

Worcester Polytechnic Institute Digital WPI

Interactive Qualifying Projects (All Years)

Interactive Qualifying Projects

April 2015

Voice Controlled Music Sequencer

Victoria Jade Valcour
Worcester Polytechnic Institute

Follow this and additional works at: <https://digitalcommons.wpi.edu/iqp-all>

Repository Citation

Valcour, V. J. (2015). *Voice Controlled Music Sequencer*. Retrieved from <https://digitalcommons.wpi.edu/iqp-all/2753>

This Unrestricted is brought to you for free and open access by the Interactive Qualifying Projects at Digital WPI. It has been accepted for inclusion in Interactive Qualifying Projects (All Years) by an authorized administrator of Digital WPI. For more information, please contact digitalwpi@wpi.edu.

VOICE CONTROLLED MUSIC SEQUENCER

Interactive Qualifying Project Report completed in partial fulfillment

of the Bachelor of Science degree at

Worcester Polytechnic Institute, Worcester, MA

Submitted to: Professor Bianchi

Ma'at Ford

Victoria Valcour

(date of submission)



Abstract

This project served as a proof-of-concept to determine the validity of a voice-controlled music sequencer. Working with Digital Performer and Dragon Dictate software, the project explored the history of voice-recognition technology, and the feasibility of this technology within a music sequencing environment.

Table of Contents

| | |
|-------------------------|----|
| Executive Summary | 4 |
| Introduction | 6 |
| Methodology | 8 |
| Results | 12 |
| Appendix A | 14 |
| Works Cited | 17 |

Executive Summary

The objective of this IQP was to develop a voice-controlled music sequencer prototype that would serve as a proof-of-concept to determine the validity of developing a more sophisticated model. A voice-controlled sequencer would allow the user to initiate and realize a number of editing and data manipulation procedures through voice commands. Execution of these commands will help facilitate the preparation of large and involved music sequences that have traditionally required time intensive editing strategies. The speech-recognition technology would allow users to easily alter properties of songs without having to use keystrokes associated with their desired command. Voice-activated music sequencers are also necessary for increasing accessibility for disabled users, who experience difficulties utilizing traditional sequencers due to visual or other physical impairments. Ultimately, a voice controlled music sequencer will make the editing procedures more representative of the interaction that occurs between a conductor and a live acoustic orchestra. This technology is necessary to increase versatility in terms of music production, as it provides more ways in which producers may create and implement innovative musical ideas and concepts. In order to meet the primary objective, the project team created a crude blueprint for a voice activated interface by linking voice software and macros (recorded actions that perform an automated computer action).

At the onset of this IQP, the project team was provided and presented with specific software to use by advisor Professor Bianchi: AppleScript, Digital Performer 7.22, and Mac speakable items. However, only certain items in AppleScript are fully compatible with the commands in Mac speakable items. Thus, the project team decided to make use of different software to meet the objective, specifically, Dragon Dictate. Dragon Dictate is a speech

recognition software that allows the user to perform tasks on the computer with custom voice commands.

The project team experimented with different types of software to fulfill the automation portion of the prototype. Mouse recording software recognizes movements that the user performs with the computer mouse and compiles them into a file that is executable. By linking this software and Dragon Dictate, the user is able to easily modify musical compositions by simply saying the name of a file that contains the respective command to be performed.

Introduction

In computer science, speech recognition is the conversion of speech to a computer command or on-screen text. In order to complete this conversion, a computer must complete a series of complex steps:

1. When a user speaks, their voice creates vibrations in the air. A computer samples these by taking precise measurements at frequent intervals of the vibrations' waves.
2. The computer filters the sampled sound to remove unwanted noise, and adjusts it to a constant volume and speed that will match the template sound sample stored in the system's memory.
3. The sound is divided into small segments that are matched with known phoneme, representations of sounds that humans make to form meaningful expression.
4. The software examines the phonemes in the context of others around them, running them through a complex statistical model and comparing them to a large library of words, phrases, and sentences. Thus, the software is able to determine what the user was saying, and as a result, executes a command (Grabianowski).

