
Add To Dictionary 23
Transcription 25
Add From File 26
Tips & Tricks 28
Index 29

Installing the WSR Toolkit

After downloading the install file and executing it, you’ll be prompted for an install directory. Once selected, the install process will begin...

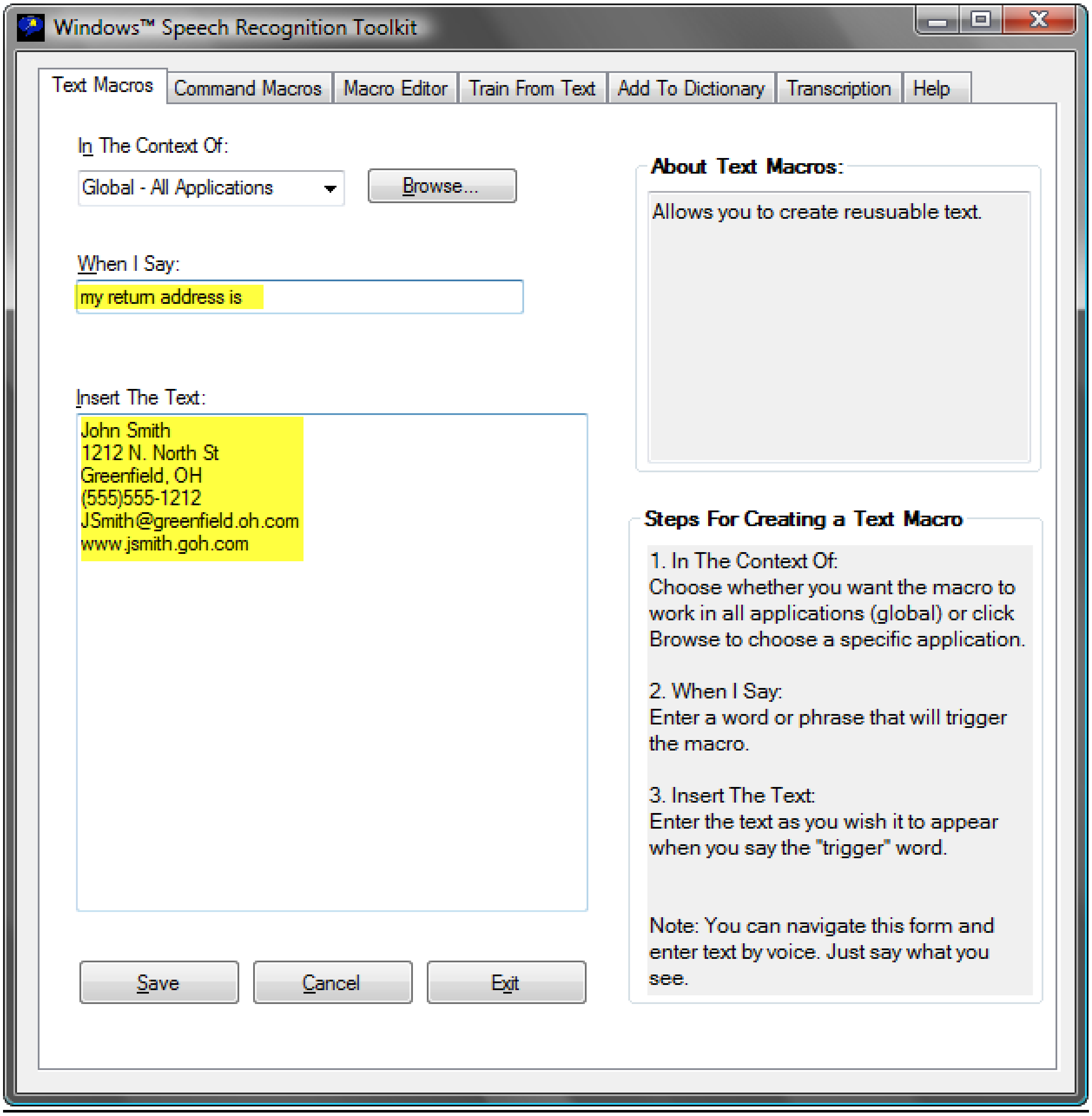
The Toolkit installation checks your machine for the Microsoft .NET 3.5 Framework (or greater) and if it’s not installed, it will ask your permission to install. After installation, there will be three shortcuts on your desktop to the WSR Toolkit application, the user’s manual and the readme.txt file.

Once the WSR Macro facility is installed and running, you'll want to set macro security level to 'low' as follows:

1. Right-click on the Windows Speech Recognition Macros icon in your system tray (bottom right corner of your computer screen)
2. Click 'Security' then 'Set security Level'. Choose 'Low'.
   This is an acceptable option while you become familiar with the WSR Toolkit and develop your macros. You can always come back to here and set the security level to 'high' as well as digitally sign your macros once you gain more familiarity with the macro system. You may decide it is desirable to leave the security level set to low.

Text Macros

On this tab you'll be able to create macros that insert text into a document or other application. Take a look at the screenshot for this tab and then we'll walk through the creation of a typical Text Macro.



In The Context Of: Global - All Applications

When I Say: my return address is

Insert The Text:

John Smith
1212 N. North St
Greenfield, OH
(555)555-1212
JSmith@greenfield.oh.com
www.jsmith.goh.com

About Text Macros:

Allows you to create reusuable text.

Steps For Creating a Text Macro

1. In The Context Of:
Choose whether you want the macro to work in all applications (global) or click Browse to choose a specific application.

2. When I Say:
Enter a word or phrase that will trigger the macro.

3. Insert The Text:
Enter the text as you wish it to appear when you say the "trigger" word.

Note: You can navigate this form and enter text by voice. Just say what you see.

(Assuming Speech Recognition is active and listening...)

You say:
When I say

[The cursor moves to the ‘When I Say’ text box area.]

(Notice the letter W is underlined in When I say. That means you can press and hold down the ALT then press the w and the cursor will move to this location. In fact, anytime you see an underlined letter you can hold down the ALT key while pressing that letter and you’ll move the cursor to that area.)

Once the cursor is in this textbox you can say (or type) your “trigger word”. That is, the word you want the Speech Macros system to listen for.

You say:
my return address is

[the text “my return address is” appears in the ‘When I Say’ textbox]

If you’re speaking to this program and not typing, you may sometime encounter recognition errors; for instance, instead of “my return address” it might think you said something like “my return add rest is”. No problem, just say “correct add rest” and you’ll be presented with choices that sound similar to what you’ve just said. All you have to do now is say the number that corresponds to the correct representation. If it still misunderstands you, say it again, if the recognition is still off, just say spell it and you can spell it letter-by-letter.

You say:
Insert the text

[The cursor moves to the ‘Insert The Text’ text box area.]

You say:
John Smith press enter
1 2 1 2 N period north street press enter
Greenfield comma o h press enter
open parenthesis 5 5 5 close parenthesis 5 5 5 hyphen 1 2 1 2 press enter
j smith at sign greenfield dot o h dot com press enter
w w w dot j smith dot g o h dot com press enter

[The text will be inserted as you talk (or type)]

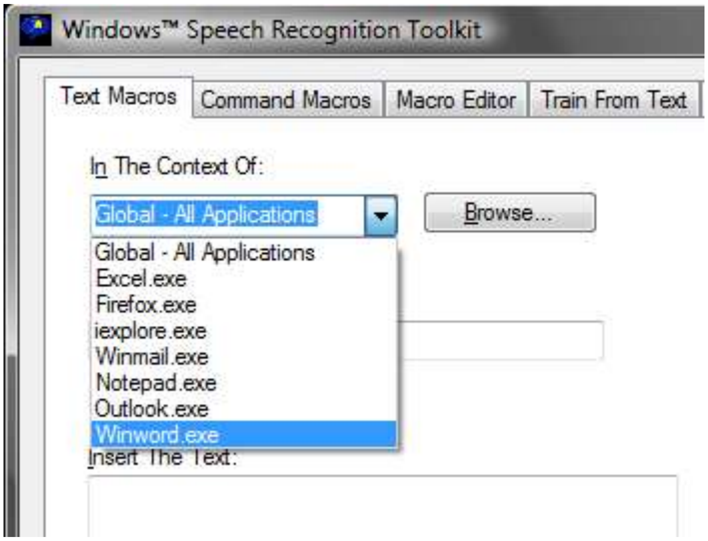
You may have to experiment a bit to get the proper recognition.

