

Fire Vox

built from the CLC-4-TTS Suite

by Charles L. Chen
August 29, 2007
ver. 5.4

Table of Contents

1. [Introduction](#)
 2. [List of Commands](#)
 3. [List of Options](#)
 4. [Additional Notes](#)
 5. [Changes](#)
-

1. Introduction

Fire Vox uses the CLC-4-TTS library functions to provide screen reading capabilities for Firefox within Firefox itself. Since Fire Vox uses both CLC-4-TTS and CLC-Utills, both extensions will have to be installed before Fire Vox will function correctly.

The code for Fire Vox is open source and is licensed under the terms of the GNU GPL.

2. List of Commands

To choose the TTS engine you wish to use, start Firefox and go to Tools, Fire Vox TTS Selection. Please note that you must install the FreeTTS engine before you can use it; not doing so will result in an error. Also note that if you do not have a Windows operating system that has SAPI 5 (such as Windows 2000 or XP), then you will not be able to use the SAPI 5 TTS engine.

The keys given here are only the default keys. These keys can be remapped to other keys by setting them through the Fire Vox Options dialog window. The Fire Vox Options dialog window can be accessed under the Tools menu of Firefox. Please note that all key commands for Fire Vox must be accompanied with CTRL and Shift. Alternatively, if the "Use Pause/Break as sticky modifier" is checked, instead of holding down CTRL and Shift, users can opt to toggle on "sticky modifier keys" by pressing the Pause/Break key. If sticky modifier keys are turned on, then all keys pressed will behave as if CTRL and Shift were being held. Users must toggle sticky modifier keys off by pressing Pause/Break again before they can type normally on a web page.

- **Auto Read: CTRL + Shift + A**

Starting at the current chunk, Fire Vox will read through the entire web page. The behavior is the same as calling Read Next Chunk multiple times, except this is automatic. To cancel out of auto reading at any time, use the Stop Reading command by pressing Ctrl + Shift + C.

- **Read Next Chunk: CTRL + Shift + F**

Will read the next chunk of content on the webpage. When the end of the webpage has been reached, Fire Vox will announce that it has reached the "End of Document." Reading the next chunk will cause Fire Vox to wrap around to the top of the page. The browser window will automatically scroll to the chunk that is being read. The chunk being read will also be highlighted if highlighting is enabled.

If a JavaScript alert popup has appeared, this command will cause Fire Vox to focus on the first button of that JavaScript alert. Usually, this will turn out to be the OK button. If you are inside a JavaScript alert popup, you can switch between buttons by pressing Tab.

- **Read Prior Chunk: CTRL + Shift + D**

Will read the prior chunk of content on the webpage. When the top of the webpage has been reached, Fire Vox will announce that it has reached the "End of Document." Reading the chunk prior to that will cause Fire Vox to wrap around to the bottom of the page. The browser window will automatically scroll to the chunk that is being read. The chunk being read will also be highlighted if highlighting is enabled.

- **Read Current Position: CTRL + Shift + P**

Reads the element that is at where the cursor is. This is a quick read and does not go through all the detailed information as Read Next Chunk and Read Previous Chunk do. This function can be used to quickly determine what has been entered in a form field.

If a JavaScript alert popup has appeared, this command will cause Fire Vox to repeat the contents of that alert.

- **Repeat Last Chunk: CTRL + Shift + E**

Will repeat the chunk that was just read. This function is a work in progress - the idea here is to allow users to go and have previously read sections repeated without having to shift back.

- **Stop Speaking: CTRL + Shift + C**

Will stop speaking.

- **Read Selected: CTRL + Shift + O**

The text that is currently selected will be read back to the user.

- **Query Current Object for More Information: CTRL + Shift + Q**

The current object will be queried for more information. What will be spoken depends on what the object is. For example, if it is an internal link, it will be identified as such and its destination section will be announced. If it is a cell in a table, the column and row headings will be announced along with its col and row number position. Querying it again will result in additional information from its parent object. In the case of the cell, another query would yield information about the table that it is in which would give the table summary and total number of columns and rows.