Throughout the last sixty years, speech recognition software has evolved and emerged with applications in various industries that include healthcare, military, and telephony. In the 1950s, the first speech recognition systems were only able to process digits. In 1952, Bell Laboratories designed the "Audrey" system, which recognized digits spoken by a single user. Ten years later at the World Fair, IBM revealed its "Shoebbox" system, which was able to understand sixteen spoken English words.

In the 1970s, interest from the United States Department of Defense allowed speech recognition technology to make major strides -- perhaps most notably in the funding of Carnegie Mellon's "Harpy" system. Harpy was able to process approximately 1,011 words, the vocabulary of an average three year old. Additionally, Bell Laboratories was able to introduce the first system that was able to interpret the voices of multiple users.

In the 1980s, the hidden Markov model was developed. This was a new statistical method that considered the probability of unknown sounds being words, rather than simply looking for sound patterns. With this model, speech recognition began to emerge in commercial applications and business settings.

In the 1990s, speech recognition software became viable for ordinary users with the aid of more efficient computer processors. The first consumer speech recognition product, Dragon Dictate, was launched in 1990. This software was able to recognize a user's continuous speech after forty-five minutes of training. In 1996, BellSouth produced the first dial-in interactive speech recognition system, "Val", which gave users information based on what they said into a telephone.

In the 2000s, speech recognition emerged into the mobile phone industry with the arrival of Google Voice Search app for the iPhone. This was particularly significant for the development of speech recognition, as the tiny size of mobile devices provided incentive to develop more advanced and alternative user-input methods (Pinola).

Methodology

The project team began by experimenting with an application that detects keystrokes, “Keyboard and Mouse Recorder” (KAMR), developed by AlphaOmega Software. The project team recorded basic keystroke commands (i.e. start, stop, play), all of which were executed properly when played back in Digital Performer. The project team tried to further this method by naming these executable files to their respective commands (i.e. a recording that clicks the play button or activates the play keystroke is named “play”) and by voicing the name of the file by using Speakable Items. However, this method was crude and tedious to perform for several reasons. Whenever the keystroke was pressed to begin recording, the operator had to immediately perform the keystroke or the mouse command. No matter how swiftly the action was performed during the recording, there was a significant delay. Additionally, the version of KAMR that the project team obtained was a trial version, so actions could only be recorded for a certain length in time. The project team’s advisor, Professor Bianchi, was willing to purchase the full version of the software, but eventually it was deduced that this method was inadequate for an eventual voice-activated music sequencer.

While the project team initially began utilizing AppleScript and Speakable Items software, it was ultimately decided to make use of other software applications. This decision was made primarily due to AppleScript being a somewhat complex programming language that takes significant time to learn. With little knowledge in AppleScript software, the project team felt that time would be better spent generating results as opposed to learning a new programming language. Furthermore, AppleScript and OS X’s built in Speakable Items worked much more

cooperatively with proprietary Apple Applications (Automator, iMovie, Safari, Finder, iMessage, etc.).

In further research on an unaffiliated Digital Performer (DP) forum concerning scriptability, the project team learned that DP is indeed scriptable in AppleScript. However, DP is not a very “well-behaved” application. In the AppleScript editor, a dictionary of commands will be published for each scriptable application. While DP does have a command dictionary, it contains only one command titled “doscript”, with no script syntax. The project team examined a 1998 MQP in which a team created an actual voice-activated music sequencer. In doing so, the project team noted that the team used a voice speaking software titled “Dragon Naturally Speaking” developed by Nuance Communications for Windows. Because this project is performed completely on Mac OS X, the project team decided to use the Macintosh alternative to Dragon Naturally Speaking, Dragon Dictate.

The project team obtained a copy of Dragon Dictate and installed it on one of the Apple computers in WPI’s Riley Lab. Dragon Dictate is a speech-to-text software that transcribes what is verbally dictated to the computer by the user. Dragon Dictate contains a tutorial for new users on how to use the software. Users create their own voice profiles when Dragon is turned on. In order to obtain optimal performance out of Dragon Dictate, our team read through all of the passages so the software could get acclimated to our voices.