Now let’s backup to the ‘In The Context Of’ textbox.

You say:
In the context of

[The cursor moves to that dropdown box and several options are displayed]

Here's something similar to what you'll see:



These are some of the more common applications you may be using. Before deciding whether you'll need to select one of these, Browse to find another or go with the default 'Global' option, some explanation on what is meant by "context" is in order.

Let's say you'd like to create a macro for Microsoft Word in which you say "back" and it will automatically move you to the previous paragraph. But what if you'd also like to create a macro in Firefox when you say "back" you'll emulate clicking the back button? You can see the need for context. "Back" could possibly mean different things in the context of different programs. So what is the 'Global' context? The name pretty much explains it. With 'Global - All Applications' selected your macro will work in all applications. In other words, the 'In The Context Of' limits the functionality of your macro to a specific application.

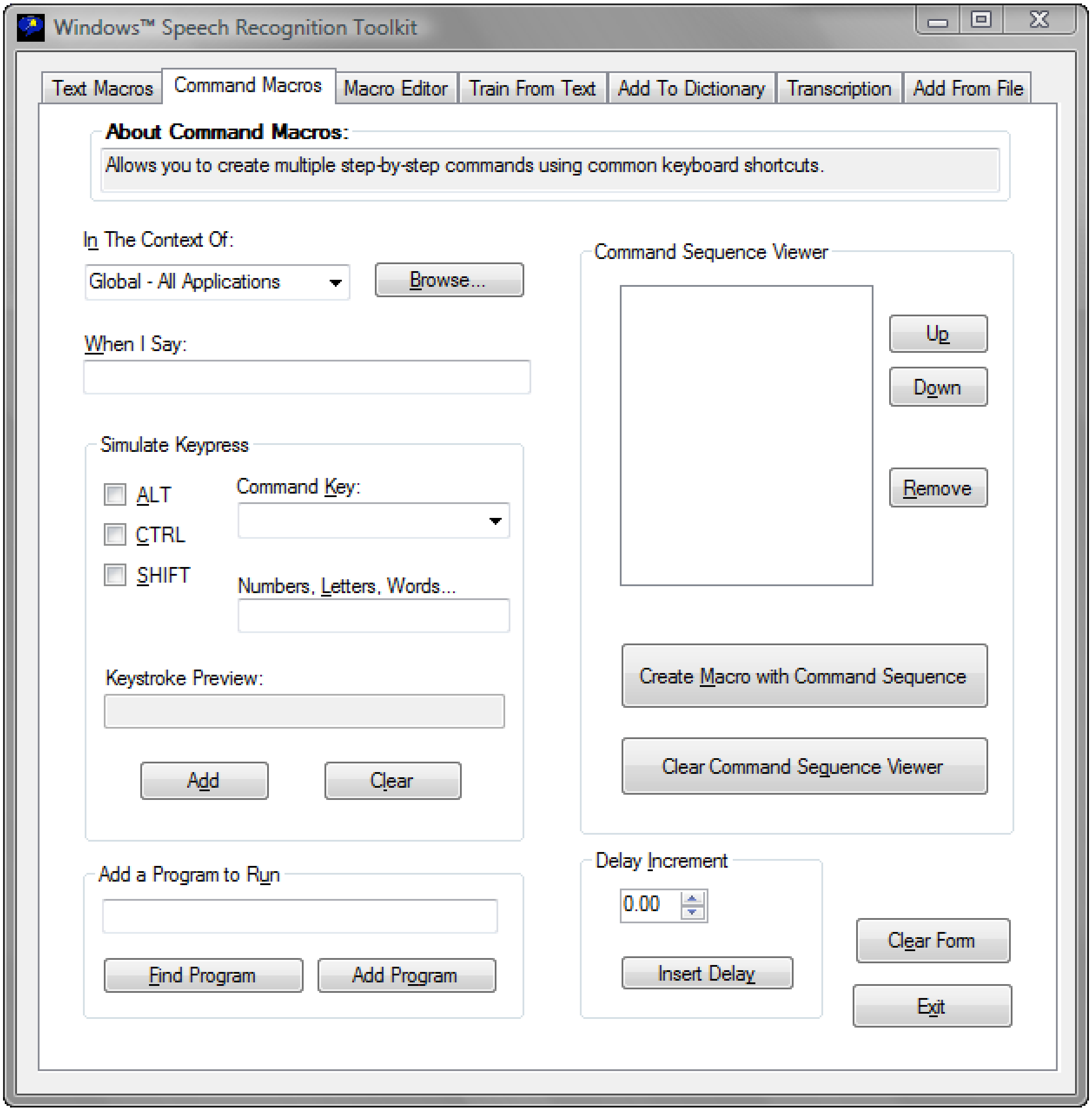
So now the question is, do you need to limit your macro to a particular application? If so, and the application you'd like to confine your macro to is listed in the list, just say the applications name; "Winword" for Microsoft Word, for instance. If it's not listed in the dropdown list, just say (or click) 'Browse'. Did you notice the underlined B in 'Browse'. As previously mentioned, that means you can also press and hold the ALT key while pressing the letter 'b'. And just for fun, you could also say "press alt b"! You'll want browse to the location of the application you'd like to choose, click on the program name and it will automatically be put in the context textbox.

If, after proofreading your Text macro, you're ready to create it, just say "Save". For reference, the 'Cancel' button allows you to clear the Text Macro tab and start fresh. The 'Exit' button will close the program.

Command Macros

This tab is possibly the most exciting of all within this application. It allows you to control your computer with voice commands that up until now have only been available with the keyboard and mouse. Don't be intimidated by all those buttons and textbox areas on this tab. We'll cover them thoroughly one-by-one. When we're done, you'll confidentially be creating sophisticated macros that do some pretty amazing stuff!

Here's the screenshot for this tab:



About Command Macros:

Allows you to create multiple step-by-step commands using common keyboard shortcuts.

In The Context Of: Global - All Applications

Delay Increment: 0.00

We've already covered the "In The Context Of" and the "When I Say" textbox areas, so if you're unsure about their use, just click the particular hyperlinked words to go there and read about them.

Let’s create a real world example (two actually) that demonstrate the many features on the Command Macros tab.

The first macro we want to create will listen for the words “surf the net” and responds by opening the Firefox browser.

We’ll keep the context global because you’ll want this macro to work no matter what you’re doing or what program you’re currently working in.

You say:
When I say

[The cursor moves to the ‘When I Say’ text box area.]

You say:
surf the net

[the text “surf the net” appears in the ‘When I Say’ textbox]

You say:
find program

[A file browsing window opens up that will allow you to point to the program you’d like to run.

You say:
mozilla firefox

[The directory will change to Mozilla Firefox - Note this may be different on your computer]

You say:
firefox

[firefox.exe is selected and the full path “C:\Program Files\Mozilla Firefox\firefox.exe” is placed in the ‘Add A Program to Run’ textbox.]