- **Spell It Out: CTRL + Shift + S**

This function is a work in progress. In future versions, I will try to adjust this function so that it only spells out the word the cursor is at, rather than the entire text of the object like it does now.

- **List of Headings Menu: CTRL + Shift + H**

Brings up a menu of submenus that contain all the levels of headings on the current document. When a heading is selected, Fire Vox will jump to that heading and read it.

- **List of Elements Menu: CTRL + Shift + L**

Brings up a menu of submenus that contain all the headings, links, images, form elements, access keys, and frames. The lists for headings, links, images, form elements, and access keys are local to the currently focused frame; the list of frames is global (includes all the frames that

are on the current web page). When an element is selected, Fire Vox will jump to that element and read it.

- **Fire Vox Options Menu: CTRL + Shift + M**
Brings up the Fire Vox Options Menu.
 - **Say Parent Node Text Content: CTRL + Shift + U**
Says the text content of the parent node of the current element. This can sometimes be useful for getting context information. [An example of such a situation can be found here.](#)
 - **Toggle Live Region Announcements: CTRL + Shift + Y**
Toggles announcements for Live Regions between "default", "tag", and "off". Default mode will cause untagged live regions to be announced - this allows users to hear live region events even if the web developer did not use the WAI-ARIA markup; however, this can sometimes be annoying if too many things are announced. Tag mode will restrict announcements only to live regions that have been marked up with WAI-ARIA; this will keep the user from being annoyed by too many untagged events happening at the same time, but the user will not be able to hear announcements to live regions that have not been tagged. Off mode will turn off live region announcements completely.
 - **Contextual Controls: CTRL + Shift + Arrow keys**
The behavior of contextual controls depends on the context in which they are used.
 - In a chat context such as ChatZilla or an AJAX based chat system that has the proper markup (such as [Reefchat](#)), pressing Up or Down will read the previous/next message in the chat transcript. Pressing Left or Right will read the previous/next message in the chat transcript that is from the same nickname as the message that the user is currently on.
-

3. List of Options

The options listed here are found in the checkboxes of the Fire Vox Options dialog window. The Fire Vox Options dialog window can be accessed under the Tools menu of Firefox or by using the Fire Vox Options Menu command (default hotkey for this command is Ctrl+Shift+M).

- **Read By Sentences**
If this is turned on, Read Next and Read Previous will cause Fire Vox to read the next / previous sentence rather than the next / previous chunk. The default is off.
- **Select Current Sentence**
If this is turned on, the sentence that is currently being read will be selected. This selection is done using the cursor, the same way that a user would select text by dragging the mouse or by holding down Shift while using the arrow keys. The default is on.

Note 1: This has no effect unless "Read By Sentences" is checked.

Note 2: This does not replace the "Highlight Chunks" option; if both are turned on, then the current chunk will be highlighted and the current sentence will be selected. If you do not want the chunk to be highlighted when using "Read By Sentences," then off "Highlight Chunks" and use only "Select Current Sentence."

- **Highlight Chunks**
If this is turned on, the chunk that is currently being read will be highlighted. The default is on.
- **Speak Events**

If this is turned on, Fire Vox will speak the text associated with events such as various items in the menus becoming selected or when an element on the page gets focused. The default is on.

- **Echo Keys**

If this is turned on, Fire Vox will echo the keys being typed. The default is on.

- **Sort Lists Alphabetically**

If this is turned on, all the Fire Vox lists will have the elements alphabetically sorted. If it is not checked, then the lists will be sorted by position. The default is off (sort by position rather than by alphabet).

- **Verbose Form Elements List**

If this is turned on, all the form elements list will have extra information that identifies what type of form element something is and what the current value of that form element is. The default is on.