The project team tested Dragon Dictate with other applications (TextEdit, Finder, and Safari) in order to see the feasibility of using it eventually with Digital Performer. The project team articulated simple commands (“delete”, “go forward”, “go back”, “open”, and “close”) at reasonable, intelligible speeds to ensure that the software understood the input clearly. Dragon

Dictate understood read commands with an approximate 90% success rate. If there is any error, the user may state “scratch that” and Dragon will provide the user with a list of possible phrases that may match what was spoken. The user would simply select one of these phrases and the information from the selected phrase would be sent to the user’s Dragon Profile for the software to “learn” from the user’s voice. The project team verbalized other commands (i.e. file, edit, format, etc.) to see if Dragon would execute them. There were some commands that Dragon would not execute, due to Dragon misinterpreting a command for another word. Ultimately, Dragon performed sufficiently with the commands that were provided off of the Dragon Dictate website for all three applications that the project team worked with.

After a few weeks of experimenting with Dragon’s interface and using its commands, the project team began to integrate Dragon Dictate with Digital Performer. The initial plan of our proof of concept was to use KAMR software to record an action, save the file, and execute the command via user voice. This proved to be a very tedious process. Additionally, Dragon Dictate was only able to open applications, and not files. Each time the project team attempted to open a file with Dragon, another window with information about that file would open. Another problem that the project team encountered using KAMR was that even when mouse recording files were executed, Digital Performer needed to be the current application running over all others. These setbacks ultimately proved that the project team’s developed concept, at that point, was not feasible.

While Dragon would not work with the recorded files for Keyboard and Mouse Recorder (KAMR), Speakable Items would. After placing some command files in the KAMR folder and resetting the Speakable Items application, the commands were successfully executed. Since the voice-to-interface engine in Speakable Items is not as complex as Dragon, commands needed to

be enunciated loudly and clearly. While this was significant progress, the project team continued to look for software that allowed the user to exert their voice less.

After researching ways to integrate Dragon with other applications, the project team found that Dragon came with its own scripting software. In Dragon Dictate, there is a list of applications with custom commands (accessible through the “commands” option) and a list of parameters for any command that is created. If a user wanted to create a command that opened the DP Effects Window, the user would create a new application context for Dragon Dictate to receive, and then adjust the command parameters. This command script editor offers an extensive amount of options in how the user chooses to execute the command.

Results

In the command list for Dragon, the project team selected Digital Performer and created a new command titled “drum roll”. The project team selected the respective keystroke that opened the drum roll command, voiced “drum roll” to the software, and the drum roll window successfully opened. This simple test serves as proof-of-concept for a voice-activated music sequencer. Provided with the versatility and simplicity of Dragon Dictate’s scripting program, the team found it much easier to communicate with the interface for Digital Performer.

To further test the capabilities of Dragon’s scripting with Digital Performer, the project team began to use the list of Digital Performer commands that were executable by keystroke and integrating them into Dragon Dictate. The new voice commands that the project team created worked seamlessly in Digital Performer.

The next step was to script the menu bar for Digital Performer. Dragon also has an option in the command scripting window for the user to activate items on the menu bar. When testing the menu items in Dragon Dictate, there were some interesting points to note. When the user voices one menu item, the menu item needs to be repeated in order to close it. The user cannot vocally switch from one menu item to another while one is currently open. If the user continues to switch while one menu is open, a glitch will occur in which Dragon will open the recited menu for a split second.

Another note from the project team’s developed method is the manner in which a user may modify music files. The project team was provided with a sample of “Adagio for Strings”, composed by Samuel Parker. When examining the music overlay for Adagio for Strings, the project team wanted to test how well the voice activation features would work with the song

itself, specifically with a command the project team created called “retrograde” (For a full list of commands that the project team created, see Appendix A). Retrograde reverses the order of musical notes, so that when played, the notes are read and played the same backwards as they are naturally played forwards. The project team selected a piece of notes from the song and voiced the command “retrograde”. As expected, the selected piece in the composition reversed the notes.