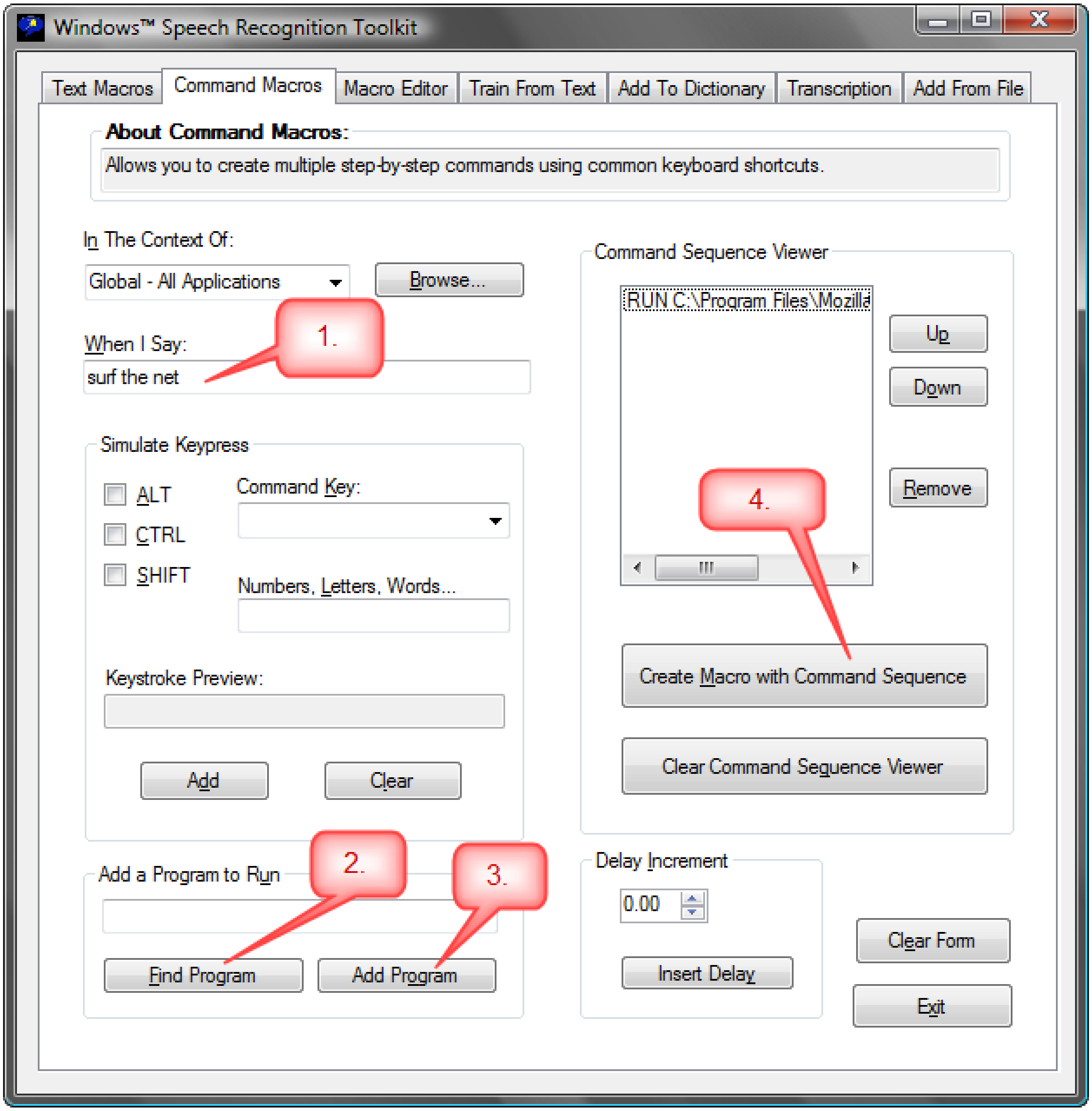
You say:
add program

[The ‘Add a Program to Run’ textbox is cleared and the ‘Command Sequence Viewer’ displays the text “RUN C:\Program Files\Mozilla Firefox\firefox.exe”.] More on the Command Sequence Viewer’ shortly...

You say:
create macro with command sequence

That’s it, you’ve created a Command Macro that will open the Firefox browser anytime you say “surf the net”! Easy as 1-2-3, right? Well, almost... ☺

Here’s a screenshot with the four different areas highlighted:



Windows™ Speech Recognition Toolkit

About Command Macros: Allows you to create multiple step-by-step commands using common keyboard shortcuts.

In The Context Of: Global - All Applications

When I Say: surf the net

Command Sequence Viewer: RUN C:\Program Files\Mozilla

Simulate Keypress: □ ALT □ CTRL □ SHIFT

Delay Increment: 0.00

Next we'll create a practical Command Macro that will only work in the context of Firefox and will do the following things when you say 'copy web address'.

1. Move the cursor to the address bar
2. Copy the web address of the current website into the clipboard

Now all you have to do is go to another application and say 'paste' and it will put that address you just copied in the application! This is nice when you'd like to email a friend a website address, for instance.

Here's how to do it:

On the Command Macros tab -

You say:
In the context of

[The dropdown box drops down revealing commonly used programs.]

You say:
firefox

[Firefox.exe is displayed in the 'In the Context Of' text area.]

You say:
alt

[The ALT checkbox will be checked]

You say:
numbers letters words

[The cursor moves to the "Numbers, Letters, Words" textbox.]

You say:
d

[The letter d is placed in the "Numbers, Letters, Words" textbox.]

The keyboard shortcut ALT-d in Firefox will move the cursor to the address bar and highlight whatever address is present.

(Notice the 'Keystroke Preview' area. This dynamically shows you the keystroke commands that are going to be built.)

You say:
add

[The ‘Keystroke Preview’ area clears and the ‘Command Sequence Viewer’ displays a new line of text.]

Now we need to insert a timing delay. This is necessary because it’s possible for the voice recognition system to emulate key presses faster than the application (in this case Firefox) can digest them. So we need to insert a negligible delay to allow the application to catch up. From experience, .25 seconds is a good delay for this type of thing.

You say:
delay increment

[The cursor moves to the ‘Delay Increment’ numerical selection area.]

Okay, now for some fancy footwork! We need to (by voice control) click the up arrow five times. That will give us a delay of .25 seconds. So you can either say “press up”, “press up”, “press up”, “press up” “press up” OR simply say “press up five times”! Pretty neat, huh?

You say:
press up five times

[The ‘Delay Increment’ shows .25]

You say:
insert delay

[The ‘Delay Increment’ area resets back to 0.00 and the ‘Command Sequence Viewer’ displays a new line of text.]

For the final sequence of our command macro, we need to emulate the keypress ctrl-c. That copies whatever is highlighted to the clipboard (so you’ll be able to paste it somewhere else).

You say:
contol

[The CTRL checkbox is checked]

You say:
numbers letters words

[The cursor moves to the “Numbers, Letters, Words” textbox.]

You say:
c

[The letter c is placed in the “Numbers, Letters, Words” textbox.]

You say:
add

[The ‘Keystroke Preview’ area clears and the ‘Command Sequence Viewer’ displays a new line of text.]

You should have three command sequences listed.

Now for a word about the ‘Command Sequence Viewer’ text area.

This is the area in which, as you build your Command Macro, allows you to essentially “preview” the sequence of commands before you actually create the macro. It’s a great tool for providing a visual double-check of your work in progress.

While on the subject of the ‘Command Sequence Viewer’ notice the Up, Down and Remove buttons. The Up and Down buttons will allow you select various lines in your Command Sequence. The Up simply moves up the list, the down highlights the next item in the list. Once an item is selected, you can say “Remove” and that particular sequence item will be removed from the list. It’s simply a way to edit your macro before saving it.

Now that you’ve reviewed your command sequence in the ‘Command Sequence Viewer’...

You say:
create macro with command sequence

And that’s it, you’re done! The macro has been saved to the Speech Macros directory and automatically loaded into the macro system which means it’s listening for your command. Go ahead, try it! Just remember, since we set the context to “firefox.exe”, you’ll have to have Firefox up as the active application.

With a little bit of creativity and thought, you’ll soon be creating voice command macros to do just about anything imaginable on your computer!

