Emacspeak
The Complete Audio Desktop

User Manual

T. V. Raman

Last Updated: 16 November 2020

This manual documents **Emacspeak, The Complete Audio Desktop** and was last updated on 16 November 2020.

Short Contents

1 Copyright 2

2 Announcing Emacspeak Manual 2nd Edition As An Open Source Project 3

3 Background 4

4 Introduction 6

5 Installation Instructions 7

6 Basic Usage 9

7 The Emacspeak Audio Desktop 19

8 Voice Lock 22

9 Using Online Help With Emacspeak 24

10 Emacs Packages 26

11 Running Terminal Based Applications 42

12 Emacspeak Commands And Options 46

13 Emacspeak Keyboard Commands 365

14 TTS Servers 366

15 Emacspeak At Twenty 369

16 Acknowledgments 381

17 Concept Index 382

18 Key Index 383

Table of Contents

1 Copyright 2

2 Announcing Emacspeak Manual 2nd Edition As An Open Source Project 3

2.1 How To Contribute To This Manual 3
2.2 Authoring Guidelines 3
2.3 Credits 3

3 Background 4

3.1 Speech-Enabling Applications 4
3.2 Audio Formatting And Aural CSS 4
3.3 Auditory Icons 4
3.4 Summary 5

4 Introduction 6

5 Installation Instructions 7

5.1 Obtaining Emacspeak 7
5.2 Quick Installation 7
5.3 Building And Testing The Speech Server 8
5.3.1 Speech Servers 8
5.3.1.1 Testing The Selected Server. 8

6 Basic Usage. 9

6.1 Emacspeak Overview 9
6.2 Using Emacs Buffers 10
6.3 Reading Without Moving 10
6.4 Speech System Commands 13
6.4.1 Character, Word And Line Echo. 13
6.4.2 Setting TTS Characteristics. 13
6.4.3 Miscellaneous Speech Commands 15
6.5 Voice Lock Mode 16
6.6 Status Commands 17

7 The Emacspeak Audio Desktop. 19

7.1 Objects Making Up The Emacspeak Desktop 19
7.2 An Object-Oriented Desktop 19
7.3 Emacspeak Specializes Aural Interaction 20
7.3.1 Audio Formatted Output 20
7.3.2 Structured Navigation: 20
7.3.3 Navigating The Desktop 20
7.3.4 Everything Is Searchable: 20

8 Voice Lock 22
8.1 How It All Works 22
8.2 What this gives 23

9 Using Online Help With Emacspeak.......... 24

10 Emacs Packages. 26
10.1 Document Authoring 26
10.1.1 Creating Well-formatted Documents 26
10.1.2 Searching, Replacing, And Spell Checking 27
Spell Checking 28
10.2 Structured Editing And Templates 28
10.2.1 Outline Editing 29
10.2.2 Template-based Authoring 30
10.2.3 Maintaining Structured Data 30
10.3 Browsing Structure 31
10.4 Web Browsing 31
10.4.1 Intro 31
10.4.2 emacs-w3 32
10.4.3 Add-ons 36
10.4.4 Conclusion 38
10.5 Messaging 38
10.6 Editing Code 38
10.7 Development Environment 38
10.8 Desktop Management 38
10.9 Personal Information Management 39
10.10 Desktop Applications 39
10.10.1 Spread Sheets 39
10.10.2 Forms Mode 39
10.10.3 OCR — Reading Print Documents 39
10.10.3.1 Emacspeak OCR Mode 40

11 Running Terminal Based Applications. 42
11.1 Char Sub-mode of Term Mode 42
11.2 Line Sub-mode of Term Mode 44
11.3 Eterm Mode Commands 45

12 Emacspeak Commands And Options 46
12.1 acss-structure 46
12.2 amixer 47
12.2.1 Amixer Commands 47
12.2.1.1 amixer 47
12.2.1.2 amixer-store 47
12.2.2 amixer Options 47
12.3 cd-tool 47
12.3.1 Cd-Tool Commands 47

12.3.1.1 cd-tool 47
12.3.2 cd-tool Options 48
12.4 dectalk-voices 48
12.4.1 Dectalk-Voices Commands 48
12.4.1.1 dectalk 48
12.4.2 dectalk-voices Options 48
12.5 dom-addons 48
12.6 dtk-interp 49
12.7 dtk-speak 49
12.7.1 Dtk-Speak Commands 49
12.7.1.1 dtk-add-cleanup-pattern 49
12.7.1.2 dtk-cloud 49
12.7.1.3 dtk-list-languages 49
12.7.1.4 dtk-local-server 49
12.7.1.5 dtk-notify-initialize 50
12.7.1.6 dtk-notify-shutdown 50
12.7.1.7 dtk-notify-stop 50
12.7.1.8 dtk-reset-state 50
12.7.1.9 dtk-select-server 50
12.7.1.10 dtk-set-character-scale 51
12.7.1.11 dtk-set-chunk-separator-syntax 51
12.7.1.12 dtk-set-language 51
12.7.1.13 dtk-set-next-language 51
12.7.1.14 dtk-set-predefined-speech-rate 51
12.7.1.15 dtk-set-preferred-language 52
12.7.1.16 dtk-set-previous-language 53
12.7.1.17 dtk-set-punctuations 53
12.7.1.18 dtk-set-punctuations-to-all 53
12.7.1.19 dtk-set-punctuations-to-some 53
12.7.1.20 dtk-set-rate 53
12.7.1.21 dtk-stop 54
12.7.1.22 dtk-toggle-allcaps-beep 54
12.7.1.23 dtk-toggle-capitalization 54
12.7.1.24 dtk-toggle-punctuation-mode 54
12.7.1.25 dtk-toggle-quiet 55
12.7.1.26 dtk-toggle-speak-nonprinting-chars 55
12.7.1.27 dtk-toggle-split-caps 55
12.7.1.28 dtk-toggle-splitting-on-white-space 55
12.7.1.29 dtk-toggle-strip-octals 56
12.7.1.30 tts-restart 56
12.7.1.31 tts-speak-version 56
12.7.2 dtk-speak Options 56
12.8 dtk-unicode 57
12.8.1 Dtk-Unicode Commands 57
12.8.1.1 dtk-unicode-customize-char 57
12.8.1.2 dtk-unicode-uncustomize-char 57
12.8.2 dtk-unicode Options 57
12.9 emacspeak 58

12.9.1 emacspeak Options 58
12.10 emacspeak-2048 62
12.10.1 Emacspeak-2048 Commands 62
12.10.1.1 emacspeak-2048-add-column 62
12.10.1.2 emacspeak-2048-add-row 62
12.10.1.3 emacspeak-2048-drop-column 62
12.10.1.4 emacspeak-2048-drop-row 62
12.10.1.5 emacspeak-2048-export 63
12.10.1.6 emacspeak-2048-import 63
12.10.1.7 emacspeak-2048-pop-state 63
12.10.1.8 emacspeak-2048-prune-stack 63
12.10.1.9 emacspeak-2048-push-state 63
12.10.1.10 emacspeak-2048-randomize-game 63
12.10.1.11 emacspeak-2048-score 63
12.10.1.12 emacspeak-2048-speak-board 63
12.10.1.13 emacspeak-2048-speak-transposed-board 64
12.11 emacspeak-abc-mode 64
12.12 emacspeak-actions 64
12.13 emacspeak-add-log 64
12.13.1 emacspeak-add-log Options 64
12.14 emacspeak-advice 65
12.14.1 Emacspeak-Advice Commands 65
12.14.1.1 emacspeak-eldoc-speak-doc 65
12.14.2 emacspeak-advice Options 65
12.15 emacspeak-amark 66
12.15.1 Emacspeak-Amark Commands 66
12.15.1.1 emacspeak-amark-add 66
12.15.1.2 emacspeak-amark-find 66
12.15.1.3 emacspeak-amark-save 66
12.16 emacspeak-analog 66
12.16.1 Emacspeak-Analog Commands 66
12.16.1.1 emacspeak-analog-backward-field-or-char 66
12.16.1.2 emacspeak-analog-forward-field-or-char 67
12.16.1.3 emacspeak-analog-next-line 67
12.16.1.4 emacspeak-analog-previous-line 67
12.16.1.5 emacspeak-analog-speak-this-field 67
12.17 emacspeak-ansi-color 67
12.18 emacspeak-apt-sources 67
12.19 emacspeak-arc 67
12.19.1 Emacspeak-Arc Commands 67
12.19.1.1 emacspeak-arc-speak-file-modification-time 67
12.19.1.2 emacspeak-arc-speak-file-name 67
12.19.1.3 emacspeak-arc-speak-file-permissions 68
12.19.1.4 emacspeak-arc-speak-file-size 68
12.20 emacspeak-auctex 68
12.20.1 Emacspeak-Auctex Commands 68
12.20.1.1 emacspeak-auctex-comma-at-end-of-word 68
12.20.1.2 emacspeak-auctex-end-of-word 68

12.20.1.3 emacspeak-auctex-lacheck-buffer-file 68
12.20.1.4 emacspeak-auctex-tex-tie-current-word 68
12.20.2 emacspeak-auctex Options 68
12.21 emacspeak-bbc 69
12.21.1 Emacspeak-Bbc Commands 69
12.21.1.1 emacspeak-bbc-get-iplayer-stream-pid 69
12.21.1.2 emacspeak-bbc-get-iplayer-stream-url 70
12.21.1.3 emacspeak-bbc-schedule 70
12.22 emacspeak-bbdb 70
12.22.1 emacspeak-bbdb Options 70
12.23 emacspeak-bibtex 70
12.24 emacspeak-bookmark 70
12.25 emacspeak-bookshare 70
12.25.1 requirements 70
12.25.2 Usage 71
12.25.3 Sample Interaction 71
12.25.4 Emacspeak-Bookshare Commands 71
12.25.4.1 emacspeak-bookshare 71
12.25.4.2 emacspeak-bookshare-action 71
12.25.4.3 emacspeak-bookshare-author-search 71
12.25.4.4 emacspeak-bookshare-browse 72
12.25.4.5 emacspeak-bookshare-browse-latest 72
12.25.4.6 emacspeak-bookshare-browse-popular 72
12.25.4.7 emacspeak-bookshare-debugInfo-handler 72
12.25.4.8 emacspeak-bookshare-download-audio 72
12.25.4.9 emacspeak-bookshare-download-audio-at-point 72
12.25.4.10 emacspeak-bookshare-download-brf 72
12.25.4.11 emacspeak-bookshare-download-brf-at-point 72
12.25.4.12 emacspeak-bookshare-download-daisy 73
12.25.4.13 emacspeak-bookshare-download-daisy-at-point 73
12.25.4.14 emacspeak-bookshare-download-epub-3 73
12.25.4.15 emacspeak-bookshare-download-epub-3-at-point . . . 73
12.25.4.16 emacspeak-bookshare-download-internal 73
12.25.4.17 emacspeak-bookshare-eww 73
12.25.4.18 emacspeak-bookshare-expand-at-point 73
12.25.4.19 emacspeak-bookshare-extract-and-view 73
12.25.4.20 emacspeak-bookshare-extract-xml 74
12.25.4.21 emacspeak-bookshare-flush-lines 74
12.25.4.22 emacspeak-bookshare-fulltext 74
12.25.4.23 emacspeak-bookshare-fulltext-search 74
12.25.4.24 emacspeak-bookshare-get-more-results 74
12.25.4.25 emacspeak-bookshare-id-search 74
12.25.4.26 emacspeak-bookshare-isbn-search 74
12.25.4.27 emacspeak-bookshare-list-preferences 74
12.25.4.28 emacspeak-bookshare-mode 75
12.25.4.29 emacspeak-bookshare-next-result 76
12.25.4.30 emacspeak-bookshare-periodical-list 76
12.25.4.31 emacspeak-bookshare-previous-result 76

12.25.4.32 emacspeak-bookshare-set-preference 76
12.25.4.33 emacspeak-bookshare-since-search 76
12.25.4.34 emacspeak-bookshare-title-search 77
12.25.4.35 emacspeak-bookshare-toc 77
12.25.4.36 emacspeak-bookshare-toc-at-point 77
12.25.4.37 emacspeak-bookshare-unpack-at-point 77
12.25.4.38 emacspeak-bookshare-url-executor 77
12.25.4.39 emacspeak-bookshare-version-handler 77
12.25.4.40 emacspeak-bookshare-view 77
12.25.4.41 emacspeak-bookshare-view-at-point 78
12.25.4.42 emacspeak-bookshare-view-page-range 78
12.25.5 emacspeak-bookshare Options 78
12.26 emacspeak-browse-kill-ring 78
12.27 emacspeak-bs 78
12.27.1 Emacspeak-Bs Commands 78
12.27.1.1 emacspeak-bs-speak-buffer-line 78
12.28 emacspeak-buff-menu 78
12.28.1 Emacspeak-Buff-Menu Commands 78
12.28.1.1 emacspeak-list-buffers-next-line 78
12.28.1.2 emacspeak-list-buffers-previous-line 79
12.28.1.3 emacspeak-list-buffers-speak-buffer-line 79
12.28.1.4 emacspeak-list-buffers-speak-buffer-name 79
12.28.2 emacspeak-buff-menu Options 79
12.29 emacspeak-c 79
12.29.1 Emacspeak-C Commands 79
12.29.1.1 emacspeak-c-speak-semantics 79
12.30 emacspeak-calc 79
12.31 emacspeak-calculator 79
12.32 emacspeak-calendar 79
12.32.1 Emacspeak-Calendar Commands 80
12.32.1.1 emacspeak-appt-repeat-announcement 80
12.32.1.2 emacspeak-calendar-setup-sunrise-sunset 80
12.32.1.3 emacspeak-calendar-speak-date 80
12.32.1.4 emacspeak-calendar-sunrise-sunset 80
12.32.2 emacspeak-calendar Options 80
12.33 emacspeak-chess 80
12.33.1 Navigating And Examining The Board 81
12.33.2 Examining Games 81
12.33.3 Emacspeak-Chess Commands 82
12.33.3.1 emacspeak-chess-east 82
12.33.3.2 emacspeak-chess-goto-target 82
12.33.3.3 emacspeak-chess-jump 82
12.33.3.4 emacspeak-chess-look-east 82
12.33.3.5 emacspeak-chess-look-king 82
12.33.3.6 emacspeak-chess-look-knight 82
12.33.3.7 emacspeak-chess-look-north 82
12.33.3.8 emacspeak-chess-look-northeast 82
12.33.3.9 emacspeak-chess-look-northwest 82

12.33.3.10 emacspeak-chess-look-south 82
12.33.3.11 emacspeak-chess-look-southeast 83
12.33.3.12 emacspeak-chess-look-southwest 83
12.33.3.13 emacspeak-chess-look-west 83
12.33.3.14 emacspeak-chess-north 83
12.33.3.15 emacspeak-chess-northeast 83
12.33.3.16 emacspeak-chess-northwest 83
12.33.3.17 emacspeak-chess-south 83
12.33.3.18 emacspeak-chess-southeast 83
12.33.3.19 emacspeak-chess-southwest 83
12.33.3.20 emacspeak-chess-speak-board 83
12.33.3.21 emacspeak-chess-speak-piece-squares 84
12.33.3.22 emacspeak-chess-speak-that-square 84
12.33.3.23 emacspeak-chess-speak-this-move 84
12.33.3.24 emacspeak-chess-speak-this-square 84
12.33.3.25 emacspeak-chess-speak-who-targets 84
12.33.3.26 emacspeak-chess-view-rank-or-file 84
12.33.3.27 emacspeak-chess-west 84
12.34 emacspeak-cider 84
12.34.1 emacspeak-cider Options 85
12.35 emacspeak-ciel 85
12.36 emacspeak-clojure 85
12.37 emacspeak-cmuscheme 85
12.38 emacspeak-comint 85
12.38.1 Emacspeak-Comint Commands 85
12.38.1.1 emacspeak-completion-pick-completion 85
12.38.1.2 emacspeak-toggle-comint-autospeak 85
12.38.1.3 emacspeak-toggle-comint-output-monitor 85
12.38.1.4 emacspeak-toggle-inaudible-or-comint-autospeak . . . 86
12.38.2 emacspeak-comint Options 86
12.39 emacspeak-company 86
12.39.1 emacspeak-company Options 86
12.40 emacspeak-compile 87
12.40.1 Emacspeak-Compile Commands 87
12.40.1.1 emacspeak-compilation-speak-error 87
12.40.2 emacspeak-compile Options 87
12.41 emacspeak-cperl 88
12.42 emacspeak-custom 88
12.42.1 Emacspeak-Custom Commands 88
12.42.1.1 emacspeak-custom-goto-group 88
12.42.1.2 emacspeak-custom-goto-toolbar 88
12.42.2 emacspeak-custom Options 88
12.43 emacspeak-dbus 90
12.43.1 Overview 90
12.43.2 Emacspeak-Dbus Commands 91
12.43.2.1 emacspeak-dbus-lock-screen 91
12.43.2.2 emacspeak-dbus-sleep-disable 91
12.43.2.3 emacspeak-dbus-sleep-enable 91

12.43.2.4 emacspeak-dbus-udisks-disable 91
12.43.2.5 emacspeak-dbus-udisks-enable 91
12.43.2.6 emacspeak-dbus-upower-disable 91
12.43.2.7 emacspeak-dbus-upower-enable 91
12.43.2.8 emacspeak-screen-saver-mode 92
12.44 emacspeak-deadgrep 92
12.45 emacspeak-debugger 92
12.46 emacspeak-desktop 92
12.47 emacspeak-dictionary 92
12.48 emacspeak-diff-mode 92
12.48.1 emacspeak-diff-mode Options 92
12.49 emacspeak-dired 94
12.49.1 Emacspeak-Dired Commands 94
12.49.1.1 emacspeak-dired-csv-open 94
12.49.1.2 emacspeak-dired-downloads 94
12.49.1.3 emacspeak-dired-epub-eww 94
12.49.1.4 emacspeak-dired-eww-open 94
12.49.1.5 emacspeak-dired-label-fields 94
12.49.1.6 emacspeak-dired-md-open 94
12.49.1.7 emacspeak-dired-midi-play 94
12.49.1.8 emacspeak-dired-open-this-file 95
12.49.1.9 emacspeak-dired-pdf-open 95
12.49.1.10 emacspeak-dired-rpm-query-in-dired 95
12.49.1.11 emacspeak-dired-show-file-type 95
12.49.1.12 emacspeak-dired-speak-file-access-time 95
12.49.1.13 emacspeak-dired-speak-file-modification-time 95
12.49.1.14 emacspeak-dired-speak-file-permissions 95
12.49.1.15 emacspeak-dired-speak-file-size 95
12.49.1.16 emacspeak-dired-speak-header-line 95
12.49.1.17 emacspeak-dired-speak-symlink-target 96
12.49.1.18 emacspeak-locate-play-results-as-playlist 96
12.49.2 emacspeak-dired Options 96
12.50 emacspeak-dismal 97
12.50.1 Emacspeak-Dismal Commands 97
12.50.1.1 emacspeak-dismal-backward-col-and-summarize 97
12.50.1.2 emacspeak-dismal-backward-row-and-summarize 97
12.50.1.3 emacspeak-dismal-col-summarize 97
12.50.1.4 emacspeak-dismal-display-cell-expression 97
12.50.1.5 emacspeak-dismal-display-cell-value 97
12.50.1.6 emacspeak-dismal-display-cell-with-col-header 98
12.50.1.7 emacspeak-dismal-display-cell-with-row-header 98
12.50.1.8 emacspeak-dismal-forward-col-and-summarize 98
12.50.1.9 emacspeak-dismal-forward-row-and-summarize 98
12.50.1.10 emacspeak-dismal-row-summarize 98
12.50.1.11 emacspeak-dismal-set-col-summarizer-list 98
12.50.1.12 emacspeak-dismal-set-row-summarizer-list 98
12.50.1.13 emacspeak-dismal-set-sheet-summarizer-list 98
12.50.1.14 emacspeak-dismal-sheet-summarize 99

12.51 emacspeak-dumb-jump 99
12.52 emacspeak-ecb 99
12.52.1 Emacspeak-Ecb Commands 99
12.52.1.1 emacspeak-ecb-speak-window-directories 99
12.52.1.2 emacspeak-ecb-speak-window-history 99
12.52.1.3 emacspeak-ecb-speak-window-methods 99
12.52.1.4 emacspeak-ecb-speak-window-sources 99
12.52.1.5 emacspeak-ecb-tree-backspace 99
12.52.1.6 emacspeak-ecb-tree-clear 99
12.53 emacspeak-eclim 99
12.54 emacspeak-ediff 100
12.54.1 Emacspeak-Ediff Commands 100
12.54.1.1 emacspeak-ediff-speak-current-difference 100
12.54.2 emacspeak-ediff Options 100
12.55 emacspeak-eglot 101
12.55.1 emacspeak-eglot Options 101
12.56 emacspeak-ein 101
12.56.1 Emacspeak-Ein Commands 101
12.56.1.1 emacspeak-ein-speak-current-cell 102
12.56.2 emacspeak-ein Options 102
12.57 emacspeak-elfeed 103
12.57.1 Emacspeak-Elfeed Commands 103
12.57.1.1 emacspeak-elfeed-eww-entry-at-point 103
12.57.1.2 emacspeak-elfeed-filter-entry-at-point 103
12.57.1.3 emacspeak-elfeed-next-entry 103
12.57.1.4 emacspeak-elfeed-previous-entry 103
12.57.1.5 emacspeak-elfeed-speak-entry-at-point 103
12.58 emacspeak-elisp-refs 104
12.59 emacspeak-elpy 104
12.60 emacspeak-elscreen 104
12.61 emacspeak-emms 104
12.61.1 Emacspeak-Emms Commands 104
12.61.1.1 emacspeak-emms-speak-current-track 104
12.62 emacspeak-enriched 104
12.62.1 Emacspeak-Enriched Commands 104
12.62.1.1 emacspeak-enriched-voiceify-faces 104
12.63 emacspeak-entertain 104
12.63.1 Emacspeak-Entertain Commands 104
12.63.1.1 emacspeak-hangman-speak-guess 104
12.63.1.2 emacspeak-hangman-speak-statistics 105
12.63.2 emacspeak-entertain Options 105
12.64 emacspeak-epa 105
12.64.1 emacspeak-epa Options 105
12.65 emacspeak-eperiodic 106
12.65.1 Emacspeak-Eperiodic Commands 106
12.65.1.1 emacspeak-eperiodic-goto-property-section 106
12.65.1.2 emacspeak-eperiodic-next-line 106
12.65.1.3 emacspeak-eperiodic-play-description 106

12.65.1.4 emacspeak-eperiodic-previous-line 106
12.65.1.5 emacspeak-eperiodic-speak-current-element 106
12.66 emacspeak-epub 106
12.66.1 Introduction 106
12.66.2 Organizing EBooks On The Emacspeak Desktop 107
12.66.3 Integrating With Project Gutenberg 107
12.66.4 Calibre Integration 108
12.66.5 Reading EBooks From The Bookshelf 108
12.66.6 Emacspeak-Epub Commands 108
12.66.6.1 emacspeak-calibre-mode 108
12.66.6.2 emacspeak-epub 108
12.66.6.3 emacspeak-epub-bookshelf-add-directory 109
12.66.6.4 emacspeak-epub-bookshelf-add-epub 109
12.66.6.5 emacspeak-epub-bookshelf-calibre-author 109
12.66.6.6 emacspeak-epub-bookshelf-calibre-search 109
12.66.6.7 emacspeak-epub-bookshelf-calibre-title 109
12.66.6.8 emacspeak-epub-bookshelf-clear 109
12.66.6.9 emacspeak-epub-bookshelf-load 110
12.66.6.10 emacspeak-epub-bookshelf-open 110
12.66.6.11 emacspeak-epub-bookshelf-open-epub 110
12.66.6.12 emacspeak-epub-bookshelf-redraw 110
12.66.6.13 emacspeak-epub-bookshelf-refresh 110
12.66.6.14 emacspeak-epub-bookshelf-remove-directory 110
12.66.6.15 emacspeak-epub-bookshelf-remove-this-book 110
12.66.6.16 emacspeak-epub-bookshelf-rename 111
12.66.6.17 emacspeak-epub-bookshelf-save 111
12.66.6.18 emacspeak-epub-browse-files 111
12.66.6.19 emacspeak-epub-calibre-dired-at-point 111
12.66.6.20 emacspeak-epub-calibre-results 111
12.66.6.21 emacspeak-epub-delete 111
12.66.6.22 emacspeak-epub-eww 111
12.66.6.23 emacspeak-epub-google 111
12.66.6.24 emacspeak-epub-gutenberg-catalog 112
12.66.6.25 emacspeak-epub-gutenberg-download 112
12.66.6.26 emacspeak-epub-locate-epubs 112
12.66.6.27 emacspeak-epub-mode 112
12.66.6.28 emacspeak-epub-next 113
12.66.6.29 emacspeak-epub-open 113
12.66.6.30 emacspeak-epub-open-with-nov 113
12.66.6.31 emacspeak-epub-previous 113
12.66.6.32 emacspeak-epub-url-executor 113
12.66.7 emacspeak-epub Options 114
12.67 emacspeak-erc 114
12.67.1 Emacspeak-Erc Commands 114
12.67.1.1 emacspeak-erc-add-name-to-monitor 114
12.67.1.2 emacspeak-erc-delete-name-from-monitor 114
12.67.1.3 emacspeak-erc-setup-cricket-rules 114
12.67.1.4 emacspeak-erc-toggle-my-monitor 114

12.67.1.5 emacspeak-erc-toggle-room-monitor 115
12.67.1.6 emacspeak-erc-toggle-speak-all-participants 115
12.67.2 emacspeak-erc Options 115
12.68 emacspeak-eshell 115
12.68.1 emacspeak-eshell Options 115
12.69 emacspeak-ess 116
12.70 emacspeak-etable 116
12.70.1 Emacspeak-Etable Commands 116
12.70.1.1 emacspeak-etable-speak-cell 116
12.71 emacspeak-eterm 117
12.71.1 Emacspeak-Eterm Commands 117
12.71.1.1 emacspeak-eterm-copy-region-to-register 117
12.71.1.2 emacspeak-eterm-define-window 117
12.71.1.3 emacspeak-eterm-describe-window 117
12.71.1.4 emacspeak-eterm-goto-line 117
12.71.1.5 emacspeak-eterm-kill-ring-save-region 117
12.71.1.6 emacspeak-eterm-maybe-send-raw 118
12.71.1.7 emacspeak-eterm-paste-register 118
12.71.1.8 emacspeak-eterm-pointer-backward-word 118
12.71.1.9 emacspeak-eterm-pointer-down 118
12.71.1.10 emacspeak-eterm-pointer-forward-word 118
12.71.1.11 emacspeak-eterm-pointer-left 118
12.71.1.12 emacspeak-eterm-pointer-right 119
12.71.1.13 emacspeak-eterm-pointer-to-bottom 119
12.71.1.14 emacspeak-eterm-pointer-to-cursor 119
12.71.1.15 emacspeak-eterm-pointer-to-left-edge 119
12.71.1.16 emacspeak-eterm-pointer-to-next-color-change . . . 119
12.71.1.17 emacspeak-eterm-pointer-to-previous-color-change . . 119
12.71.1.18 emacspeak-eterm-pointer-to-right-edge 119
12.71.1.19 emacspeak-eterm-pointer-to-top 120
12.71.1.20 emacspeak-eterm-pointer-up 120
12.71.1.21 emacspeak-eterm-search-backward 120
12.71.1.22 emacspeak-eterm-search-forward 120
12.71.1.23 emacspeak-eterm-set-filter-window 120
12.71.1.24 emacspeak-eterm-set-focus-window 120
12.71.1.25 emacspeak-eterm-set-marker 121
12.71.1.26 emacspeak-eterm-speak-cursor 121
12.71.1.27 emacspeak-eterm-speak-pointer 121
12.71.1.28 emacspeak-eterm-speak-pointer-char 121
12.71.1.29 emacspeak-eterm-speak-pointer-line 121
12.71.1.30 emacspeak-eterm-speak-pointer-word 121
12.71.1.31 emacspeak-eterm-speak-predefined-window 121
12.71.1.32 emacspeak-eterm-speak-screen 121
12.71.1.33 emacspeak-eterm-speak-window 122
12.71.1.34 emacspeak-eterm-toggle-filter-window 122
12.71.1.35 emacspeak-eterm-toggle-focus-window 122
12.71.1.36 emacspeak-eterm-toggle-pointer-mode 122
12.71.1.37 emacspeak-eterm-toggle-review 122

12.71.1.38 emacspeak-eterm-yank-window 122
12.71.1.39 emacspeak-toggle-eterm-autospeak 122
12.71.2 emacspeak-eterm Options 123
12.72 emacspeak-eudc 123
12.72.1 Emacspeak-Eudc Commands 123
12.72.1.1 emacspeak-eudc-send-mail 123
12.73 emacspeak-evil 123
12.73.1 Emacspeak-Evil Commands 123
12.73.1.1 emacspeak-evil-toggle-evil 123
12.73.2 emacspeak-evil Options 123
12.74 emacspeak-eww 124
12.74.1 Structured Navigation 124
12.74.2 Filtering Content Using The DOM 125
12.74.3 Diving Into (Focusing) On Specific Content 126
12.74.4 Updated Commands For Following Links 126
12.74.5 Table Browsing 126
12.74.6 Miscellaneous Commands 127
12.74.7 Filtering Content Using XSLT And XPath 127
12.74.8 EWW And EBooks On The Emacspeak Audio Desktop . . 129
12.74.9 Emacspeak-Eww Commands 129
12.74.9.1 emacspeak-eww-add-mark 129
12.74.9.2 emacspeak-eww-curl-play-media-at-point 129
12.74.9.3 emacspeak-eww-delete-mark 129
12.74.9.4 emacspeak-eww-dive-into-div 129
12.74.9.5 emacspeak-eww-fillin-form-field 129
12.74.9.6 emacspeak-eww-google-knowledge-card 130
12.74.9.7 emacspeak-eww-links-rel 130
12.74.9.8 emacspeak-eww-marks-load 130
12.74.9.9 emacspeak-eww-marks-save 130
12.74.9.10 emacspeak-eww-masquerade 130
12.74.9.11 emacspeak-eww-next-dd 130
12.74.9.12 emacspeak-eww-next-dl 130
12.74.9.13 emacspeak-eww-next-dt 130
12.74.9.14 emacspeak-eww-next-element 131
12.74.9.15 emacspeak-eww-next-element-from-history 131
12.74.9.16 emacspeak-eww-next-element-like-this 131
12.74.9.17 emacspeak-eww-next-h 131
12.74.9.18 emacspeak-eww-next-h1 131
12.74.9.19 emacspeak-eww-next-h2 131
12.74.9.20 emacspeak-eww-next-h3 131
12.74.9.21 emacspeak-eww-next-h4 132
12.74.9.22 emacspeak-eww-next-h5 132
12.74.9.23 emacspeak-eww-next-h6 132
12.74.9.24 emacspeak-eww-next-li 132
12.74.9.25 emacspeak-eww-next-ol 132
12.74.9.26 emacspeak-eww-next-p 132
12.74.9.27 emacspeak-eww-next-table 132
12.74.9.28 emacspeak-eww-next-ul 133

12.74.9.29 emacspeak-eww-open-mark 133
12.74.9.30 emacspeak-eww-play-media-at-point 133
12.74.9.31 emacspeak-eww-previous-dd 133
12.74.9.32 emacspeak-eww-previous-dl 133
12.74.9.33 emacspeak-eww-previous-dt 133
12.74.9.34 emacspeak-eww-previous-element 134
12.74.9.35 emacspeak-eww-previous-element-from-history . . . 134
12.74.9.36 emacspeak-eww-previous-element-like-this 134
12.74.9.37 emacspeak-eww-previous-h 134
12.74.9.38 emacspeak-eww-previous-h1 134
12.74.9.39 emacspeak-eww-previous-h2 134
12.74.9.40 emacspeak-eww-previous-h3 134
12.74.9.41 emacspeak-eww-previous-h4 135
12.74.9.42 emacspeak-eww-previous-h5 135
12.74.9.43 emacspeak-eww-previous-h6 135
12.74.9.44 emacspeak-eww-previous-li 135
12.74.9.45 emacspeak-eww-previous-ol 135
12.74.9.46 emacspeak-eww-previous-p 135
12.74.9.47 emacspeak-eww-previous-table 135
12.74.9.48 emacspeak-eww-previous-ul 136
12.74.9.49 emacspeak-eww-reading-settings 136
12.74.9.50 emacspeak-eww-restore 136
12.74.9.51 emacspeak-eww-shell-command-on-url-at-point . . . 136
12.74.9.52 emacspeak-eww-smart-tabs 136
12.74.9.53 emacspeak-eww-smart-tabs-add 136
12.74.9.54 emacspeak-eww-smart-tabs-load 136
12.74.9.55 emacspeak-eww-smart-tabs-save 137
12.74.9.56 emacspeak-eww-speak-this-element 137
12.74.9.57 emacspeak-eww-table-data 137
12.74.9.58 emacspeak-eww-table-next-cell 137
12.74.9.59 emacspeak-eww-table-next-row 137
12.74.9.60 emacspeak-eww-table-previous-cell 137
12.74.9.61 emacspeak-eww-table-previous-row 137
12.74.9.62 emacspeak-eww-table-speak-cell 138
12.74.9.63 emacspeak-eww-table-speak-dimensions 138
12.74.9.64 emacspeak-eww-tags-at-point 138
12.74.9.65 emacspeak-eww-update-title 138
12.74.10 emacspeak-eww Options 138
12.75 emacspeak-feeds 139
12.75.1 Emacspeak-Feeds Commands 139
12.75.1.1 emacspeak-feeds-add-feed 139
12.75.1.2 emacspeak-feeds-archive-feeds 139
12.75.1.3 emacspeak-feeds-atom-display 139
12.75.1.4 emacspeak-feeds-browse 140
12.75.1.5 emacspeak-feeds-fastload-feeds 140
12.75.1.6 emacspeak-feeds-opml-display 140
12.75.1.7 emacspeak-feeds-restore-feeds 140
12.75.1.8 emacspeak-feeds-rss-display 140