Appendix A

The screenshot displays the 'Commands' window, which is organized into several sections:

- Left Pane:** A list of application categories including Calendar, Digital Performer, Dragon Dictate, Finder, Keyboard And M..., Mail, Messages, Microsoft Word, Notes, Reminders, Safari, TextEdit, and Global.
- Command List:** A table with columns for 'Active', 'Command', and 'Type'. The 'Adjust Beat Detection' command is selected.
- Right Pane:** A detailed view of the 'Adjust Beat Detection' command, showing its context, type, and menu path.

| Active | Command | Type |
|-------------------------------------|----------------------------------------------|-----------|
| <input checked="" type="checkbox"/> | Adjust Beat Detection | Menu Item |
| <input checked="" type="checkbox"/> | Adjust Beat Sensitivity | Menu Item |
| <input checked="" type="checkbox"/> | Adjust Bite Volume Minus Point Five Decibels | Menu Item |
| <input checked="" type="checkbox"/> | Adjust Bite Volume Plus Point Five Decibels | Menu Item |
| <input checked="" type="checkbox"/> | Adjust Pitch Segmentation | Menu Item |
| <input checked="" type="checkbox"/> | Adjust Sequence to Soundbite Tempo | Menu Item |
| <input checked="" type="checkbox"/> | Adjust Soundbites to Sequence Tempo | Menu Item |
| <input checked="" type="checkbox"/> | Align and Spot audio | Menu Item |
| <input checked="" type="checkbox"/> | Analyze Soundbite tempo | Menu Item |
| <input checked="" type="checkbox"/> | Arpeggiator | Menu Item |
| <input checked="" type="checkbox"/> | Audio | Menu Item |
| <input checked="" type="checkbox"/> | Audio Performance | Menu Item |
| <input checked="" type="checkbox"/> | Audio Pitch Correction | Menu Item |
| <input checked="" type="checkbox"/> | Audio Plugins | Menu Item |
| <input checked="" type="checkbox"/> | Background Processing | Keystroke |
| <input checked="" type="checkbox"/> | Bite Volume and Gain | Menu Item |
| <input checked="" type="checkbox"/> | Bounce Settings | Menu Item |
| <input checked="" type="checkbox"/> | Bounce to Disk | Menu Item |
| <input checked="" type="checkbox"/> | Burn CD from Disk Image | Menu Item |
| <input checked="" type="checkbox"/> | Capture Realtime MIDI effects | Menu Item |
| <input checked="" type="checkbox"/> | Change Velocity | Menu Item |
| <input checked="" type="checkbox"/> | Chunks | Keystroke |
| <input checked="" type="checkbox"/> | Clear Bite Gain | Menu Item |
| <input checked="" type="checkbox"/> | Clear Bite Volume | Menu Item |