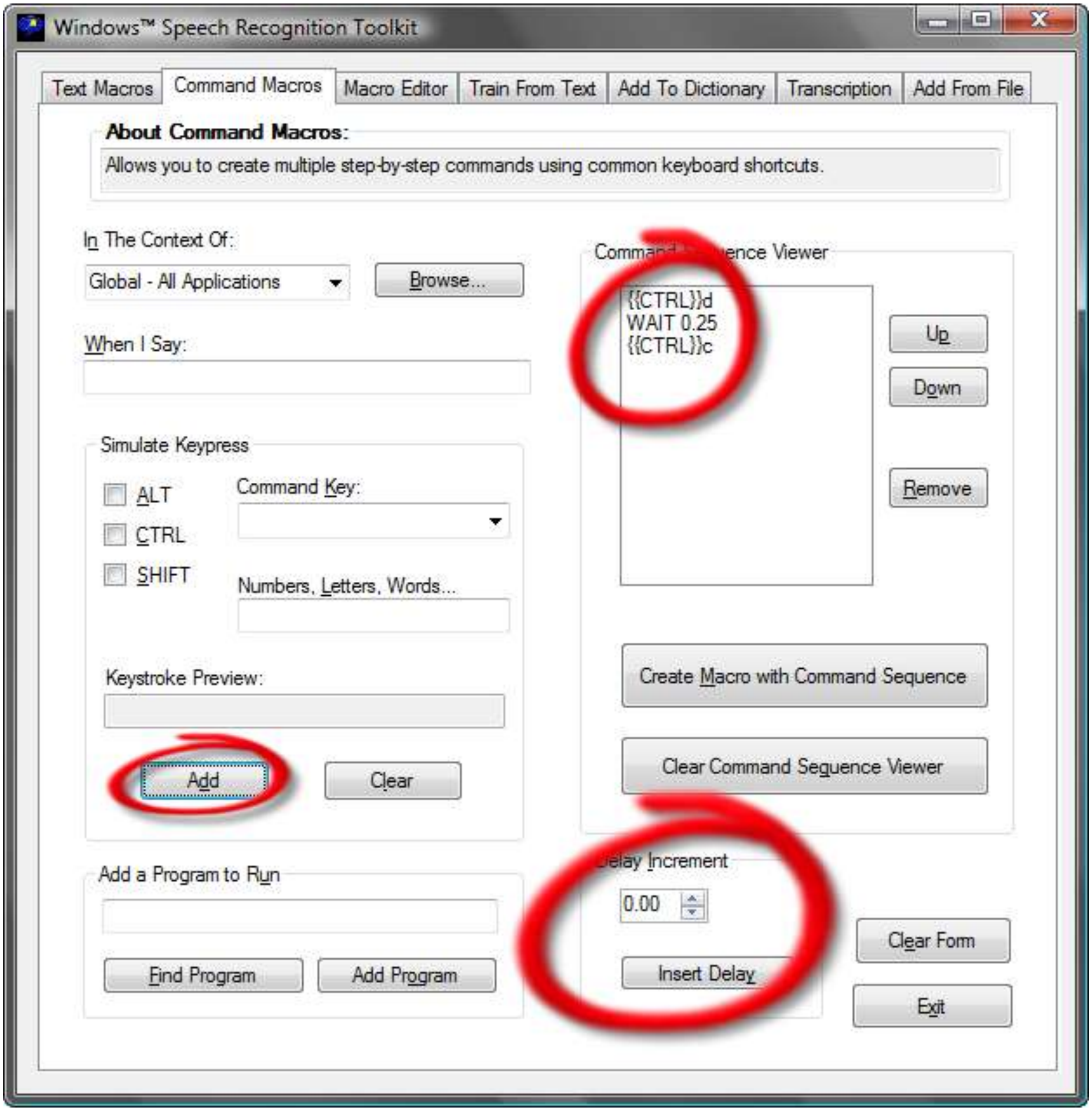
Got a question? Still scratching your head about some of this? Just post your question to one of the WSR macro forums and more than likely you’ll have an answer in short order.

Here are a couple of good forums:

http://www.msspeech-forum.com

http://tech.groups.yahoo.com/group/ms-speech/

Here’s the screenshot, highlighting some key areas and text:



Windows™ Speech Recognition Toolkit

About Command Macros:

Allows you to create multiple step-by-step commands using common keyboard shortcuts.

In The Context Of: Global - All Applications

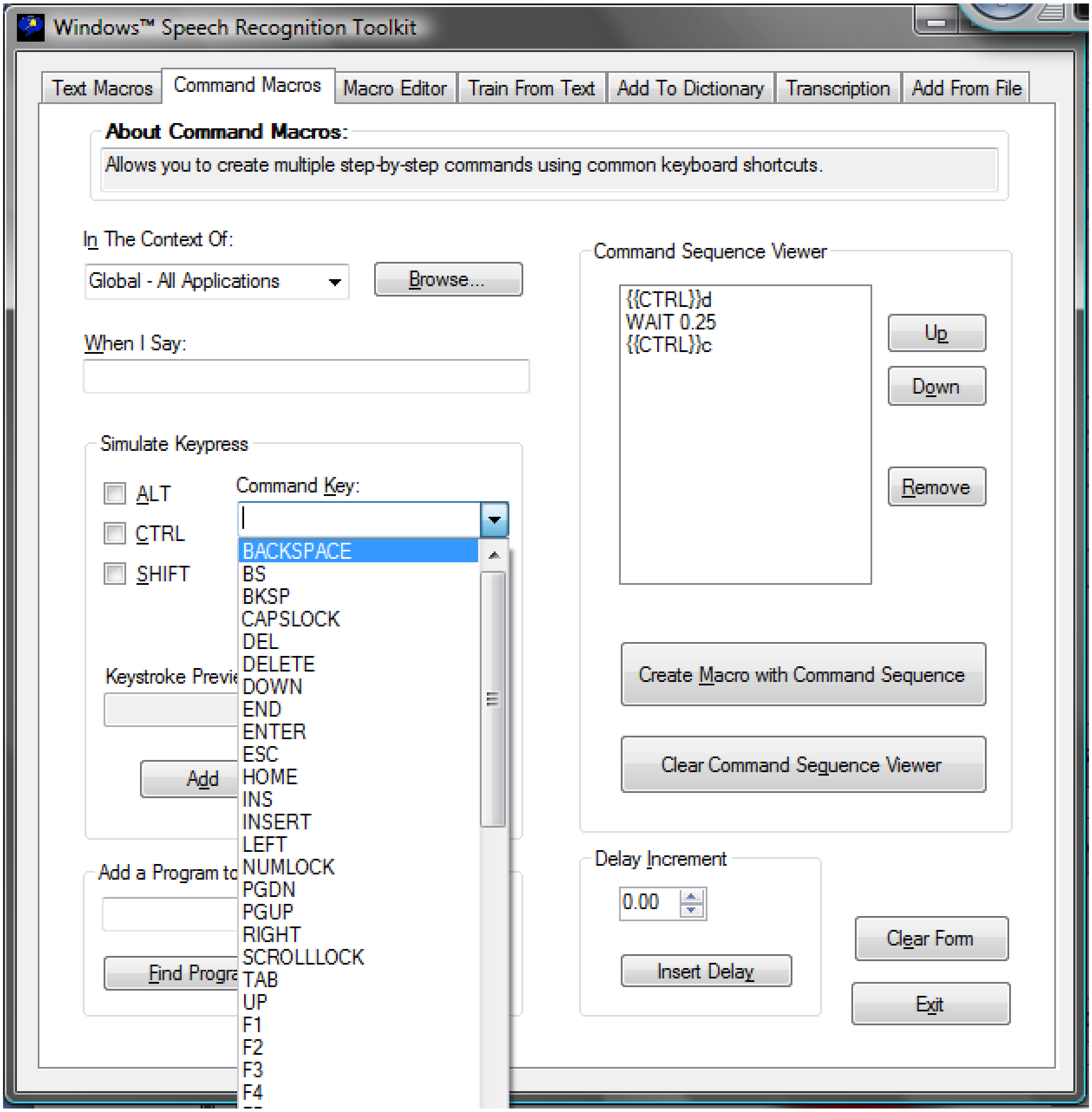
Command Sequence Viewer

```
{{CTRL}}d
WAIT 0.25
{{CTRL}}c
```

Delay Increment: 0.00

Take another quick look at the ‘Command Key’ dropdown box. This dropdown box contains every “special key” that can be used with Windows. For example the ‘F1’ key.

Here’s a screenshot with a partial listing of command keys displayed:



Windows™ Speech Recognition Toolkit

About Command Macros: Allows you to create multiple step-by-step commands using common keyboard shortcuts.