12.75.2 emacspeak-feeds Options 140
12.76 emacspeak-filtertext 141
12.76.1 Emacspeak-Filtertext Commands 141
12.76.1.1 emacspeak-filtertext 141
12.76.1.2 emacspeak-filtertext-mode 141
12.76.1.3 emacspeak-filtertext-revert 141
12.77 emacspeak-fix-interactive 141
12.77.1 Emacspeak-Fix-Interactive Commands 142
12.77.1.1 emacspeak-fix-all-recent-commands 142
12.77.1.2 emacspeak-fix-commands-loaded-from 142
12.78 emacspeak-flycheck 142
12.78.1 emacspeak-flycheck Options 142
12.79 emacspeak-flymake 143
12.79.1 emacspeak-flymake Options 143
12.80 emacspeak-flyspell 143
12.80.1 emacspeak-flyspell Options 144
12.81 emacspeak-folding 144
12.82 emacspeak-forge 144
12.82.1 emacspeak-forge Options 144
12.83 emacspeak-forms 145
12.83.1 Emacspeak-Forms Commands 145
12.83.1.1 emacspeak-forms-find-file 145
12.83.1.2 emacspeak-forms-flush-unwanted-records 145
12.83.1.3 emacspeak-forms-rerun-filter 145
12.83.1.4 emacspeak-forms-speak-field 145
12.83.1.5 emacspeak-forms-summarize-current-position 145
12.83.1.6 emacspeak-forms-summarize-current-record 145
12.84 emacspeak-geiser 145
12.85 emacspeak-gh-explorer 145
12.85.1 Emacspeak-Gh-Explorer Commands 145
12.85.1.1 emacspeak-gh-explorer-next 146
12.85.1.2 emacspeak-gh-explorer-previous 146
12.86 emacspeak-gnuplot 146
12.87 emacspeak-gnus 146
12.87.1 Emacspeak-Gnus Commands 146
12.87.1.1 emacspeak-gnus-personal-gmail-last-week 146
12.87.1.2 emacspeak-gnus-personal-gmail-recent 146
12.87.1.3 emacspeak-gnus-summary-catchup-quietly-and-exit . . 146
12.87.2 emacspeak-gnus Options 146
12.88 emacspeak-go-mode 153
12.89 emacspeak-gomoku 153
12.89.1 Emacspeak-Gomoku Commands 153
12.89.1.1 emacspeak-gomoku-display-statistics 153
12.89.1.2 emacspeak-gomoku-goto-x-y 153
12.89.1.3 emacspeak-gomoku-show-current-column 153
12.89.1.4 emacspeak-gomoku-show-current-negative-diagonal . . 153
12.89.1.5 emacspeak-gomoku-show-current-positive-diagonal . . 153
12.89.1.6 emacspeak-gomoku-show-current-row 153

12.89.1.7 emacspeak-gomoku-speak-emacs-previous-move . . . 153
12.89.1.8 emacspeak-gomoku-speak-humans-previous-move . . 154
12.89.1.9 emacspeak-gomoku-speak-number-of-moves 154
12.89.1.10 emacspeak-gomoku-speak-square 154
12.90 emacspeak-google 154
12.90.1 Emacspeak-Google Commands 154
12.90.1.1 emacspeak-google-extract-from-cache 154
12.90.1.2 emacspeak-google-knowledge-search 154
12.90.1.3 emacspeak-google-on-this-site 154
12.90.1.4 emacspeak-google-open-link 154
12.90.1.5 emacspeak-google-show-toolbelt 155
12.90.1.6 emacspeak-google-sign-in 155
12.90.1.7 emacspeak-google-sign-out 155
12.90.1.8 emacspeak-google-similar-to-this-page 155
12.90.1.9 emacspeak-google-toolbelt-change 155
12.90.1.10 emacspeak-google-toolbelt-change-Shopping 155
12.90.1.11 emacspeak-google-toolbelt-change-blog 155
12.90.1.12 emacspeak-google-toolbelt-change-books 155
12.90.1.13 emacspeak-google-toolbelt-change-books-format . . 155
12.90.1.14 emacspeak-google-toolbelt-change-books-type 155
12.90.1.15 emacspeak-google-toolbelt-change-books-viewability . . 156
12.90.1.16 emacspeak-google-toolbelt-change-commercial 156
12.90.1.17 emacspeak-google-toolbelt-change-commercial-prices . . 156
12.90.1.18 emacspeak-google-toolbelt-change-date-filter 156
12.90.1.19 emacspeak-google-toolbelt-change-forums 156
12.90.1.20 emacspeak-google-toolbelt-change-group-discussions . . 156
12.90.1.21 emacspeak-google-toolbelt-change-images 156
12.90.1.22 emacspeak-google-toolbelt-change-in-depth 156
12.90.1.23 emacspeak-google-toolbelt-change-literal 156
12.90.1.24 emacspeak-google-toolbelt-change-news 156
12.90.1.25 emacspeak-google-toolbelt-change-non-commercial . . 157
12.90.1.26 emacspeak-google-toolbelt-change-patents 157
12.90.1.27 emacspeak-google-toolbelt-change-places 157
12.90.1.28 emacspeak-google-toolbelt-change-recent 157
12.90.1.29 emacspeak-google-toolbelt-change-recipes 157
12.90.1.30 emacspeak-google-toolbelt-change-reviews 157
12.90.1.31 emacspeak-google-toolbelt-change-social 157
12.90.1.32 emacspeak-google-toolbelt-change-sort-by-date . . . 157
12.90.1.33
emacspeak-google-toolbelt-change-structured-snippets 157
12.90.1.34 emacspeak-google-toolbelt-change-timeline 157
12.90.1.35 emacspeak-google-toolbelt-change-timeline-high . . 158
12.90.1.36 emacspeak-google-toolbelt-change-timeline-low . . . 158
12.90.1.37 emacspeak-google-toolbelt-change-video 158
12.90.1.38 emacspeak-google-toolbelt-change-video-duration . . 158
12.90.1.39
emacspeak-google-toolbelt-change-web-history-not-visited . . 158

12.90.1.40 emacspeak-google-toolbelt-change-web-history-visited 158
12.90.1.41 emacspeak-google-tts 158
12.90.1.42 emacspeak-google-tts-region 158
12.90.1.43 emacspeak-google-what-is-my-ip 159
12.90.1.44 emacspeak-google-who-links-to-this-page 159
12.90.2 emacspeak-google Options 159
12.91 emacspeak-gridtext 159
12.91.1 Emacspeak-Gridtext Commands 159
12.91.1.1 emacspeak-gridtext-apply 159
12.91.1.2 emacspeak-gridtext-load 159
12.91.1.3 emacspeak-gridtext-save 160
12.92 emacspeak-gtags 160
12.93 emacspeak-gud 160
12.94 emacspeak-haskell 160
12.94.1 emacspeak-haskell Options 160
12.95 emacspeak-helm 161
12.96 emacspeak-hide 161
12.96.1 Emacspeak-Hide Commands 161
12.96.1.1 emacspeak-hide-or-expose-all-blocks 161
12.96.1.2 emacspeak-hide-or-expose-block 161
12.96.1.3 emacspeak-hide-speak-block-sans-prefix 161
12.97 emacspeak-hide-lines 162
12.98 emacspeak-hideshow 162
12.99 emacspeak-hydra 162
12.99.1 Emacspeak-Hydra Commands 162
12.99.1.1 emacspeak-hydra-toggle-talkative 162
12.100 emacspeak-ibuffer 162
12.100.1 Emacspeak-Ibuffer Commands 162
12.100.1.1 emacspeak-ibuffer-speak-buffer-line 162
12.101 emacspeak-ido 162
12.101.1 emacspeak-ido Options 162
12.102 emacspeak-iedit 163
12.102.1 emacspeak-iedit Options 163
12.103 emacspeak-indium 163
12.104 emacspeak-info 163
12.104.1 Emacspeak-Info Commands 163
12.104.1.1 emacspeak-info-next-section 163
12.104.1.2 emacspeak-info-previous-section 163
12.104.1.3 emacspeak-info-speak-header 164
12.104.1.4 emacspeak-info-wizard 164
12.104.2 emacspeak-info Options 164
12.105 emacspeak-ispell 165
12.105.1 emacspeak-ispell Options 165
12.106 emacspeak-ivy 165
12.106.1 emacspeak-ivy Options 166
12.107 emacspeak-jabber 167
12.107.1 Emacspeak-Jabber Commands 167

12.107.1.1 emacspeak-jabber-chat-next-message . . . 167
12.107.1.2 emacspeak-jabber-chat-previous-message . . . 167
12.107.1.3 emacspeak-jabber-chat-speak-this-message . . . 167
12.107.1.4 emacspeak-jabber-popup-roster . . . 167
12.107.1.5 emacspeak-jabber-speak-recent-message . . . 167
12.107.2 emacspeak-jabber Options . . . 167
12.108 emacspeak-jdee . . . 169
12.109 emacspeak-js2 . . . 169
12.109.1 emacspeak-js2 Options . . . 169
12.110 emacspeak-keymap . . . 170
12.110.1 emacspeak-keymap Options . . . 170
12.111 emacspeak-kmacro . . . 172
12.112 emacspeak-librivox . . . 172
12.112.1 Usage . . . 172
12.112.2 Emacspeak-Librivox Commands . . . 172
12.112.2.1 emacspeak-librivox . . . 172
12.112.2.2 emacspeak-librivox-play . . . 173
12.112.2.3 emacspeak-librivox-search-by-author . . . 173
12.112.2.4 emacspeak-librivox-search-by-genre . . . 173
12.112.2.5 emacspeak-librivox-search-by-title . . . 173
12.112.3 emacspeak-librivox Options . . . 173
12.113 emacspeak-lispy . . . 173
12.113.1 Overview . . . 174
12.113.2 emacspeak-lispy Options . . . 174
12.114 emacspeak-lua . . . 175
12.115 emacspeak-m-player . . . 175
12.115.1 Usage . . . 175
12.115.2 Emacspeak-M-Player Commands . . . 175
12.115.2.1 emacspeak-m-player . . . 175
12.115.2.2 emacspeak-m-player-add-autopan . . . 175
12.115.2.3 emacspeak-m-player-add-autosat . . . 175
12.115.2.4 emacspeak-m-player-add-equalizer . . . 175
12.115.2.5 emacspeak-m-player-add-filter . . . 176
12.115.2.6 emacspeak-m-player-add-ladspa . . . 176
12.115.2.7 emacspeak-m-player-alt-src-step . . . 176
12.115.2.8 emacspeak-m-player-amark-add . . . 176
12.115.2.9 emacspeak-m-player-amark-jump . . . 176
12.115.2.10 emacspeak-m-player-amark-save . . . 176
12.115.2.11 emacspeak-m-player-apply-reverb-preset . . . 176
12.115.2.12 emacspeak-m-player-backward-10min . . . 177
12.115.2.13 emacspeak-m-player-backward-10s . . . 177
12.115.2.14 emacspeak-m-player-backward-1min . . . 177
12.115.2.15 emacspeak-m-player-balance . . . 177
12.115.2.16 emacspeak-m-player-beginning-of-track . . . 177
12.115.2.17 emacspeak-m-player-bind-accelerator . . . 177
12.115.2.18 emacspeak-m-player-clear-filters . . . 177
12.115.2.19 emacspeak-m-player-command . . . 177
12.115.2.20 emacspeak-m-player-customize-options . . . 177

12.115.2.21 emacspeak-m-player-delete-filter 178
12.115.2.22 emacspeak-m-player-delete-ladspa 178
12.115.2.23 emacspeak-m-player-display-metadata 178
12.115.2.24 emacspeak-m-player-display-percent 178
12.115.2.25 emacspeak-m-player-display-position 178
12.115.2.26 emacspeak-m-player-double-speed 178
12.115.2.27 emacspeak-m-player-edit-reverb 178
12.115.2.28 emacspeak-m-player-end-of-track 178
12.115.2.29 emacspeak-m-player-equalizer-control 178
12.115.2.30 emacspeak-m-player-equalizer-preset 179
12.115.2.31 emacspeak-m-player-faster 179
12.115.2.32 emacspeak-m-player-forward-10min 179
12.115.2.33 emacspeak-m-player-forward-10s 179
12.115.2.34 emacspeak-m-player-forward-1min 179
12.115.2.35 emacspeak-m-player-from-media-history 179
12.115.2.36 emacspeak-m-player-get-length 179
12.115.2.37 emacspeak-m-player-half-speed 180
12.115.2.38 emacspeak-m-player-left-channel 180
12.115.2.39 emacspeak-m-player-load 180
12.115.2.40 emacspeak-m-player-load-file 180
12.115.2.41 emacspeak-m-player-load-playlist 180
12.115.2.42 emacspeak-m-player-locate-media 180
12.115.2.43 emacspeak-m-player-mode 181
12.115.2.44 emacspeak-m-player-mode-line 182
12.115.2.45 emacspeak-m-player-next-track 182
12.115.2.46 emacspeak-m-player-pan 183
12.115.2.47 emacspeak-m-player-pause 183
12.115.2.48 emacspeak-m-player-persist-process 183
12.115.2.49 emacspeak-m-player-play-rss 183
12.115.2.50 emacspeak-m-player-play-tracks-jump 183
12.115.2.51 emacspeak-m-player-play-tree-up 183
12.115.2.52 emacspeak-m-player-pop-to-player 183
12.115.2.53 emacspeak-m-player-previous-track 183
12.115.2.54 emacspeak-m-player-quit 184
12.115.2.55 emacspeak-m-player-reset-options 184
12.115.2.56 emacspeak-m-player-reset-speed 184
12.115.2.57 emacspeak-m-player-restore-process 184
12.115.2.58 emacspeak-m-player-right-channel 184
12.115.2.59 emacspeak-m-player-scale-speed 184
12.115.2.60 emacspeak-m-player-seek-absolute 184
12.115.2.61 emacspeak-m-player-seek-percentage 184
12.115.2.62 emacspeak-m-player-seek-relative 185
12.115.2.63 emacspeak-m-player-set-clip-end 185
12.115.2.64 emacspeak-m-player-set-clip-start 185
12.115.2.65 emacspeak-m-player-shuffle 185
12.115.2.66 emacspeak-m-player-slave-command 185
12.115.2.67 emacspeak-m-player-slower 185
12.115.2.68 emacspeak-m-player-stream-info 185

12.115.2.69 emacspeak-m-player-url 186
12.115.2.70 emacspeak-m-player-using-hrtf 186
12.115.2.71 emacspeak-m-player-using-openal 186
12.115.2.72 emacspeak-m-player-volume-change 186
12.115.2.73 emacspeak-m-player-volume-down 186
12.115.2.74 emacspeak-m-player-volume-set 186
12.115.2.75 emacspeak-m-player-volume-up 187
12.115.2.76 emacspeak-m-player-write-clip 187
12.115.2.77 emacspeak-m-player-youtube-player 187
12.115.2.78 emacspeak-multimedia 187
12.115.3 emacspeak-m-player Options 189
12.116 emacspeak-magit 189
12.116.1 emacspeak-magit Options 189
12.117 emacspeak-make-mode 196
12.117.1 emacspeak-make-mode Options 196
12.118 emacspeak-man 197
12.118.1 Emacspeak-Man Commands 197
12.118.1.1 emacspeak-man-browse-man-page 197
12.118.1.2 emacspeak-man-speak-this-section 197
12.118.2 emacspeak-man Options 197
12.119 emacspeak-markdown 197
12.119.1 emacspeak-markdown Options 197
12.120 emacspeak-maths 198
12.120.1 Setup 198
12.120.2 Technical Overview 198
12.120.3 Basic Usage: 198
12.120.4 Emacspeak-Maths Commands 199
12.120.4.1 emacspeak-maths-depth 199
12.120.4.2 emacspeak-maths-down 199
12.120.4.3 emacspeak-maths-enter 199
12.120.4.4 emacspeak-maths-enter-guess 199
12.120.4.5 emacspeak-maths-flush-output 199
12.120.4.6 emacspeak-maths-left 199
12.120.4.7 emacspeak-maths-restart 199
12.120.4.8 emacspeak-maths-right 199
12.120.4.9 emacspeak-maths-root 199
12.120.4.10 emacspeak-maths-shutdown 200
12.120.4.11 emacspeak-maths-speak-alt 200
12.120.4.12 emacspeak-maths-spoken-mode 200
12.120.4.13 emacspeak-maths-start 200
12.120.4.14 emacspeak-maths-switch-to-output 200
12.120.4.15 emacspeak-maths-up 201
12.121 emacspeak-message 201
12.121.1 emacspeak-message Options 201
12.122 emacspeak-metapost 202
12.123 emacspeak-midge 202
12.124 emacspeak-mines 202
12.124.1 Emacspeak-Mines Commands 202

12.124.1.1 emacspeak-mines-beginning-of-row 202
12.124.1.2 emacspeak-mines-end-of-row 202
12.124.1.3 emacspeak-mines-goto 202
12.124.1.4 emacspeak-mines-jump-to-uncovered-cell 203
12.124.1.5 emacspeak-mines-speak-board 203
12.124.1.6 emacspeak-mines-speak-cell 203
12.124.1.7 emacspeak-mines-speak-mark-count 203
12.124.1.8 emacspeak-mines-speak-neighbors 203
12.124.1.9 emacspeak-mines-speak-uncovered-count 203
12.125 emacspeak-mspools 203
12.126 emacspeak-muggles 203
12.126.1 Using Hydras 203
12.126.2 Implemented Muggles 204
12.126.3 Emacspeak-Muggles Commands 204
12.126.3.1 emacspeak-muggles-brightness/body 204
12.126.3.2 emacspeak-muggles-hideshow/body 204
12.126.3.3 emacspeak-muggles-ido-yank 205
12.126.3.4 emacspeak-muggles-lispy-or-sp 205
12.126.3.5 emacspeak-muggles-maths-navigator/body 205
12.126.3.6 emacspeak-muggles-navigate/body 206
12.126.3.7 emacspeak-muggles-org-nav/body 206
12.126.3.8 emacspeak-muggles-org-table/body 206
12.126.3.9 emacspeak-muggles-toggle-option/body 207
12.126.3.10 emacspeak-muggles-undo-only/undo-redo/body . . 207
12.126.3.11 emacspeak-muggles-vuiet/body 208
12.126.3.12 emacspeak-muggles-yank-pop/body 208
12.127 emacspeak-muse 209
12.127.1 emacspeak-muse Options 209
12.128 emacspeak-navi-mode 209
12.129 emacspeak-net-utils 210
12.130 emacspeak-newsticker 210
12.131 emacspeak-nov 210
12.132 emacspeak-nxml 210
12.132.1 Emacspeak-Nxml Commands 210
12.132.1.1 emacspeak-nxml-summarize-outline 210
12.132.2 emacspeak-nxml Options 210
12.133 emacspeak-ocr 213
12.133.1 Emacspeak-Ocr Commands 213
12.133.1.1 emacspeak-ocr 213
12.133.1.2 emacspeak-ocr-backward-page 214
12.133.1.3 emacspeak-ocr-customize 214
12.133.1.4 emacspeak-ocr-flipflop-and-recognize-image 214
12.133.1.5 emacspeak-ocr-forward-page 214
12.133.1.6 emacspeak-ocr-mode 214
12.133.1.7 emacspeak-ocr-name-document 216
12.133.1.8 emacspeak-ocr-open-working-directory 216
12.133.1.9 emacspeak-ocr-page 216
12.133.1.10 emacspeak-ocr-read-current-page 217

12.133.1.11 emacspeak-ocr-recognize-image 217
12.133.1.12 emacspeak-ocr-save-current-page 217
12.133.1.13 emacspeak-ocr-scan-and-recognize 217
12.133.1.14 emacspeak-ocr-scan-image 217
12.133.1.15 emacspeak-ocr-scan-photo 217
12.133.1.16 emacspeak-ocr-set-compress-image-options 217
12.133.1.17 emacspeak-ocr-set-scan-image-options 218
12.133.1.18 emacspeak-ocr-write-document 218
12.133.2 emacspeak-ocr Options 218
12.134 emacspeak-org 219
12.134.1 Emacspeak-Org Commands 219
12.134.1.1 emacspeak-org-capture-link 219
12.134.1.2 emacspeak-org-popup-input 219
12.134.1.3 emacspeak-org-popup-input-buffer 219
12.134.1.4
emacspeak-org-table-speak-both-headers-and-element 219
12.134.1.5 emacspeak-org-table-speak-column-header 219
12.134.1.6
emacspeak-org-table-speak-column-header-and-element 219
12.134.1.7 emacspeak-org-table-speak-coordinates 219
12.134.1.8 emacspeak-org-table-speak-current-element 219
12.134.1.9 emacspeak-org-table-speak-row-header 220
12.134.1.10 emacspeak-org-table-speak-row-header-and-element . . 220
12.134.2 emacspeak-org Options 220
12.135 emacspeak-orgalist 224
12.136 emacspeak-origami 224
12.137 emacspeak-outline 224
12.137.1 Emacspeak-Outline Commands 224
12.137.1.1 emacspeak-outline-speak-backward-heading 225
12.137.1.2 emacspeak-outline-speak-forward-heading 225
12.137.1.3 emacspeak-outline-speak-next-heading 225
12.137.1.4 emacspeak-outline-speak-previous-heading 225
12.137.1.5 emacspeak-outline-speak-this-heading 225
12.138 emacspeak-package 225
12.138.1 Emacspeak-Package Commands 225
12.138.1.1 emacspeak-package-next-line 225
12.138.1.2 emacspeak-package-previous-line 225
12.138.1.3 emacspeak-package-summarize-line 225
12.138.2 emacspeak-package Options 226
12.139 emacspeak-paradox 227
12.139.1 Emacspeak-Paradox Commands 227
12.139.1.1 emacspeak-paradox-summarize-line 227
12.139.2 emacspeak-paradox Options 227
12.140 emacspeak-perl 227
12.141 emacspeak-pianobar 227
12.141.1 PIANOBAR == Pandora Client for Emacs 227
12.141.2 Emacspeak Usage: 227
12.141.3 Emacspeak-Pianobar Commands 228

12.141.3.1 emacspeak-pianobar . . . 228
12.141.3.2 emacspeak-pianobar-command . . . 228
12.141.3.3 emacspeak-pianobar-electric-mode-toggle . . . 228
12.141.3.4 emacspeak-pianobar-next-preset . . . 228
12.141.3.5 emacspeak-pianobar-previous-preset . . . 228
12.141.3.6 emacspeak-pianobar-send-raw . . . 228
12.141.3.7 emacspeak-pianobar-switch-to-preset . . . 228
12.141.3.8 emacspeak-pianobar-volume-down . . . 229
12.141.3.9 emacspeak-pianobar-volume-up . . . 229
12.142 emacspeak-popup . . . 229
12.142.1 emacspeak-popup Options . . . 229
12.143 emacspeak-proced . . . 229
12.143.1 Emacspeak-Proced Commands . . . 229
12.143.1.1 emacspeak-proced-jump-to-process . . . 229
12.143.1.2 emacspeak-proced-next-field . . . 229
12.143.1.3 emacspeak-proced-next-line . . . 229
12.143.1.4 emacspeak-proced-previous-field . . . 229
12.143.1.5 emacspeak-proced-previous-line . . . 229
12.143.1.6 emacspeak-proced-speak-args . . . 230
12.143.1.7 emacspeak-proced-speak-field . . . 230
12.143.1.8 emacspeak-proced-speak-that-field . . . 230
12.143.1.9 emacspeak-proced-speak-this-field . . . 230
12.144 emacspeak-project . . . 230
12.145 emacspeak-projectile . . . 230
12.146 emacspeak-pronounce . . . 230
12.146.1 Emacspeak-Pronounce Commands . . . 230
12.146.1.1 emacspeak-pronounce-clear-dictionaries . . . 230
12.146.1.2 emacspeak-pronounce-define-local-pronunciation . . 231
12.146.1.3 emacspeak-pronounce-define-pronunciation . . . 231
12.146.1.4 emacspeak-pronounce-define-template-pronunciation . . 231
12.146.1.5 emacspeak-pronounce-dispatch . . . 231
12.146.1.6 emacspeak-pronounce-edit-pronunciations . . . 231
12.146.1.7 emacspeak-pronounce-load-dictionaries . . . 231
12.146.1.8 emacspeak-pronounce-refresh-pronunciations . . . 232
12.146.1.9 emacspeak-pronounce-save-dictionaries . . . 232
12.146.1.10 emacspeak-pronounce-toggle-use-of-dictionaries . . 232
12.146.2 emacspeak-pronounce Options . . . 232
12.147 emacspeak-py . . . 232
12.148 emacspeak-pydoc . . . 232
12.149 emacspeak-python . . . 232
12.150 emacspeak-racer . . . 233
12.150.1 emacspeak-racer Options . . . 233
12.151 emacspeak-racket . . . 233
12.152 emacspeak-re-builder . . . 233
12.152.1 emacspeak-re-builder Options . . . 233
12.153 emacspeak-reftex . . . 233
12.154 emacspeak-related . . . 233
12.155 emacspeak-rg . . . 233

12.156 emacspeak-rmail . . . 234
12.156.1 Emacspeak-Rmail Commands . . . 234
12.156.1.1 emacspeak-rmail-speak-current-message-labels . . . 234
12.156.1.2 emacspeak-rmail-summarize-current-message . . . 234
12.157 emacspeak-rpm-spec . . . 234
12.158 emacspeak-rst . . . 234
12.158.1 emacspeak-rst Options . . . 234
12.159 emacspeak-ruby . . . 235
12.160 emacspeak-rust-mode . . . 235
12.161 emacspeak-sage . . . 235
12.161.1 Emacspeak-Sage Commands . . . 235
12.161.1.1 emacspeak-sage-describe-symbol . . . 236
12.161.1.2 emacspeak-sage-get-output . . . 236
12.161.1.3 emacspeak-sage-get-output-as-latex . . . 236
12.161.1.4 emacspeak-sage-speak-output . . . 236
12.162 emacspeak-sdcv . . . 236
12.163 emacspeak-ses . . . 236
12.163.1 Emacspeak-Ses Commands . . . 236
12.163.1.1 emacspeak-ses-backward-column-and-summarize . . 236
12.163.1.2 emacspeak-ses-backward-row-and-summarize . . . 236
12.163.1.3 emacspeak-ses-forward-column-and-summarize . . . 236
12.163.1.4 emacspeak-ses-forward-row-and-summarize . . . 237
12.163.1.5 emacspeak-ses-summarize-cell . . . 237
12.163.1.6 emacspeak-ses-summarize-current-cell . . . 237
12.164 emacspeak-setup . . . 237
12.165 emacspeak-sgml-mode . . . 237
12.166 emacspeak-sh-script . . . 237
12.167 emacspeak-shx . . . 237
12.168 emacspeak-slime . . . 237
12.168.1 emacspeak-slime Options . . . 237
12.169 emacspeak-smart-window . . . 238
12.170 emacspeak-smartparens . . . 238
12.170.1 emacspeak-smartparens Options . . . 238
12.171 emacspeak-solitaire . . . 238
12.171.1 Emacspeak-Solitaire Commands . . . 238
12.171.1.1 emacspeak-solitaire-show-column . . . 238
12.171.1.2 emacspeak-solitaire-show-row . . . 238
12.171.1.3 emacspeak-solitaire-speak-coordinates . . . 238
12.171.1.4 emacspeak-solitaire-speak-row . . . 239
12.171.1.5 emacspeak-solitaire-speak-stones . . . 239
12.172 emacspeak-sounds . . . 239
12.172.1 Emacspeak-Sounds Commands . . . 239
12.172.1.1 emacspeak-audio-setup . . . 239
12.172.1.2 emacspeak-sounds-select-theme . . . 239
12.172.1.3 emacspeak-toggle-auditory-icons . . . 240
12.172.2 emacspeak-sounds Options . . . 240
12.173 emacspeak-speak . . . 240
12.173.1 Emacspeak-Speak Commands . . . 240

12.173.1.1 emacspeak–choose-completion 240
12.173.1.2 emacspeak–persist-variable 240
12.173.1.3 emacspeak-blink-matching-open 241
12.173.1.4 emacspeak-completions-move-to-completion-group . . 241
12.173.1.5 emacspeak-dial-dtk 241
12.173.1.6 emacspeak-execute-repeatedly 241
12.173.1.7 emacspeak-goto-percent 241
12.173.1.8 emacspeak-mark-backward-mark 241
12.173.1.9 emacspeak-minibuffer-choose-completion 241
12.173.1.10 emacspeak-minibuffer-next-completion 242
12.173.1.11 emacspeak-minibuffer-previous-completion 242
12.173.1.12 emacspeak-open-info 242
12.173.1.13 emacspeak-owindow-next-line 242
12.173.1.14 emacspeak-owindow-previous-line 242
12.173.1.15 emacspeak-owindow-scroll-down 242
12.173.1.16 emacspeak-owindow-scroll-up 242
12.173.1.17 emacspeak-owindow-speak-line 243
12.173.1.18 emacspeak-read-next-line 243
12.173.1.19 emacspeak-read-next-word 243
12.173.1.20 emacspeak-read-previous-line 243
12.173.1.21 emacspeak-read-previous-word 243
12.173.1.22 emacspeak-shell-command 243
12.173.1.23 emacspeak-silence 244
12.173.1.24 emacspeak-speak-and-skip-extent-upto-char 244
12.173.1.25 emacspeak-speak-and-skip-extent-upto-this-char . . 244
12.173.1.26 emacspeak-speak-browse-buffer-by-style 244
12.173.1.27 emacspeak-speak-buffer 244
12.173.1.28 emacspeak-speak-buffer-filename 245
12.173.1.29 emacspeak-speak-buffer-interactively 245
12.173.1.30 emacspeak-speak-char 245
12.173.1.31 emacspeak-speak-char-name 245
12.173.1.32 emacspeak-speak-completions-if-available 245
12.173.1.33 emacspeak-speak-continuously 246
12.173.1.34 emacspeak-speak-current-column 246
12.173.1.35 emacspeak-speak-current-field 246
12.173.1.36 emacspeak-speak-current-kill 246
12.173.1.37 emacspeak-speak-current-mark 246
12.173.1.38 emacspeak-speak-current-percentage 247
12.173.1.39 emacspeak-speak-current-window 247
12.173.1.40 emacspeak-speak-date-as-seconds 247
12.173.1.41 emacspeak-speak-display-char 247
12.173.1.42 emacspeak-speak-face-interval-and-move 247
12.173.1.43 emacspeak-speak-front-of-buffer 248
12.173.1.44 emacspeak-speak-header-line 248
12.173.1.45 emacspeak-speak-help 248
12.173.1.46 emacspeak-speak-help-interactively 248
12.173.1.47 emacspeak-speak-line 248
12.173.1.48 emacspeak-speak-line-interactively 249