| <input checked="" type="checkbox"/> | Clear Files after Align | Menu Item |
| <input checked="" type="checkbox"/> | Clear Files after Align copy | Menu Item |
| <input checked="" type="checkbox"/> | Clear Loops | Menu Item |
| <input checked="" type="checkbox"/> | Clear Pitch | Menu Item |
| <input checked="" type="checkbox"/> | Clear Pitch Control Points | Menu Item |
| <input checked="" type="checkbox"/> | Clear Soundbite tempo | Menu Item |
| <input checked="" type="checkbox"/> | Clear Sync Points | Menu Item |
| <input checked="" type="checkbox"/> | Close | Keystroke |
| <input checked="" type="checkbox"/> | Control Panel | Keystroke |
| <input checked="" type="checkbox"/> | Copy Beats | Menu Item |
| <input checked="" type="checkbox"/> | Copy Sequence Tempo to Soundbite | Menu Item |
| <input checked="" type="checkbox"/> | Count | Keystroke |
| <input checked="" type="checkbox"/> | Counter | Keystroke |
| <input checked="" type="checkbox"/> | Create Soundbites from beats | Menu Item |
| <input checked="" type="checkbox"/> | De Flam | Menu Item |
| <input checked="" type="checkbox"/> | Delete Fades | Menu Item |
| <input checked="" type="checkbox"/> | Deslect all | Keystroke |
| <input checked="" type="checkbox"/> | Digital Performer | Menu Item |
| <input checked="" type="checkbox"/> | Dither | Menu Item |
| <input checked="" type="checkbox"/> | Double Soundbite tempo | Menu Item |
| <input checked="" type="checkbox"/> | Drum Editor | Keystroke |
| <input checked="" type="checkbox"/> | Duplicate Soundbite | Menu Item |
| <input checked="" type="checkbox"/> | Edit | Menu Item |
| <input checked="" type="checkbox"/> | Edit Audio Export Formats | Menu Item |
| <input checked="" type="checkbox"/> | Edit Bounce Again Settings | Menu Item |
| <input checked="" type="checkbox"/> | Edit in Waveform Editor | Menu Item |
| <input checked="" type="checkbox"/> | Effects | Keystroke |
| <input checked="" type="checkbox"/> | Erase | Keystroke |
| <input checked="" type="checkbox"/> | Extract tempo from MIDI | Menu Item |
| <input checked="" type="checkbox"/> | Fade | Menu Item |
| <input checked="" type="checkbox"/> | Fade Again | Menu Item |
| <input checked="" type="checkbox"/> | Fade In | Menu Item |
| <input checked="" type="checkbox"/> | Fade Out | Menu Item |
| <input checked="" type="checkbox"/> | File | Menu Item |
| <input checked="" type="checkbox"/> | Find Tempo Factor Near Sequence Tempo | Menu Item |
| <input checked="" type="checkbox"/> | Freeze Selected Tracks | Menu Item |