In The Context Of: Global - All Applications

Command Sequence Viewer:

```
{{CTRL}}d
WAIT 0.25
{{CTRL}}c
```

Delay Increment: 0.00

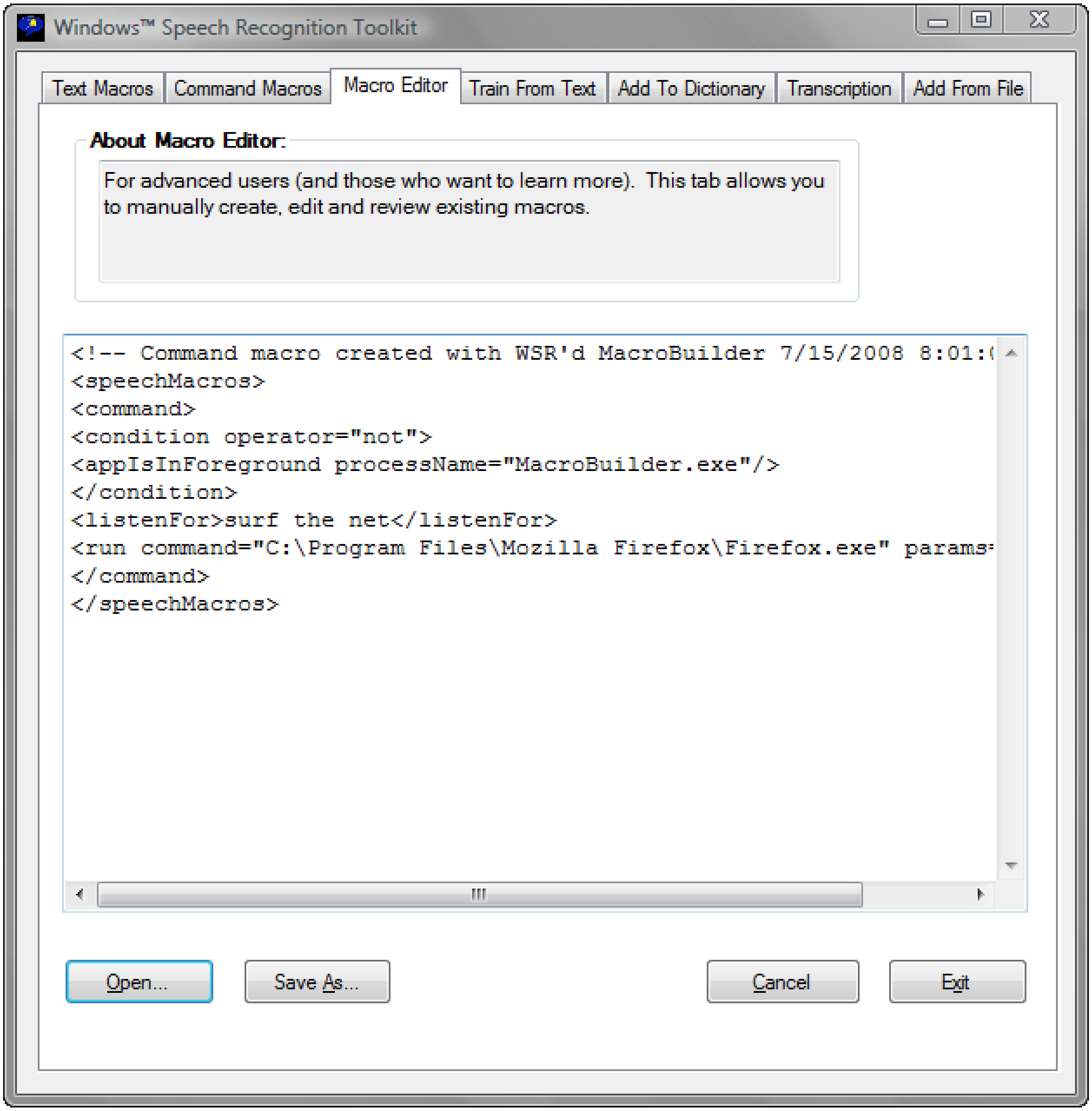
This is how and where you’d craft command sequences like ‘ALT-HOME’, ‘SHIFT-F10’, etc...

The overall concept with command macros is simple. If you know the keyboard shortcut key(s) necessary to accomplish your task, you can simply insert them here, in sequence, with delays to accomplish your task via voice. Don’t know the keyboard shortcut? Search our additional documentation for complete lists of the most popular applications keyboard shortcuts.

Macro Editor

If you’re curious about Windows Speech Recognition Macros and want to learn more, you’re in the right place. On this tab you’ll get to see the inner workings and details of the macro language itself. If you’re not interested in this stuff, just skip over this section for now. You can always come if you need to dig a little deeper.

Let’s jump right in with a screenshot of the Firefox macro created in the Command Macros tutorial:



For advanced users (and those who want to learn more). This tab allows you to manually create, edit and review existing macros.

```
<!-- Command macro created with WSR'd MacroBuilder 7/15/2008 8:01:(
<speechMacros>
<command>
<condition operator="not">
<appIsInForeground processName="MacroBuilder.exe"/>
</condition>
<listenFor>surf the net</listenFor>
<run command="C:\Program Files\Mozilla Firefox\Firefox.exe" params=
</command>
</speechMacros>
```

Don't worry about all that code for now. Just know this is a great place to come if for some reason you're having a problem with a particular macro. You may decide to post a question online and you can easily copy your macro code from here and paste it to a website for help.

The 'Open' button will allow you to browse to your Speech Macros directory and select any file you macro file you wish. Note: Macro files end with the extension .WSRMac.

The 'Save As' button will allow you to save changes to your macro. Careful though. You should have a good idea what your doing before doing any macro editing.

The 'Cancel' button clears any text in the macro textbox.

'Exit' will close the program.

For those wanting to know more, here are some of the (currently) available XML tags available for Speech Macros along with a brief description.

Note these "tags" are case sensitive, you must use exact upper/lowercase as shown here.

<speechMacros>

All macro files must use this tag and end with a </speechMacros> tag. Only one set of these is allowed for each macro file.

<command>

Required for every command and must be terminated with a </command> tag.

<listenFor>

This tag is what is used by the macro facility to listen for what is being spoken. It must be enclosed in a </listenFor> tag.

For instance: <listenFor>surf the net</listenFor>

When the computer "hears" the word or phrase contained within this tag it will execute everything within the preceding <command> tags.

<sendKeys>

Using this tag you can "send keys" to a particular application. It must be terminated with a </sendKeys> tag. In Microsoft Word you may want to send the keys CTRL-p to bring up the print dialog. In your macro code it would look like this:

```
<sendKeys>{{CTRL}}p</sendKeys>
```

You can find more on the syntax of the send keys here:

http://blogs.msdn.com/robch/archive/2008/05/07/sending-keys-using-custom-wsrmacros.aspx

<insertText>

Like the other tags so far, this one also needs to be properly terminated with a </insertText> tag.

Like the <sendKeys> tag, this tag will insert any text between the opening and closing tag. Unlike <sendKeys> you can have plenty of space in between. For example:

```
<insertText>
Line 1
Line 2

Line 3 was blank and this is line 4
</insertText>
```

<run>

This tag allows you to run commands similar to executing them from the command line.

The tag:

```
<run command="C:\Program Files\Mozilla Firefox\Firefox.exe" params=""/>
```

will run the Firefox browser. Notice you can also pass the program parameters with the “params” attribute. This would be useful if for instance you’d like to start notepad and open a particular text file. It would look something like this:

```
<run command="notepad.exe" params="MyFile.txt"/>
```

Note this tag is terminated differently than the others so far. There isn’t a </run> tag.

<appIsInForeground>

```
<appIsInForeground processName="notepad.exe"/>
```

Allows you to restrict the speech macro command to a specific application.

You can also use “windowTitle” and “windowTitleContains” attributes instead of the “processName” attribute.

So if you were interested in having a macro only execute when you had a specific document open in Word, the tag might look something like this:

```
<appIsInForeground windowTitleContains=”MyDoc.doc”/>
```

<speak>

This tells the speech recognition system to talk back to you! Whatever text you put between the speak tags will be audible on the speakers.

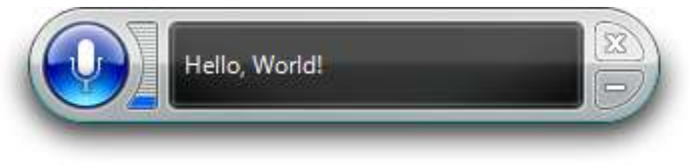
```
<speak>the voice box is a wonderful instrument</speak>
```

<setFeedbackText>

The following code:

```
<setTextFeedback>Hello, World!</setTextFeedback>
```

Would produce this in the feedback text area:



Hello, World!

<switchToApp>

This tag allows you to switch focus to an already running application.

```
<switchToApp windowTitleContains=”MyDoc.doc”/>
```

would switch to your MyDoc document. You can also use the windowTitle attribute here.