12.173.1.49 emacspeak-speak-line-number 249
12.173.1.50 emacspeak-speak-line-set-column-filter 249
12.173.1.51 emacspeak-speak-message-again 249
12.173.1.52 emacspeak-speak-message-at-time 250
12.173.1.53 emacspeak-speak-microseconds-since-epoch 250
12.173.1.54 emacspeak-speak-milliseconds-since-epoch 250
12.173.1.55 emacspeak-speak-minibuffer 250
12.173.1.56 emacspeak-speak-minor-mode-line 250
12.173.1.57 emacspeak-speak-mode-line 251
12.173.1.58 emacspeak-speak-next-face-chunk 251
12.173.1.59 emacspeak-speak-next-field 251
12.173.1.60 emacspeak-speak-next-personality-chunk 251
12.173.1.61 emacspeak-speak-next-window 251
12.173.1.62 emacspeak-speak-other-buffer 251
12.173.1.63 emacspeak-speak-other-window 252
12.173.1.64 emacspeak-speak-overlay-properties 252
12.173.1.65 emacspeak-speak-page 252
12.173.1.66 emacspeak-speak-page-interactively 252
12.173.1.67 emacspeak-speak-paragraph 253
12.173.1.68 emacspeak-speak-paragraph-interactively 253
12.173.1.69 emacspeak-speak-preceding-char 253
12.173.1.70 emacspeak-speak-predefined-window 253
12.173.1.71 emacspeak-speak-previous-face-chunk 254
12.173.1.72 emacspeak-speak-previous-field 254
12.173.1.73 emacspeak-speak-previous-personality-chunk 254
12.173.1.74 emacspeak-speak-previous-window 254
12.173.1.75 emacspeak-speak-rectangle 255
12.173.1.76 emacspeak-speak-region 255
12.173.1.77 emacspeak-speak-rest-of-buffer 255
12.173.1.78 emacspeak-speak-run-shell-command 255
12.173.1.79 emacspeak-speak-seconds-since-epoch 256
12.173.1.80 emacspeak-speak-sentence 256
12.173.1.81 emacspeak-speak-set-display-table 256
12.173.1.82 emacspeak-speak-sexp 256
12.173.1.83 emacspeak-speak-sexp-interactively 256
12.173.1.84 emacspeak-speak-skim-buffer 256
12.173.1.85 emacspeak-speak-spaces-at-point 257
12.173.1.86 emacspeak-speak-spell-current-word 257
12.173.1.87 emacspeak-speak-text-range 257
12.173.1.88 emacspeak-speak-this-face-chunk 257
12.173.1.89 emacspeak-speak-this-personality-chunk 257
12.173.1.90 emacspeak-speak-time 257
12.173.1.91 emacspeak-speak-version 258
12.173.1.92 emacspeak-speak-visual-line 258
12.173.1.93 emacspeak-speak-voice-annotate-paragraphs 258
12.173.1.94 emacspeak-speak-which-function 258
12.173.1.95 emacspeak-speak-window-information 258
12.173.1.96 emacspeak-speak-word 258

12.173.1.97 emacspeak-speak-word-interactively 259
12.173.1.98 emacspeak-speak-world-clock 259
12.173.1.99 emacspeak-switch-to-reference-buffer 259
12.173.1.100 emacspeak-toggle-action-mode 259
12.173.1.101 emacspeak-toggle-audio-indentation 259
12.173.1.102 emacspeak-toggle-character-echo 260
12.173.1.103 emacspeak-toggle-header-line 260
12.173.1.104 emacspeak-toggle-line-echo 260
12.173.1.105 emacspeak-toggle-mail-alert 260
12.173.1.106 emacspeak-toggle-show-point 260
12.173.1.107 emacspeak-toggle-speak-line-invert-filter 261
12.173.1.108 emacspeak-toggle-speak-messages 261
12.173.1.109 emacspeak-toggle-word-echo 261
12.173.1.110 emacspeak-view-notifications 261
12.173.1.111 emacspeak-voiceify-rectangle 261
12.173.1.112 emacspeak-voiceify-region 262
12.173.1.113 emacspeak-zap-tts 262
12.173.2 emacspeak-speak Options 262
12.174 emacspeak-speedbar 263
12.174.1 Emacspeak-Speedbar Commands 263
12.174.1.1 emacspeak-speedbar-click 264
12.174.1.2 emacspeak-speedbar-goto-speedbar 264
12.175 emacspeak-sql 264
12.176 emacspeak-sudoku 264
12.176.1 Emacspeak-Sudoku Commands 264
12.176.1.1 emacspeak-sudoku-board-columns-summarize 264
12.176.1.2 emacspeak-sudoku-board-distribution-summarize . . 264
12.176.1.3 emacspeak-sudoku-board-rows-summarize 264
12.176.1.4 emacspeak-sudoku-board-sub-squares-summarize . . 264
12.176.1.5 emacspeak-sudoku-board-summarizer 264
12.176.1.6 emacspeak-sudoku-down-sub-square 265
12.176.1.7 emacspeak-sudoku-erase-current-column 265
12.176.1.8 emacspeak-sudoku-erase-current-row 265
12.176.1.9 emacspeak-sudoku-erase-current-sub-square 265
12.176.1.10 emacspeak-sudoku-hint 265
12.176.1.11 emacspeak-sudoku-history-pop 265
12.176.1.12 emacspeak-sudoku-history-push 265
12.176.1.13 emacspeak-sudoku-how-many-remaining 265
12.176.1.14 emacspeak-sudoku-next-sub-square 265
12.176.1.15 emacspeak-sudoku-previous-sub-square 266
12.176.1.16 emacspeak-sudoku-speak-current-cell-coordinates . . 266
12.176.1.17 emacspeak-sudoku-speak-current-cell-value 266
12.176.1.18 emacspeak-sudoku-speak-current-column 266
12.176.1.19 emacspeak-sudoku-speak-current-row 266
12.176.1.20 emacspeak-sudoku-speak-current-sub-square 266
12.176.1.21 emacspeak-sudoku-speak-remaining-in-column . . 266
12.176.1.22 emacspeak-sudoku-speak-remaining-in-row 266
12.176.1.23 emacspeak-sudoku-speak-remaining-in-sub-square . . 266

12.176.1.24 emacspeak-sudoku-up-sub-square 266
12.177 emacspeak-supercite 267
12.178 emacspeak-syslog 267
12.178.1 emacspeak-syslog Options 267
12.179 emacspeak-tab-bar 268
12.179.1 emacspeak-tab-bar Options 268
12.180 emacspeak-table 268
12.181 emacspeak-table-ui 268
12.181.1 Emacspeak-Table-Ui Commands 268
12.181.1.1 emacspeak-table-copy-current-element-to-kill-ring . . 268
12.181.1.2 emacspeak-table-copy-current-element-to-register . . 268
12.181.1.3 emacspeak-table-copy-to-clipboard 269
12.181.1.4 emacspeak-table-display-table-in-region 269
12.181.1.5 emacspeak-table-find-csv-file 269
12.181.1.6 emacspeak-table-find-file 269
12.181.1.7 emacspeak-table-goto 270
12.181.1.8 emacspeak-table-goto-bottom 270
12.181.1.9 emacspeak-table-goto-left 270
12.181.1.10 emacspeak-table-goto-right 270
12.181.1.11 emacspeak-table-goto-top 270
12.181.1.12 emacspeak-table-mode 271
12.181.1.13 emacspeak-table-next-column 273
12.181.1.14 emacspeak-table-next-row 274
12.181.1.15 emacspeak-table-paste-from-clipboard 274
12.181.1.16 emacspeak-table-previous-column 274
12.181.1.17 emacspeak-table-previous-row 274
12.181.1.18 emacspeak-table-search 275
12.181.1.19 emacspeak-table-search-column 275
12.181.1.20 emacspeak-table-search-headers 275
12.181.1.21 emacspeak-table-search-row 275
12.181.1.22 emacspeak-table-select-automatic-speaking-method . . 275
12.181.1.23 emacspeak-table-sort-on-current-column 276
12.181.1.24 emacspeak-table-speak-both-headers-and-element . . 276
12.181.1.25 emacspeak-table-speak-column-filtered 276
12.181.1.26 emacspeak-table-speak-column-header-and-element . . 276
12.181.1.27 emacspeak-table-speak-coordinates 276
12.181.1.28 emacspeak-table-speak-current-element 276
12.181.1.29 emacspeak-table-speak-dimensions 277
12.181.1.30 emacspeak-table-speak-row-filtered 277
12.181.1.31 emacspeak-table-speak-row-header-and-element . . 277
12.181.1.32 emacspeak-table-ui-filter-load 277
12.181.1.33 emacspeak-table-ui-filter-save 277
12.181.1.34 emacspeak-table-view-csv-buffer 277
12.181.1.35 emacspeak-table-view-csv-url 278
12.182 emacspeak-tabulate 278
12.182.1 Emacspeak-Tabulate Commands 278
12.182.1.1 emacspeak-tabulate-region 278
12.183 emacspeak-tapestry 278

12.183.1 Emacspeak-Tapestry Commands 278
12.183.1.1 emacspeak-speak-window-layout 278
12.183.1.2 emacspeak-tapestry-describe-tapestry 279
12.183.1.3 emacspeak-tapestry-select-window-by-name 279
12.184 emacspeak-tar 279
12.184.1 Emacspeak-Tar Commands 279
12.184.1.1 emacspeak-tar-speak-file-date 279
12.184.1.2 emacspeak-tar-speak-file-permissions 279
12.184.1.3 emacspeak-tar-speak-file-size 279
12.185 emacspeak-tcl 279
12.186 emacspeak-tempo 280
12.187 emacspeak-tetris 280
12.187.1 Emacspeak-Tetris Commands 280
12.187.1.1 emacspeak-tetris-goto-bottom-row 280
12.187.1.2 emacspeak-tetris-goto-top-row 280
12.187.1.3 emacspeak-tetris-speak-column 280
12.187.1.4 emacspeak-tetris-speak-coordinates 280
12.187.1.5 emacspeak-tetris-speak-current-shape 280
12.187.1.6
emacspeak-tetris-speak-current-shape-and-coordinates 280
12.187.1.7 emacspeak-tetris-speak-next-shape 280
12.187.1.8 emacspeak-tetris-speak-row 280
12.187.1.9 emacspeak-tetris-speak-row-number 281
12.187.1.10 emacspeak-tetris-speak-score 281
12.187.1.11 emacspeak-tetris-speak-x-coordinate 281
12.188 emacspeak-texinfo 281
12.189 emacspeak-threes 281
12.189.1 Emacspeak-Threes Commands 281
12.189.1.1 emacspeak-threes-0 282
12.189.1.2 emacspeak-threes-1 282
12.189.1.3 emacspeak-threes-2 282
12.189.1.4 emacspeak-threes-3 282
12.189.1.5 emacspeak-threes-export 282
12.189.1.6 emacspeak-threes-import 282
12.189.1.7 emacspeak-threes-pop-state 282
12.189.1.8 emacspeak-threes-prune-stack 282
12.189.1.9 emacspeak-threes-push-state 283
12.189.1.10 emacspeak-threes-score 283
12.189.1.11 emacspeak-threes-speak-board 283
12.189.1.12 emacspeak-threes-speak-empty-count 283
12.189.1.13 emacspeak-threes-speak-next 283
12.189.1.14 emacspeak-threes-speak-transposed-board 283
12.190 emacspeak-tide 283
12.190.1 emacspeak-tide Options 283
12.191 emacspeak-todo-mode 283
12.192 emacspeak-transient 284
12.192.1 Introduction 284
12.192.2 Browsing Contents Of transient–show 284

12.192.3 Emacspeak-Transient Commands 284
12.192.3.1 emacspeak-transient-mode 284
12.192.4 emacspeak-transient Options 284
12.193 emacspeak-twittering 286
12.193.1 Emacspeak-Twittering Commands 286
12.193.1.1 emacspeak-twittering-jump-to-following-url 286
12.193.1.2 emacspeak-twittering-speak-this-tweet 286
12.193.1.3 emacspeak-twittering-twarc 286
12.194 emacspeak-typo 286
12.195 emacspeak-url-template 286
12.195.1 Emacspeak-Url-Template Commands 286
12.195.1.1 emacspeak-url-template-fetch 286
12.195.1.2 emacspeak-url-template-help 287
12.195.1.3 emacspeak-url-template-load 287
12.195.1.4 emacspeak-url-template-nls-add-to-wishlist 287
12.195.1.5 emacspeak-url-template-save 287
12.195.2 emacspeak-url-template Options 287
12.196 emacspeak-vdiff 287
12.196.1 Emacspeak-Vdiff Commands 288
12.196.1.1 emacspeak-vdiff-speak-other-hunk 288
12.196.1.2 emacspeak-vdiff-speak-other-line 288
12.196.1.3 emacspeak-vdiff-speak-this-hunk 288
12.196.2 emacspeak-vdiff Options 288
12.197 emacspeak-view 288
12.197.1 Emacspeak-View Commands 288
12.197.1.1 emacspeak-view-line-to-top 288
12.198 emacspeak-vm 288
12.198.1 Emacspeak-Vm Commands 289
12.198.1.1 emacspeak-vm-browse-message 289
12.198.1.2 emacspeak-vm-catch-up-all-messages 289
12.198.1.3 emacspeak-vm-locate-subject-line 289
12.198.1.4 emacspeak-vm-mode-line 289
12.198.1.5 emacspeak-vm-speak-labels 289
12.198.1.6 emacspeak-vm-speak-message 289
12.198.1.7 emacspeak-vm-toggle-html-mime-demotion 289
12.198.1.8 emacspeak-vm-yank-header 289
12.198.2 emacspeak-vm Options 289
12.199 emacspeak-vterm 290
12.199.1 Usage 290
12.199.2 emacspeak-vterm Options 290
12.200 emacspeak-vuiet 291
12.200.1 Emacspeak-Vuiet Commands 291
12.200.1.1 emacspeak-vuiet-track-info 291
12.201 emacspeak-wdired 291
12.202 emacspeak-we 291
12.202.1 Emacspeak-We Commands 291
12.202.1.1 emacspeak-we-class-filter-and-follow 291
12.202.1.2 emacspeak-we-class-filter-and-follow-link 292

12.202.1.3 emacspeak-we-count-matches 292
12.202.1.4 emacspeak-we-count-nested-tables 292
12.202.1.5 emacspeak-we-count-tables 292
12.202.1.6 emacspeak-we-extract-by-class 292
12.202.1.7 emacspeak-we-extract-by-class-list 293
12.202.1.8 emacspeak-we-extract-by-id 293
12.202.1.9 emacspeak-we-extract-by-id-list 293
12.202.1.10 emacspeak-we-extract-by-role 293
12.202.1.11 emacspeak-we-extract-matching-urls 294
12.202.1.12 emacspeak-we-extract-nested-table 294
12.202.1.13 emacspeak-we-extract-nested-table-list 294
12.202.1.14 emacspeak-we-extract-speakable 294
12.202.1.15 emacspeak-we-extract-table-by-match 295
12.202.1.16 emacspeak-we-extract-table-by-position 295
12.202.1.17 emacspeak-we-extract-tables-by-match-list 295
12.202.1.18 emacspeak-we-extract-tables-by-position-list 295
12.202.1.19 emacspeak-we-follow-and-extract-main 296
12.202.1.20 emacspeak-we-follow-and-filter-by-id 296
12.202.1.21 emacspeak-we-junk-by-class 296
12.202.1.22 emacspeak-we-junk-by-class-list 296
12.202.1.23 emacspeak-we-style-filter 297
12.202.1.24 emacspeak-we-toggle-xsl-keep-result 297
12.202.1.25 emacspeak-we-url-expand-and-execute 297
12.202.1.26 emacspeak-we-url-rewrite-and-follow 297
12.202.1.27 emacspeak-we-xpath-filter-and-follow 297
12.202.1.28 emacspeak-we-xpath-junk-and-follow 298
12.202.1.29 emacspeak-we-xsl-toggle 298
12.202.1.30 emacspeak-we-xslt-apply 298
12.202.1.31 emacspeak-we-xslt-filter 298
12.202.1.32 emacspeak-we-xslt-junk 299
12.202.1.33 emacspeak-we-xslt-select 299
12.202.2 emacspeak-we Options 299
12.203 emacspeak-websearch 299
12.203.1 Emacspeak-Websearch Commands 299
12.203.1.1 emacspeak-websearch-accessible-google 299
12.203.1.2 emacspeak-websearch-amazon-search 299
12.203.1.3 emacspeak-websearch-ask-jeeves 299
12.203.1.4 emacspeak-websearch-biblio-search 300
12.203.1.5 emacspeak-websearch-citeseer-search 300
12.203.1.6 emacspeak-websearch-dispatch 300
12.203.1.7 emacspeak-websearch-foldoc-search 300
12.203.1.8 emacspeak-websearch-google 300
12.203.1.9 emacspeak-websearch-google-feeling-lucky 300
12.203.1.10 emacspeak-websearch-google-news 300
12.203.1.11 emacspeak-websearch-google-search-in-date-range . . 301
12.203.1.12 emacspeak-websearch-google-with-toolbelt 301
12.203.1.13 emacspeak-websearch-gutenberg 301
12.203.1.14 emacspeak-websearch-help 301

12.203.1.15 emacspeak-websearch-merriam-webster-search .. 301
12.203.1.16 emacspeak-websearch-wikipedia-search 301
12.203.1.17 emacspeak-websearch-youtube-search 301
12.204 emacspeak-webspace 301
12.204.1 Emacspeak-Webspace Commands 301
12.204.1.1 emacspeak-webspace-feed-reader 302
12.204.1.2 emacspeak-webspace-filter 302
12.204.1.3 emacspeak-webspace-headlines................... 302
12.204.1.4 emacspeak-webspace-headlines-browse 302
12.204.1.5 emacspeak-webspace-headlines-update 302
12.204.1.6 emacspeak-webspace-mode 302
12.204.1.7 emacspeak-webspace-open 303
12.204.1.8 emacspeak-webspace-yank-link.................... 303
12.204.2 emacspeak-webspace Options......................... 303
12.205 emacspeak-widget.. 303
12.205.1 Emacspeak-Widget Commands....................... 303
12.205.1.1 emacspeak-widget-browse-widget-interactively ... 303
12.205.1.2 emacspeak-widget-help 303
12.205.1.3 emacspeak-widget-summarize-parent............. 303
12.205.1.4 emacspeak-widget-summarize-widget-under-point .. 304
12.205.1.5 emacspeak-widget-update-from-minibuffer 304
12.205.2 emacspeak-widget Options 304
12.206 emacspeak-windmove 304
12.207 emacspeak-winring.. 305
12.208 emacspeak-wizards.. 305
12.208.1 Emacspeak-Wizards Commands...................... 305
12.208.1.1 emacspeak-annotate-add-annotation 305
12.208.1.2 emacspeak-clipboard-copy 305
12.208.1.3 emacspeak-clipboard-paste 305
12.208.1.4 emacspeak-copy-current-file...................... 306
12.208.1.5 emacspeak-curl.................................. 306
12.208.1.6 emacspeak-customize............................ 306
12.208.1.7 emacspeak-describe-emacspeak 306
12.208.1.8 emacspeak-emergency-tts-restart................. 307
12.208.1.9 emacspeak-frame-label-or-switch-to-labelled-frame .. 307
12.208.1.10 emacspeak-kill-buffer-quietly 307
12.208.1.11 emacspeak-learn-emacs-mode................... 307
12.208.1.12 emacspeak-link-current-file 307
12.208.1.13 emacspeak-next-frame-or-buffer................. 308
12.208.1.14 emacspeak-previous-frame-or-buffer............. 308
12.208.1.15 emacspeak-select-this-buffer-next-display 308
12.208.1.16 emacspeak-select-this-buffer-other-window-display .. 308
12.208.1.17 emacspeak-select-this-buffer-previous-display.... 309
12.208.1.18 emacspeak-show-property-at-point.............. 309
12.208.1.19 emacspeak-show-style-at-point.................. 309
12.208.1.20 emacspeak-skip-blank-lines-backward 309
12.208.1.21 emacspeak-skip-blank-lines-forward............. 310
12.208.1.22 emacspeak-speak-hostname..................... 310

12.208.1.23 emacspeak-speak-popup-messages 310
12.208.1.24 emacspeak-speak-show-active-network-interfaces . . 310
12.208.1.25 emacspeak-speak-telephone-directory 310
12.208.1.26 emacspeak-speak-this-buffer-next-display 311
12.208.1.27 emacspeak-speak-this-buffer-other-window-display . . 311
12.208.1.28 emacspeak-speak-this-buffer-previous-display . . . 311
12.208.1.29 emacspeak-ssh-tts-restart 312
12.208.1.30 emacspeak-submit-bug 312
12.208.1.31 emacspeak-symlink-current-file 312
12.208.1.32 emacspeak-view-emacspeak-news 312
12.208.1.33 emacspeak-view-emacspeak-tips 312
12.208.1.34 emacspeak-wizards-add-autoload-cookies 313
12.208.1.35 emacspeak-wizards-alpha-vantage-quotes 313
12.208.1.36 emacspeak-wizards-bindings-from-org 313
12.208.1.37 emacspeak-wizards-bindings-to-org 313
12.208.1.38 emacspeak-wizards-braille 313
12.208.1.39 emacspeak-wizards-cleanup-shell-path 313
12.208.1.40 emacspeak-wizards-color-at-point 314
12.208.1.41 emacspeak-wizards-color-diff-at-point 314
12.208.1.42 emacspeak-wizards-color-wheel 314
12.208.1.43 emacspeak-wizards-colors 314
12.208.1.44 emacspeak-wizards-count-slides-in-region 315
12.208.1.45 emacspeak-wizards-customize-saved 315
12.208.1.46 emacspeak-wizards-cycle-to-next-buffer 315
12.208.1.47 emacspeak-wizards-cycle-to-previous-buffer 315
12.208.1.48 emacspeak-wizards-display-pod-as-manpage 315
12.208.1.49 emacspeak-wizards-enumerate-matching-commands . . 316
12.208.1.50 emacspeak-wizards-enumerate-matching-faces . . . 316
12.208.1.51 emacspeak-wizards-enumerate-obsolete-faces 316
12.208.1.52
emacspeak-wizards-enumerate-uncovered-commands 316
12.208.1.53 emacspeak-wizards-enumerate-unmapped-faces . . 316
12.208.1.54 emacspeak-wizards-espeak-line 316
12.208.1.55 emacspeak-wizards-espeak-region 316
12.208.1.56 emacspeak-wizards-espeak-string 317
12.208.1.57 emacspeak-wizards-eww-buffer-list 317
12.208.1.58 emacspeak-wizards-exec-path-from-shell 317
12.208.1.59 emacspeak-wizards-execute-asynchronously 317
12.208.1.60 emacspeak-wizards-execute-emacspeak-command . . 317
12.208.1.61 emacspeak-wizards-find-file-as-root 317
12.208.1.62 emacspeak-wizards-find-grep 318
12.208.1.63 emacspeak-wizards-find-longest-line-in-region . . . 318
12.208.1.64
emacspeak-wizards-find-longest-paragraph-in-region 318
12.208.1.65 emacspeak-wizards-find-shortest-line-in-region . . 318
12.208.1.66 emacspeak-wizards-finder-find 318
12.208.1.67 emacspeak-wizards-finder-mode 318
12.208.1.68 emacspeak-wizards-fix-read-only-text 319

12.208.1.69 emacspeak-wizards-fix-typo 319
12.208.1.70 emacspeak-wizards-frame-colors 319
12.208.1.71 emacspeak-wizards-free-geo-ip 319
12.208.1.72 emacspeak-wizards-gen-fn-decl 319
12.208.1.73 emacspeak-wizards-generate-finder 320
12.208.1.74 emacspeak-wizards-generate-voice-sampler 320
12.208.1.75 emacspeak-wizards-google-headlines 320
12.208.1.76 emacspeak-wizards-google-news 320
12.208.1.77 emacspeak-wizards-how-many-matches 320
12.208.1.78 emacspeak-wizards-iex-show-financials 321
12.208.1.79 emacspeak-wizards-iex-show-metadata 321
12.208.1.80 emacspeak-wizards-iex-show-news 321
12.208.1.81 emacspeak-wizards-iex-show-price 321
12.208.1.82 emacspeak-wizards-iex-show-quote 321
12.208.1.83 emacspeak-wizards-iex-show-tops 322
12.208.1.84 emacspeak-wizards-iex-this-financials 322
12.208.1.85 emacspeak-wizards-iex-this-news 322
12.208.1.86 emacspeak-wizards-iex-this-price 322
12.208.1.87 emacspeak-wizards-list-voices 322
12.208.1.88 emacspeak-wizards-midi-using-m-score 322
12.208.1.89 emacspeak-wizards-mlb-standings 322
12.208.1.90 emacspeak-wizards-move-and-speak 323
12.208.1.91 emacspeak-wizards-nba-standings 323
12.208.1.92 emacspeak-wizards-next-bullet 323
12.208.1.93 emacspeak-wizards-next-interactive-defun 323
12.208.1.94 emacspeak-wizards-noaa-weather 323
12.208.1.95 emacspeak-wizards-occur-header-lines 324
12.208.1.96 emacspeak-wizards-pdf-open 324
12.208.1.97 emacspeak-wizards-previous-bullet 324
12.208.1.98 emacspeak-wizards-quote 324
12.208.1.99 emacspeak-wizards-scratch 325
12.208.1.100 emacspeak-wizards-set-colors 325
12.208.1.101 emacspeak-wizards-shell 325
12.208.1.102 emacspeak-wizards-shell-by-key 325
12.208.1.103 emacspeak-wizards-shell-command-on-current-file . . 326
12.208.1.104 emacspeak-wizards-shell-directory-reset 326
12.208.1.105 emacspeak-wizards-shell-directory-set 327
12.208.1.106 emacspeak-wizards-shell-toggle 327
12.208.1.107 emacspeak-wizards-show-defined-voices 327
12.208.1.108 emacspeak-wizards-show-eval-result 327
12.208.1.109 emacspeak-wizards-show-face 327
12.208.1.110 emacspeak-wizards-show-memory-used 327
12.208.1.111 emacspeak-wizards-speak-iso-datetime 327
12.208.1.112 emacspeak-wizards-spot-words 328
12.208.1.113 emacspeak-wizards-squeeze-blanks 328
12.208.1.114 emacspeak-wizards-swap-fg-and-bg 328
12.208.1.115 emacspeak-wizards-term 328
12.208.1.116 emacspeak-wizards-terminal 328

12.208.1.117 emacspeak-wizards-toggle-mm-dd-yyyy-date-pronouncer . . . 329
12.208.1.118 emacspeak-wizards-toggle-yyyymmdd-date-pronouncer 329
12.208.1.119 emacspeak-wizards-tramp-open-location 329
12.208.1.120 emacspeak-wizards-tune-in-radio-browse 329
12.208.1.121 emacspeak-wizards-tune-in-radio-search 329
12.208.1.122 emacspeak-wizards-units 329
12.208.1.123 emacspeak-wizards-vc-n 329
12.208.1.124 emacspeak-wizards-vc-view-mode 330
12.208.1.125 emacspeak-wizards-vc-viewer 330
12.208.1.126 emacspeak-wizards-vc-viewer-refresh 330
12.208.1.127 emacspeak-wizards-view-buffers-filtered-by-m-player-mode . . 330
12.208.1.128 emacspeak-wizards-view-buffers-filtered-by-this-mode 330
12.208.1.129 emacspeak-wizards-voice-sampler 331
12.208.1.130 emacspeak-wizards-ytel-download 331
12.208.1.131 emacspeak-wizards-ytel-play-at-point 331
12.208.2 emacspeak-wizards Options 331
12.209 emacspeak-woman 332
12.209.1 emacspeak-woman Options 332
12.210 emacspeak-xkcd 333
12.210.1 Emacspeak-Xkcd Commands 333
12.210.1.1 emacspeak-xkcd-open-explanation-browser 333
12.211 emacspeak-xref 333
12.212 emacspeak-xslt 333
12.212.1 Emacspeak-Xslt Commands 333
12.212.1.1 emacspeak-xslt-view 333
12.212.1.2 emacspeak-xslt-view-file 333
12.212.1.3 emacspeak-xslt-view-region 333
12.212.1.4 emacspeak-xslt-view-xml 333
12.213 emacspeak-yaml 334
12.214 emacspeak-yasnippet 334
12.215 emacspeak-ytel 334
12.215.1 Emacspeak-Ytel Commands 334
12.215.1.1 emacspeak-ytel-download 334
12.215.1.2 emacspeak-ytel-play-at-point 334
12.215.1.3 emacspeak-ytel-yank 334
12.216 espeak-voices 334
12.216.1 Espeak-Voices Commands 334
12.216.1.1 espeak 334
12.216.2 espeak-voices Options 335
12.217 extra-muggles 335
12.217.1 Using Hydras 335
12.217.2 Extra Muggles 335
12.217.3 Extra-Muggles Commands 335

12.217.3.1 emacspeak-muggles-emacspeak-m-player-mode-map-cmd . . . 335
12.217.3.2 emacspeak-muggles-info-summary/body 336
12.217.3.3 emacspeak-muggles-m-player/body 337
12.217.3.4 emacspeak-muggles-outliner/body 338
12.217.3.5 emacspeak-muggles-pianobar-key-map-cmd 339
12.217.3.6 emacspeak-muggles-smartparens/body 339
12.217.3.7 emacspeak-muggles-view/body 340
12.217.3.8 emacspeak-origami/body 341
12.218 g-utils 341
12.218.1 g-utils Options 341
12.219 gm-nnir 342
12.219.1 Gm-Nnir Commands 342
12.219.1.1 gm-nnir-group-make-gmail-group 342
12.219.1.2 gm-nnir-group-make-nnir-group 342
12.220 gmaps 342
12.220.1 Gmaps Commands 342
12.220.1.1 gmaps 342
12.220.1.2 gmaps-bicycling-directions 343
12.220.1.3 gmaps-directions 343
12.220.1.4 gmaps-driving-directions 343
12.220.1.5 gmaps-locations-load 343
12.220.1.6 gmaps-locations-save 343
12.220.1.7 gmaps-mode 343
12.220.1.8 gmaps-place-details 344
12.220.1.9 gmaps-place-reviews 344
12.220.1.10 gmaps-places-nearby 344
12.220.1.11 gmaps-places-search 344
12.220.1.12 gmaps-set-current-filter 345
12.220.1.13 gmaps-set-current-location 345
12.220.1.14 gmaps-set-current-radius 345
12.220.1.15 gmaps-transit-directions 345
12.220.1.16 gmaps-walking-directions 345
12.220.2 gmaps Options 346
12.221 ladspa 346
12.221.1 Ladspa Commands 346
12.221.1.1 ladspa 346
12.221.1.2 ladspa-analyse-plugin-at-point 346
12.221.1.3 ladspa-edit-control 346
12.221.1.4 ladspa-instantiate 346
12.221.1.5 ladspa-mode 346
12.222 mac-voices 347
12.222.1 mac-voices Options 347
12.223 outloud-voices 347
12.223.1 Outloud-Voices Commands 347
12.223.1.1 outloud 347
12.223.2 outloud-voices Options 347
12.224 plain-voices 347

12.225 soundscape . . . 348
12.225.1 Soundscape Commands . . . 348
12.225.1.1 soundscape . . . 348
12.225.1.2 soundscape-kill . . . 348
12.225.1.3 soundscape-listener . . . 349
12.225.1.4 soundscape-listener-shutdown . . . 349
12.225.1.5 soundscape-remote . . . 349
12.225.1.6 soundscape-restart . . . 349
12.225.1.7 soundscape-stop . . . 349
12.225.1.8 soundscape-theme . . . 349
12.225.1.9 soundscape-toggle . . . 350
12.225.1.10 soundscape-update-mood . . . 350
12.225.2 soundscape Options . . . 350
12.226 sox . . . 350
12.226.1 Sox Commands . . . 351
12.226.1.1 sox-add-effect . . . 351
12.226.1.2 sox-delete-effect-at-point . . . 351
12.226.1.3 sox-edit-effect-at-point . . . 351
12.226.1.4 sox-mode . . . 351
12.226.1.5 sox-open-file . . . 351
12.226.1.6 sox-play . . . 352
12.226.1.7 sox-refresh . . . 352
12.226.1.8 sox-save . . . 352
12.226.1.9 sox-set-effect . . . 352
12.226.1.10 sox-show-timestamp . . . 352
12.226.1.11 sox-stop . . . 352
12.227 sox-gen . . . 352
12.227.1 Binaural Beats Using SoX . . . 352
12.227.1.1 High-Level Commands For Pre-Defined Binaural Beats . . . 353
12.227.2 Sox-Gen Commands . . . 353
12.227.2.1 sox-beats-binaural . . . 353
12.227.2.2 sox-binaural . . . 354
12.227.2.3 sox-chakras . . . 354
12.227.2.4 sox-relax . . . 354
12.227.2.5 sox-rev-up . . . 354
12.227.2.6 sox-slide-binaural . . . 354
12.227.2.7 sox-tone-binaural . . . 354
12.227.2.8 sox-tone-slide-binaural . . . 355
12.227.2.9 sox-turn-down . . . 355
12.227.2.10 sox-wind-down . . . 355
12.227.3 sox-gen Options . . . 355
12.228 tetris . . . 355
12.229 toy-braille . . . 355
12.230 voice-setup . . . 356
12.230.1 Voice-Setup Commands . . . 356
12.230.1.1 voice-lock-mode . . . 356
12.230.1.2 voice-lock-mode–turn-on . . . 357

12.230.1.3 voice-setup-describe-personality 357
12.230.1.4 voice-setup-toggle-silence-personality 357
12.230.2 voice-setup Options 357
12.231 xbacklight 359
12.231.1 Xbacklight Commands 359
12.231.1.1 xbacklight-black 359
12.231.1.2 xbacklight-decrement 359
12.231.1.3 xbacklight-get 360
12.231.1.4 xbacklight-increment 360
12.231.1.5 xbacklight-set 360
12.231.1.6 xbacklight-white 360
12.232 URL Templates 360

13 Emacspeak Keyboard Commands 365

14 TTS Servers 366

14.1 High-level Overview 366
14.1.1 Commands That Queue Output 366
14.1.2 Commands That Set State 368

15 Emacspeak At Twenty 369

15.1 Turning Twenty 369
15.2 Using UNIX With Speech Output — 1994 369
15.3 Key Enabler — Emacs And Lisp Advice 369
15.4 Key Component — Text To Speech (TTS) 370
15.5 Emacspeak And Software Development 370
15.5.1 Programming Defensively 371
15.6 Emacspeak And Authoring Documents 371
15.7 Emacspeak And The Early Days Of The Web 372
15.8 Audio Formatting — Generalizing Aural CSS 373
15.9 Conversational Gestures For The Audio Desktop 373
15.9.1 Speech-Enabling Interactive Games 374
15.10 Accessing Media Streams 375
15.11 EBooks — Ubiquitous Access To Books 375
15.12 Leveraging Computational Tools — From SQL And R To IPython Notebooks ... 376
15.13 Social Web — EMail, Instant Messaging, Blogging And Tweeting Using Open Protocols ... 376
15.14 The RESTful Web — Web Wizards And URL Templates For Faster Access ... 377
15.15 Mashing It Up — Leveraging Evolving Web APIs 378
15.16 Conclusion — Turning Twenty 379
15.17 References 379

16 Acknowledgments 381

17 Concept Index 382

18 Key Index 383

1 Copyright

This manual documents Emacspeak, a speech extension to Emacs.