Adjust Beat Detection Details:

- Context: Digital Performer 7.22
- Type: Menu Item
- Menu: Audio > Adjust Beat Detection...

Commands

| Active | Command | Type |
|-------------------------------------|------------------------------|-----------|
| <input checked="" type="checkbox"/> | Groove quantize | Keystroke |
| <input checked="" type="checkbox"/> | Halve Soundbite tempo | Menu Item |
| <input checked="" type="checkbox"/> | Heal Separation | Keystroke |
| <input checked="" type="checkbox"/> | Help | Menu Item |
| <input checked="" type="checkbox"/> | Humanize | Menu Item |
| <input checked="" type="checkbox"/> | Import Audio | Keystroke |
| <input checked="" type="checkbox"/> | Insert Loop | Menu Item |
| <input checked="" type="checkbox"/> | Invert Pitch | Menu Item |
| <input checked="" type="checkbox"/> | Layering | Menu Item |
| <input checked="" type="checkbox"/> | Layering Move Backward | Menu Item |
| <input checked="" type="checkbox"/> | Layering Move Forward | Menu Item |
| <input checked="" type="checkbox"/> | Layering Move to Back | Menu Item |
| <input checked="" type="checkbox"/> | Layering Move to Front | Menu Item |
| <input checked="" type="checkbox"/> | Load | Menu Item |
| <input checked="" type="checkbox"/> | MIDI Keys | Keystroke |
| <input checked="" type="checkbox"/> | Merge | Keystroke |
| <input checked="" type="checkbox"/> | Merge Soundbites | Menu Item |
| <input checked="" type="checkbox"/> | Meter Bridge | Menu Item |
| <input checked="" type="checkbox"/> | Mixing Board | Keystroke |
| <input checked="" type="checkbox"/> | Move to Original Time Stamp | Menu Item |
| <input checked="" type="checkbox"/> | Move to User Time Stamp | Menu Item |
| <input checked="" type="checkbox"/> | Multi Bounce | Menu Item |
| <input checked="" type="checkbox"/> | New | Keystroke |
| <input checked="" type="checkbox"/> | New Sequence | Menu Item |
| <input checked="" type="checkbox"/> | New V-Rack | Menu Item |
| <input checked="" type="checkbox"/> | Normalize | Menu Item |
| <input checked="" type="checkbox"/> | Open | Keystroke |
| <input checked="" type="checkbox"/> | Open Selected Effects | Keystroke |
| <input checked="" type="checkbox"/> | Paste Multiple | Keystroke |
| <input checked="" type="checkbox"/> | Paste Repeat | Keystroke |
| <input checked="" type="checkbox"/> | Paste Repeat Multiple | Keystroke |
| <input checked="" type="checkbox"/> | Pause | Keystroke |
| <input checked="" type="checkbox"/> | Play | Keystroke |
| <input checked="" type="checkbox"/> | Play Selection | Menu Item |
| <input checked="" type="checkbox"/> | Print Window | Keystroke |
| <input checked="" type="checkbox"/> | Project | Menu Item |
| <input checked="" type="checkbox"/> | Quantize | Keystroke |
| <input checked="" type="checkbox"/> | Quantize Pitch | Menu Item |
| <input checked="" type="checkbox"/> | Quick Scribe Editor | Keystroke |
| <input checked="" type="checkbox"/> | Redo Next Action | Keystroke |
| <input checked="" type="checkbox"/> | Redo Previous Action | Keystroke |
| <input checked="" type="checkbox"/> | Region | Menu Item |
| <input checked="" type="checkbox"/> | Reload Soundbite | Menu Item |
| <input checked="" type="checkbox"/> | Repeat | Keystroke |
| <input checked="" type="checkbox"/> | Replace Soundbite | Menu Item |
| <input checked="" type="checkbox"/> | Retrograde | Menu Item |
| <input checked="" type="checkbox"/> | Reveal in Finder | Menu Item |
| <input checked="" type="checkbox"/> | Reverse Time | Menu Item |
| <input checked="" type="checkbox"/> | Rewind | Keystroke |
| <input checked="" type="checkbox"/> | Run Last Bounce Again | Menu Item |
| <input checked="" type="checkbox"/> | Save | Keystroke |
| <input checked="" type="checkbox"/> | Scale Expression | Menu Item |
| <input checked="" type="checkbox"/> | Scale Tempos | Menu Item |
| <input checked="" type="checkbox"/> | Scale Time | Menu Item |
| <input checked="" type="checkbox"/> | Select all Tracks in Range | Keystroke |
| <input checked="" type="checkbox"/> | Sequence Editor | Keystroke |
| <input checked="" type="checkbox"/> | Set Bite Gain | Menu Item |
| <input checked="" type="checkbox"/> | Set Dub Align and Spot Audio | Menu Item |
| <input checked="" type="checkbox"/> | Set Dub File | Menu Item |
| <input checked="" type="checkbox"/> | Set Gap between SoundBites | Menu Item |

Adjust Beat Detection

Command Description

Context:

Type:

Menu:

Commands

| Active | Command | Type |
|-------------------------------------|--------------------------------------------------------|-----------|
| <input checked="" type="checkbox"/> | Set Guide Align and Spot Audio | Menu Item |
| <input checked="" type="checkbox"/> | Set Guide File | Menu Item |
| <input checked="" type="checkbox"/> | Set Pitch Mode for Selected Bites | Menu Item |
| <input checked="" type="checkbox"/> | Set Pitch Mode for Selected Bites Vocals | Menu Item |
| <input checked="" type="checkbox"/> | Set Pitch Mode for Track and Selected Bites | Menu Item |
| <input checked="" type="checkbox"/> | Set Pitch Mode for Track and Selected Bites Instrument | Menu Item |
| <input checked="" type="checkbox"/> | Set Pitch Mode for Track and Selected Bites Vocals | Menu Item |
| <input checked="" type="checkbox"/> | Set Sync Point at First Beat | Menu Item |
| <input checked="" type="checkbox"/> | Set Sync Points | Menu Item |
| <input checked="" type="checkbox"/> | Set Tempo from file name | Menu Item |
| <input checked="" type="checkbox"/> | Set Track Pitch Mode Instrument | Menu Item |
| <input checked="" type="checkbox"/> | Set Track Pitch Mode Vocals | Menu Item |
| <input checked="" type="checkbox"/> | Set Track Pitch mode | Menu Item |
| <input checked="" type="checkbox"/> | Set User Time Stamp from Sequence | Menu Item |
| <input checked="" type="checkbox"/> | Setup | Menu Item |
| <input checked="" type="checkbox"/> | Shift | Keystroke |
| <input checked="" type="checkbox"/> | Shortcuts | Keystroke |
| <input checked="" type="checkbox"/> | Smooth Audio Edits | Menu Item |
| <input checked="" type="checkbox"/> | Smooth Audio Edits Again | Menu Item |
| <input checked="" type="checkbox"/> | Snip | Keystroke |
| <input checked="" type="checkbox"/> | Soundbite tempo | Menu Item |
| <input checked="" type="checkbox"/> | Spectral Effects | Menu Item |
| <input checked="" type="checkbox"/> | Splice | Keystroke |
| <input checked="" type="checkbox"/> | Split Audio | Keystroke |
| <input checked="" type="checkbox"/> | Split Notes | Keystroke |
| <input checked="" type="checkbox"/> | Split at counter | Keystroke |
| <input checked="" type="checkbox"/> | Step Record | Menu Item |
| <input checked="" type="checkbox"/> | Strip Silence | Menu Item |
| <input checked="" type="checkbox"/> | Studio | Menu Item |
| <input checked="" type="checkbox"/> | Take Automation Snapshot | Menu Item |
| <input checked="" type="checkbox"/> | Take Automation Snapshot Again | Menu Item |
| <input checked="" type="checkbox"/> | Time Stamps | Menu Item |
| <input checked="" type="checkbox"/> | Toggle Bite Volume Bypass | Menu Item |
| <input checked="" type="checkbox"/> | Tools | Keystroke |
| <input checked="" type="checkbox"/> | Tools | Keystroke |
| <input checked="" type="checkbox"/> | Tracks | Keystroke |
| <input checked="" type="checkbox"/> | Transpose | Keystroke |
| <input checked="" type="checkbox"/> | Trim | Keystroke |
| <input checked="" type="checkbox"/> | Trim Audio | Keystroke |
| <input checked="" type="checkbox"/> | Trim End | Keystroke |
| <input checked="" type="checkbox"/> | Trim start | Keystroke |
| <input checked="" type="checkbox"/> | UnFreeze Selected Tracks | Menu Item |
| <input checked="" type="checkbox"/> | Undo | Keystroke |
| <input checked="" type="checkbox"/> | Undo History | Keystroke |
| <input checked="" type="checkbox"/> | Use External Waveform Editor | Menu Item |
| <input checked="" type="checkbox"/> | VocAlign | Menu Item |
| <input checked="" type="checkbox"/> | Window | Menu Item |
| <input checked="" type="checkbox"/> | event list | Keystroke |
| <input checked="" type="checkbox"/> | graphic editor | Keystroke |
| <input checked="" type="checkbox"/> | lyrics | Keystroke |
| <input checked="" type="checkbox"/> | markers | Keystroke |
| <input checked="" type="checkbox"/> | movie | Keystroke |
| <input checked="" type="checkbox"/> | play | Keystroke |
| <input checked="" type="checkbox"/> | play play | Keystroke |
| <input checked="" type="checkbox"/> | sound bites | Keystroke |

Set Gap between SoundBites

Command Description

Context:

Type:

Menu:

Works Cited

"DP AppleScript Documentation." *Motunation*. N.p., 21 Nov. 2004. Web. 23 Aug. 2013.

"Dragon Dictate for Mac Command Cheat Sheet." *Dragon Dictate*. Nuance, n.d. Web. 23 Aug. 2013.

Grabianowski, Ed. "How Speech Recognition Works." *HowStuffWorks*. N.p., n.d. Web. 23 Aug. 2013.

Pinola, Melanie. "Speech Recognition Through the Decades: How We Ended Up With Siri." *ComputerWorld*. IDG Magazines Norge, 11 Mar. 2011. Web. 23 Aug. 2013.