- **Adjust Window on HTML Focus**

If this is turned on, Fire Vox will try to always scroll the window whenever something on the page gets focused. This keeps the focused item as close to the top of the page as possible (as long as scrolling allows it), and thus making magnifier use somewhat easier; however, this auto adjust feature causes a serious "skipping" problem for mouse users since clicking on a link will give the link focus which will then scroll the screen, usually before the link can become activated. The result of this "skipping" is that users will miss their initial mouse click much of the time. Keyboard only users do not experience this problem as it only affects the mouse. The default is off.

- **Use CSS Speech Properties**

If this is turned on, Fire Vox will try to use any embedded CSS style sheets that relate to speech properties. This support is still partial, but I am trying to develop workarounds to the technical limitations imposed by Firefox and the speech engines. [A detailed discussion of the technical issues involved with implemented CSS speech properties can be found here.](#) I believe that the "say-instead" property should be added to the CSS3 speech module and CLC-4-TTS supports this property; [here is a demo page that shows "say-instead" in action.](#)

- **Auto detect language**

If this is turned on, Fire Vox will try to detect the language the page is using and choose an appropriate voice for it. So if there is a Chinese page being displayed and you have a Chinese voice installed, Fire Vox will use the Chinese voice for that page. Note that this will only work if you have an appropriate voice installed.

- **Use Pause/Break as sticky modifier**

If this is turned on, pressing Pause/Break will toggle sticky modifiers on and off. When sticky modifiers are on, any key press is treated as if it were accompanied by CTRL and Shift. When sticky modifiers are off, key presses are treated normally. Note that using Pause/Break to turn on sticky modifiers will only work if this option is checked. Also note that if sticky modifiers are turned on, they must be turned off before you can type normally.

4. Additional Notes

[Please check out the tutorials for more information.](#)

5. Changes

- 1.0 - Initial release: Engine based on preparsing a page and changing the DOM to be easier to navigate. Manipulating the DOM may cause some slight browser instability and page distortion.
- 1.1 - Unstable experimental build: Moved to doing parsing on the fly. Not released because of instability (often misses text that should be read, highlighting broken, no autoscroll).
- 1.2 - Dynamic parsing engine: all identification is done on the fly. There are no nodes added, deleted, or replaced in the DOM. All known issues from 1.1 were resolved. In fact, this is much stabler and safer than the initial release with the preparsing method. Framed pages are readable now. Focus shifting is also implemented - this means that when a link is being read/was the last thing read, pressing Enter will actually follow the link since it will be the element on the page that has focus.
- 1.3 - Enhancements: Querying objects for more information is now available. Initial version of the spell it out function has been implemented. Overall stability has been greatly enhanced.
- 2.0 - Fire Vox Debut: Added the ability to announce menu items. Major refactoring of existing code. Released it as Fire Vox since this is not just a library demo anymore.
- 2.1 - Experimental build: Echoed keys for the address bar blank.
- 2.2 - Experimental build: Attempted to add in form support. Discovered that the current way of handling objects made form support impossible. Refactored the object handling routine. Got radio buttons to work. Still had significant bugs.
- 2.3 - Form support added: Worked out the problems in 2.2. Added support for input blank fields that are of text type.
- 2.4 - Improved web page handling. Will skip SCRIPT elements now.
- 2.5 - Much more robust - far less likely to get stuck now.
- 2.6 - Auto read function added - Finally, Fire Vox can read through a page without the user having to do Read Next Chunk repeatedly.
- 2.7 - Will now automatically announce elements when focused. JavaScript popup alerts are now handled. Read at caret added. Headings menu added.
- 2.8 - Fire Vox Options dialog created. Entire key handling system was redone to make user configurable keys. Persistent storage of options is now possible.
- 2.9 - List of elements is now available. List of headings now divides headings by their respective levels.
- 3.0 - Major Milestone Release: List of elements now includes headings, links, images, form elements, access keys, and frames. There is now a safety check to ensure that users do not use the keys reserved for the Firefox browser (G, I, R, W). The "skipping" problem that occurs with users who have speak events on and use the mouse to click has now been resolved by allowing users to turn off "Adjust window on HTML element focused" - the default is off, the reason the option is that while the auto adjustment does cause skipping for mouse users, keyboard only users do not experience the problem and having auto adjustment makes it easier to use a magnifier since the object of interest will be kept as near to the top of the screen as possible.
- 3.1 - Support for FreeTTS added. Fire Vox is now cross platform compatible and will work on Windows, Linux, Mac, and pretty much any operating system that will support Java and Firefox!
- 3.2 - Bug fix for getting stuck under some conditions. Highly visible bug in Deer Park. This bug is inactive (or very close to inactive) in Firefox 1.0.4 and earlier.
- 3.3 - Changed logic for cursor handling to reflect the new quirks related to cursor/focus/input handling in Firefox 1.5.
- 3.4 - Changed variable names used to handle chrome events to correct problems with certain events not speaking properly because of the new variable names used in Firefox 1.5.
- 3.5 - Major stability enhancement. Much less likely to get stuck reading in a loop (which was a problem in older versions on LiveJournal).
- 3.6 - Version bump to maintain compatibility with Firefox 1.5.0.1.