2 Announcing Emacspeak Manual 2nd Edition As An Open Source Project

This is to announce the launch of a new open source project to create a user manual for Emacspeak — an Emacs speech extension that provides a complete audio desktop.

2.1 How To Contribute To This Manual

This manual is organized as a series of chapters, with each chapter in a separate file. If you feel capable of contributing to a specific section, send out a message to the Emacspeak mailing list `mailto:emacspeak@cs.vassar.edu`. You can then start adding content to a local copy of the chapter to which you are contributing. When you feel you have something to submit, mail out the file to the emacspeak mailing list — I'll integrate new content as it comes in.

2.2 Authoring Guidelines

For this manual to hang together and make sense to the new user at whom it is targeted, contributors need to stick to a consistent style. If you plan to contribute content, you should take some time to read the existing sections — note that many of these are skeletal and the first contributions will be to flesh these sections out.

If you are familiar with texinfo, go ahead and mark up your content using texinfo. If you are not, simply author the documentation you create as plain formatted ASCII. If you do submit files as texinfo source, make sure to validate them at your end first by running the files through `makeinfo` — badly created or malformed texinfo source takes more time to fix than marking up straight text.

2.3 Credits

This initial version draws heavily from the original Emacspeak user manual, and includes contributions from Jim Van Zandt and Jason White. Authors who contribute complete sections will be acknowledged here as well as in the specific section they author.

3 Background

Emacspeak was originally developed in late 1994 and released as Open Source in May 1995. Since then, the system has been regularly updated every six months to provide an up-to-date *Audio Desktop.* Here is a brief overview of some of the significant aspects of the system, and the lessons learnt from its development and use. The work on Emacspeak was presented at CHI96 and the co-located Assets96 conference in Vancouver, BC. This overview is being written nearly 15 years later to trace the impact of the work.

3.1 Speech-Enabling Applications

The underlying thesis behind AsTeR (Audio System For Technical Readings) and later Emacspeak is that information is display-independent. This leads to the insight that producing auditory renderings of information starting from the true source of that information often produces better renderings than those that result from working from a modality-specific representation; thus, attempting to speak visually rendered information can often prove sub-optimal. AsTeR applied these ideas to documents authored in LaTeX; Emacspeak generalized these ideas to user interfaces.

Emacspeak was therefore designed from the ground-up to enable applications generate their own spoken feedback, rather than having an external software program construct the spoken feedback by responding to events in its environment.

In Emacspeak, theory meets practice to deliver a working implementation; Emacspeak leverages the power of Emacs and its embedded Lisp interpreter to inject spoken feedback into applications that run within Emacs. For a detailed overview on how the *advice* mechanism in Emacs is used, see the original Assets96 paper, as well as the chapter on Emacspeak in the OReilly publication entitled Beautiful Code (`http://emacspeak.sourceforge.net/raman/publications/bc-emacspeak/publish-emacspeak-bc.html`).

3.2 Audio Formatting And Aural CSS

AsTeR introduced the notion of *audio formatting* a concept analogous to the well-understood notion of visual formatting. The work on AsTeR coincided with the Web being at its infancy. As the Web evolved to acquire Cascaded Style Sheets (CSS), ideas from AsTeR were used to define Aural CSS as an appendix to CSS1 in 1995. Emacspeak proved an ideal platform to prototype the ideas within Aural CSS — first within the Emacs/W3 browser. Around this time, Emacs itself evolved to support multiple fonts and *font-locking* to implement syntax coloring for various types of content. Emacspeak applied the ideas of audio formatting to create the auditory analog of *font-lock* — Emacspeak calls this *voice locking.*

Voice locking in Emacspeak continues to be a unique feature among systems that provide auditory feedback. Later in 1997, Emacspeak's implementation was overhauled to use Aural CSS for all aspects of voice-locking, rather than just for Web content.

3.3 Auditory Icons

Emacspeak augments spoken feedback with short auditory icons that vastly speed up interaction. Combined with audio formatting, the resulting experience is analogous to moving from a monochrome character-cell display to a high-quality color display — the overall user experience is rich in comparison. This enabled Emacspeak to explore innovative means of auditory communication — as an example, see my Assets-98 paper entitled *Conversational Gestures For The Audio Desktop* that details how one can play Tetris on the Emacspeak desktop.

3.4 Summary

The lessons learnt from developing Emacspeak are many — here are a few highlights:

- A model for browsing tabular data — see relevant chapter in my book *Auditory User Interfaces*
- Audio formatting and Aural CSS
- Auditory icons for efficient feedback.
- Web widgets for rapid task completion on the Web.

And many more than will fit this margin.

4 Introduction

Emacspeak provides a complete audio desktop by speech-enabling all of Emacs.

In the past, screen reading programs have allowed visually impaired users to get feedback using synthesized speech. Such programs have been commercially available for a long time. Most of them originally ran on PC's under DOS, and have moved over to the Windows environment. However, screen-readers for the UNIX environment have been conspicuous in their absence. Note that this is now changing with the availability of console-level Linux screenreaders such as `speakup`. Such Linux screenreaders provide the same level of UNIX accessibility provided in the late 80's by PC terminal emulators running a DOS screenreader. This means that most visually impaired computer users face the additional handicap of being DOS-impaired — a far more serious problem:-)

Emacspeak is an emacs subsystem that provides complete speech access. It is *not* a screen-reader — rather, it is a complete user environment with built-in speech feedback. Emacspeak has a significant advantage; since it runs inside Emacs, a structure-sensitive, fully customizable environment, Emacspeak has more context-specific information about what it is speaking than its screenreader counterparts. This is why Emacspeak is not a "screenreader", it is a system that produces speech output.

A Traditional screen-reader speaks the content of the screen, leaving it to the user to interpret the visual layout. Emacspeak, on the other hand, treats speech as a first-class output modality; it speaks the information in a manner that is easy to comprehend when listening.

The basic concepts used by Emacspeak are simple; all interactive Emacs commands have been adapted to provide speech feedback. Hence, you use Emacs as normal; Emacspeak works behind the scene to give audio feedback in addition to updating the screen.

Emacspeak consists of a core speech system that provides speech and audio services to the rest of the Emacspeak desktop; application-specific extensions provide context-specific spoken feedback using these services. Emacspeak currently comes with speech extensions for several popular Emacs subsystems and editing modes. I would like to thank their respective authors for their wonderful work which makes Emacs more than a text editor[1].

[1] I have now been using Emacspeak under Linux as the only source of speech feedback since 1994.

5 Installation Instructions

This chapter gives brief and detailed installation instructions for configuring, installing and starting Emacspeak.

5.1 Obtaining Emacspeak

Emacspeak is available on the Internet at:

WWW `http://emacspeak.sourceforge.net`

Git Repository
`https://github.com/tvraman/emacspeak`

Mail List `emacspeak@cs.vassar.edu`

List Request
`emacspeak-request@cs.vassar.edu`

The Emacspeak mailing list is maintained by Greg E. Priest-Dorman. If you are using Emacspeak, you should join the list by sending mail to the request address, `mailto:emacspeak-request@cs.vassar.edu`.

5.2 Quick Installation

Here are the quick installation instructions. See the next section for detailed installation instructions.

Packages for Linux distributions such as Debian typically become available on the WWW a few weeks or months after a new version is released. The instructions below are for building and installing Emacspeak from the source distribution. If you install one of the prepackaged distributions, use the install instructions that come with that package.

- Obtain the source code — either by downloading the tar.bz2 file for the latest release — or by cloning the git repository.

  ```
  git clone https://github.com/tvraman/emacspeak
  ```

- Change to the `emacspeak` directory.
- Type '`make config`' to configure the sources.
- Type '`make`' to compile the files.
- Next, decide which text-to-speech engine you will be using, and proceed to install that engine. Your choices are:
  - Open Source ESpeak on Linux. Install the ESpeak packages for your system, then compile the Emacspeak ESpeak server by doing:

    ```
    cd servers/native-espeak
    make
    ```

  - ViaVoice Outloud (AKA Eloquent). You need to purchase this engine from the voxin site. That purchase will give you install-ready packages for installing the speech engine as well as Emacspeak. See `https://voxin.oralux.net/rss.xml` for the latest packages, and `https://voxin.oralux.net` for the main Voxin Web site.

  - On the Mac, you can use the builtin Mac TTS engine — emacspeak comes with a speech server for that TTS engine written in Python.
- Having installed and configured the TTS engine of your choice, and having built the associated speech server, set Emacspeak up to use that engine by setting environment variable `DTK_PROGRAM`. If using `bash` as your shell, add the line

  ```
  export DTK_PROGRAM=<engine-name>
  ```

  to your `.bash_profile`.

  As an example, to use ESpeak, add

  ```
  export DTK_PROGRAM=espeak
  ```

- Run it by adding the line

  ```
  (load-file "<emacspeak-dir>/lisp/emacspeak-setup.el")
  ```

  to the top of your .emacs file.

  In the above, <*emacspeak-dir*> refers to the directory where you unpacked the sources.

See the next section for details on building and testing the speech server.

5.3 Building And Testing The Speech Server

5.3.1 Speech Servers

Speech servers are located in the `emacspeak/servers` directory.

- ESpeak: `servers/espeak`. This is a TCL script that uses a library built in `servers/native-espeak/`.
- Dectalk: `servers/dtk-exp`. This is a TCL script that does not depend on any native code.
- Outloud: `servers/outloud` or `servers/32-outloud` (for 64-bit machines). This is a TCL script that uses the library built in `servers/linux-outloud`. Note that a checkout from GitHub gives you a prebuilt library — however you will need to purchase the TTS engine from `http://voxin.oralux.net`.
- Mac: `servers/mac`. This is a Python script that binds to the native Mac TTS.

5.3.1.1 Testing The Selected Server.

Once you have picked the TTS engine to use, run the selected server script at a shell prompt, e.g. for the `espeak` engine, execute:

```
./servers/espeak
```

This will result in a TCL prompt. Here, you can test the TTS engine by typing

```
q "this is a test."
d
```

You should hear the TTS engine speak the text.

Type *`s`* to stop speech. You should see a TCL prompt when you execute it.

Quit this TCL session by typing *`C-d`* (an end-of-file character).

6 Basic Usage.

This chapter gives an overview of how to use Emacspeak; for a full listing of Emacspeak keybindings, see See Section 12.9 [emacspeak], page 58. Note: This documentation should be used in conjunction with the online Emacs info pages that extensively document Emacs itself. These sections briefly describe the speech-enabling extensions. However, they should not be considered a substitute for reading the Emacs manual. How successfully you use Emacspeak will depend on how well you learn your Emacs.

All Emacs navigation and editing commands have been speech enabled. Thus, moving to the next or previous word, line or paragraph results in the text around point being spoken. Exactly how much text is spoken is determined by the amount by which you moved.

In addition, Emacspeak provides basic reading functions that can be invoked to listen to chunks of text without moving.

6.1 Emacspeak Overview

Emacspeak provides a small number of core services around which the remainder of the audio interface is constructed. These essential features of the software are briefly outlined in the following paragraphs; the commands by which they can be controlled will be described later in the manual.

Apart from providing a fluent spoken interface to all of Emacs' basic editing functions, Emacspeak also includes software modules which add speech feedback to a range of applications that can be run from within Emacs. In this sense, Emacspeak amounts to much more than a talking text editor; indeed, it can more aptly be characterized as a true "audio desktop", in which speech is treated as a first-class output modality.

Emacspeak implements a special minor mode, known as "voice lock mode" (see Section 6.5 [Voice-lock], page 16) which uses distinct speech characteristics to provide aural highlighting of specific textual constructs, such as comments in program code, quoted strings and reserved words See ⟨undefined⟩ [Voice Lock], page ⟨undefined⟩. This facility is further extended when Emacspeak is used with the `EWW` and `W3` Web browsers, to enable the semantic and structural distinctions captured by the HTML markup to be communicated efficiently See Section 10.4 [Web Browsing], page 31.

It is often desirable to exercise control over the pronunciation of a word (E.G. a technical term or a reserved word in a programming language) within specific contexts. Emacspeak maintains pronunciation dictionaries for this purpose, which may be customized by the user. Moreover, individual dictionaries can be activated selectively, depending for example on the current major mode or the name of the file which is being visited See Section 12.146 [emacspeak-pronounce], page 230.

In addition to spoken feedback, Emacspeak can generate "auditory icons" — short sound cues which alert the user to significant events, for example the opening or deletion of a file, the completion of an action, the arrival of an electronic mail message or the creation of a completion buffer. Sound cues act as a supplement to the spoken interface, and are especially valuable to the experienced user in facilitating rapid interaction. Note that in order to support auditory icons, the computer must be equipped with sound hardware for which the operating system has been correctly configured See Section 12.172 [emacspeak-sounds], page 239.

6.2 Using Emacs Buffers

While typing in an Emacs buffer, hitting space speaks the recently typed word. I use completion all the time; so Emacspeak will speak the completion just inserted as well as the next possible completion. In Emacs, use *`M-x load-library RETURN completion RETURN`* for loading the completion package.

The standard Emacs prompting functions have also been speech-enabled. Emacs prompts with available lists of completions in response to partial input wherever appropriate — all forms of completion provide speech feedback.

In addition, Emacspeak provides a number of commands for reading portions of the current buffer, getting status information, and modifying Emacspeak's state.

All of the commands are documented in the subsequent sections. They can be classified into types:

- Emacspeak commands for listening to chunks of information. The names of these commands all start with the common prefix `emacspeak-speak-`. All Emacspeak commands are bound to the keymap *emacspeak-keymap* and are accessed with the key *`C-e`*[1]. Thus, the Emacspeak command *emacspeak-speak-line* is bound to *`l`* in keymap *emacspeak-keymap* and can be accessed with the keystroke *`C-e l`*. If for some reason you wish to use some key other than *`C-e`* as the common keyboard prefix for all Emacspeak commands, set the variable *emacspeak-prefix*.
- The second category of commands provided by Emacspeak manipulate the state of the speech device. The names of these commands start with the common prefix `dtk-` and are bound in keymap *emacspeak-dtk-submap*.

  You can access these commands via the prefix *`C-e d`*[2]. Thus, the command *dtk-set-rate* is bound to *`r`* in keymap *emacspeak-dtk-submap* and can be executed by pressing *`C-e d r`*.

Emacs has extensive online help; so does emacspeak. Please use it.

This info manual is only to get you started. You can get a summary of Emacspeak's features by pressing *`C-h C-e`*

6.3 Reading Without Moving

Emacspeak speaks information as you move around within a buffer. How much text is spoken depends on how you move, thus, when you move by words, you hear the current word; when you move by paragraphs, you hear the current paragraph spoken. In addition, the following commands allow you to listen to information without moving point (point is emacs terminology for the editing cursor).

Reading without moving point:

`C-e c` `emacspeak-speak-char`

Speak character under point. Pronounces character phonetically unless called with a PREFIX arg.

[1] C-e is mnemonic for Emacspeak.

[2] Historically, d was mnemonic for Dectalk; note that nothing in Emacspeak is DECTalk specific.

C-e w `emacspeak-speak-word`

Speak current word. With prefix ARG, speaks the rest of the word from point. Negative prefix arg speaks from start of word to point. If executed on the same buffer position a second time, the word is spelled instead of being spoken.

C-e l `emacspeak-speak-line`

Speaks current line. With prefix ARG, speaks the rest of the line from point. Negative prefix optional arg speaks from start of line to point. Voicifies if option 'voice-lock-mode' is on. Indicates indentation with a tone or a spoken cue if audio indentation is in use. Indicates position of point with an aural highlight if option 'emacspeak-show-point' is turned on — see command `emacspeak-show-point` bound to *C-e C-d*. Lines that start hidden blocks of text, e.g. outline header lines, or header lines of blocks created by command 'emacspeak-hide-or-expose-block' are indicated with auditory icon ellipses.

C-e UP `emacspeak-read-previous-line`

Read previous line, specified by an offset, without moving. Default is to read the previous line.

C-e DOWN `emacspeak-read-next-line`

Read next line, specified by an offset, without moving. Default is to read the next line.

C-e p `emacspeak-speak-paragraph`

Speak paragraph. With prefix arg, speaks rest of current paragraph. Negative prefix arg will read from start of current paragraph to point. If voice-lock-mode is on, then it will use any defined personality.

C-e r `emacspeak-speak-region`

Speak current region delimited by *point* and *mark*. When called from a program, argument START and END specify region to speak.

C-e cap R `emacspeak-speak-rectangle`

Speak a rectangle of text. Rectangle is delimited by point and mark. When call from a program, arguments specify the START and END of the rectangle.

C-e b `emacspeak-speak-buffer`

Speak current buffer contents. With prefix ARG, speaks the rest of the buffer from point. Negative prefix arg speaks from start of buffer to point. If voice lock mode is on, the paragraphs in the buffer are voice annotated first, see command `emacspeak-speak-voice-annotate-paragraphs`. This provides the auditory equivalent of *dropped caps* from visual typography.

C-e n `emacspeak-speak-rest-of-buffer`

Speak remainder of the buffer starting at point

C-e / `emacspeak-speak-this-buffer-other-window-display`

Speak this buffer as displayed in a different frame. Emacs allows you to display the same buffer in multiple windows or frames. These different windows can display different portions of the buffer. This is equivalent to leaving a book open at places at once. This command allows you to listen to the places where you have left the book open. The number used to invoke this command specifies which of the displays you wish to speak. Typically you will have two or at most three such displays open. The current display is 0, the next is 1, and so on. Optional argument ARG specifies the display to speak.

C-e LEFT `emacspeak-speak-this-buffer-previous-display`

Speak this buffer as displayed in a 'previous' window. See documentation for command `emacspeak-speak-this-buffer-other-window-display` for the meaning of 'previous'.

C-e RIGHT `emacspeak-speak-this-buffer-next-display`

Speak this buffer as displayed in a 'previous' window. See documentation for command `emacspeak-speak-this-buffer-other-window-display` for the meaning of 'previous'.

C-e [`emacspeak-speak-page`

Speak a page. With prefix ARG, speaks rest of current page. Negative prefix arg will read from start of current page to point. If option 'voice-lock-mode' is on, then it will use any defined personality.

C-e DIGIT `emacspeak-speak-predefined-window`

Speak one of the first 10 windows on the screen. In general, you'll never have Emacs split the screen into more than two or three. Argument ARG determines the 'other' window to speak. Speaks entire window irrespective of point. Semantics of 'other' is the same as for the builtin Emacs command 'other-window'.

C-e C-n `emacspeak-speak-next-window`

Speak the next window.

C-e C-p `emacspeak-speak-previous-window`

Speak the previous window.

emacspeak-speak-other-window

Speak contents of 'other' window. Speaks entire window irrespective of point. Semantics of 'other' is the same as for the builtin Emacs command 'other-window'. Optional argument ARG specifies 'other' window to speak.

ESCAPE UP `emacspeak-owindow-previous-line`

Move to the next line in the other window and speak it. Numeric prefix arg COUNT specifies number of lines to move.

ESCAPE DOWN

`emacspeak-owindow-next-line`

Move to the next line in the other window and speak it. Numeric prefix arg COUNT can specify number of lines to move.

ESCAPE next

`emacspeak-owindow-scroll-up`

Scroll up the window that command 'other-window' would move to. Speak the window contents after scrolling.

ESCAPE *prior*

`emacspeak-owindow-scroll-down`

Scroll down the window that command ‘other-window’ would move to. Speak the window contents after scrolling.

emacspeak-speak-sexp

Speak current sexp. With prefix ARG, speaks the rest of the sexp from point. Negative prefix arg speaks from start of sexp to point.

C-e meta C-@

`emacspeak-speak-spaces-at-point` Speak the white space at point.

6.4 Speech System Commands

This section documents Emacspeak's various user commands for controlling the text to speech (TTS) system.

6.4.1 Character, Word And Line Echo.

By default, Emacspeak speaks characters as they are typed — this is called character echo; Words are spoken as they are completed — this is called word echo. Emacspeak can also optionally speak each line as it is typed — this is called line echo.

Character, word and line echo can be toggled — either in the current buffer — or for all buffers (globally). To toggle the specific echo functionality for all buffers, precede the specific command with *C-u*. Note that in the documentation below, this use of *C-u* is indicated using the common Emacs terminology of *prefix arg* or *interactive prefix arg.*

C-e d k `emacspeak-toggle-character-echo`

Toggle state of Emacspeak character echo. Interactive PREFIX arg means toggle the global default value, and then set the current local value to the result.

C-e d w `emacspeak-toggle-word-echo`

Toggle state of Emacspeak word echo. Interactive PREFIX arg means toggle the global default value, and then set the current local value to the result.

C-e d l `emacspeak-toggle-line-echo`

Toggle state of Emacspeak line echo. Interactive PREFIX arg means toggle the global default value, and then set the current local value to the result.

6.4.2 Setting TTS Characteristics.

Emacspeak user commands can set different characteristics of the speech output such as speech rate and punctuations mode.

Emacspeak provides a number of settings that affect how attributes of the text such as capitalization are conveyed. These include settings that produce a short tone for each upper case letter, as well as a smart mode for speaking mixed case words which is especially useful when programming. These settings can be made locally in a given buffer or be applied to all buffers by preceding these commands with *C-u*.

C-e d r `dtk-set-rate`

Set speaking RATE for the TTS. Interactive PREFIX arg means set the global default value, and then set the current local value to the result.

`C-e d f` `dtk-set-character-scale`

Set scale FACTOR for speech rate. Speech rate is scaled by this factor when speaking characters. Interactive PREFIX arg means set the global default value, and then set the current local value to the result.

This function is advised.

Before-advice 'emacspeak-auto': Automatically defined advice to speak interactive prompts.

`C-e d DIGIT`
`dtk-set-predefined-speech-rate`

Set speech rate to one of nine predefined levels using digit keys *`0`* through *`9`*. Interactive PREFIX arg says to set the rate globally.

`C-e d p` `dtk-set-punctuations`

Set punctuation mode to MODE. Possible values are 'some', 'all', or 'none'. Interactive PREFIX arg means set the global default value, and then set the current local value to the result.

`C-e d m` `dtk-set-pronunciation-mode`

Set pronunciation MODE. This command is valid only for newer Dectalks, e.g. the Dectalk Express. Possible values are 'math, name, europe, spell', all of which can be turned on or off. Argument STATE specifies new state.

`C-e d s` `dtk-toggle-split-caps`

Toggle split caps mode. Split caps mode is useful when reading Hungarian notation in program source code. Interactive PREFIX arg means toggle the global default value, and then set the current local value to the result.

`C-e d c` `dtk-toggle-capitalization`

Toggle capitalization. when set, capitalization is indicated by a short beep. Interactive PREFIX arg means toggle the global default value, and then set the current local value to the result.

`C-e d cap C`
`dtk-toggle-allcaps-beep`

Toggle allcaps-beep. When set, allcaps words are indicated by a short beep. Interactive PREFIX arg means toggle the global default value, and then set the current local value to the result. Note that allcaps-beep is a very useful thing when programming. However it is irritating to have it on when reading documents.

In addition, Emacspeak can convey the indentation of lines as they are spoken — This is relevant when programming and is the default when working with program source.

`C-e d i` `emacspeak-toggle-audio-indentation`

Toggle state of Emacspeak audio indentation. Interactive PREFIX arg means toggle the global default value, and then set the current local value to the result. Specifying the method of indentation as 'tones' results in the speech-server producing a tone whose length is a function of the line's indentation. Specifying 'speak' results in the number of initial spaces being spoken.

Indentation feedback style is set by option `emacspeak-audio-indentation-method`

The default value is `"speak"`

See variable `emacspeak-audio-indentation-methods` for possible values. Automatically becomes local in any buffer where it is set.

6.4.3 Miscellaneous Speech Commands

Speech can be stopped using command `dtk-stop` — though in normal use, the action of moving the cursor will stop ongoing speech. The speech server can be stopped and restarted for cases where the user wants to switch to a different server — or in the rare case to nuke a runaway speech server.

`C-e s` dtk-stop

Stop speech now.

`C-e d q` `dtk-toggle-quiet`

Toggle state of the speech device between being quiet and talkative. Useful if you want to continue using an Emacs session that has emacspeak loaded but wish to make the speech shut up. Optional argument PREFIX specifies whether speech is turned off in the current buffer or in all buffers.

`C-e C-s` `dtk-emergency-restart`

Use this to nuke the currently running dtk server and restart it. Useful if you want to switch to another synthesizer while emacspeak is running. Also useful for emergency stopping of speech.

Finally, here are the remaining commands available via the TTS related keymap *`C-e d`*.

`C-e d a` `dtk-add-cleanup-pattern`

Add this pattern to the list of repeating patterns that are cleaned up. Optional interactive prefix arg deletes this pattern if previously added. Cleaning up repeated patterns results in emacspeak speaking the pattern followed by a repeat count instead of speaking all the characters making up the pattern. Thus, by adding the repeating pattern '.' (this is already added by default) emacspeak will say "aw fifteen dot" when speaking the string "..............." instead of "period period period period".

`C-e d d` `dtk-select-server`

Select a speech server interactively. This will be the server that is used when you next call either M-x dtk-initialize or C-e C-s. Argument PROGRAM specifies the speech server program.

`C-e d SPACE`

`dtk-toggle-splitting-on-white-space`

Toggle splitting of speech on white space. This affects the internal state of emacspeak that decides if we split text purely by clause boundaries, or also include whitespace. By default, emacspeak sends a clause at a time to the speech device. This produces fluent speech for normal use. However in modes such as 'shell-mode' and some programming language modes, clause markers appear infrequently, and this can result in large amounts of text being sent to the speech device at once, making the system unresponsive when asked to stop talking. Splitting on white space makes emacspeak's stop command responsive. However, when splitting on white space, the speech sounds choppy since the synthesizer is getting a word at a time.

C-e d RETURN
`dtk-set-chunk-separator-syntax`

Interactively set how text is split in chunks. See the Emacs documentation on syntax tables for details on how characters are classified into various syntactic classes. Argument S specifies the syntax class.

C-e d t `emacspeak-dial-dtk`

Prompt for and dial a phone NUMBER with the Dectalk.

C-e d cap V
`tts-speak-version`

Use this to find out which version of the TTS firmware you are running.

C-e d z `emacspeak-zap-dtk`

Send this command to the TTS engine directly.

6.5 Voice Lock Mode

The status of voice lock mode can be toggled on and off in the current buffer by issuing the command *C-e d v* (M-x voice-lock-toggle).

Voice-lock is turned on automatically in all buffers that support it when Emacspeak is started.

Emacspeak can be set to disable voice lock in all of the major modes that support it. To do so, insert the following statement into your Emacs initialization file:

```
(global-voice-lock-mode -1)
```

The characteristics of the different voice personalities used by voice lock mode vary according to the capabilities of the speech synthesizer. The definitions applicable to the Dectalk family of synthesizers are contained in `dectalk-voices.el`, which is supplied as part of the Emacspeak distribution. The Emacspeak distribution also contains voice definitions for many other synthesizers.

Using voice lock mode, Emacspeak also supports many of the aural style properties defined in level 2 of the World Wide Web Consortium's Cascading Style Sheet specification (see `http://www.w3.org/TR/REC-CSS2/`. Thus, when Emacspeak is running in conjunction with a cooperating user agent, such as William Perry's Emacspeak/W3 web browser, the rendering of HTML documents can be regulated by style sheets. Examples of style rules which employ the CSS audio properties can be found in the default style sheet which is supplied in the Emacs/W3 distribution.

6.6 Status Commands

The following commands provide miscellaneous information.

C-e a `emacspeak-speak-message-again`

Speak the last message from Emacs once again.

C-e m `emacspeak-speak-mode-line`

Speak the mode-line.

C-e cap M `emacspeak-speak-minor-mode-line`

Speak the minor mode-information.

C-e SPC `emacspeak-speak-header-line`

Speak the header-line.

C-e C-w `emacspeak-speak-window-information`

Speaks information about current windows.

C-e t `emacspeak-speak-time`

Speak the time.

C-e cap V `emacspeak-speak-version`

Announce version information for running Emacspeak.

C-e f `emacspeak-speak-buffer-filename`

Speak name of file being visited in current buffer. Speak default directory if invoked in a dired buffer, or when the buffer is not visiting any file.

C-e h `emacspeak-speak-help`

Speak help buffer if one present. With prefix arg, speaks the rest of the buffer from point. Negative prefix arg speaks from start of buffer to point.

C-e k `emacspeak-speak-current-kill`

Speak the current kill entry. This is the text that will be yanked in by the next *C-y*. Prefix numeric arg, COUNT, specifies that the text that will be yanked as a result of a *C-y* followed by count-1 *M-y* be spoken. The kill number that is spoken says what numeric prefix arg to give to command `yank`.

C-e v `emacspeak-view-register`

Display the contents of a register, and then speak it.

C-e C-@ `emacspeak-speak-current-mark`

Speak the line containing the mark. With no argument, speaks the line containing the mark — this is where 'exchange-point-and-mark' C-x C-x would jump. Numeric prefix arg 'COUNT' speaks line containing mark 'n' where 'n' is one less than the number of times one has to jump using 'set-mark-command' to get to this marked position. The location of the mark is indicated by an aural highlight achieved by a change in voice personality.

C-e C-l `emacspeak-speak-line-number`

Speak the line number of the current line.

C-e = `emacspeak-speak-current-column`
Speak the current column.

C-e % `emacspeak-speak-current-percentage`
Announce the percentage into the current buffer.

7 The Emacspeak Audio Desktop.

This chapter describes the Emacspeak audio desktop and gives tips and tricks for making use of many of Emacs' powerful features.

The desktop is the work area where you organize the tools of your trade and the information objects relevant to your current activities. In the conventional world of visual GUI (Graphical User Interface)-based computing, these tools and information objects manifest themselves as a collection of icons organized in a two-dimensional work-area — this organization is designed to place frequently used objects within easy reach.

Notice that organizing one's work area in terms of visual icons arranged in a two-dimensional area where such an organization is optimized for the available "conversational gestures" of pointing and clicking is an artifact of visual interaction.

In the spirit of a truly speech-enabled application, Emacspeak does not simply provide you spoken access to a particular presentation of your work environment that was initially designed with the "sign language" of visual interaction in mind. Instead, Emacspeak enables you to work with documents and other information objects in a manner that is optimized to aural, eyes-free interaction. A necessary consequence of this setup is that users accustomed to the purely visual manifestation of today's electronic desktop do not immediately perceive the Emacspeak environment as an electronic desktop. This section of the manual hopes to introduce you to a work-style that encourages a different perspective on how one interacts with the computer in performing day-to-day computing tasks.

The end result in my case has been a marked increase in personal productivity.

7.1 Objects Making Up The Emacspeak Desktop

A "buffer" is the basic building block of the Emacspeak desktop. Any information presented by Emacs is placed in a "buffer". For example, when perusing this manual within Emacs, the "file" containing the documentation is presented in a "buffer". All information objects such as WWW pages, email messages, output from user interaction with command-line shells etc., are presented by Emacs in individual "buffers".