Along this same type of functionality there are the following additional tags:

```
<minimizeApp/>
<maximizeApp/>
<restoreApp/>
<closeApp/>
```

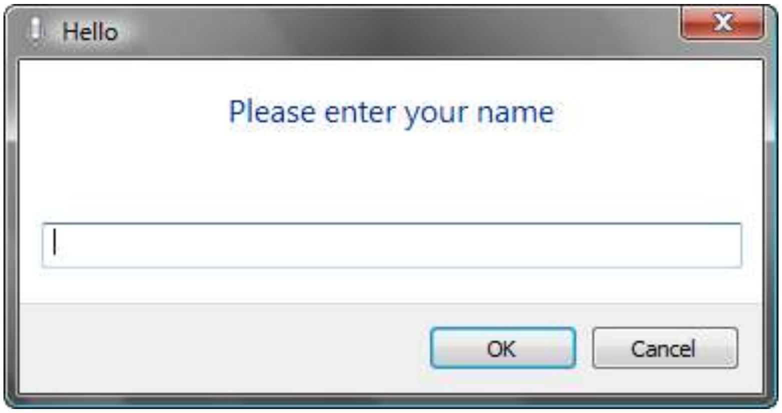
These also support the windowTitleContains and windowTitle attributes

<prompt>

Getting input from the user can be accomplished with the prompt tag like so:

```
<prompt title="Hello" resultState="ans">Please enter your name</prompt>
```

This will produce a prompt box like this:



Please enter your name

The resultState=”ans” will store the input from the user in the {[ans]} variable that can be used in your macro. Note: you can name the resultState variable anything you like.

Two similar tags the alert and confirm:

```
<alert title=”” timeout=””/>
<confirm title=”” timeout=””/>
```

<waitFor>

Allows you to pause macro execution

```
<waitFor seconds=”.25”/>
```

Other acceptable attributes include “anyStateNameIsSet” and “allStateNamesAreSet” but the seconds attribute is most often used.

<emulateRecognition>

Exactly as the tags name implies, it allows you to emulate a spoken command.

```
<emulateRecognition>stop listening</emulateRecognition>
```

Would send the recognizer the text “stop listening” just as if it had been spoken.

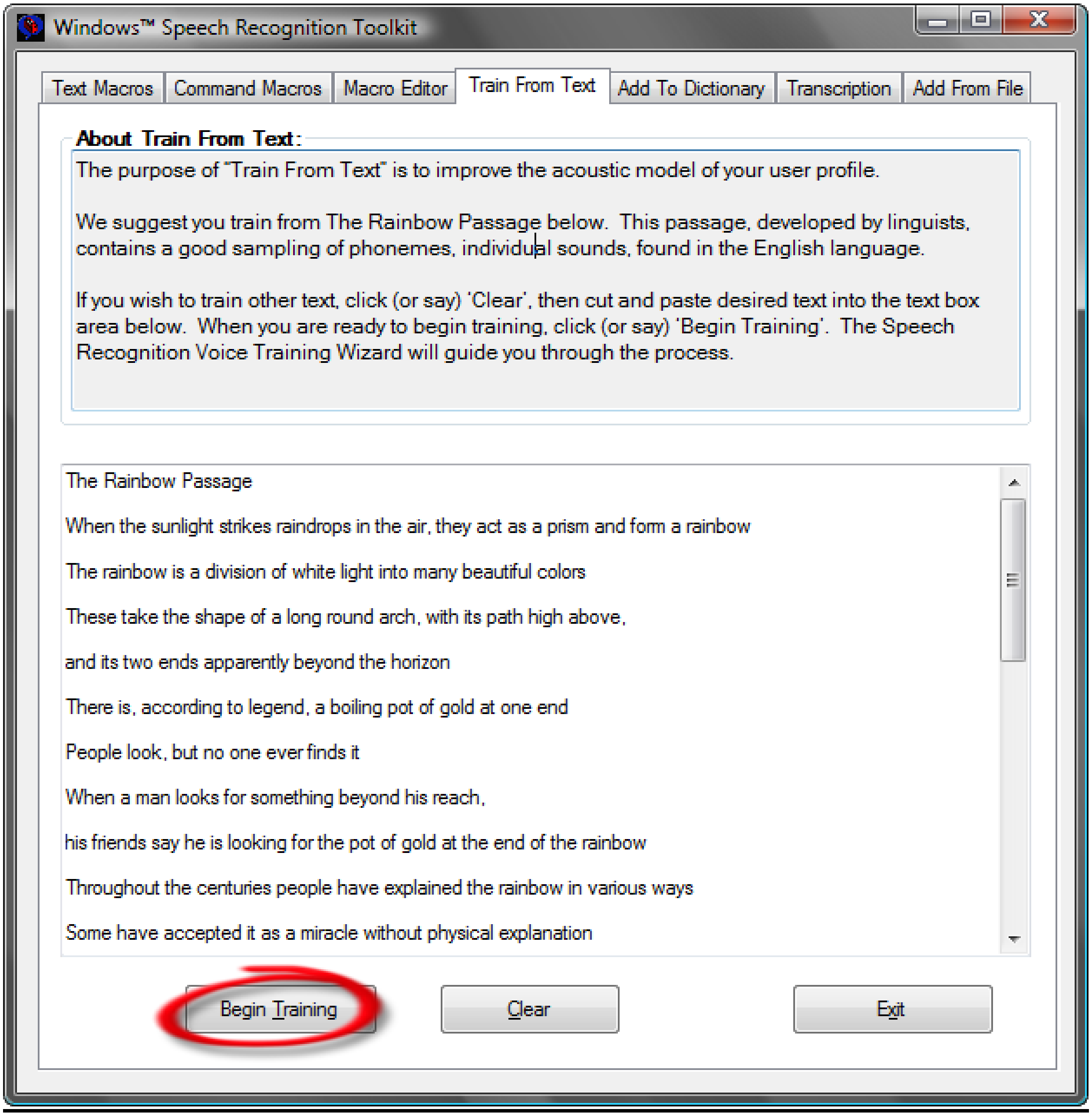
If you’re still hungry for more, don’t worry. There is plenty that hasn’t been described in this brief overview of macro tags. We’ve only touched on the most common ones. Check out some of the online groups and blogs. There’s a growing base of information available.

Train From Text

The purpose of “Train From Text” is to improve the acoustic model of your user profile. The “Rainbow Passage” is suggested as it was developed by linguists and contains all of the phonemes, individual sounds (the *b* in bat, for example), found in the English language.

If you wish to train from text other than the Rainbow Passage, choose ‘Clear’, then copy/paste your selected text into the textbox area.

When you are ready to begin choose ‘Begin Training’.



About Train From Text:

The purpose of "Train From Text" is to improve the acoustic model of your user profile.

We suggest you train from The Rainbow Passage below. This passage, developed by linguists, contains a good sampling of phonemes, individual sounds, found in the English language.

If you wish to train other text, click (or say) 'Clear', then cut and paste desired text into the text box area below. When you are ready to begin training, click (or say) 'Begin Training'. The Speech Recognition Voice Training Wizard will guide you through the process.

The Rainbow Passage

When the sunlight strikes raindrops in the air, they act as a prism and form a rainbow

The rainbow is a division of white light into many beautiful colors

These take the shape of a long round arch, with its path high above,

and its two ends apparently beyond the horizon

There is, according to legend, a boiling pot of gold at one end

People look, but no one ever finds it

When a man looks for something beyond his reach,

his friends say he is looking for the pot of gold at the end of the rainbow

Throughout the centuries people have explained the rainbow in various ways

Some have accepted it as a miracle without physical explanation

Add To Dictionary

The purpose of “Add To Dictionary” tab is to help the speech recognition engine recognize difficult to understand words and phrases.

If you find certain words or phrases consistently being misunderstood, you can add these to your personal speech dictionary, as well as record a pronunciation. This will increase the chances of accurate recognition for these particular words and phrases.