- 3.7 - Bug fix for Java FreeTTS not speaking properly in Fire Vox Options.
 - 3.8 - CSS speech property support added. Stability enhancement for auto reading; it should now be completely impossible to get stuck in a loop while auto reading.
 - 3.9 - Bug fixes for HTML element focus detection.
 - 4.0 - Added "Say Parent Node Text Content" function.
 - 4.1 - The list of links under the list of elements works better now. The Fire Vox Options menu will now save when Enter is used to close out of it.
 - 4.2 - Alert boxes stop speaking when the alert box is closed. Speech properties in external CSS stylesheets now work. An automatic page refresh will no longer cause users to lose their place. Select boxes work. Overall stability enhancements.
 - 4.3 - Preliminary support for Orca added. Stability enhancement for cursor control. Rules for when cursor matching should be attempted have been greatly improved. One very noticeable effect is that "skip to main content" links work much better now. Autoread is easier to stop; any key press will turn autoread off.
 - 4.4 - Added the option for reading by sentences instead of by chunks.
 - 4.5 - Added the option for auto detecting the language on the page.
 - 4.6 - Version bump for compatibility with Firefox 2.0. Basic code needed to support the Firefox Options dialog added, but not enabled since it is still incomplete.
 - 4.7 - Adds a workaround hack for a [Firefox 2.0 cursor bug](#).
 - 4.8 - Bug fix for a problem where the language will automatically switch to US English when reading by sentence. Bug fix for a problem where reading starts on the next chunk if a user clicks a sentence within the current chunk when reading by sentence.
 - 4.9 - Fire Vox is now the world's first assistive technology to support Live Regions tagged with WAI-ARIA.
 - 5.0 - Stability enhancements (won't get lost if the current element disappears, better performance) and support for more WAI-ARIA live region properties.
 - 5.1 - Stability enhancements (better handling of untagged live regions, better handling in the list of elements menu for images that are part of links, headings, etc.).
 - 5.2 - Preliminary support for ChatZilla. Introduction of context controls.
 - 5.3 - Support for WAI-ARIA channel, labelledby, describedby, and busy. Better table handling. Smarter highlight behavior; automatic unhighlight if an area that is not highlighted is clicked. Ability to use "pause/break" to toggle on sticky mod keys (no need to hold CTRL + Shift) added.
 - 5.4 - Added support for the following WAI-ARIA widgets: application, checkbox, checkboxtristate, img, listbox, log, menu, menubar, menuitem, progressbar, radio, radiogroup, slider, tree, and treeitem.
-