Buffers provide a base level of user interaction; Emacs derives its power by allowing applications to specialize buffers to enable specific types of user-interaction that is optimized for a specific class of information.

7.2 An Object-Oriented Desktop

The basic "buffer object" can be specialized by Emacs applications to provide optimal interaction. This kind of specialization makes the Emacs environment an object-oriented environment; thus, the basic conversational gesture of "move to the next statement" can be assigned behavior that is appropriate to the content that the user is currently navigating. As an example of such specialization, Emacs provides "specialized modes" for working with English text, programming languages, markup source e.g. HTML (Hypertext Markup Language) or LaTeX documents and so on.

7.3 Emacspeak Specializes Aural Interaction

The content-specific user interaction described above is a very powerful feature of Emacs, and this is where Emacspeak derives its power. Traditionally, the ability to create buffers specialized for working with specific content-types has been used by the Emacs community to develop versatile programming environments, messaging applications such as mail and news readers, and authoring environments. The clean design present in all of these Emacs extensions in terms of separating application functionality from the user-interface, combined with the availability of the entire source code making up these packages under the open-source model has laid the ground-work for developing Emacspeak as a versatile aural counterpart to the product of years of software engineering that has been invested by the Emacs community. In short, Emacspeak would not exist in its present shape or form without this prior effort.

7.3.1 Audio Formatted Output

Emacspeak takes advantage of the content-specific knowledge available within specialized buffers to produce “audio formatted” output designed to optimize user interaction. A basic consequence of the above is “voice locking” in specialized modes; a more interesting consequence is the implementation of Aural Cascading Style Sheets (ACSS) in conjunction with the Emacs EWW (Emacs Web Wowser) and W3 Web browsers.

7.3.2 Structured Navigation:

Emacspeak also exploits content-specific knowledge to provide structured navigation of different types of electronic content. In many cases, such structured navigation is an extension of what Emacs provides by default; in other cases, Emacspeak implements the necessary extensions to provide the level of structural navigation needed to work efficiently in an eyes-free environment.

Notable among such structured navigation is Emacs’ powerful outline feature.

An example of content-sensitive navigation is provided by the `imenu` package which dynamically creates a “table of contents” based on the content that is being displayed in a given buffer. Emacs *Selective Display* lets one easily hide and expose portions of a buffer based on indentation — this feature can be used to advantage when working with program source code.

7.3.3 Navigating The Desktop

In addition to navigating individual information objects, the Emacspeak environment provides speech-enabled navigation of the various buffers that are currently open on the Emacspeak desktop via Emacs’ built-in list-buffers feature. Emacs’ dired — directory editor — for browsing the file system, along with the new speedbar package that combines features from dired and imenu round off the suite of navigational tools.

7.3.4 Everything Is Searchable:

Emacs derives one final advantage from using buffers as the basic building block for the entire desktop. Every Emacs buffer is searchable via a uniform and powerful search interface. Emacs’ incremental search works efficiently and consistently to enable you locate “objects” of interest either within a given document or to locate a given object from among the various objects that are currently open on the Emacspeak desktop. This is **very powerful** — where a GUI (Graphical User Interface) user is typically limited to quickly locating an object from a relatively small collection — the size of the collection being a direct function of available display real-estate — the Emacspeak user can typically work with a far larger collection of objects. This is well-suited to the eyes-free environment, where display real-estate has no meaning; so bringing up a list of currently open buffers and performing an incremental search to locate a specific buffer is just as efficient independent of whether you have a few dozen or a few hundred buffers open.

To illustrate the above, my typical working Emacs session lasts between two and three weeks — over that time I typically accumulate several hundred open buffers holding a large variety of content ranging from program source code to email messages and WWW pages.

Ubiquitous search in the eyes-free environment is critical — as a comparison, when using a conventional, purely visual WWW browser, users have no means of easily “searching” for say the “submit” button on a WWW page. This inability is a minor annoyance in visual interaction, and the typical mouse-enabled user **never** uses the find dialog to find a submit button — it is simply more efficient to point at the submit button given the eye’s ability to quickly scan the two-dimensional display. This luxury is absent in an eyes-free environment; as a consequence, blind users confronted by the combination of a visual interface and screen-reader are typically limited to either tabbing through all the controls on a WWW page, or using the sub-optimal find dialog.

8 Voice Lock

See `http://tvraman.github.io/emacspeak/blog/voice-lock-refreshed.html` for a high-level overview of how Emacspeak Voice-Lock has evolved over the years.

1. Emacspeak defines a number of voice overlays such as `voice-bolden`, and `voice-lighten` that can be applied to a given voice to change what it sounds like.
2. Voice overlays are defined in terms of Aural CSS (ACSS (`http://www.w3.org/tr/css2/aural.html`)) to keep them independent of a specific TTS engine.
3. For each such overlay there is a corresponding `<`*`overlay-name`*`>-settings` variable that can be customized via custom.
4. The numbers in `voice-bolden-settings` as an example:

Setting	Value
family	nil
average-pitch	1
pitch-range	6
stress	6
richness	nil
punctuation	nil

Unset values (`nil`) show up as “unspecified” in the customize interface.

1. Do not directly customize `voice-bolden` and friends, instead customize the corresponding `voice-bolden-settings`, since that ensures that all voices that are defined in terms of `voice-bolden` get correctly updated.
2. Discovering what to customize:

Command `emacspeak-show-personality-at-point` (bound by default to C-e M-v) will show you the value of properties personality and face at point. Describe-variable on these names should tell you what to customize; so as an example:

Put point on a comment line, and hit *`C-e M-v`*: you will hear

```
Personality emacspeak-voice-lock-comment-personality
Face font-lock-comment-delimiter-face
```

Describe-variable of `emacspeak-voice-lock-comment-personality` gives:

```
emacspeak-voice-lock-comment-personality's value is acss-p0-s0-all

Documentation: Personality used for font-lock-comment-face This
personality uses voice-monotone whose effect can be changed globally
by customizing voice-monotone-settings.
```

8.1 How It All Works

Here is a brief explanation of the connection between `voice-bolden` and its associated `voice-bolden-settings`.

1. Voice settings are initially in `voice-bolden-settings` which is a list of numbers.
2. That list of numbers needs to be translated to appropriate device-specific codes to send to the TTS engine.

3. You do not want to do this translation *each* time you speak something.
4. So when `voice-bolden` is defined, the definition happens in two steps:
   - The list of settings is stored away in `voice-bolden-settings`,
   - A corresponding voice-name is generated — '`acss-a<n>-p<n>-r<n>-s<n>`' and the corresponding control codes to send to the device are stored away in a hash-table keyed by the above symbol.
   - Finally, `voice-bolden` is assigned the above symbol.

8.2 What this gives

1. The ability to customize the voice via custom by editing the list of numbers in `voice-bolden-settings`
2. When that list is edited, `voice-bolden` is arranged to be updated automatically.

The following additional commands from module See Section 12.208 [emacspeak-wizards], page 305, are useful when designing aural styles.

1. `emacspeak-wizards-generate-voice-sampler`

   Generate a buffer containing text that demonstrates the effect of various aural settings.
2. `emacspeak-wizards-voice-sampler`

   Applied specified aural style to text in current region.

9 Using Online Help With Emacspeak.

Emacs provides an extensive online help system for helping you learn about various aspects of using Emacs. Emacspeak provides online help for its various extensions using this same help system. This chapter explains how to use the online help facilities in order to empower you in discovering powerful and versatile working techniques that will make you more and more productive in your day to day computing.

The online help options are accessed via the *C-h* prefix key, which must be followed by an additional letter or control character to designate the kind of help desired. For example, *C-h t* `help-with-tutorial` visits the Emacs tutorial in a new buffer; *C-h i* `info` enters the Info documentation system, from which you can read Texinfo manuals that have been installed on your system, including the Emacs and Emacspeak documentation; and *C-h k* `describe-key` provides a description of the Emacs function which is bound to the next key that you type. For learning about the various options that are available via the *C-h* mechanism described above, view the online help for command `help-for-help` bound to *C-h C-h* — using what has been described so far, you would achieve this by pressing *C-h k* followed by *C-h C-h*.

Emacspeak users should note that online help is typically displayed in a separate Emacs window. Where it makes sense to do so, Emacspeak will automatically speak the displayed help. Once you've asked for help, you can hear the displayed documentation as many times as you wish using Emacspeak command `emacspeak-speak-help` bound to *C-e h*. If you want to move through the displayed help a line at a time, switch to the buffer where the help is displayed — the buffer is called `*Help*`.

Often, in adding an auditory interface to an Emacs extension, such as a web browser or mail reader, Emacspeak defines additional commands and key bindings which enhance the functionality of the spoken feedback provided by the application. This manual does not purport to document all such commands. It is important, therefore, when learning to use the various Emacs extensions which comprise the 'audio desktop' (see Chapter 7 [Audio Desktop], page 19) that you take advantage of online help to obtain details of any context-specific features provided by Emacspeak. The following two commands are of particular importance in this regard:

- *C-h m* `describe-mode` explains which major and minor modes are currently in effect, and lists the commands and key bindings associated with them.
- *C-h b* `describe-bindings` lists all of the key bindings which are currently defined.

The importance of these help functions can be illustrated by the Emacs/W3 web browser. When point is positioned inside a table, certain key bindings are established with which you can access Emacspeak commands that make it possible to read the rows and columns of the table and explore its structure efficiently. To get a description of these key bindings, you can use W3 to visit the sample HTML file supplied as part of the Emacspeak distribution, and, after having moved point onto the first row of the table, issue the command *C-h m* `describe-mode` to create a help buffer containing an explanation of the features offered by W3 mode.

Emacspeak supplements the online help facilities available within Emacs by defining several commands of its own, as follows:

- *C-h c-e* `describe-emacspeak` presents a list of standard Emacspeak commands.

- *C-e F* `emacspeak-view-emacspeak-faq` opens a new buffer containing the Emacspeak FAQ, a list of frequently asked questions about Emacspeak together with their answers.
- *C-e <F1>* `emacspeak-learn-mode` enters a mode in which the function of every key that you type is spoken; this mode can be terminated with the *C-g* `keyboard-quit` command.

10 Emacs Packages.

Emacs — The extensible, self-documenting editor, derives its functionality from its powerful extension mechanism. This extension mechanism is used to implement many user-level applications such as mail readers, web browsers, software development environments and so on. This chapter gives directions on how to locate the right Emacs package for addressing specific tasks. The chapter is organized into logical sections that each pertain to a specific class of tasks; in individual subsections within a section give a brief overview of particular Emacs packages that have been speech-enabled.

For a categorized list of speech-enabled applications on theEmacspeak desktop, see `http://emacspeak.sf.net/applications.html`. In Emacs 24 and later, you can use Emacs' builtin package manager to install and update packages.

10.1 Document Authoring

The Emacspeak environment provides a rich collection of structured document authoring tools. These are well-suited for working in an eyes-free environment — you clearly do not want to use a **What You See Is What You Get** (WYSIWYG) authoring tool if you cannot see what you're getting. Structure-based authoring tools allow you to focus on the act of content creation, leaving the minutiae of visual layout to the computer.

10.1.1 Creating Well-formatted Documents

Before authoring a document, decide its primary audience. If the document contains relatively simple content, e.g., no mathematical equations etc. and is primarily targeted at the web, you are probably better off using HTML. You can create well-structured HTML documents with the help of package `nxml-mode` for editing XML documents. Another option is to use `org-mode` to create a Wiki-like text document that can be easily published to multiple output formats including HTML.

Packages `org-mode` and `nxml-mode` are speech-enabled by Emacspeak to provide auditory icons, structured navigation and outlines, as well as voice locking for audio formatted feedback as you work.

If the document being authored is more complex, you are usually better off creating it in LaTeX. Note that LaTeX documents can be converted to HTML either via package `tex4ht` — available on the WWW.

The TeX family of typesetting languages is suitable for producing well-formatted documents in an eyes-free environment. Unlike WYSIWYG environments, the author of a TeX or LaTeX document works with the content of the document, leaving it to the formatting system (TeX) to format the document for good visual presentation.

The `AUCTeX` package is an Emacs extension that facilitates authoring and maintaining structured documents in TeX and LaTeX. Package `bibtex` facilitates maintenance and use of `bibtex` bibliography databases. The Texinfo package allows creation of software documentation that is suitable for both printing as well as online viewing as hypertext. Emacspeak speech-enables packages `AUCTeX`, `bibtex` and `texinfo` to provide convenient spoken feedback as you create documents. For details on using these packages, see their accompanying online info documentation.

As the document preparation system of choice, Emacspeak supports a fluent speech-enabled interface to editing and formatting LaTeX documents. This interface is provided by speech-enabling *AUCTeX* mode.

Mode *AUCTeX* provides efficient keyboard shortcuts for inserting and maintaining LaTeX markup as a document is being authored. All of these editing commands provide succinct auditory feedback when used with Emacspeak. The syntax coloring provided by this mode is extended to provide *voice locking* — consequently, Emacspeak uses different voices to speak the embedded markup to set it apart from the content.

Mode *AUCTeX* can be used to create empty document templates and to insert document content at the appropriate places in the template. The mode also enables structured navigation of the document as it is being edited. Emacspeak speech-enables these template creation and structured navigation commands to produce auditory icons and succinct spoken feedback. For example, while editing, the user can quickly browse through the sections of the document and have each section title spoken. Document elements such as paragraphs and bulleted lists can be manipulated as logical units. These features are especially relevant in an eyes-free environment where the user needs to select logical parts of the document without having to point at portions of a visual display.

10.1.2 Searching, Replacing, And Spell Checking

Incremental search, a process by which the system prompts the user for a search string and moves the selection to the next available match while allowing the user to add more characters to the search string, is the search technique of choice among most Emacs users. As the system successively finds each match, it provides the user the option of continuing the search. Incremental search is a more complex instance of traditional search interaction because in addition to either stopping or continuing the search, the user can modify the current search in a number of ways including specifying a longer (or shorter) search string.

All of the user commands available during incremental search are documented in the online Emacs info manual. These are speech-enabled by Emacspeak to provide spoken prompts as the dialog begins; auditory icons indicate a search hit or search miss as the search progresses. Along with auditory icons *search-hit* and *search-miss* the user also hears the current line spoken, and in the case of a search hit, the matching text is *aurally* highlighted by using the standard audio formatting technique of changing voice characteristic. This feedback proves extremely effective when the search pattern appears several times on a single line; the user is unambiguously cued to the current match.

Search and replace actions are an extension to the basic conversational gestures of a search dialog. In addition to specifying a search string, the user also specifies a replacement string. On the Emacspeak desktop, this functionality is provided by command *query-replace*. The speech-enabled version of this interaction prompts the user for the search and replacement texts. The auditory feedback during the interactive search and replacement process parallels that described in the case of incremental search. Audio formatting to indicate the occurrence that is about to be replaced proves an effective means of avoiding erroneous modifications to the text being edited. As an example, consider using command *query-replace* to locate and replace the second occurrence of **foo** with **bar** in the text

‘`Do not change this fool, but change this food.`’

When the search matches the first occurrence of **foo** in word *fool*, the aural highlighting helps the user in answering “no” in response to question “should this occurrence be replaced". In addition to allowing the user to supply a simple "yes or no" answer for each match, command *query-replace* also allows the user to specify a number of other valid answers as described in the online Emacs documentation.

Spell Checking

A more complex instance of conversational gesture "search and replace" is exhibited by standard spell checking dialogues. Spell checking differs from the search and replace dialog described above in that the search and replacement text is guessed by the system based on an available dictionary. Words that are not found in the dictionary are flagged as potential spelling errors, and the system offers an interactive search and replace dialog for each of these possible errors. During this dialog, the system successively selects each occurrence of the possibly erroneous word and offers a set of possible replacements. Unlike in the case of simple search and replace, more than one possible replacement string is offered, since a potential spelling error can be corrected by more than one word appearing in the dictionary.

In the visual interface, such spell checking dialogues are realized by displaying the available choices in a pop-up window and allowing the user to pick a correction Once a correction is selected, the user is offered the choice of interactively replacing the erroneous word with the correction.

The spell checking interface on the Emacspeak desktop is speech-enabled to provide fluent auditory feedback. The visual interface parallels that described above and is provided by package `ispell` which is part of the standard Emacs distribution. Emacspeak provides a spoken prompt that is composed of the line containing the possibly erroneous word (which is aurally highlighted to set it apart from the rest of the text on that line) and the available corrections. Each correction is prefixed with a number that the user can use to select it. Once a correction is selected, the interaction continues with the query and replace interaction described earlier. The speech interface to the spell checker is as fluent as the visual interface. Notice that Emacspeak users do not need to concern themselves with the details of the visual display such as "the corrections are displayed in a window at the top of the screen".

In addition to the standard spell checker described above, newer versions of Emacs include an "on-the-fly" spell checker that flags erroneous words as they are typed. Emacspeak speech-enables package `flyspell` so that such erroneous words are aurally highlighted.

10.2 Structured Editing And Templates

Editing documents based on the inherent structure present in the electronic encoding can be very efficient when using spoken interaction. We described mode *AUCTeX* — a specialized interface to authoring LaTeX documents as a special instance of such structured editing in see Section 10.1 [Document Authoring], page 26.

The Emacspeak desktop allows the user to efficiently author and maintain an electronic document based either on the structure present in the markup (as in the case of mode AUCTeX) or on special outlining constructs that allow the user to impose a desired logical structure on the document. This section describes the effect of speech-enabling such editing tools and points out the advantages in using these in a speech oriented interface.

Template-based authoring — a technique that allows the user to create a document by inserting contents into appropriate positions in a predefined template — goes hand in hand with such structured editing. Finally, structured editing can vastly simplify the creation and maintenance of structured data, for example, the data present in a UNIX password file. Such data files are in fact nothing more than a collection of database records, where each record (or line) consists of a set of fields delimited by a special character. Maintaining such files without exploiting the underlying structure often tends to be error prone. We describe editing modes that can exploit such record structure to provide a fluent editing interface. Finally, we outline a speech-enabled interface to a spreadsheet application as a complex instance of such structured data editing.

10.2.1 Outline Editing

All of the various outline editing interfaces on the Emacs desktop allow the user to *hide* or *show* the contents at the different levels of a possibly nested tree structure. Components of this tree structure can be manipulated as a unit, e.g., entire subtrees can be deleted or copied. Outline editing thus provides an efficient means of obtaining quick overviews of a document.

The visual interface displays such hidden content as a series of ellipses following the visible outline heading. Emacspeak produces auditory icon *ellipses* when speaking such outline headings.

The basic *outline mode* allows the user to specify the syntax and level of outline header lines as a regular expression. This simple technique can be used to advantage in the structured navigation of large electronic texts such as those available on the Internet from online book projects such as project Gutenberg and the Internet Wiretap. For example, when this feature is activated while reading the electronic text of a Shakespearean play, the different acts can be recognized as separate nodes in the logical structure of the document. The user can then hide the document body with a single keystroke, navigate the outline headings to find a particular act, and have that portion rendered either visually or aurally. Hiding an outline level produces auditory icon *close-object*; exposing a hidden level produces auditory icon *open-object*. For details on using mode `outline`, see the relevant section of the online Emacs info manual.

The basic outline facility described above is applicable to all content being edited or browsed on the Emacspeak desktop. In addition, Emacspeak has other specialized outline editing modes such as *folding mode* that provide extended outlining facilities. In mode *folding*, the user can create (possibly nested) *folds* — logical containers of content that are delimited by a special *fold mark*. The fold mark is typically a text string that is chosen based on the type of content that is being manipulated. Thus, when *folding* a C˜program source file, fold marks are created from C˜comments. The user can *open* or *close* any or all folds in a document, and these actions are accompanied by auditory icons *open-object* and *close-object*. By entering a fold, all editing actions are restricted to the contents of that fold; this proves a simple yet convenient way of constraining editing actions such as search and replace to specific portions of large documents. Folds can be manipulated as a unit and can be deleted, copied or moved.

Mode *folding* proves especially effective in maintaining large software modules. The technique can be used to advantage by creating folds for different sections in a module and by further placing each function appearing in a particular section in a fold of its own.

Complex functions can themselves be folded into sections where each section reflects a different stage in the algorithm implemented by that function. Thus, the technique of folding can be used as an effective aid in *literate* programming. I typically write software modules by first creating an outline structure using folds that reflect the various components of that module. Next, I populate each fold with the function signatures and documentation for the functions in each section. When I am satisfied with the overall architecture of the module, I fill in the function skeletons with actual program code. This technique is used extensively in maintaining the Emacspeak code base.

10.2.2 Template-based Authoring

Emacspeak supports two powerful template-based authoring subsystems that enable the user to quickly create and fill in templates. *Dmacro* (short for "dynamic macros") allows the user to define and invoke template-based macros that are specialized for creating different types of content. For example, when programming in C, the user can invoke dynamic macros that insert skeletons of standard C constructs with a few keystrokes. This form of editing has numerous advantages in creating consistently structured code when developing large software modules. Emacspeak speech-enables mode *dmacro* to provide succinct spoken feedback as templates are created and filled. The user invokes *dmacro* via command *insert dmacro*, which is typically bound to a single key. This results in a dialog where the user is prompted to pick one of the dynamic macros available in the current context. If the users choice can be uniquely completed, that completion is spoken; otherwise, the list of possible completions based on the available partial input is spoken, accompanied by auditory icon *help*.

An alternative template-editing facility is provided by mode *tempo* This mode is designed to be used in creating template-based editing tools for specific markup languages; a good example is mode *html-helper*, a mode for creating and updating HTML documents for publishing on the WWW (see see Section 10.1 [Document Authoring], page 26).

10.2.3 Maintaining Structured Data

Consider the following entry from file `/etc/passwd` on my laptop.

'`aster:KoUxwQ2:501:100:Aster Labrador:/home/aster:/bin/bash`'

File `/etc/passwd` is a simple instance of a text file that stores structured data records as a series of fields delimited by a special character. Each item in the file acquires *meaning* from the *position* in which it occurs for example, the fifth field contains the **user name**, Aster Labrador. More generally, structured data where each field in a record has *meaning* is found throughout the desktop in applications ranging from entries in a rolodex to rows in a spreadsheet.

Typically, users do not directly edit the stored representation of the data. Instead, application front-ends provide a more human-centric (and hopefully less error prone) user interface for modifying and maintaining the data. Thus, spreadsheet applications present the data as a two dimensional table that is automatically updated to reflect changes in the underlying data. The two dimensional table is perhaps the most commonly found visual front-end to structured data tables with row and column headers prove a succinct way of implicitly displaying the *meaning* along with the *value* of the fields making up each data record.

10.3 Browsing Structure

This section describes packages that allow you to browse structured information — these are distinct from the tools described in Section 10.2 [Structured Editing], page 28, in that they are typically used for working with content that is read-only e.g., online documentation.

10.4 Web Browsing

Note that there is newer material at `http://emacspeak.blogspot.com/2014/05/emacspeak-eww-updates-for-complete.html` and in See Section 12.74 [emacspeak-eww], page 124. For Emacs 24.4 and later, note that the built-in Web browser EWW is also the browser of choice for emacspeak.

— T. V. Raman, October 2014.

This document, “The State of Web Browsing in Emacspeak” describes the primary web browsers in use under emacspeak and ways they might be used more efficiently.

Version 1.0 — February, 2007.

Copyright © 2007 Robert D. Crawford `rdc1x@comcast.net`

10.4.1 Intro

Web browsing in emacspeak can be made as simple or as complicated as you wish to make it. On the one hand, all that need be done is to open a web browser and fetch a particular url. This works fine in many cases with exceptions being things like html tables. On the other hand, if you spend a little time learning a particular browser and its functionality, web browsing can be made more efficient and the web more navigable.

This document is intended to be an introduction to the 2 primary browsers that run under emacs: emacs/w3 and EWW. I also intend to introduce a couple of add-on packages intended to make life easier and to address specific shortcomings.

The primary reason for this documentation is the fact that a lot of the documentation of the various functions tells exactly what it does, but not why or in what circumstance one might use it. Hopefully I can remedy that here.

Who this document is intended for

While this document is geared toward emacspeak users, it might be helpful for other users as well. I have tried to note where functions are specific to emacspeak.

Assumptions

This document assumes emacs is installed. If reading the section on a particular browser and trying out the associated functions, it, of course, assumes that the browser is installed and working. For features specific to emacspeak, a working install of emacspeak is necessary. It is far beyond the scope of this document to help with the installation of these programs.

For help with installing any of the above bits of software, the following mailing lists and / or newsgroups are available:

- emacs - `gnu.emacs.help`
- emacs/w3 - the mailing list for w3 is pretty dead. Sorry. The documentation is, however, not terrible.
- emacspeak - see `http://emacspeak.sf.net`

It is highly suggested that the mailing list archives, google, and the relevant documentation be consulted before posting messages to any mailing list. Nothing is more irritating than answering the same questions over and over. Those of you with kids know what I am talking about.

It is also assumed that the reader is comfortable with using emacs itself. Understanding the convention for communicating keystrokes to run commands, navigating documents, and the like are not covered.

10.4.2 emacs-w3

emacs/w3 is a web browser written completely in emacs lisp. It has some really nice features applicable to the emacspeak community such as the ability to navigate tables and support for the w3c's aural cascading stylesheets.

emacs/w3 advantages

As mentioned above, the ability to navigate tables is a super help. emacs/w3 also has support for cascading stylesheets. This allows incredible control of voices used for what would normally be visual attributes of the text such as bold, italics, preformatted text and the like.

Another advantage of emacs/w3 is that it is written completely in emacs lisp. With some effort, emacs/w3 is very customizable and quite extendable.

emacs/w3 disadvantages

Rendering can be slow. Sometimes it can be painfully, excruciatingly slow. That might be a slight exaggeration, but slow it is. This is because it is written in lisp... something I mentioned just above as a strength. It is a trade-off, but one that some see as worth it. There are, however, some things that can be done to speed up the browsing process. See See [`emacspeak-w3-xsl-transform`], page 35, for more information.

Another disadvantage is that emacs/w3 chokes on some pages. Sometimes it gives error messages and doesn't display anything. Sometimes it does this to some people and doesn't do it to others as we saw on the emacspeak mailing list a short time ago. Sometimes it gives error messages and renders the page anyway.

One other major annoyance of emacs/w3 is that sometimes it simply stops doing anything while rendering a page. There is a way I have found to get around this. I hit *`C-g`*. I usually wait five to ten seconds and then simply quit, using *`C-g`*. Not always, but usually, the page has already completed downloading and is being rendered and it therefore is not a problem.

emacs/w3 also has no bookmark functionality. This can be remedied in several ways. One simple way, mentioned below, is to use the emacs package bmk-mgr. Another way is to use org mode with remember which is the method used by Dr. Raman, the author of emacspeak. See See [bmk-mgr], page 37, later in this manual, for more information.

History back and next in the browser also seem to be broken but this is not generally a problem for me as I never look back.

emacs/w3 native functions

Many functions for efficient navigation of the internet are native to emacs/w3. Some of the most useful functions are listed below

Unlike emacs-w3m, the information presented by `describe-mode` is very complete, but a little terse. All the function names are listed and asking for help on particular functions works well. That being said, use this list to augment, not replace, the built-in help that is available in emacs.

`C-f` will open a new buffer containing the cell point is in. In most instances, this works very well. Imagine you are looking at a page that is divided into 4 distinct areas: a cell at the top of the page that contains a banner and some navigation, and a "body" area that is divided into three sections consisting of more navigation, an article, and advertisement. If point is in the main article cell, using `C-f` will open another buffer that contains only the text of that cell, the article you are interested in.

One caveat is that this does not always work as advertised. Sometimes the leftmost character of each line is missing. At least it makes for interesting reading. Usually when I have this problem I simply exit that buffer and linearize the tables in the original page.

The `m` key executes a very useful command. It will complete a link on the page. Imagine that you are reading through a document and you hear a link that you need to visit. You could tab through all the links until you hear the one you want or you could hit the `m` key and enter the link text at the prompt. Completion is available and it is not case-sensitive. Efficient, no?

The period in a cell will speak the contents of that particular cell. This command is, in my opinion, most useful when navigating tables with cells that have only one paragraph or less. I tend to not read whole articles in this manner because, inevitably, someone will interrupt me and I will lose my place.

The equals key, while in a table cell will give you the cell information. It tells you the row and column position, the size of the table, and at what nesting level the table is.

The pipe key, is used to read the table column. As this command seems to read the rectangle the column is in, this command is most useful when used in a table where there is no column spanning, i.e. all rows and columns are uniform.

Here is a list of table navigation commands:

- `C-e +` moves to the beginning of the next table row.
- `C-e -` moves to the beginning of the previous table row.
- `C-e <` moves to the beginning of the table.
- `C-e >` moves to the end of the table.
- `C-e =` moves to the top of the table column.
- `C-e <DOWN>` moves to the next cell down in the same column.
- `C-e <UP>` moves to the previous cell in the same column.

- `C-e <LEFT>` moves to the previous cell in the same row.
- `C-e <RIGHT>` moves to the next cell in the same row.

As you can see, table navigation in w3 can be easy and fun.

emacspeak specific functions for emacs/w3

An incredible amount of work has been done by the emacspeak community to make emacs/w3 accessible to those with visual impairments. Here is an explanation of some of those functions.

The command `C-t` will toggle the visibility of table borders. This command might be useful where you want to hear all punctuation symbols on a page but the table characters get in the way.

The quote key will execute a command that allows you to skim the contents of the buffer. it will read the page, paragraph-by-paragraph, pausing between paragraphs to prompt you to move on by pressing `SPACE`. If you hit `SPACE` in the middle of a paragraph, it skips to the next paragraph.

Another skimming command is bound to the `z` key. This will allow you to zip through web pages by logical blocks such as div, paragraph, and table tags.

Using the imenu facilities is another way of skimming the document and getting to the information you desire. imenu works especially well for well-structured documents. The first thing that need be done is to copy the `w3-imenu.el` file from the `contrib` directory of the `w3` directory to somewhere in your load path. I am using the cvs version of w3 and my w3 directory is under `/home/rdc/sourceforge`. The easiest thing to do is probably to do an `M-x locate` and search for `w3-imenu.el` to see where it is. After locating the file, move it into your load path. In my case I have it under `/home/rdc/share/emacs/site-lisp/`.

There are two ways to use the imenu facilities: automatically and manually. Since I do not use imenu on every site, I prefer to invoke it manually to save the time required to build the index.

Once things are in place, invoke imenu with the `j` key. This will ask you for an index position. Hitting `TAB` will give you a list of the possible index positions. Another way of navigating the document would now be to use the keys `M-n` and `M-p` to go to the next and previous index positions. Note that you have to build an index for a page before you can use these commands.

`Cap A` and `cap R` serve the same function. `Cap A` browses the Atom feed at point and `cap R` browses the rss feed at point. This is useful to sample the feed, so to speak, before going through the trouble of configuring your feed reader to fetch the feed. It might also be useful to grab the headlines from a page and present them in a more concise, readable format. If using the sort-tables xsl transform, there will be a link at the top of the page if there is an rss feed available. See See [emacspeak atom and emacspeak rss], page 37, for more information.

Google provides many useful tools for web surfers. The following commands are useful to access much google goodness.

`Cap C` extracts the current page from the google cache. With a prefix argument it will extract the link under point. This is useful for those times when a particular site is down...

maybe it is in the cache... maybe it is not. It can also be used for when particular pages are removed from a site like in the case of a government conspiracy. Are we at war with Eurasia or East Asia?

The slash key will search google for pages similar to the current page.

The command `g` will do a google search restricted to the site of the document.

The `l` command googles for who links to this page.

The command `t` runs the url under point through the google transcoder. This is useful for sites that are heavy on the use of tables and the xsl transforms are not helpful. It also works on some sites that use javascript to go to the next page in the story, such as Reuters. Using a prefix argument with this command will untranscode the url under point for pages that are currently transcoded.

`Cap T` will jump to the first occurrence of the title in the document. Multiple consecutive executions of this command will jump to further occurrences. This command is probably one of the most useful timesavers while web browsing.

`M-s` jumps to the "submit" button for the form you are editing.

`M-r` plays the media stream at point with the default media player.

The `y` command will rewrite the url of the url under point. This is useful for those sites you frequent. Often, sites that have printer friendly content have a specific way in which the url is written. For The first time you run this command in a particular buffer you are prompted for a pattern to use. The pattern is in the form of

‘`("from string" "to string")`’

The opening paren is supplied. Remember to quote the strings or you will get an error. From this point on, until you kill the buffer in which you wrote the rule, hitting `y` on a link will use this rewrite rule to visit the page. If you mistyped the rule, providing a numeric argument will allow you to rewrite the rewrite rule. I love alliteration.