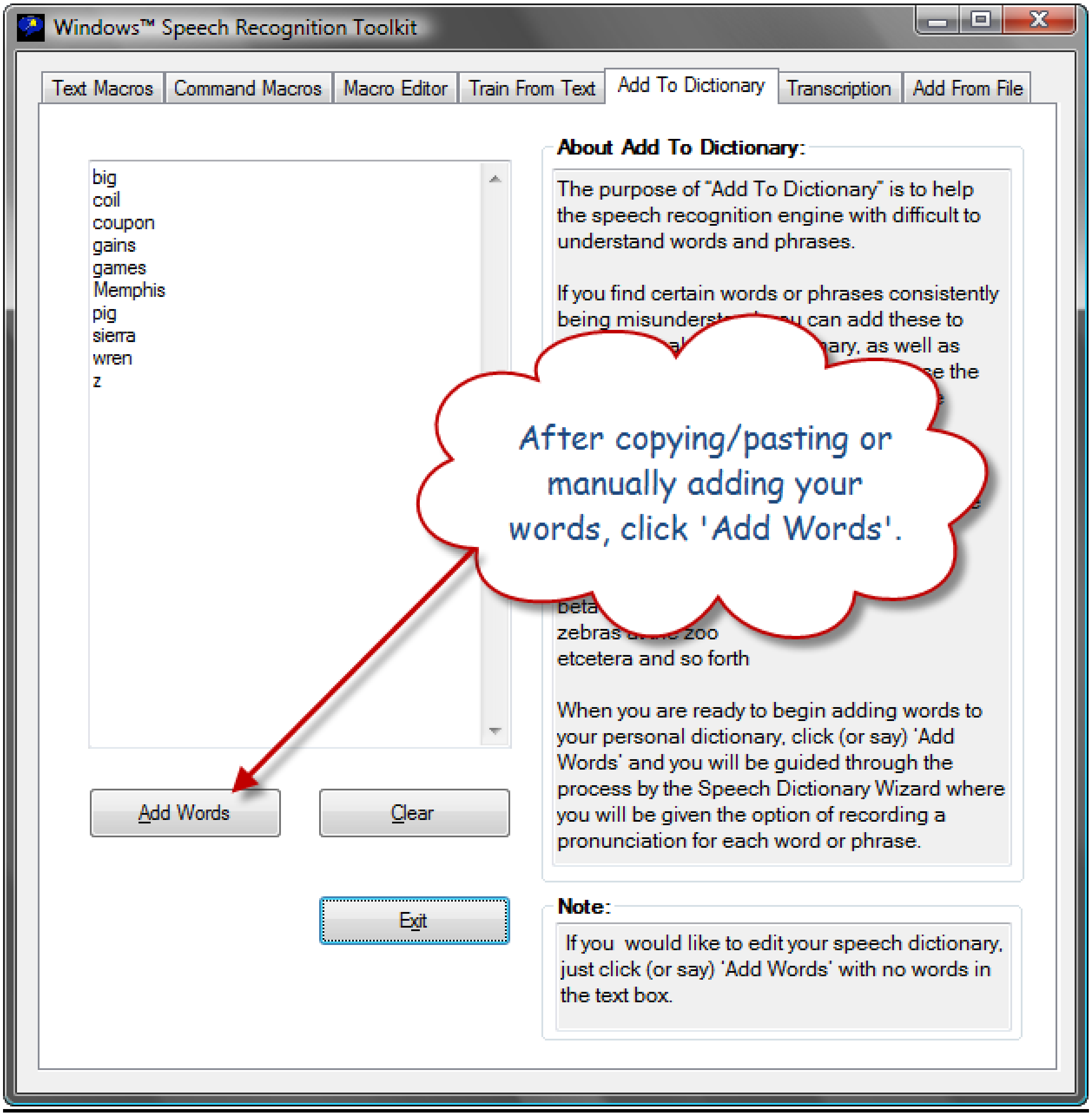
To use this feature, copy and paste words and/or phrases in the text box area. Each word or phrase should be on a separate line, for instance:

alpha
beta
zebras at the zoo
etcetera and so forth

When you are ready to begin adding these words to your personal dictionary, click (or say) ‘Add Words’ and you will be guided through the process by the Speech Dictionary Wizard where you will be given the option of recording a pronunciation for each word or phrase.

Important Note: If you would like to edit (change or delete) words from your speech dictionary, leave the list blank and say (or click) ‘Add To Dictionary’.

Here's a screenshot of the 'Add To Dictionary' tab loaded with some of my troublesome words. For instance the word *coupon;* instead of saying *coo-pon*, I prefer to say *q-pon*. The only way to get speech recognition to recognize this was to add the word to the dictionary with a recorded pronunciation. This is important to note. It is highly recommended that you always record a pronunciation when adding words to the dictionary. It will result in more accurate recognition than just adding words.



big
coil
coupon
gains
games
Memphis
pig
sierra
wren
z

About Add To Dictionary:

The purpose of "Add To Dictionary" is to help the speech recognition engine with difficult to understand words and phrases.

If you find certain words or phrases consistently being misunderst... ...u can add these to ...al... ...ary, as well as ...e the

beta
zebras a... zoo
etcetera and so forth

When you are ready to begin adding words to your personal dictionary, click (or say) 'Add Words' and you will be guided through the process by the Speech Dictionary Wizard where you will be given the option of recording a pronunciation for each word or phrase.

Note:

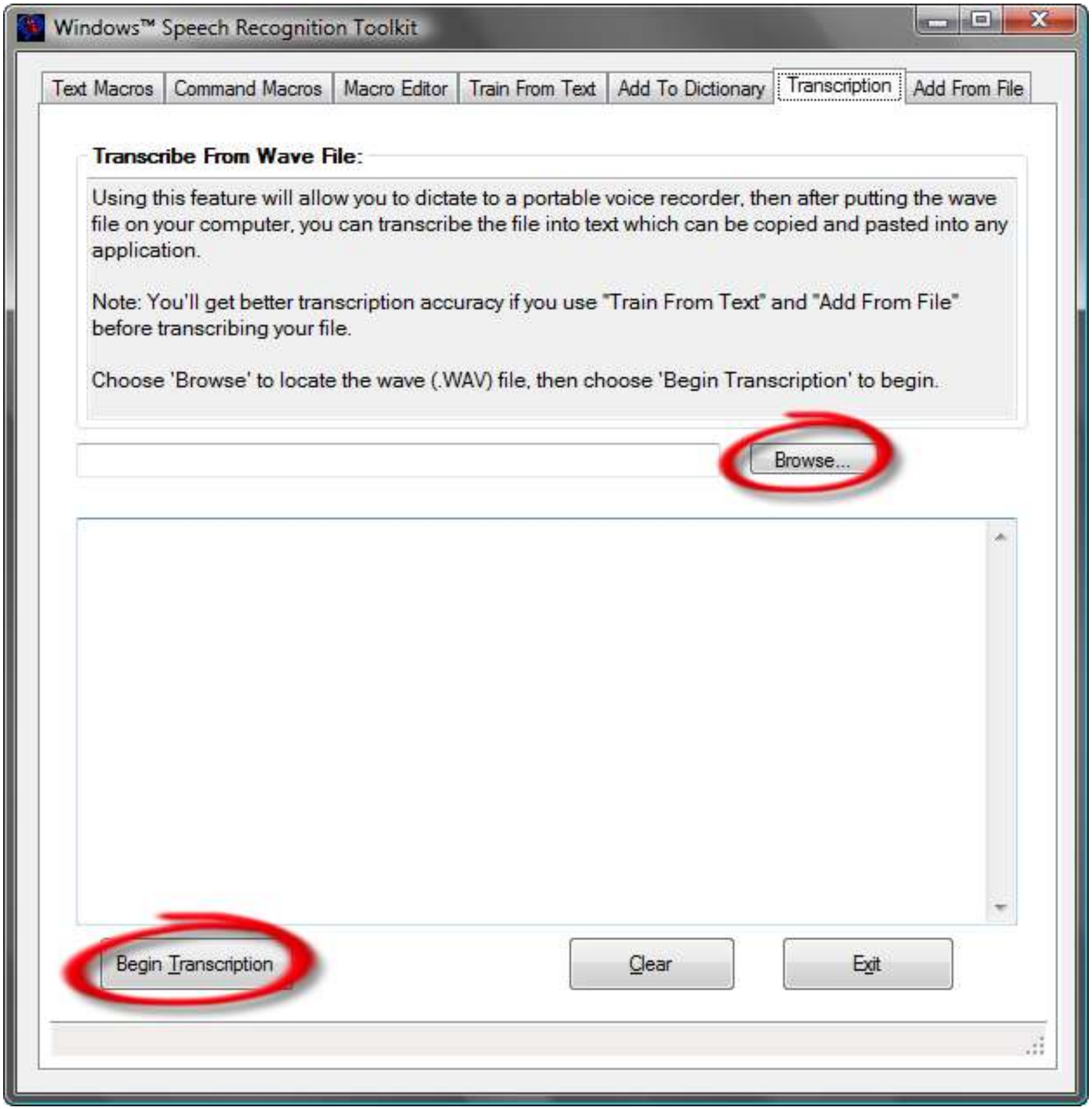
If you would like to edit your speech dictionary, just click (or say) 'Add Words' with no words in the text box.