Saving the best for last, `e` is the xsl map prefix. As I mentioned in the section on emacs-w3m, xsl transforms are some powerful magic that takes a web page and transforms it in some way. Linearizing tables is a good example, and the one I use most often.

The keystroke `e a` prompts for an xsl transform to apply to the current page. If you know the name of the particular transform you want you can use tab completion to select it. Otherwise, you can hit `TAB` to get a buffer that contains the list of choices.

If you know that you want a particular transform done automatically you can use the command `e s` to select a transform and then `e o` to turn xslt on (the same command will turn xslt off). Then, every page opened from that point on will have the transform applied.

There is the variable `emacspeak-w3-xsl-transform` that can be set via the usual methods. This variable specifies a transform to use before displaying a web page. There is an advantage to turning on xsl transforms all the time. If you use the `identity.xsl`, the `linearize-tables.xsl` or the `sort-tables.xsl` it can actually speed up rendering of the page. This is because the transforms provide clean and balanced html to the renderer. Additionally, using `sort-tables.xsl` or `linearize-tables.xsl` will provide a little more boost as rendering nested tables is particularly difficult for a web browser.

Sometimes it is just easier to read the printer friendly version of a story instead of having to linearize the tables and search for the content. Also, some sites, like the New York Times, I believe, make you navigate several pages to read the whole story, but if you select a "Print this story" link you can read the entire story on one page formatted without a lot of the cruft on the normal page. This is where the *e Cap P* command comes in. It will extract all the print streams from the current document.

Closely associated commands are *e r* and *e Cap R* which extract the media streams from the current page and from the link under point, respectively.

The command *e y* is another command that is useful for frequented sites. It does the same as the *y* command above in that it rewrites the url at point and follows it. In addition, it filters the output by a particular CSS class.

The command *e e* does more magic to the url at point. It processes the url using a specific function. For example, it can be used in retrieving radio content from the BBC. If you execute *C-e u* and type in *BBC Channel On Demand* or use tab completion to get the same, and then type in *radio4* you will be presented with a page containing a plethora of links to other pages containing information about particular shows. On these pages there is, somewhere, a link that will play the program. If you hit enter on one of the links on the first page, you will be taken to one of these description pages. By using the *e e* command on a link you cut out this middle step and proceed directly to playing the program you are interested in.

If there is no executor defined for a current buffer, hitting `TAB` after *e e* will give you a list of possibilities to choose from. One nice feature of this function is that it can be used for any function. If you cannot remember the keystroke that will play the url under point in `emacspeak-m-player`, but you know what it is called, you can hit *e e* and then enter the name of the function. Nice.

The *e f* command will run the current page through an XPath filter and return the results. For more information on XPath, see `http://en.wikipedia.org/wiki/XPath`. If you wanted to see only the links on a page, when prompted enter *//a* and you will be returned every link on the page. If you wanted to see only the contents of "p" tags, you would enter *//p*. This can be useful for many things, form elements included. Giving this command a prefix argument will reverse the filter, giving you everything but the content of the specified tag.

A related command can be invoked with the *e p* keystroke. This command does the same as the filter above but works on the url under point.

emacs/w3 tips and tricks

As I mentioned above, using *C-g* when it seems the browser is not responding will often display the page with no ill effects. Your mileage may vary. Taxes, tags and title are extra.

Also mentioned above is the use of sort-tables or identity as an xsl transform to speed up the rendering of pages. Every little bit helps.

Another useful tip is the use of the *k* key. This key will place the current url in the kill-ring for later yanking. If a page will not render correctly, using *k* will get the url and allow me to pass it to emacs-w3m. The counterpart to this command, *cap K* puts the url under point in the kill-ring.

10.4.3 Add-ons

Some of these are emacspeak specific, some are not. You can usually tell by the name.

emacspeak url template

I love this package. Since changing my primary browser to emacs/w3 I have really been giving the `url-template` package a workout. The `url-template` package contains templates that prompt you for information to supply to various sites to retrieve information without all the fuss of having to go to the site and navigate it. One really nice thing about url-templates is the fact that they need not be web pages. Media streams can also be made into url-templates.

The way to get to the templates is with the command *`C-e u`*. A `TAB` at the prompt will give you a list of the available templates. You should go now and have a look at the info manual section on See Section 12.232 [URL Templates], page 360, and read it. I'll wait here.

By the way, the ones I find most useful are the "Google Hits", "emacswiki search" "NPR On Demand" and "Weather forecast from Weather Underground".

emacspeak atom and emacspeak rss

These are fairly simple rss and atom browsers for the emacspeak desktop. Using the Customize interface you add feeds in the form of titles and urls. Then you call the readers with *`C-e C-u`* for rss feeds and use *`M-x emacspeak-atom-browse`* for atom feeds. There is also emacspeak support for newsticker, an rss / atom reader that is a part of emacs 22, but I have never used it.

I personally use See Info file `gnus`, node '`Top`', for rss feeds but setting gnus up for only that purpose is like hunting rabbits with a bazooka.

emacspeak websearch

`emacspeak-websearch` provides more search options than you can shake a stick at; `emacspeak-websearch` provides search for dictionaries, news sites, software sites, google tools, weather, currency converter and much more. It can be accessed with the keystroke *`C-e ?`*. At the prompt, you can enter another question mark to get a list of the available search options. You will then be prompted for the necessary information. One of the nice things about this package is that it attempts to jump to and read the most relevant information on the result page. Module `emacspeak-websearch` is complemented by module `emacspeak-url-template`; that module provides URL templates that prompt for and retrieve the relevant information from complex Web pages.

See Section 12.203 [emacspeak-websearch], page 299, and See Section 12.195 [emacspeak-url-template], page 286, for more information.

bmk-mgr

This is a newcomer to the emacspeak world. In the interest of full disclosure, I am the one that wrote the emacspeak module that makes this package accessible. It is a bookmarks manager that works with both emacs/w3 and emacs-w3m. As of the writing of this document there are still some issues, especially when using it on emacs version 22, but those are being worked on. I think this is a good solution to the problem of emacs/w3 not having bookmarks functionality and providing one central bookmark location for those who regularly use both browsers. See `http://www.emacswiki.org/cgi-bin/wiki/EmacsBmkMgr`, for more information.

10.4.4 Conclusion

emacspeak makes the internet not only accessible to those with visual impairments, but it makes browsing and information retrieval quick and efficient. If a user will spend a little time up front to learn the tools available to access the web, the increase in efficiency and ability will more than make up for the time spent. The nice thing about these tools is that you can integrate them in your day-to-day as you have the time. While it is not necessary to use everything mentioned in the above document, if you add some of these tools to your repertoire you will not be sorry.

In the end, no one makes you use a hammer to drive nails but it sure beats using a banana.

10.5 Messaging

Working with messaging applications involves both authoring and browsing content. Emacspeak provides a rich set of speech-enabled messaging tools. Further, all of the tools described in the previous sections integrate smoothly with the messaging applications described here; this means that you do not need to re-learn a new set of work habits when dealing with content in your messaging application.

10.6 Editing Code

Files containing program source code form a very specific class of **structured** documents. Unlike documents meant for human consumption that are often only loosely structured, program source (as a concession to the computer's intolerance of lack of structure) are per force well-structured and adhere to a fairly stringent syntax.

The Emacs environment provides editing modes that are specific to creating and maintaining software written in most popular programming languages. Many of these editing modes are speech-enabled by Emacspeak. Speech-enabling these modes includes providing a rich set of navigational commands that allow you to move through the source efficiently. In addition, Emacspeak's core **voice-lock** facilities are used to produce audio formatted output — this helps you spot errors quickly.

10.7 Development Environment

In addition to providing specialized editing modes for creating and maintaining program source, Emacs provides a rich set of software development tools that can be combined to create powerful Integrated Development Environments (IDE). These IDEs are speech-enabled by Emacspeak to provide a versatile and powerful environment for eyes-free software development.

10.8 Desktop Management

Emacs provides an integrated environment for performing all of ones day-to-day computing tasks ranging from electronic messaging to software development. The environment derives its power from the fact that this integration allows for content to be handled across different tasks in a seamless manner. In order to work effectively with large Emacs sessions with many documents and applications open at the same time, the Emacspeak desktop provides a powerful collection of desktop management tools designed to help the user easily locate objects that pertain to a given task.

10.9 Personal Information Management

This section describes speech-enabled tools designed to aid in personal information management such as maintaining a daily calendar.

10.10 Desktop Applications

10.10.1 Spread Sheets

Spreadsheet applications present a two dimensional view of structured data where the field values are (possibly) mutually dependent. On the Emacspeak desktop, a speech-enabled spreadsheet application can be used to manipulate such data-driven documents ranging from simple cheque books and expense reports to complex investment portfolios. Where the traditional visual interface to spreadsheets is typically independent of the semantics of the data stored in the spreadsheet, the speech-enabled interface is derived from the *meaning* of the various fields making up the data. When presenting such information on a visual display, implicit visual layout can be used to cue the user to the meaning of different data fields.

On the other hand, in the case of an actively scrolling auditory display, the spoken output needs to explicitly convey both the value and interpretation of the different data items. In addition, the interface needs to enable an active dialog between user and application where the user is able to query the system about the possible meaning of a particular item of data.

Finally, the aural interface needs to enable *multiple* views of the display. In the visual interface, such *multiple* views are automatically enabled by the two dimensional layout combined with the eye's ability to move rapidly around the layout structure. Thus, while viewing any particular row of a portfolio, one can immediately see the current total value as well as the net gain or loss. The Emacs spread-sheet package `dismal` can be retrieved from `ftp://cs.nyu.edu/pub/local/fox/dismal`.

10.10.2 Forms Mode

Forms mode an Emacs mode designed to edit structured data records like the line shown from file `/etc/passwd` presents a user-friendly visual interface that displays the field name along with the field value. The user can edit the field value and save the file, at which point the data is written out using the underlying `:` delimited representation. Mode *forms* provides a flexible interface to associating meaning to the fields of such structured data files. For details on it use, see the forms-mode section of the online Emacs info documentation.

10.10.3 OCR — Reading Print Documents

Module `emacspeak-ocr` implements an OCR (Optical Character Recognition) front-end for the Emacspeak desktop.

Page image is acquired using tools from package `SANE` (Scanner Access Now Easy). The acquired image is run through the OCR engine if one is available, and the results placed in a buffer that is suitable for browsing the results. This buffer is placed in mode `emacspeak-ocr-mode` a specialized mode for reading and scanning documents.

10.10.3.1 Emacspeak OCR Mode

Emacspeak OCR mode is a special major mode for document scanning and OCR.

Pre-requisites:

- A working scanner back-end like SANE on Linux.
- An OCR engine.

Make sure your scanner back-end works, and that you have the utilities to scan a document and acquire an image as a tiff file. Then set variable *emacspeak-ocr-scan-image-program* to point at this utility. By default, this is set to 'scanimage' which is the image scanning utility provided by SANE.

By default, this front-end attempts to compress the acquired tiff image; make sure you have a utility like `tiffcp`. Variable *emacspeak-ocr-compress-image* is set to 'tiffcp' by default; if you use something else, you should customize this variable.

Next, make sure you have an OCR engine installed and working. By default this front-end assumes that OCR is available as `/usr/bin/ocr`.

Once you have ensured that acquiring an image and applying OCR to it work independently of Emacs, you can use this Emacspeak front-end to enable easy OCR access from within Emacspeak.

The Emacspeak OCR front-end is launched by command *emacspeak-ocr* bound to *`C-e C-o`*.

This command switches to a special buffer that has OCR commands bounds to single keystrokes — see the key-binding list at the end of this description. Use Emacs online help facility to look up help on these commands.

Mode *emacspeak-ocr-mode* provides the necessary functionality to scan, OCR, read and save documents. By default, scanned images and the resulting text are saved under directory `~/ocr`; see variable *emacspeak-ocr-working-directory*. Invoking command *emacspeak-ocr-open-working-directory* bound to *`d`* will open this directory.

By default, the document being scanned is named 'untitled'. You can name the document by using command *emacspeak-ocr-name-document* bound to *`n`*. The document name is used in constructing the name of the image and text files.

Here is a list of all emacspeak OCR commands along with their key-bindings and a brief description:

`digit` *emacspeak-ocr-page* Jumps to specified page in the OCR output.

`c` *emacspeak-ocr-set-compress-image-options* Interactively update image compression options. Prompts with current setting in the minibuffer. Setting persists for current Emacs session.

`i` *emacspeak-ocr-set-scan-image-options* Interactively update scan image options. Prompts with current setting in the minibuffer. Setting persists for current Emacs session.

`spc` *emacspeak-ocr-read-current-page* Speaks current page.

s *emacspeak-ocr-save-current-page* Saves current page as a text file.

p *emacspeak-ocr-page* Prompts for a page number and moves to the specified page.

] *emacspeak-ocr-forward-page* Move forward to the next page.

[*emacspeak-ocr-backward-page* Move back to the previous page.

d *emacspeak-ocr-open-working-directory* Open directory containing the results of OCR.

n *emacspeak-ocr-name-document* Name current document.

o *emacspeak-ocr-recognize-image* Launch OCR engine on a scanned image.

i *emacspeak-ocr-scan-image* Acquire an image using `scanimage`.

RET *emacspeak-ocr-scan-and-recognize* Scan and recognize a page.

w *emacspeak-ocr-write-document* Write all pages of current document to a text file.

q *bury-buffer* Bury the OCR buffer.

c *emacspeak-ocr-customize* Customize Emacspeak OCR settings.

? *describe-mode* Describe OCR mode.

11 Running Terminal Based Applications.

You can use the terminal emulator mode to run arbitrary terminal-based programs from within Emacs. You open a terminal emulator buffer using *M-x term*, with an extra carriage return to accept the default shell (such as `bash`). (Incidentally, don't confuse this command with *M-x terminal-emulator*, which starts an older terminal emulator mode not supported by Emacspeak.)

Three kinds of commands are used within the terminal emulator. Normal term commands use a prefix of *C-c*. The emacspeak commands for eterm mode use a prefix of *C-t*. Anything else is a normal shell command.

There are two sub-modes of term mode: char sub-mode and line sub-mode. In char sub-mode, emacspeak will only speak the final chunk of output — typically the last line displayed. Each character typed (except 'term-escape-char') is sent immediately. Use char sub-mode for screen oriented programs like `vi` or `pine`.

In line sub-mode, program output is spoken if user option `eterm-autospeak` is turned on. When you type a return at the end of the buffer, that line is sent as input, while return not at end copies the rest of the line to the end and sends it. When using terminal line mode with option `eterm-autospeak` turned on, speech feedback is similar to that obtained in regular `shell-mode` buffers.

The default is char sub-mode. You can switch to line sub-mode with *C-c C-j* (recall that control J is a linefeed), and back to char sub-mode with *C-c C-k* (think of character spelled with a K).

Note: Use char-mode with the terminal emulator for running screen-oriented programs like Lynx or Pine. For regular shell interaction just use *M-x shell* instead of using the terminal emulator.

11.1 Char Sub-mode of Term Mode

In char sub-mode of term, each character you type is sent directly to the inferior process without intervention from emacs, except for the escape character (usually *C-c*).

Here are some of the useful commands for the char sub-mode. Note that the usual commands for killing a buffer or switching buffers do not work in this mode, so new key bindings are supplied. The first five commands are different ways of leaving this mode.

C-c C-j
M-x term-line-mode
: Switch to line sub-mode of term mode.

C-c o
M-x other-window
: Select the next window on this frame. All windows on current frame are arranged in a cyclic order. This command selects the next window in that order. If there are no other windows, this command does nothing.

C-c C-f
M-x find-file
: Switch to a buffer visiting a file, creating one if none already exists.

C-c 0
M-x delete-window

Remove current window from the display.

C-c k
M-x kill-buffer

Kill the current buffer.

C-c C-x C-c
M-x save-buffers-kill-emacs

Offer to save each buffer, then kill this Emacs process.

C-c C-d
M-x list-directory

Display a list of files in or matching DIRNAME, a la 'ls'. DIRNAME is globbed by the shell if necessary. Prefix arg (*C-u*) means supply -l switch to `ls`. The list appears in a second window.

C-c 1
M-x delete-other-windows

Delete all other windows in the frame, making the current window fill its frame.

C-c C-c
M-x term-send-raw

Send the last character typed through the terminal-emulator without any interpretation.

C-c (
M-x start-kbd-macro

Record subsequent keyboard input, defining a keyboard macro. The commands are recorded even as they are executed. Use *C-c)* to finish recording and make the macro available. Use *M-x* `name-last-kbd-macro` to give it a permanent name. Prefix arg (*C-u*) means append to last macro defined; This begins by re-executing that macro as if you had typed it again.

C-c)
M-x end-kbd-macro

Finish defining a keyboard macro. The definition was started by *C-c (*. The macro is now available for use via *C-c* `e`, or it can be given a name with *M-x* *name-last-kbd-macro* and then invoked under that name.

C-c e
M-x call-last-kbd-macro

Call the last keyboard macro that you defined with *C-c (*. A prefix argument serves as a repeat count. Zero means repeat until error.

You can get a list of all the key sequences with a C-c prefix by typing *C-c C-h* while in this sub-mode. Some of those commands are only available in the char sub-mode, while others are generally available.

11.2 Line Sub-mode of Term Mode

In line sub-mode of term mode, emacs editing commands work normally, until you type `RETURN` which sends the current line to the inferior process.

Here are some of the useful commands for the line sub-mode of the term mode. In addition, the usual commands for handling a buffer work in this mode (*`C-x o`* to switch windows, *`C-x k`* to kill a buffer, *`C-x f`* to find a file, and so forth).

`C-c C-k`
`M-x term-char-mode`
Switch to char sub-mode of term mode.

`C-c C-z`
`M-x term-stop-subjob`
Stop the current subjob. Resume the subjob in the foreground with the ordinary command `fg`, or run it in the background with `bg`. WARNING: if there is no current subjob, you can end up suspending the top-level process running in the buffer. If you accidentally do this, use *`M-x term-continue-subjob`* to resume the process. (This is not a problem with most shells, including `bash`, since they ignore this signal.)

`C-c C-\`
`M-x term-quit-subjob`
Send quit signal to the current subjob.

`C-c C-c`
`M-x term-interrupt-subjob`
Interrupt the current subjob.

`C-c C-w`
`M-x backward-kill-word`
Kill characters backward until encountering the end of a word.

`C-c C-u`
`M-x term-kill-input`
Kill all text from last stuff output by interpreter to point.

`C-c C-a`
`M-x term-bol`
Goes to the beginning of line, then skips past the prompt, if any. If a prefix argument is given (*`C-u`*), then no prompt skip — go straight to column 0.

`C-c C-d`
`M-x term-send-eof`
Send an end of file character (EOF) to the current buffer's process.

You can get a list of all the key sequences with a *`C-c`* prefix by typing *`C-c C-h`* while in this sub-mode. Some of those commands are only available in the line sub-mode, while others are generally available.

11.3 Eterm Mode Commands

The eterm mode maintains a pointer, which is not necessarily the same as the terminal's cursor. It is intended to be used in eterm's char submode. In char submode, `C-t ,` (that's control-t followed by comma) will tell you where the eterm pointer is. `C-t C-i` will tell you where the terminal's cursor is. The top left corner of the window is "row 0 column 0".

The eterm pointer can be moved with `C-t <` (to the top of the screen), `C-t >` (to the bottom of the screen), `C-t n` (to the next line), `C-t p` (to the previous line), and `C-t .` (to the cursor). Each of these also speaks the line the pointer moves to. You can also search forward with `C-t s`.

These commands speak without moving the pointer: `C-t l` (current line), `C-t w` (current word), `C-t c` (current character), and `C-t SPACE` (from eterm pointer to cursor).

You may enter review mode with `C-t q`. In review mode, you can search the buffer and speak its contents, without disturbing the terminal. Commands for moving the pointer are similar to normal editing commands, but without a control key: `n` and `p` for next and previous line, `f` and `b` for forward and back by characters, `<` and `>` for the beginning or end of the buffer. `c`, `w`, and `l` speak the current character, word, and line. `s` searches forward (not incrementally). A comma speaks the pointer location. A period moves the pointer to the terminal cursor. Return to normal term mode by typing `q`.

12 Emacspeak Commands And Options

Overview

This chapter documents the various Emacspeak modules and is auto-generated from the Emacspeak source code. It is meant to be a complete reference to the emacspeak implementation while also providing high-level usage summaries of some of the larger Emacspeak modules.

Each section provides a high-level overview of that module, followed by detailed description of the commands and options defined in that module.

Emacspeak modules can be classified as follows:

1. Modules that interface with various TTS engines, e.g. `espeak-voices` and `dectalk-voices`.
2. Modules that implement core Emacspeak functionality, e.g., `emacspeak-speak`, `emacspeak-keymap`, `voice-lock` and `emacspeak-advice`.
3. Emacspeak extensions that speech-enable various emacs packages, these are all named using the convention `emacspeak-<`*`packagename`*`>`.

The first two of the above are mostly of interest when extending Emacspeak, or to learn how things are implemented. The third category is useful in understanding how emacspeak works with a given package; thus, when learning to use the Emacs Web Browser (EWW), read the EWW documentation, then read See Section 12.74 [emacspeak-eww], page 124.

This chapter documents a total of 1169 commands and 863 options.

12.1 acss-structure

The CSS Speech Style Sheet specification defines a number of abstract device independent voice properties. A setting conforming to the CSS speech specification can be represented in elisp as a structure.

We will refer to this structure as a "speech style". This structure needs to be mapped to device dependent codes to produce the desired effect. This module forms a bridge between emacs packages that wish to implement audio formatting and Emacspeak's TTS module. Emacspeak produces voice change effects by examining the value of text-property 'personality.

Think of a buffer of formatted text along with the text-property 'personality appropriately set as a "aural display list". Module voice-setup.el help applications like EWW produce audio-formatted output by calling function acss-personality-from-speech-style with a "speech-style" –a structure as defined in this module and get back a symbol that they then assign to the value of property 'personality. Emacspeak's rendering engine then does the needful at the time speech is produced. Function acss-personality-from-speech-style does the following: Takes as input a "speech style" (1) Computes a symbol that will be used henceforth to refer to this specific speech style. (2) Examines emacspeak's internal voice table to see if this speech style has a voice already defined. If so it returns immediately. Otherwise, it does the additional work of defining a -voice for future use. See module voice-setup.el to see how voices are defined independent of a given TTS engine.

12.2 amixer

Provide an emacs front-end to amixer, the sound mixer in ALSA that is used to configure the audio device.

The main entry point is command `emacspeak-audio-setup` bound to *`C-e)`*. When called for the first time, this command builds up a database of available controls on the default audio device. These control names are then available for completion in the minibuffer. Pick a desired control, e.g., `"master playback volume"`, and this displays a prompt with the current value. Enter the new value and press `<RETURN>`. To reset all controls to their default values, Press *`C-j`*.

12.2.1 Amixer Commands

12.2.1.1 amixer

`amixer` (**&optional** *refresh*) [Command]

Manipulate ALSA settings.
Interactive prefix arg refreshes cache.

(fn &optional REFRESH)

12.2.1.2 amixer-store

`amixer-store` [Command]

Persist current amixer settings.

12.2.2 amixer Options

`User Option` *amixer-device* [Variable]

ALSA Control Device.

12.3 cd-tool

Provide an emacs front-end to cdtool. cdtool can be obtained as an rpm check using rpmfind or from its home site at sunsite.unc.edu /pub/Linux/apps/sound/cdrom/cli This module also provides the ability to play or save clips from a CD if you have cdda2wav installed. cdda2wav is a cd to wav converter.

12.3.1 Cd-Tool Commands

12.3.1.1 cd-tool

`cd-tool` [Command]

`C-e DEL`

`<fn> DEL`

Front-end to CDTool.
Bind this function to a convenient key-
Emacspeak users automatically have

this bound to <DEL> in the emacspeak keymap.

Key	Action
—	——
+	Next Track
-	Previous Track
SPC	Pause or Resume
e	Eject
=	Shuffle
i	CD Info
p	Play
s	Stop
t	track
c	clip
cap C	Save clip to disk

12.3.2 cd-tool Options

`User Option` *cd-tool-start-command* [Variable]

Name of cdstart command; most likely either "cdstart" or "cdplay".

12.4 dectalk-voices

This module defines the various voices used in voice-lock mode. This module is Dectalk specific.

12.4.1 Dectalk-Voices Commands

12.4.1.1 dectalk

`dectalk` [Command]

`C-e d C-d`

`<fn> d C-d`
Select Dectalk TTS.

12.4.2 dectalk-voices Options

`User Option` *dectalk-default-speech-rate* [Variable]

Default speech rate .

12.5 dom-addons

Useful additional functions for dom.el

12.6 dtk-interp

All requests to the speech server are factored out into this module. These calls are declared here as defun so they are inlined by the byte compiler. This keeps the code efficient, but gives us the flexibility to call out to different speech servers.

12.7 dtk-speak

Defines the TTS interface. Here, prefix dtk is synonymous with tts.

12.7.1 Dtk-Speak Commands

12.7.1.1 dtk-add-cleanup-pattern

`dtk-add-cleanup-pattern` (**&optional** *delete*) [Command]

`C-e d a`

`<fn> d a`

Add this pattern to the list of repeating patterns.
 Optional interactive prefix arg deletes
this pattern if previously added. Thus, by adding the repeating
pattern '.' (this is already added by default) emacspeak
will say ''aw fifteen dot'' when speaking the string
''...............'' instead of ''period period period period
''

(fn &optional DELETE)

12.7.1.2 dtk-cloud

`dtk-cloud` [Command]

`C-e d C-c`

`<fn> d C-c`
Select preferred Cloud TTS server.

12.7.1.3 dtk-list-languages

`dtk-list-languages` [Command]
Say the available languages.

12.7.1.4 dtk-local-server

`dtk-local-server` (*program* **&optional** *prompt-port*) [Command]

`C-e d L`

`<fn> d L`

Select and start an local speech server interactively. Local server
lets Emacspeak on a remote host connect back via SSH port forwarding
for instance. Argument PROGRAM specifies the speech server

program. Port defaults to dtk-local-server-port

(fn PROGRAM &optional PROMPT-PORT)

12.7.1.5 dtk-notify-initialize

`dtk-notify-initialize` [Command]

C-e d C-n

<fn> d C-n
Initialize notification TTS stream.

12.7.1.6 dtk-notify-shutdown

`dtk-notify-shutdown` [Command]

C-e d C-s

<fn> d C-s
Shutdown notification TTS stream.

12.7.1.7 dtk-notify-stop

`dtk-notify-stop` [Command]

Stop speech on notification stream.

12.7.1.8 dtk-reset-state

`dtk-reset-state` [Command]

C-e d R

<fn> d R

Restore sanity to the Dectalk.
Typically used after the Dectalk has been power cycled.

12.7.1.9 dtk-select-server

`dtk-select-server` (*program* **&optional** *device*) [Command]

C-e d d

<fn> d d

Select a speech server interactively.
When called interactively, restarts speech server.
Argument PROGRAM specifies the speech server program.
Optional arg device sets up environment variable
ALSA_DEFAULT to specified device before starting the server.

(fn PROGRAM &optional DEVICE)

12.7.1.10 dtk-set-character-scale

`dtk-set-character-scale` (*factor* **&optional** *prefix*) [Command]

C-e d f

<fn> d f

Set scale FACTOR for speech rate.
Speech rate is scaled by this factor
when speaking characters.
Interactive PREFIX arg means set the global default value, and then set the current local value to the result.

(fn FACTOR &optional PREFIX)

12.7.1.11 dtk-set-chunk-separator-syntax

`dtk-set-chunk-separator-syntax` (*s*) [Command]

C-e d RET

<fn> d RET
Interactively set how text is split in chunks.
See the Emacs documentation on syntax tables for details on how characters are classified into various syntactic classes.
Argument S specifies the syntax class.

(fn S)

12.7.1.12 dtk-set-language

`dtk-set-language` (*lang*) [Command]

C-e d S

<fn> d S

Set language to lang.

(fn LANG)

12.7.1.13 dtk-set-next-language

`dtk-set-next-language` [Command]

C-e d N

<fn> d N

Switch to the next language

12.7.1.14 dtk-set-predefined-speech-rate

`dtk-set-predefined-speech-rate` (**&optional** *prefix*) [Command]

C-e d 9

C-e d 8

C-e d 7

C-e d 6

C-e d 5

C-e d 4

C-e d 3

C-e d 2

C-e d 1

C-e d 0

<fn> d 9

<fn> d 8

<fn> d 7

<fn> d 6

<fn> d 5

<fn> d 4

<fn> d 3

<fn> d 2

<fn> d 1

<fn> d 0

Set speech rate to one of nine predefined levels.
Interactive PREFIX arg says to set the rate globally.
Formula used is:
rate = dtk-speech-rate-base + dtk-speech-rate-step * level.

(fn &optional PREFIX)

12.7.1.15 dtk-set-preferred-language

`dtk-set-preferred-language` (*alias lang*) [Command]

Set the alias of the preferred language:
For example if alias="en" lang="en_GB",
then the following call:
 dtk-set-language("en")
will set "en_GB".

(fn ALIAS LANG)

12.7.1.16 dtk-set-previous-language

`dtk-set-previous-language` [Command]

C-e d P

<fn> d P

Switch to the previous language

12.7.1.17 dtk-set-punctuations

`dtk-set-punctuations` (*mode* **&optional** *prefix*) [Command]

C-e d p

<fn> d p

Set punctuation mode to MODE.
Possible values are 'some', 'all', or 'none'.
Interactive PREFIX arg means set the global default value, and then set the current local value to the result.

(fn MODE &optional PREFIX)

12.7.1.18 dtk-set-punctuations-to-all

`dtk-set-punctuations-to-all` (**&optional** *prefix*) [Command]

Set punctuation mode to all.
Interactive PREFIX arg sets punctuation mode globally.

(fn &optional PREFIX)

12.7.1.19 dtk-set-punctuations-to-some

`dtk-set-punctuations-to-some` (**&optional** *prefix*) [Command]

Set punctuation mode to some.
Interactive PREFIX arg sets punctuation mode globally.

(fn &optional PREFIX)

12.7.1.20 dtk-set-rate

`dtk-set-rate` (*rate* **&optional** *prefix*) [Command]

C-e d r

<fn> d r

Set speaking RATE for the tts.
Interactive PREFIX arg means set the global default value, and then set the current local value to the result.

(fn RATE &optional PREFIX)

12.7.1.21 dtk-stop

`dtk-stop` (**&optional** *all*) [Command]

`C-e s`

`<fn> s`

Stop speech now.
Optional arg ‘all’ or interactive call silences notification stream as well.

(fn &optional ALL)

12.7.1.22 dtk-toggle-allcaps-beep

`dtk-toggle-allcaps-beep` (**&optional** *prefix*) [Command]

`C-e d C`

`<fn> d C`

Toggle allcaps-beep.
when set, allcaps words are indicated by a
short beep. Interactive PREFIX arg means toggle the global default
value, and then set the current local value to the result.
Note that allcaps-beep is a very useful thing when programming.
However it is irritating to have it on when reading documents.

12.7.1.23 dtk-toggle-capitalization

`dtk-toggle-capitalization` (**&optional** *prefix*) [Command]

`C-e d c`

`<fn> d c`

Toggle capitalization.
when set, capitalization is indicated by a
short beep. Interactive PREFIX arg means toggle the global default
value, and then set the current local value to the result.

12.7.1.24 dtk-toggle-punctuation-mode

`dtk-toggle-punctuation-mode` (**&optional** *prefix*) [Command]

`C-e d ,`

`<fn> d ,`

Toggle punctuation mode between "some" and "all".
Interactive PREFIX arg makes the new setting global.

(fn &optional PREFIX)

12.7.1.25 dtk-toggle-quiet

`dtk-toggle-quiet` (**&optional** *prefix*) [Command]

Toggles state of dtk-quiet.
Turning on this switch silences speech.
Optional interactive prefix arg causes this setting to become global.

12.7.1.26 dtk-toggle-speak-nonprinting-chars

`dtk-toggle-speak-nonprinting-chars` (**&optional** *prefix*) [Command]

C-e d n

<fn> d n

Toggle speak-nonprinting-chars.
Interactive PREFIX arg means toggle the global default
value, and then set the current local value to the result.

12.7.1.27 dtk-toggle-split-caps

`dtk-toggle-split-caps` (**&optional** *prefix*) [Command]

C-e d s

<fn> d s

Toggle split caps mode.
Split caps mode is useful when reading
Hungarian notation in program source code. Interactive PREFIX arg
means toggle the global default value, and then set the current local
value to the result.