After copying/pasting or manually adding your words, click 'Add Words'.

Transcription

An important note on transcription: There will be a noticeable increase in recognition accuracy if you use the 'Train From Text' and 'Add From Text' features before using transcription.

If you have a portable voice recorder, this feature may be just what you're looking for. With a portable voice recorder, you can dictate notes, memos and record thoughts and ideas while you're away from the computer. Once you return, just copy the sound file to your computer, convert to a wave (.WAV) file (note some voice recorders save files as .wav), then choose the 'Browse' button to point the transcription module to the wave file and choose 'Begin Transcription'. You'll soon see your words appearing in the transcription textbox! Once the transcription is complete, the entire text will automatically be selected and copied onto the clipboard, ready for you to paste into a more suitable application.



Windows™ Speech Recognition Toolkit

Transcribe From Wave File:

Using this feature will allow you to dictate to a portable voice recorder, then after putting the wave file on your computer, you can transcribe the file into text which can be copied and pasted into any application.

Note: You'll get better transcription accuracy if you use "Train From Text" and "Add From File" before transcribing your file.

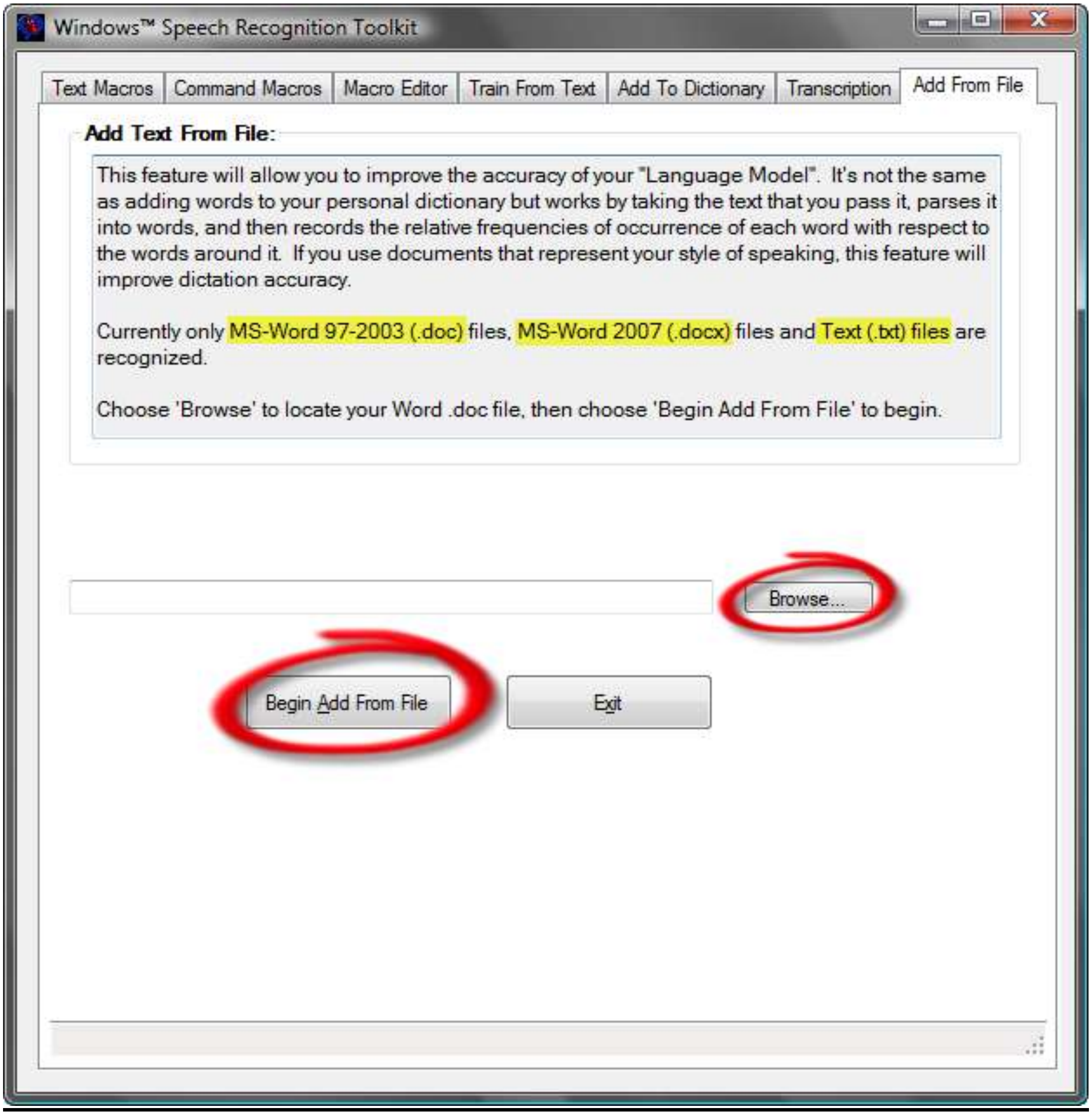
Choose 'Browse' to locate the wave (.WAV) file, then choose 'Begin Transcription' to begin.

Add From File

This feature is best described as follows:

“Add from File” allows you to improve the accuracy of your "Language Model". It's not the same as adding words to your personal dictionary but works by taking the text that you pass it, parses it into words, and then records the relative frequencies of occurrence of each word with respect to the words around it. If you use documents that represent your style of speaking, this feature will improve dictation accuracy.”

Choose ‘Browse’ to find the document you wish to add, then choose ‘Begin Add from File’. The file will be opened and read into the language processor and quickly added to your language model.



Add Text From File:

This feature will allow you to improve the accuracy of your "Language Model". It's not the same as adding words to your personal dictionary but works by taking the text that you pass it, parses it into words, and then records the relative frequencies of occurrence of each word with respect to the words around it. If you use documents that represent your style of speaking, this feature will improve dictation accuracy.

Currently only MS-Word 97-2003 (.doc) files, MS-Word 2007 (.docx) files and Text (.txt) files are recognized.

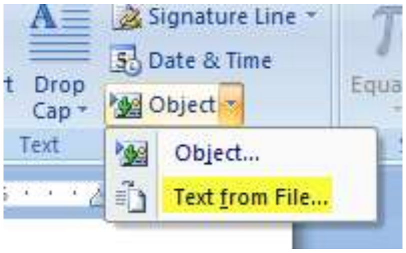
Choose 'Browse' to locate your Word .doc file, then choose 'Begin Add From File' to begin.

Tips & Tricks

This section contains shortcuts and tips for using the WSR Toolkit.

- **Using 'Add From File' With Multiple Files**
  Currently the 'Add From File' feature only analyzes one file at a time. It may become burdensome if you have many files you would like for it to analyze. You can, of course, create one file from multiple files. Within Microsoft Word, here's how:

  1. Create a new, blank document
  2. Choose the 'Insert' tab (ALT-N)
  3. In the 'Text' ribbon box, choose 'Object', 'Text from File'

  You can now select multiple files and they will be concatenated into one large file. Save this file within MS-Word. You can now browse on the 'Add From File' tab to this large file.

- **Increasing Speech Recognition Accuracy**
  Following is a recommended procedure for increasing speech recognition accuracy with the Toolkit. (Note: it assumes you are using a 'new' speech profile)

  1. Set up the microphone in Windows Vista (you'll read the "Peter dictates to his computer..." passage)
  2. Run WSR Toolkit and read the Rainbow Passage on the 'Train From Text' tab.
  3. Select the 'Add From File' tab and browse/select one of your MS-Word document or plain text file that represents your style of writing/speaking.

  Additional tips for increased recognition accuracy:

  1. Be diligent about correcting mistakes using "correct that" instead of simply retyping.
  2. Add words (or phrases) to your personal speech dictionary (WSR Toolkit 'Add To Dictionary' tab) that are repeatedly misrecognized. First try adding them without pronunciation, then if it still gets it wrong, delete and re-add the word (or phrase) with pronunciation.

Index

Command Key, 15
Command Sequence Viewer
    Up, Down & Remove Buttons, 13
context
    Global Context, 7
copy web address, 11
insert a timing delay, 12
Keystroke Preview, 11
Microsoft .NET 3.5 Framework, 4
Numbers, Letters, Words” textbox, 11
surf the net, 9
trigger word, 6
WSR, 2