12.7.1.28 dtk-toggle-splitting-on-white-space

`dtk-toggle-splitting-on-white-space` [Command]

C-e d SPC

<fn> d SPC
Toggle splitting of speech on white space.
This affects the internal state of emacspeak that decides if we split
text purely by clause boundaries, or also include
whitespace. By default, emacspeak sends a clause at a time
to the speech device. This produces fluent speech for
normal use. However in modes such as 'shell-mode' and some
programming language modes, clause markers appear
infrequently, and this can result in large amounts of text
being sent to the speech device at once, making the system
unresponsive when asked to stop talking. Splitting on white
space makes emacspeak's stop command responsive. However,
when splitting on white space, the speech sounds choppy
since the synthesizer is getting a word at a time.

12.7.1.29 dtk-toggle-strip-octals

`dtk-toggle-strip-octals` (**&optional** *prefix*) [Command]

`C-e d o`

`<fn> d o`

Toggle stripping of octals.
Interactive prefix arg means
 toggle the global default value, and then set the current local
value to the result.

12.7.1.30 tts-restart

`tts-restart` [Command]

`C-e C-s`

`<fn> C-s`

Use this to nuke the currently running TTS server and restart it.

12.7.1.31 tts-speak-version

`tts-speak-version` [Command]

`C-e d V`

`<fn> d V`

Speak version.

12.7.2 dtk-speak Options

`User Option` *dtk-cleanup-repeats* [Variable]

List of repeating patterns to clean up. Use command 'dtk-add-cleanup-pattern' bound to C-e d a to add more patterns.

If more than 3 consecutive occurrences of a specified pattern is found, the TTS engine replaces it with a repeat count.

`User Option` *dtk-cloud-server* [Variable]

Set this to your preferred cloud TTS server.

`User Option` *dtk-local-engine* [Variable]

Engine we use for our local TTS server.

`User Option` *dtk-speak-nonprinting-chars* [Variable]

Option that specifies handling of non-printing chars. Non nil value means non printing characters should be spoken as their octal value. Set this to t to avoid a dectalk bug that makes the speech box die if it seems some accented characters in certain contexts.

`User Option` *dtk-speech-rate-base* [Variable]

Value of lowest tolerable speech rate.

`User Option` *dtk-speech-rate-step* [Variable]

Value of speech rate increment. Determines step size when setting speech rate via command 'dtk-set-predefined-speech-rate'.

`User Option` *dtk-use-tones* [Variable]

Allow tones to be turned off.

`User Option` *tts-notification-device* [Variable]

Virtual ALSA device to use for notifications stream.

12.8 dtk-unicode

This Provides Unicode support to the speech layer.

12.8.1 Dtk-Unicode Commands

12.8.1.1 dtk-unicode-customize-char

`dtk-unicode-customize-char` (*char replacement*) [Command]

Add a custom replacement string for CHAR.

When called interactively, CHAR defaults to the character after point.

(fn CHAR REPLACEMENT)

12.8.1.2 dtk-unicode-uncustomize-char

`dtk-unicode-uncustomize-char` (*char*) [Command]

Delete custom replacement for CHAR.

When called interactively, CHAR defaults to the character after point.

(fn CHAR)

12.8.2 dtk-unicode Options

`User Option` *dtk-unicode-character-replacement-alist* [Variable]

Replacements for characters.

`User Option` *dtk-unicode-name-transformation-rules-alist* [Variable]

Alist of character name transformation rules.

`User Option` *dtk-unicode-process-utf8* [Variable]

Turn this off when working with TTS engines that handle UTF8.

`User Option` *dtk-unicode-untouched-charsets* [Variable]

Characters of these charsets are ignored by dtk-unicode-replace-chars.

12.9 emacspeak

Emacspeak extends Emacs to be a fully functional audio desktop. This is the main emacspeak module. It actually does very little: It sets up Emacs to load package-specific Emacspeak modules as each package is loaded. It implements function emacspeak which loads the rest of the system.

12.9.1 emacspeak Options

`User Option` *emacspeak-play-emacspeak-startup-icon* [Variable]

If set to T, emacspeak plays its icon as it launches.

`User Option` *voice-bold* [Variable]

Personality used for bold This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-bold-italic* [Variable]

Personality used for bold-italic This personality uses voice-bolden-and-animate whose effect can be changed globally by customizing voice-bolden-and-animate-settings.

`User Option` *voice-builtin-personality* [Variable]

Personality used for font-lock-builtin-face This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-button* [Variable]

Personality used for button This personality uses voice-bolden-medium whose effect can be changed globally by customizing voice-bolden-medium-settings.

`User Option` *voice-comment-delimiter-personality* [Variable]

Personality used for font-lock-comment-delimiter-face This personality uses voice-smoothen-medium whose effect can be changed globally by customizing voice-smoothen-medium-settings.

`User Option` *voice-comment-personality* [Variable]

Personality used for font-lock-comment-face This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-constant-personality* [Variable]

Personality used for font-lock-constant-face This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-doc-personality* [Variable]

Personality used for font-lock-doc-face This personality uses voice-monotone-medium whose effect can be changed globally by customizing voice-monotone-medium-settings.

`User Option` *voice-error* [Variable]

Personality used for error This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

`User Option` *voice-file-name-shadow* [Variable]

Personality used for file-name-shadow This personality uses voice-monotone-medium whose effect can be changed globally by customizing voice-monotone-medium-settings.

`User Option` *voice-fixed-pitch* [Variable]

Personality used for fixed-pitch This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-fringe* [Variable]

Personality used for fringe This personality uses voice-monotone-medium whose effect can be changed globally by customizing voice-monotone-medium-settings.

`User Option` *voice-function-name-personality* [Variable]

Personality used for font-lock-function-name-face This personality uses voice-bolden-medium whose effect can be changed globally by customizing voice-bolden-medium-settings.

`User Option` *voice-header-line* [Variable]

Personality used for header-line This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-help-argument-name* [Variable]

Personality used for help-argument-name This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

`User Option` *voice-highlight* [Variable]

Personality used for highlight This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

`User Option` *voice-hydra-face-amaranth* [Variable]

Personality used for hydra-face-amaranth This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

`User Option` *voice-hydra-face-blue* [Variable]

Personality used for hydra-face-blue This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-hydra-face-pink* [Variable]

Personality used for hydra-face-pink This personality uses voice-bolden-medium whose effect can be changed globally by customizing voice-bolden-medium-settings.

`User Option` *voice-hydra-face-red* [Variable]

Personality used for hydra-face-red This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-hydra-face-teal* [Variable]

Personality used for hydra-face-teal This personality uses voice-lighten-medium whose effect can be changed globally by customizing voice-lighten-medium-settings.

`User Option` *voice-isearch* [Variable]
Personality used for isearch This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-isearch-fail* [Variable]
Personality used for isearch-fail This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-italic* [Variable]
Personality used for italic This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

`User Option` *voice-keyword-personality* [Variable]
Personality used for font-lock-keyword-face This personality uses voice-animate-extra whose effect can be changed globally by customizing voice-animate-extra-settings.

`User Option` *voice-link* [Variable]
Personality used for link This personality uses voice-bolden-medium whose effect can be changed globally by customizing voice-bolden-medium-settings.

`User Option` *voice-link-visited* [Variable]
Personality used for link-visited This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-match* [Variable]
Personality used for match This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-negation-char-personality* [Variable]
Personality used for font-lock-negation-char-face This personality uses voice-brighten-extra whose effect can be changed globally by customizing voice-brighten-extra-settings.

`User Option` *voice-outline-1* [Variable]
Personality used for outline-1 This personality uses voice-bolden-extra whose effect can be changed globally by customizing voice-bolden-extra-settings.

`User Option` *voice-outline-2* [Variable]
Personality used for outline-2 This personality uses voice-bolden-medium whose effect can be changed globally by customizing voice-bolden-medium-settings.

`User Option` *voice-outline-3* [Variable]
Personality used for outline-3 This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-outline-4* [Variable]
Personality used for outline-4 This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-outline-5* [Variable]
Personality used for outline-5 This personality uses voice-lighten-medium whose effect can be changed globally by customizing voice-lighten-medium-settings.

`User Option` *voice-outline-6* [Variable]

Personality used for outline-6 This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-preprocessor-personality* [Variable]

Personality used for font-lock-preprocessor-face This personality uses voice-monotone-medium whose effect can be changed globally by customizing voice-monotone-medium-settings.

`User Option` *voice-query-replace* [Variable]

Personality used for query-replace This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-regexp-grouping-backslash* [Variable]

Personality used for font-lock-regexp-grouping-backslash This personality uses voice-smoothen-extra whose effect can be changed globally by customizing voice-smoothen-extra-settings.

`User Option` *voice-regexp-grouping-construct* [Variable]

Personality used for font-lock-regexp-grouping-construct This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

`User Option` *voice-region* [Variable]

Personality used for region This personality uses voice-brighten whose effect can be changed globally by customizing voice-brighten-settings.

`User Option` *voice-secondary-selection* [Variable]

Personality used for secondary-selection This personality uses voice-lighten-medium whose effect can be changed globally by customizing voice-lighten-medium-settings.

`User Option` *voice-shadow* [Variable]

Personality used for shadow This personality uses voice-monotone-medium whose effect can be changed globally by customizing voice-monotone-medium-settings.

`User Option` *voice-shr-link* [Variable]

Personality used for shr-link This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-string-personality* [Variable]

Personality used for font-lock-string-face This personality uses voice-lighten-extra whose effect can be changed globally by customizing voice-lighten-extra-settings.

`User Option` *voice-success* [Variable]

Personality used for success This personality uses voice-brighten-extra whose effect can be changed globally by customizing voice-brighten-extra-settings.

`User Option` *voice-type-personality* [Variable]

Personality used for font-lock-type-face This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

`User Option` *voice-underline* [Variable]

Personality used for underline This personality uses voice-lighten-extra whose effect can be changed globally by customizing voice-lighten-extra-settings.

`User Option` *voice-variable-name-personality* [Variable]

Personality used for font-lock-variable-name-face This personality uses voice-bolden-extra whose effect can be changed globally by customizing voice-bolden-extra-settings.

`User Option` *voice-variable-pitch* [Variable]

Personality used for variable-pitch This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

`User Option` *voice-warning* [Variable]

Personality used for warning This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

`User Option` *voice-warning-personality* [Variable]

Personality used for font-lock-warning-face This personality uses voice-bolden-and-animate whose effect can be changed globally by customizing voice-bolden-and-animate-settings.

12.10 emacspeak-2048

Speech-enable 2048 Game

12.10.1 Emacspeak-2048 Commands

12.10.1.1 emacspeak-2048-add-column

`emacspeak-2048-add-column` [Command]

Add a column to the current board.

12.10.1.2 emacspeak-2048-add-row

`emacspeak-2048-add-row` [Command]

Add a row to the current board.

12.10.1.3 emacspeak-2048-drop-column

`emacspeak-2048-drop-column` [Command]

Drop last row from the current board.

12.10.1.4 emacspeak-2048-drop-row

`emacspeak-2048-drop-row` [Command]

Drop last row from the current board.

12.10.1.5 emacspeak-2048-export

`emacspeak-2048-export` (**&optional** *prompt*) [Command]

Exports game stack to a file.
Optional interactive prefix arg prompts for a file.
Note that the file is overwritten silently.

(fn &optional PROMPT)

12.10.1.6 emacspeak-2048-import

`emacspeak-2048-import` (**&optional** *prompt*) [Command]

Import game.
Optional interactive prefix arg prompts for a filename.

(fn &optional PROMPT)

12.10.1.7 emacspeak-2048-pop-state

`emacspeak-2048-pop-state` [Command]

Reset state from stack.

12.10.1.8 emacspeak-2048-prune-stack

`emacspeak-2048-prune-stack` (*drop*) [Command]

Prune game stack to specified length.

(fn DROP)

12.10.1.9 emacspeak-2048-push-state

`emacspeak-2048-push-state` [Command]

Push current game state on stack.

12.10.1.10 emacspeak-2048-randomize-game

`emacspeak-2048-randomize-game` (**&optional** *count*) [Command]

Puts game in a randomized new state.

(fn &optional COUNT)

12.10.1.11 emacspeak-2048-score

`emacspeak-2048-score` [Command]

Show total on board.

12.10.1.12 emacspeak-2048-speak-board

`emacspeak-2048-speak-board` [Command]

Speak board.

12.10.1.13 emacspeak-2048-speak-transposed-board

`emacspeak-2048-speak-transposed-board` [Command]
Speak board column-wise.

12.11 emacspeak-abc-mode

ABC-MODE == Specialized mode for editing ABC Music notation. See `http://www.lesession.co.uk/abc/abc_notation.htm` for details. This package speech-enables abc-mode.

12.12 emacspeak-actions

Define mode-specific actions. Actions are defined by adding them to hook emacspeak-<mode-name>-actions-hook

12.13 emacspeak-add-log

Speech-enable Changelog mode

12.13.1 emacspeak-add-log Options

`User Option` *voice-change-log-acknowledgment* [Variable]
Personality used for change-log-acknowledgment This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

`User Option` *voice-change-log-conditionals* [Variable]
Personality used for change-log-conditionals This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

`User Option` *voice-change-log-date* [Variable]
Personality used for change-log-date This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-change-log-email* [Variable]
Personality used for change-log-email This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-change-log-file* [Variable]
Personality used for change-log-file This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-change-log-function* [Variable]
Personality used for change-log-function This personality uses voice-bolden-extra whose effect can be changed globally by customizing voice-bolden-extra-settings.

`User Option` *voice-change-log-list* [Variable]
Personality used for change-log-list This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-change-log-name* [Variable]
Personality used for change-log-name This personality uses voice-lighten-extra whose effect can be changed globally by customizing voice-lighten-extra-settings.

12.14 emacspeak-advice

This module defines the advice forms for making the core of Emacs speak Advice forms that are specific to Emacs subsystems do not belong here! I violate this at present by advising completion. Note that we needed to advice a lot more for Emacs 19 and Emacs 20 than we do for Emacs 21 and Emacs 22. As of August 2007, this file is being purged of advice forms not needed in Emacs 22.

12.14.1 Emacspeak-Advice Commands

12.14.1.1 emacspeak-eldoc-speak-doc

`emacspeak-eldoc-speak-doc` [Command]

C-, ,

C-x @ a ,

Speak Eldoc documentation if available.

12.14.2 emacspeak-advice Options

`User Option` *emacspeak-speak-errors* [Variable]

Specifies if error messages are cued.

`User Option` *emacspeak-speak-tooltips* [Variable]

Enable to get tooltips spoken.

`User Option` *voice-completions-annotations* [Variable]

Personality used for completions-annotations This personality uses voice-annotate whose effect can be changed globally by customizing voice-annotate-settings.

`User Option` *voice-completions-common-part* [Variable]

Personality used for completions-common-part This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-completions-first-difference* [Variable]

Personality used for completions-first-difference This personality uses voice-brighten whose effect can be changed globally by customizing voice-brighten-settings.

`User Option` *voice-eldoc-highlight-function-argument* [Variable]

Personality used for eldoc-highlight-function-argument This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-log-edit-header* [Variable]

Personality used for log-edit-header This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-log-edit-summary* [Variable]

Personality used for log-edit-summary This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-log-edit-unknown-header* [Variable]

Personality used for log-edit-unknown-header This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-minibuffer-prompt* [Variable]

Personality used for minibuffer-prompt This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

12.15 emacspeak-amark

Structure emacspeak-amark holds a bookmark into an mp3 file path: fully qualified pathname to file being marked name: Bookmark tag Position: time offset from start

This library will be used from emacspeak-m-player, emacspeak-mplayer and friends to set and jump to bookmarks.

12.15.1 Emacspeak-Amark Commands

12.15.1.1 emacspeak-amark-add

`emacspeak-amark-add` (*path name position*) [Command]

Add an AMark to the buffer local list of AMarks. AMarks are
bookmarks in audio content. If there is an existing amark of the
given name, it is updated with path and position.

(fn PATH NAME POSITION)

12.15.1.2 emacspeak-amark-find

`emacspeak-amark-find` (*name*) [Command]

Return matching AMark if found in buffer-local AMark list.

(fn NAME)

12.15.1.3 emacspeak-amark-save

`emacspeak-amark-save` [Command]

Save buffer-local AMarks in directory of currently playing resource.

12.16 emacspeak-analog

Speech-enables package analog –convenient log analyzer

12.16.1 Emacspeak-Analog Commands

12.16.1.1 emacspeak-analog-backward-field-or-char

`emacspeak-analog-backward-field-or-char` [Command]

Move back to next field if field specification is available.
Otherwise move to previous char.
Speak field or char moved to.

12.16.1.2 emacspeak-analog-forward-field-or-char

`emacspeak-analog-forward-field-or-char` [Command]

Move forward to next field if field specification is available.
Otherwise move to next char.
Speak field or char moved to.

12.16.1.3 emacspeak-analog-next-line

`emacspeak-analog-next-line` [Command]

Move down and speak current field.

12.16.1.4 emacspeak-analog-previous-line

`emacspeak-analog-previous-line` [Command]

Move up and speak current field.

12.16.1.5 emacspeak-analog-speak-this-field

`emacspeak-analog-speak-this-field` [Command]

Speak current field.

12.17 emacspeak-ansi-color

Module ansi-color (bundled with Emacs 21) handles ansi escape sequences and turns them into appropriate faces. This is useful in things like shell buffers. This module maps ansi codes to the appropriate voices.

12.18 emacspeak-apt-sources

This module speech-enables apt-sources.el that is included in the debian-el package and provides a major mode for editing APT's sources.list file.

12.19 emacspeak-arc

Auditory interface to archive mode This lets Emacs manipulate package files such as .zip and .jar files.

12.19.1 Emacspeak-Arc Commands

12.19.1.1 emacspeak-arc-speak-file-modification-time

`emacspeak-arc-speak-file-modification-time` [Command]

Speak modification time of the file on current line

12.19.1.2 emacspeak-arc-speak-file-name

`emacspeak-arc-speak-file-name` [Command]

Speak the name of the file on current line

12.19.1.3 emacspeak-arc-speak-file-permissions

`emacspeak-arc-speak-file-permissions` [Command]

Speak permissions of file current entry

12.19.1.4 emacspeak-arc-speak-file-size

`emacspeak-arc-speak-file-size` [Command]

Speak the size of the file on current line

12.20 emacspeak-auctex

Speech-enables the AucTeX package. AucTeX, now available from ELPA, has been my authoring environment of choice for writing LaTeX since 1991.

12.20.1 Emacspeak-Auctex Commands

12.20.1.1 emacspeak-auctex-comma-at-end-of-word

`emacspeak-auctex-comma-at-end-of-word` [Command]

Move to the end of current word and add a comma.

12.20.1.2 emacspeak-auctex-end-of-word

`emacspeak-auctex-end-of-word` (*arg*) [Command]

move to end of word

(fn ARG)

12.20.1.3 emacspeak-auctex-lacheck-buffer-file

`emacspeak-auctex-lacheck-buffer-file` [Command]

Run Lacheck on current buffer.

12.20.1.4 emacspeak-auctex-tex-tie-current-word

`emacspeak-auctex-tex-tie-current-word` (*n*) [Command]

Tie the next n words.

(fn N)

12.20.2 emacspeak-auctex Options

`User Option` *voice-latex-bold-personality* [Variable]

Personality used for font-latex-bold-face This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-latex-doctex-documentation-personality* [Variable]

Personality used for font-latex-doctex-documentation-face This personality uses voice-monotone-medium whose effect can be changed globally by customizing voice-monotone-medium-settings.

`User Option` *voice-latex-doctex-preprocessor-personality* [Variable]
Personality used for font-latex-doctex-preprocessor-face This personality uses voice-brighten-medium whose effect can be changed globally by customizing voice-brighten-medium-settings.

`User Option` *voice-latex-italic-personality* [Variable]
Personality used for font-latex-italic-face This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

`User Option` *voice-latex-math-personality* [Variable]
Personality used for font-latex-math-face This personality uses voice-brighten-extra whose effect can be changed globally by customizing voice-brighten-extra-settings.

`User Option` *voice-latex-sedate-personality* [Variable]
Personality used for font-latex-sedate-face This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

`User Option` *voice-latex-string-personality* [Variable]
Personality used for font-latex-string-face This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-latex-subscript-personality* [Variable]
Personality used for font-latex-subscript-face This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

`User Option` *voice-latex-superscript-personality* [Variable]
Personality used for font-latex-superscript-face This personality uses voice-brighten whose effect can be changed globally by customizing voice-brighten-settings.

`User Option` *voice-latex-verbatim-personality* [Variable]
Personality used for font-latex-verbatim-face This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-latex-warning-personality* [Variable]
Personality used for font-latex-warning-face This personality uses voice-bolden-and-animate whose effect can be changed globally by customizing voice-bolden-and-animate-settings.

12.21 emacspeak-bbc

BBC developer API (backstage) is now history. that implementation is in obsolete/emacspeak-bbc-backstage.el This module contains a light-weight client.

12.21.1 Emacspeak-Bbc Commands

12.21.1.1 emacspeak-bbc-get-iplayer-stream-pid

`emacspeak-bbc-get-iplayer-stream-pid` (*pid*) [Command]
Stream using get_iplayer.

(fn PID)

12.21.1.2 emacspeak-bbc-get-iplayer-stream-url

`emacspeak-bbc-get-iplayer-stream-url` (*url*) [Command]

Stream using get_iplayer.

(fn URL)

12.21.1.3 emacspeak-bbc-schedule

`emacspeak-bbc-schedule` [Command]

Browse BBC Schedule from get_iplayer radio cache

12.22 emacspeak-bbdb

Speech-enables BBDB. I have used BBDB to manage email address and contact information since 1991.

12.22.1 emacspeak-bbdb Options

`User Option` *voice-bbdb-field-name* [Variable]

Personality used for bbdb-field-name This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-bbdb-name* [Variable]

Personality used for bbdb-name This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-bbdb-organization* [Variable]

Personality used for bbdb-organization This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

12.23 emacspeak-bibtex

Speech extensions for bibtex mode.

12.24 emacspeak-bookmark

Speech enable bookmarks

12.25 emacspeak-bookshare

BOOKSHARE == http://www.bookshare.org provides book access to print-disabled users. It provides a simple Web API http://developer.bookshare.org This module implements an Emacspeak Bookshare client.

12.25.1 requirements

- You need to get your own API key
- You need Emacs built with libxml2 support

12.25.2 Usage

The main entry point is command `emacspeak-bookshare` bound to *C-e C-b*. This creates a special *Bookshare Interaction* buffer that is placed in *emacspeak-bookshare-mode*. Se the help for that mode on detailed usage instructions and key-bindings.

12.25.3 Sample Interaction

Assuming you have correctly setup your API key:

- Customize group `emacspeak-bookshare` by pressing *C-h G*.
- Press *C-e C-b* to open or switch to the Bookshare buffer.
- Perform a search `a` or *t* for author or title search.
- You will be prompted for your Bookshare password if this is the first time.
- The password will be saved to your configured `auth-source` — usually `~/.authinfo.gpg`. You can also use `password-store[.]`
- The results of the search appear in the Bookshare buffer. Audio formatting and auditory icons convey if a result is already available locally.
- If not available locally, press *D* to download the content.
- Press *U* to unpack the downloaded content.
- Press *e* to display the entire book.
- Press *c* to display the table of contents.
- Now, use all of EWW See Section 12.74 [emacspeak-eww], page 124, extensions and profit!

12.25.4 Emacspeak-Bookshare Commands

12.25.4.1 emacspeak-bookshare

`emacspeak-bookshare` [Command]

C-e C-b

<fn> C-b

Bookshare Interaction.

12.25.4.2 emacspeak-bookshare-action

`emacspeak-bookshare-action` [Command]

Call action specified by invoking key.

12.25.4.3 emacspeak-bookshare-author-search

`emacspeak-bookshare-author-search` (*query* **&optional** *category*) [Command]

Perform a Bookshare author search.
Interactive prefix arg filters search by category.

(fn QUERY &optional CATEGORY)

12.25.4.4 emacspeak-bookshare-browse

`emacspeak-bookshare-browse` [Command]

Browse Bookshare.

12.25.4.5 emacspeak-bookshare-browse-latest

`emacspeak-bookshare-browse-latest` [Command]

Return latest books.

12.25.4.6 emacspeak-bookshare-browse-popular

`emacspeak-bookshare-browse-popular` (**&optional** *category*) [Command]

Browse popular books.
Optional interactive prefix arg prompts for a category to use as a filter.

(fn &optional CATEGORY)

12.25.4.7 emacspeak-bookshare-debugInfo-handler

`emacspeak-bookshare-debugInfo-handler` (**&rest** *arguments*) [Command]

Do nothing and return nil.
This function accepts any number of ARGUMENTS, but ignores them.

(fn &rest ARGUMENTS)

12.25.4.8 emacspeak-bookshare-download-audio

`emacspeak-bookshare-download-audio` (*id target*) [Command]

Download audio format of specified book to target location.

(fn ID TARGET)

12.25.4.9 emacspeak-bookshare-download-audio-at-point

`emacspeak-bookshare-download-audio-at-point` [Command]

Download audio version of book under point.
Target location is generated from author and title.

12.25.4.10 emacspeak-bookshare-download-brf

`emacspeak-bookshare-download-brf` (*id target*) [Command]

Download Daisy format of specified book to target location.

(fn ID TARGET)

12.25.4.11 emacspeak-bookshare-download-brf-at-point

`emacspeak-bookshare-download-brf-at-point` [Command]

Download Braille version of book under point.
Target location is generated from author and title.

12.25.4.12 emacspeak-bookshare-download-daisy

`emacspeak-bookshare-download-daisy` (*id target*) [Command]

Download Daisy format of specified book to target location.

(fn ID TARGET)

12.25.4.13 emacspeak-bookshare-download-daisy-at-point

`emacspeak-bookshare-download-daisy-at-point` [Command]

Download Daisy version of book under point.
Target location is generated from author and title.

12.25.4.14 emacspeak-bookshare-download-epub-3

`emacspeak-bookshare-download-epub-3` (*id target*) [Command]

Download epub-3 format of specified book to target location.

(fn ID TARGET)

12.25.4.15 emacspeak-bookshare-download-epub-3-at-point

`emacspeak-bookshare-download-epub-3-at-point` [Command]

Download epub-3 version of book under point.
Target location is generated from author and title.

12.25.4.16 emacspeak-bookshare-download-internal

`emacspeak-bookshare-download-internal` (*url target*) [Command]

Download content to target location.

(fn URL TARGET)

12.25.4.17 emacspeak-bookshare-eww

`emacspeak-bookshare-eww` (*directory*) [Command]

Render complete book using EWW

(fn DIRECTORY)

12.25.4.18 emacspeak-bookshare-expand-at-point

`emacspeak-bookshare-expand-at-point` [Command]

Expand entry at point by retrieving metadata.
Once retrieved, memoize to avoid multiple retrievals.

12.25.4.19 emacspeak-bookshare-extract-and-view

`emacspeak-bookshare-extract-and-view` (*url*) [Command]

Extract content referred to by link under point, and render via the browser.

(fn URL)

12.25.4.20 emacspeak-bookshare-extract-xml

`emacspeak-bookshare-extract-xml` (*url*) [Command]

Extract content referred to by link under point, and return an XML buffer.

(fn URL)

12.25.4.21 emacspeak-bookshare-flush-lines

`emacspeak-bookshare-flush-lines` (*regexp*) [Command]

Flush lines matching regexp in Bookshare buffer.

(fn REGEXP)

12.25.4.22 emacspeak-bookshare-fulltext

`emacspeak-bookshare-fulltext` (*directory*) [Command]

Display fulltext contents of book in specified directory.
Useful for fulltext search in a book.

(fn DIRECTORY)

12.25.4.23 emacspeak-bookshare-fulltext-search

`emacspeak-bookshare-fulltext-search` (*query*) [Command]

Perform a Bookshare fulltext search.

(fn QUERY)

12.25.4.24 emacspeak-bookshare-get-more-results

`emacspeak-bookshare-get-more-results` [Command]

Get next page of results for last query.

12.25.4.25 emacspeak-bookshare-id-search

`emacspeak-bookshare-id-search` (*query*) [Command]

Perform a Bookshare id search.

(fn QUERY)

12.25.4.26 emacspeak-bookshare-isbn-search

`emacspeak-bookshare-isbn-search` (*query*) [Command]

Perform a Bookshare isbn search.

(fn QUERY)

12.25.4.27 emacspeak-bookshare-list-preferences

`emacspeak-bookshare-list-preferences` [Command]

Return preference list.

12.25.4.28 emacspeak-bookshare-mode

`emacspeak-bookshare-mode` [Command]

A Bookshare front-end for the Emacspeak desktop.

The Emacspeak Bookshare front-end is launched by command
emacspeak-bookshare bound to C-e C-b

This command switches to a special buffer that has Bookshare
commands bounds to single keystrokes– see the key-binding list at
the end of this description. Use Emacs online help facility to
look up help on these commands.

emacspeak-bookshare-mode provides the necessary functionality to
Search and download Bookshare material, Manage a local library of
downloaded Bookshare content, And commands to easily read newer
Daisy books from Bookshare.

Here is a list of all emacspeak Bookshare commands with their key-bindings:
a Author Search
A Author/Title Search
t Title Search
s Full Text Search
d Date Search
b Browse

key binding
— ——-

ESC Prefix Command
SPC emacspeak-bookshare-expand-at-point
+ emacspeak-bookshare-action
/ emacspeak-bookshare-action
3 emacspeak-bookshare-download-epub-3-at-point
A emacspeak-bookshare-download-audio-at-point
B emacspeak-bookshare-download-brf-at-point
C emacspeak-bookshare-fulltext
D emacspeak-bookshare-download-daisy-at-point
E emacspeak-bookshare-eww
I emacspeak-bookshare-action
P emacspeak-bookshare-action
S emacspeak-bookshare-action
U emacspeak-bookshare-unpack-at-point
V emacspeak-bookshare-view-at-point
[backward-page
] forward-page
a emacspeak-bookshare-action

b emacspeak-bookshare-browse
c emacspeak-bookshare-toc-at-point
d emacspeak-bookshare-action
e emacspeak-bookshare-eww
f emacspeak-bookshare-flush-lines
i emacspeak-bookshare-action
j next-line
k previous-line
l .. m emacspeak-bookshare-action
p emacspeak-bookshare-action
q bury-buffer
s .. t emacspeak-bookshare-action
v emacspeak-bookshare-view

M-n emacspeak-bookshare-next-result
M-p emacspeak-bookshare-previous-result

In addition to any hooks its parent mode 'special-mode' might have run,
this mode runs the hook 'emacspeak-bookshare-mode-hook', as the final or penultimate step
during initialization.

12.25.4.29 emacspeak-bookshare-next-result

`emacspeak-bookshare-next-result` [Command]

Move to next result.

12.25.4.30 emacspeak-bookshare-periodical-list

`emacspeak-bookshare-periodical-list` [Command]

Return list of periodicals.

12.25.4.31 emacspeak-bookshare-previous-result

`emacspeak-bookshare-previous-result` [Command]

Move to previous result.

12.25.4.32 emacspeak-bookshare-set-preference

`emacspeak-bookshare-set-preference` (*preference-id value*) [Command]

Set preference preference-id to value.

(fn PREFERENCE-ID VALUE)

12.25.4.33 emacspeak-bookshare-since-search

`emacspeak-bookshare-since-search` (*query* **&optional** *category*) [Command]

Perform a Bookshare date search.
Optional interactive prefix arg filters by category.

(fn QUERY &optional CATEGORY)

12.25.4.34 emacspeak-bookshare-title-search

`emacspeak-bookshare-title-search` (*query* **&optional** *category*) [Command]

Perform a Bookshare title search.
Interactive prefix arg filters search by category.

(fn QUERY &optional CATEGORY)

12.25.4.35 emacspeak-bookshare-toc

`emacspeak-bookshare-toc` (*directory*) [Command]

View TOC for book in specified directory.

(fn DIRECTORY)

12.25.4.36 emacspeak-bookshare-toc-at-point

`emacspeak-bookshare-toc-at-point` [Command]

View TOC for book at point.
Make sure it's downloaded and unpacked first.

12.25.4.37 emacspeak-bookshare-unpack-at-point

`emacspeak-bookshare-unpack-at-point` [Command]

Unpack downloaded content if necessary.

12.25.4.38 emacspeak-bookshare-url-executor

`emacspeak-bookshare-url-executor` (*url*) [Command]

Custom URL executor for use in Bookshare TOC.

(fn URL)

12.25.4.39 emacspeak-bookshare-version-handler

`emacspeak-bookshare-version-handler` (**&rest** *arguments*) [Command]

Do nothing and return nil.
This function accepts any number of ARGUMENTS, but ignores them.

(fn &rest ARGUMENTS)

12.25.4.40 emacspeak-bookshare-view

`emacspeak-bookshare-view` (*directory*) [Command]

View book in specified directory.

(fn DIRECTORY)

12.25.4.41 emacspeak-bookshare-view-at-point

`emacspeak-bookshare-view-at-point` [Command]

View book at point.

Make sure it's downloaded and unpacked first.

12.25.4.42 emacspeak-bookshare-view-page-range

`emacspeak-bookshare-view-page-range` (*url*) [Command]

Play pages in specified page range from URL.

(fn URL)

12.25.5 emacspeak-bookshare Options

`User Option` *emacspeak-bookshare-api-key* [Variable]

Web API key for this application. See http://developer.bookshare.org/docs for details on how to get an API key.

`User Option` *emacspeak-bookshare-directory* [Variable]

Customize this to the root of where books are organized.

12.26 emacspeak-browse-kill-ring

Browse the kill ring using browse-kill-ring.el - interactively insert items from kill-ring (by Colin Walters)

12.27 emacspeak-bs

speech-enable bs.el – an alternative to Emacs' default list-buffers

12.27.1 Emacspeak-Bs Commands

12.27.1.1 emacspeak-bs-speak-buffer-line

`emacspeak-bs-speak-buffer-line` [Command]

Speak information about this buffer

12.28 emacspeak-buff-menu

Speech-enable buffer-menus.

12.28.1 Emacspeak-Buff-Menu Commands

12.28.1.1 emacspeak-list-buffers-next-line

`emacspeak-list-buffers-next-line` (*count*) [Command]

Speech enabled buffer menu navigation

(fn COUNT)

12.28.1.2 emacspeak-list-buffers-previous-line

`emacspeak-list-buffers-previous-line` (*count*) [Command]

Speech enabled buffer menu navigation

(fn COUNT)

12.28.1.3 emacspeak-list-buffers-speak-buffer-line

`emacspeak-list-buffers-speak-buffer-line` [Command]

Speak information about this buffer

12.28.1.4 emacspeak-list-buffers-speak-buffer-name

`emacspeak-list-buffers-speak-buffer-name` [Command]

Speak the name of the buffer on this line

12.28.2 emacspeak-buff-menu Options

`User Option` *voice-buffer-menu-buffer* [Variable]

Personality used for buffer-menu-buffer This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

12.29 emacspeak-c

Make C and C++ mode more emacspeak friendly Works with both boring c-mode and the excellent cc-mode

12.29.1 Emacspeak-C Commands

12.29.1.1 emacspeak-c-speak-semantics

`emacspeak-c-speak-semantics` [Command]

Speak the C semantics of this line.

12.30 emacspeak-calc

This module extends the Emacs Calculator. Extensions are minimal. We force a calc-load-everything, And use an after advice on this function To fix all of calc's interactive functions

12.31 emacspeak-calculator

Speech enable desktop calculator

12.32 emacspeak-calendar

This module speech enables the Emacs Calendar. Speech enabling is not the same as speaking the screen: This is an excellent example of the advantages of speech-enabled interaction.

12.32.1 Emacspeak-Calendar Commands

12.32.1.1 emacspeak-appt-repeat-announcement

`emacspeak-appt-repeat-announcement` [Command]

C-e A

<fn> A

Speaks the most recently displayed appointment message if any.

12.32.1.2 emacspeak-calendar-setup-sunrise-sunset

`emacspeak-calendar-setup-sunrise-sunset` [Command]

Set up geo-coordinates using Google Maps reverse geocoding.
To use, configure variable gmaps-my-address via M-x customize-variable.

12.32.1.3 emacspeak-calendar-speak-date

`emacspeak-calendar-speak-date` [Command]

Speak the date under point when called in Calendar Mode.

12.32.1.4 emacspeak-calendar-sunrise-sunset

`emacspeak-calendar-sunrise-sunset` (*address* **&optional** *arg*) [Command]

Display sunrise/sunset for specified address.

(fn ADDRESS &optional ARG)

12.32.2 emacspeak-calendar Options

`User Option` *voice-calendar-today* [Variable]

Personality used for calendar-today This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-diary* [Variable]

Personality used for diary This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-holiday* [Variable]

Personality used for holiday This personality uses voice-brighten-extra whose effect can be changed globally by customizing voice-brighten-extra-settings.

12.33 emacspeak-chess

The Emacs Chess package provides a rich environment for playing and exploring Chess Games. That package comes with a light-weight module that announces moves.

This module aims do do much more, including:

- Navigate the board along various axies with audio-formatted output.
- Browse games via rich audio-formatted output.

- Speech-enable all interactive commands provided by the Chess package.
- Enable various means of exploring the state of game, perhaps with a view to being able to spot patterns from listening to the output.

12.33.1 Navigating And Examining The Board

The board can be navigated along the 8 compass directions. Arrow keys move to the appropriate squares on the board. `/` and `\` move down the diagonals. `[` and `]` move up the respective diagonals.

- Move North: `emacspeak-chess-north` bound to *`up`*.
- Move South: `emacspeak-chess-south` bound to *`down`*.
- Move West: `emacspeak-chess-west` bound to *`left`*.
- Move East: `emacspeak-chess-east` bound to *`right`*.

You can also move along the diagonals:

- Move Northwest: `emacspeak-chess-northwest` bound to `[`.
- Move Northeast: `emacspeak-chess-northeast` bound to `]`.
- Move Southwest: `emacspeak-chess-southwest` bound to `/`.
- Move Southeast: `emacspeak-chess-southeast` bound to `\`.

You can also jump to a given board position by:

- Jump: `emacspeak-chess-jump` bound to `j`.
- Target of last move: `emacspeak-chess-goto-target` bound to `t`.
- Look: `emacspeak-chess-speak-that-square` bound to `l`.
- Review current square: `;`.
- Review Board: `emacspeak-chess-speak-board` bound to `z`. Useful in end-state.
- Locate pieces: `emacspeak-chess-speak-piece-squares` bound to `s`. Specify piece as a single char — `w` speaks all white pieces, `l` speaks all black pieces, use SAN notation char for a specific piece. Use `a` to hear entire board, this last is most useful when the game is an end-state.
- Discover who targets a square: `emacspeak-chess-speak-who-targets` bound to `w`. Argument `piece` is similar to the previously listed command.

You can obtain views of the board along the rows and diagonals, as well as a *knight's perspective* :

- Viewers: `v` followed by the directional navigation keys speaks the squares in that direction from the current square. `vn` speaks the squares from a knight's perspective, i.e. the squares the Knight would see from the current position. similarly, `vk` speaks the non-empty squares seen by the King from the current position. `v` followed by a rank-or-file char speaks that complete rank or file.

12.33.2 Examining Games

Package *Chess* allows one to browse through a game using `,` and `.` — `chess-display-move-backward` and `chess-display-move-forward`. Emacspeak speech-enables these commands to speak the move that led to the currently displayed state of the game. Finally, `m` speaks the *current move* being displayed.

12.33.3 Emacspeak-Chess Commands

12.33.3.1 emacspeak-chess-east

`emacspeak-chess-east` [Command]

Move east one step.

12.33.3.2 emacspeak-chess-goto-target

`emacspeak-chess-goto-target` [Command]

Jump to the most recent target square.

12.33.3.3 emacspeak-chess-jump

`emacspeak-chess-jump` (*coord*) [Command]

Jump to square specified as coord.

(fn COORD)

12.33.3.4 emacspeak-chess-look-east

`emacspeak-chess-look-east` [Command]

Look east

12.33.3.5 emacspeak-chess-look-king

`emacspeak-chess-look-king` [Command]

Look along non-empty squares reachable by the king from current position.

12.33.3.6 emacspeak-chess-look-knight

`emacspeak-chess-look-knight` [Command]

Look along non-empty squares reachable by a knight from current position.

12.33.3.7 emacspeak-chess-look-north

`emacspeak-chess-look-north` [Command]

Look north

12.33.3.8 emacspeak-chess-look-northeast

`emacspeak-chess-look-northeast` [Command]

Look northeast

12.33.3.9 emacspeak-chess-look-northwest

`emacspeak-chess-look-northwest` [Command]

Look northwest

12.33.3.10 emacspeak-chess-look-south

`emacspeak-chess-look-south` [Command]

Look south

12.33.3.11 emacspeak-chess-look-southeast

`emacspeak-chess-look-southeast` [Command]

Look southeast

12.33.3.12 emacspeak-chess-look-southwest

`emacspeak-chess-look-southwest` [Command]

Look southwest

12.33.3.13 emacspeak-chess-look-west

`emacspeak-chess-look-west` [Command]

Look west

12.33.3.14 emacspeak-chess-north

`emacspeak-chess-north` [Command]

Move north one step.

12.33.3.15 emacspeak-chess-northeast

`emacspeak-chess-northeast` [Command]

Move northeast one step.

12.33.3.16 emacspeak-chess-northwest

`emacspeak-chess-northwest` [Command]

Move northwest one step.

12.33.3.17 emacspeak-chess-south

`emacspeak-chess-south` [Command]

Move south one step.

12.33.3.18 emacspeak-chess-southeast

`emacspeak-chess-southeast` [Command]

Move southeast one step.

12.33.3.19 emacspeak-chess-southwest

`emacspeak-chess-southwest` [Command]

Move southwest one step.

12.33.3.20 emacspeak-chess-speak-board

`emacspeak-chess-speak-board` [Command]

Speak complete board. Only useful in end-states.

12.33.3.21 emacspeak-chess-speak-piece-squares

`emacspeak-chess-speak-piece-squares` (*piece*) [Command]

Prompt for a piece (single char) and speak its locations on the board. Piece is specified as a char using SAN notation. Use ‘w’ for all whites pieces, ‘l’ for all black pieces, and ‘a’ for entire board..

(fn PIECE)

12.33.3.22 emacspeak-chess-speak-that-square

`emacspeak-chess-speak-that-square` (*coord*) [Command]

Speak square at specified coord.

(fn COORD)

12.33.3.23 emacspeak-chess-speak-this-move

`emacspeak-chess-speak-this-move` [Command]

Speak move at current display index.

12.33.3.24 emacspeak-chess-speak-this-square

`emacspeak-chess-speak-this-square` [Command]

Speak square under point.

12.33.3.25 emacspeak-chess-speak-who-targets

`emacspeak-chess-speak-who-targets` (*piece*) [Command]

Speak description of squares that can target current square. Argument ‘piece’ specifies piece-or-color as in command emacspeak-chess-speak-piece-squares

(fn PIECE)

12.33.3.26 emacspeak-chess-view-rank-or-file

`emacspeak-chess-view-rank-or-file` [Command]

View a complete rank or file from white’s perspective.

12.33.3.27 emacspeak-chess-west

`emacspeak-chess-west` [Command]

Move west one step.

12.34 emacspeak-cider

Speech-Enable CIDER — Clojure IDE

12.34.1 emacspeak-cider Options

`User Option` *voice-cider-repl-stderr-facevoice-animate* [Variable]

Personality used for cider-repl-stderr-facevoice-animate This personality uses nil whose effect can be changed globally by customizing nil-settings.

12.35 emacspeak-ciel

Package ciel provides vim's "copy inside" and "clear inside" commands. Emacspeak binds these commands to <Super i> and <Super o>. This module speech-enables ciel.

12.36 emacspeak-clojure

CLOJURE-mode: Specialized mode for Clojure programming.

12.37 emacspeak-cmuscheme

speech-enable scheme support

12.38 emacspeak-comint

comint == command interaction. Advice comint and friends to speak.

12.38.1 Emacspeak-Comint Commands

12.38.1.1 emacspeak-completion-pick-completion

`emacspeak-completion-pick-completion` [Command]

Pick completion and return safely where we came from.

12.38.1.2 emacspeak-toggle-comint-autospeak

`emacspeak-toggle-comint-autospeak` (**&optional** *prefix*) [Command]

Toggle state of Emacspeak comint autospeak.
When turned on, comint output is automatically spoken. Turn this on if
you want your shell to speak its results. Interactive
PREFIX arg means toggle the global default value, and then
set the current local value to the result.

(fn &optional PREFIX)

12.38.1.3 emacspeak-toggle-comint-output-monitor

`emacspeak-toggle-comint-output-monitor` (**&optional** *prefix*) [Command]

`C-e o`

`<fn> o`

Toggle state of Emacspeak comint monitor.
When turned on, comint output is automatically spoken. Turn this on if
you want your shell to speak its results. Interactive

PREFIX arg means toggle the global default value, and then
set the current local value to the result.

12.38.1.4 emacspeak-toggle-inaudible-or-comint-autospeak

`emacspeak-toggle-inaudible-or-comint-autospeak` [Command]

`C-e C-q`

`<fn> C-q`

Toggle comint-autospeak when in a comint or vterm buffer.
Otherwise call voice-setup-toggle-silence-personality which toggles the
personality under point.

12.38.2 emacspeak-comint Options

`User Option` *emacspeak-comint-autospeak* [Variable]

Says if comint output is automatically spoken. You can use ‘emacspeak-toggle-comint-autospeak‘ bound to M-x emacspeak-toggle-comint-autospeak to toggle this setting.

`User Option` *voice-comint-highlight-input* [Variable]

Personality used for comint-highlight-input This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-comint-highlight-prompt* [Variable]

Personality used for comint-highlight-prompt This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

12.39 emacspeak-company

COMPANY -mode: Complete Anything Support for emacs.

This module provides an Emacspeak Company Front-end, And advises the needed interactive commands in Company. It adds an emacspeak-specific front-end `emacspeak-company-frontend` to the value of company-frontends. Note that *company-frontends* is a user-customizable option and ends up getting saved by emacs along with other custom settings. Function `emacspeak-company-frontend` handles providing spoken feedback, and leaves it to other frontends on *company-frontends* to generate their own feedback.

12.39.1 emacspeak-company Options

`User Option` *voice-company-echo* [Variable]

Personality used for company-echo This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-company-echo-common* [Variable]

Personality used for company-echo-common This personality uses voice-bolden-medium whose effect can be changed globally by customizing voice-bolden-medium-settings.

`User Option` *voice-company-preview* [Variable]
: Personality used for company-preview This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-company-preview-common* [Variable]
: Personality used for company-preview-common This personality uses voice-lighten-medium whose effect can be changed globally by customizing voice-lighten-medium-settings.

`User Option` *voice-company-preview-search* [Variable]
: Personality used for company-preview-search This personality uses voice-brighten whose effect can be changed globally by customizing voice-brighten-settings.

`User Option` *voice-company-template-field* [Variable]
: Personality used for company-template-field This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

12.40 emacspeak-compile

This module makes compiling code from inside Emacs speech friendly. It is an example of how a little amount of code can make Emacspeak even better.

12.40.1 Emacspeak-Compile Commands

12.40.1.1 emacspeak-compilation-speak-error

`emacspeak-compilation-speak-error` [Command]
: Speech feedback about the compilation error.

12.40.2 emacspeak-compile Options

`User Option` *voice-compilation-column-number* [Variable]
: Personality used for compilation-column-number This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

`User Option` *voice-compilation-error* [Variable]
: Personality used for compilation-error This personality uses voice-animate-extra whose effect can be changed globally by customizing voice-animate-extra-settings.

`User Option` *voice-compilation-info* [Variable]
: Personality used for compilation-info This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-compilation-line-number* [Variable]
: Personality used for compilation-line-number This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

`User Option` *voice-compilation-mode-line-exit* [Variable]
: Personality used for compilation-mode-line-exit This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

`User Option` *voice-compilation-mode-line-fail* [Variable]
Personality used for compilation-mode-line-fail This personality uses voice-brighten whose effect can be changed globally by customizing voice-brighten-settings.

`User Option` *voice-compilation-mode-line-run* [Variable]
Personality used for compilation-mode-line-run This personality uses voice-annotate whose effect can be changed globally by customizing voice-annotate-settings.

`User Option` *voice-compilation-warning* [Variable]
Personality used for compilation-warning This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

12.41 emacspeak-cperl

Provide additional advice to CPerl mode

12.42 emacspeak-custom

Advise custom to speak. most of the work is actually done by emacspeak-widget.el which speech-enables the widget libraries.

12.42.1 Emacspeak-Custom Commands

12.42.1.1 emacspeak-custom-goto-group

`emacspeak-custom-goto-group` [Command]
Jump to custom group when in a customization buffer.

12.42.1.2 emacspeak-custom-goto-toolbar

`emacspeak-custom-goto-toolbar` [Command]
Jump to custom toolbar when in a customization buffer.

12.42.2 emacspeak-custom Options

`User Option` *voice-custom-button* [Variable]
Personality used for custom-button This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-custom-button-mouse* [Variable]
Personality used for custom-button-mouse This personality uses voice-bolden-medium whose effect can be changed globally by customizing voice-bolden-medium-settings.

`User Option` *voice-custom-button-pressed* [Variable]
Personality used for custom-button-pressed This personality uses voice-bolden-extra whose effect can be changed globally by customizing voice-bolden-extra-settings.

`User Option` *voice-custom-button-pressed-unraised* [Variable]
Personality used for custom-button-pressed-unraised This personality uses voice-bolden-extra whose effect can be changed globally by customizing voice-bolden-extra-settings.

User Option *voice-custom-button-unraised* [Variable]
Personality used for custom-button-unraised This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

User Option *voice-custom-changed* [Variable]
Personality used for custom-changed This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

User Option *voice-custom-comment* [Variable]
Personality used for custom-comment This personality uses voice-monotone-medium whose effect can be changed globally by customizing voice-monotone-medium-settings.

User Option *voice-custom-comment-tag* [Variable]
Personality used for custom-comment-tag This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

User Option *voice-custom-documentation* [Variable]
Personality used for custom-documentation This personality uses voice-brighten-medium whose effect can be changed globally by customizing voice-brighten-medium-settings.

User Option *voice-custom-face-tag* [Variable]
Personality used for custom-face-tag This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

User Option *voice-custom-group-subtitle* [Variable]
Personality used for custom-group-subtitle This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

User Option *voice-custom-group-tag* [Variable]
Personality used for custom-group-tag This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

User Option *voice-custom-group-tag-1* [Variable]
Personality used for custom-group-tag-1 This personality uses voice-lighten-medium whose effect can be changed globally by customizing voice-lighten-medium-settings.

User Option *voice-custom-invalid* [Variable]
Personality used for custom-invalid This personality uses voice-animate-extra whose effect can be changed globally by customizing voice-animate-extra-settings.

User Option *voice-custom-link* [Variable]
Personality used for custom-link This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

User Option *voice-custom-modified* [Variable]
Personality used for custom-modified This personality uses voice-lighten-medium whose effect can be changed globally by customizing voice-lighten-medium-settings.

`User Option` *voice-custom-rogue* [Variable]

Personality used for custom-rogue This personality uses voice-bolden-and-animate whose effect can be changed globally by customizing voice-bolden-and-animate-settings.

`User Option` *voice-custom-saved* [Variable]

Personality used for custom-saved This personality uses voice-smoothen-extra whose effect can be changed globally by customizing voice-smoothen-extra-settings.

`User Option` *voice-custom-set* [Variable]

Personality used for custom-set This personality uses voice-smoothen-medium whose effect can be changed globally by customizing voice-smoothen-medium-settings.

`User Option` *voice-custom-state* [Variable]

Personality used for custom-state This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

`User Option` *voice-custom-themed* [Variable]

Personality used for custom-themed This personality uses voice-brighten whose effect can be changed globally by customizing voice-brighten-settings.

`User Option` *voice-custom-variable-button* [Variable]

Personality used for custom-variable-button This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

`User Option` *voice-custom-variable-obsolete* [Variable]

Personality used for custom-variable-obsolete This personality uses voice-smoothen-extra whose effect can be changed globally by customizing voice-smoothen-extra-settings.

`User Option` *voice-custom-variable-tag* [Variable]

Personality used for custom-variable-tag This personality uses voice-bolden-medium whose effect can be changed globally by customizing voice-bolden-medium-settings.

`User Option` *voice-custom-visibility* [Variable]

Personality used for custom-visibility This personality uses voice-annotate whose effect can be changed globally by customizing voice-annotate-settings.

12.43 emacspeak-dbus

Loading this module sets up Emacspeak to respond to DBus notifications. This module needs to be loaded explicitly from the user's init file after emacspeak has been started.

12.43.1 Overview

This module provides integration via DBus for the following:

- Respond to network coming up or going down — `(nm-enable)`.
- Respond to screen getting locked/unlocked by gnome-screen-saver — `(emacspeak-dbus-watch-screen-lock)`.
- Respond to laptop going to sleep or waking up — `(emacspeak-dbus-sleep-enable)`.

- Respond to insertion/ejection of removable storage — `(emacspeak-dbus-udisks-enable)`.
- Watch for power devices — `(emacspeak-dbus-upower-enable)`.
- An interactive command `emacspeak-dbus-lock-screen` bound to *C-, C-d* to lock the screen using DBus. Note: this key-binding is available only if this module is loaded.

Add calls to the desired functions from the above list to the emacs startup file after this module has been loaded. See relevant hooks for customizing behavior. Note that each of the sleep/wake-up, UDisks2 and network/up-down can be separately enabled/disabled, and the actions customized via appropriately named hook functions.

12.43.2 Emacspeak-Dbus Commands

12.43.2.1 emacspeak-dbus-lock-screen

`emacspeak-dbus-lock-screen` [Command]

Lock screen using DBus.

12.43.2.2 emacspeak-dbus-sleep-disable

`emacspeak-dbus-sleep-disable` [Command]

Disable integration with login1 daemon. Does nothing if
already disabled.

12.43.2.3 emacspeak-dbus-sleep-enable

`emacspeak-dbus-sleep-enable` [Command]

Enable integration with Login1. Does nothing if already enabled.

12.43.2.4 emacspeak-dbus-udisks-disable

`emacspeak-dbus-udisks-disable` [Command]

Disable integration with UDisks2 daemon. Does nothing if
already disabled.

12.43.2.5 emacspeak-dbus-udisks-enable

`emacspeak-dbus-udisks-enable` [Command]

Enable integration with UDisks2. Does nothing if already enabled.

12.43.2.6 emacspeak-dbus-upower-disable

`emacspeak-dbus-upower-disable` [Command]

Disable integration with UPower daemon. Does nothing if
already disabled.

12.43.2.7 emacspeak-dbus-upower-enable

`emacspeak-dbus-upower-enable` [Command]

Enable integration with UPower. Does nothing if already enabled.

12.43.2.8 emacspeak-screen-saver-mode

`emacspeak-screen-saver-mode` [Command]

A light-weight mode for the '*Emacspeak Screen Saver *' buffer.
This is a hidden buffer that is made current so we automatically
switch to a screen-saver soundscape.

In addition to any hooks its parent mode 'special-mode' might have run,
this mode runs the hook 'emacspeak-screen-saver-mode-hook', as the final or penultimate step
during initialization.

key binding
— ——-

12.44 emacspeak-deadgrep

DEADGREP == Front-end to ripgrep.

12.45 emacspeak-debugger

DEBUGGER == Emacs Interactive Debugger. Speech-enable the debugger by speech-enabling interactive commands.

12.46 emacspeak-desktop

advice desktop package

12.47 emacspeak-dictionary

Speech-enables emacs client for accessing dictionary server at dict.org:2628

12.48 emacspeak-diff-mode

DIFF-MODE support.

12.48.1 emacspeak-diff-mode Options

`User Option` *voice-diff-added* [Variable]

Personality used for diff-added This personality uses voice-brighten whose effect can be changed globally by customizing voice-brighten-settings.

`User Option` *voice-diff-changed* [Variable]

Personality used for diff-changed This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

`User Option` *voice-diff-context* [Variable]

Personality used for diff-context This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-diff-file-header* [Variable]

Personality used for diff-file-header This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-diff-function* [Variable]

Personality used for diff-function This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

`User Option` *voice-diff-header* [Variable]

Personality used for diff-header This personality uses voice-bolden-extra whose effect can be changed globally by customizing voice-bolden-extra-settings.

`User Option` *voice-diff-hunk-header* [Variable]

Personality used for diff-hunk-header This personality uses voice-bolden-medium whose effect can be changed globally by customizing voice-bolden-medium-settings.

`User Option` *voice-diff-index* [Variable]

Personality used for diff-index This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-diff-indicator-added* [Variable]

Personality used for diff-indicator-added This personality uses voice-annotate whose effect can be changed globally by customizing voice-annotate-settings.

`User Option` *voice-diff-indicator-changed* [Variable]

Personality used for diff-indicator-changed This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-diff-indicator-removed* [Variable]

Personality used for diff-indicator-removed This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

`User Option` *voice-diff-nonexistent* [Variable]

Personality used for diff-nonexistent This personality uses voice-lighten-extra whose effect can be changed globally by customizing voice-lighten-extra-settings.

`User Option` *voice-diff-refine-added* [Variable]

Personality used for diff-refine-added This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-diff-refine-changed* [Variable]

Personality used for diff-refine-changed This personality uses voice-brighten-medium whose effect can be changed globally by customizing voice-brighten-medium-settings.

`User Option` *voice-diff-refine-removed* [Variable]

Personality used for diff-refine-removed This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

`User Option` *voice-diff-removed* [Variable]

Personality used for diff-removed This personality uses voice-smoothen-extra whose effect can be changed globally by customizing voice-smoothen-extra-settings.

12.49 emacspeak-dired

This module speech enables dired. It reduces the amount of speech you hear: Typically you hear the file names as you move through the dired buffer Voicification is used to indicate directories, marked files etc.

12.49.1 Emacspeak-Dired Commands

12.49.1.1 emacspeak-dired-csv-open

`emacspeak-dired-csv-open` [Command]

Open CSV file on current dired line.

12.49.1.2 emacspeak-dired-downloads

`emacspeak-dired-downloads` [Command]

C-. d

C-' d

C-x @ s d

Open Downloads directory.

12.49.1.3 emacspeak-dired-epub-eww

`emacspeak-dired-epub-eww` [Command]

Open epub on current line in EWW

12.49.1.4 emacspeak-dired-eww-open

`emacspeak-dired-eww-open` [Command]

Open HTML file on current dired line.

12.49.1.5 emacspeak-dired-label-fields

`emacspeak-dired-label-fields` [Command]

Labels the fields of the listing in the dired buffer.
Currently is a no-op unless
unless 'dired-listing-switches' contains -l

12.49.1.6 emacspeak-dired-md-open

`emacspeak-dired-md-open` [Command]

Preview markdown file on current dired line.

12.49.1.7 emacspeak-dired-midi-play

`emacspeak-dired-midi-play` [Command]

Play midi file on current dired line.

12.49.1.8 emacspeak-dired-open-this-file

`emacspeak-dired-open-this-file` [Command]

Smart dired opener. Invokes appropriate Emacspeak handler on current file in DirEd.

12.49.1.9 emacspeak-dired-pdf-open

`emacspeak-dired-pdf-open` [Command]

Open PDF file on current dired line.

12.49.1.10 emacspeak-dired-rpm-query-in-dired

`emacspeak-dired-rpm-query-in-dired` [Command]

Run rpm -qi on current dired entry.

12.49.1.11 emacspeak-dired-show-file-type

`emacspeak-dired-show-file-type` (**&optional** *file deref-symlinks*) [Command]

Displays type of current file by running command file.
Like Emacs' built-in dired-show-file-type but allows user to customize
options passed to command 'file'.

(fn &optional FILE DEREF-SYMLINKS)

12.49.1.12 emacspeak-dired-speak-file-access-time

`emacspeak-dired-speak-file-access-time` [Command]

Speak access time of the current file.

12.49.1.13 emacspeak-dired-speak-file-modification-time

`emacspeak-dired-speak-file-modification-time` [Command]

Speak modification time of the current file.

12.49.1.14 emacspeak-dired-speak-file-permissions

`emacspeak-dired-speak-file-permissions` [Command]

Speak the permissions of the current file.

12.49.1.15 emacspeak-dired-speak-file-size

`emacspeak-dired-speak-file-size` [Command]

Speak the size of the current file.
On a directory line, run du -s on the directory to speak its size.

12.49.1.16 emacspeak-dired-speak-header-line

`emacspeak-dired-speak-header-line` [Command]

Speak the header line of the dired buffer.

12.49.1.17 emacspeak-dired-speak-symlink-target

`emacspeak-dired-speak-symlink-target` [Command]

Speaks the target of the symlink on the current line.

12.49.1.18 emacspeak-locate-play-results-as-playlist

`emacspeak-locate-play-results-as-playlist` (**&optional** *shuffle*) [Command]

Treat locate results as a play-list.
Optional interactive prefix arg shuffles playlist.

(fn &optional SHUFFLE)

12.49.2 emacspeak-dired Options

`User Option` *voice-dired-broken-symlink* [Variable]

Personality used for dired-broken-symlink This personality uses 'voice-monotone whose effect can be changed globally by customizing 'voice-monotone-settings.

`User Option` *voice-dired-directory* [Variable]

Personality used for dired-directory This personality uses voice-bolden-medium whose effect can be changed globally by customizing voice-bolden-medium-settings.

`User Option` *voice-dired-flagged* [Variable]

Personality used for dired-flagged This personality uses voice-animate-extra whose effect can be changed globally by customizing voice-animate-extra-settings.

`User Option` *voice-dired-header* [Variable]

Personality used for dired-header This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

`User Option` *voice-dired-ignored* [Variable]

Personality used for dired-ignored This personality uses voice-lighten-extra whose effect can be changed globally by customizing voice-lighten-extra-settings.

`User Option` *voice-dired-mark* [Variable]

Personality used for dired-mark This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-dired-marked* [Variable]

Personality used for dired-marked This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-dired-perm-write* [Variable]

Personality used for dired-perm-write This personality uses voice-lighten-extra whose effect can be changed globally by customizing voice-lighten-extra-settings.

`User Option` *voice-dired-set-id* [Variable]

Personality used for dired-set-id This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

`User Option` *voice-dired-special* [Variable]

Personality used for dired-special This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-dired-symlink* [Variable]

Personality used for dired-symlink This personality uses voice-animate-extra whose effect can be changed globally by customizing voice-animate-extra-settings.

`User Option` *voice-dired-warning* [Variable]

Personality used for dired-warning This personality uses voice-animate-extra whose effect can be changed globally by customizing voice-animate-extra-settings.

12.50 emacspeak-dismal

### emacspeak-dismal: No Commentary

12.50.1 Emacspeak-Dismal Commands

12.50.1.1 emacspeak-dismal-backward-col-and-summarize

`emacspeak-dismal-backward-col-and-summarize` (*cols*) [Command]

Move backward by arg columns
(the previous column by default)and summarize it.

(fn COLS)

12.50.1.2 emacspeak-dismal-backward-row-and-summarize

`emacspeak-dismal-backward-row-and-summarize` (*rows*) [Command]

Move backward by arg rows
(the previous row by default)and summarize it.

(fn ROWS)

12.50.1.3 emacspeak-dismal-col-summarize

`emacspeak-dismal-col-summarize` [Command]

Summarizes a col using the specification in list
emacspeak-dismal-col-summarizer-list

12.50.1.4 emacspeak-dismal-display-cell-expression

`emacspeak-dismal-display-cell-expression` [Command]

Display the expression in the message area

12.50.1.5 emacspeak-dismal-display-cell-value

`emacspeak-dismal-display-cell-value` [Command]

Display the cell value in the message area

12.50.1.6 emacspeak-dismal-display-cell-with-col-header

`emacspeak-dismal-display-cell-with-col-header` [Command]

Display current cell along with its column header.
The 'column header' is the entry in row 0.

12.50.1.7 emacspeak-dismal-display-cell-with-row-header

`emacspeak-dismal-display-cell-with-row-header` [Command]

Displays current cell along with its row header.
The 'row header' is the entry in column 0.

12.50.1.8 emacspeak-dismal-forward-col-and-summarize

`emacspeak-dismal-forward-col-and-summarize` (*cols*) [Command]

Move forward by arg columns
(the next column by default)and summarize it.

(fn COLS)

12.50.1.9 emacspeak-dismal-forward-row-and-summarize

`emacspeak-dismal-forward-row-and-summarize` (*rows*) [Command]

Move forward by arg rows
(the next row by default)and summarize it.

(fn ROWS)

12.50.1.10 emacspeak-dismal-row-summarize

`emacspeak-dismal-row-summarize` [Command]

Summarizes a row using the specification in list
emacspeak-dismal-row-summarizer-list

12.50.1.11 emacspeak-dismal-set-col-summarizer-list

`emacspeak-dismal-set-col-summarizer-list` [Command]

Specify or reset col summarizer list.

12.50.1.12 emacspeak-dismal-set-row-summarizer-list

`emacspeak-dismal-set-row-summarizer-list` [Command]

Specify or reset row summarizer list.

12.50.1.13 emacspeak-dismal-set-sheet-summarizer-list

`emacspeak-dismal-set-sheet-summarizer-list` [Command]

Specify or reset sheet summarizer list.

12.50.1.14 emacspeak-dismal-sheet-summarize

`emacspeak-dismal-sheet-summarize` [Command]

Summarizes a sheet using the specification in list
emacspeak-dismal-sheet-summarizer-list

12.51 emacspeak-dumb-jump

DUMB-JUMP == Jump to imputed cross-references in source code. This module speech-enables dumb-jump

12.52 emacspeak-ecb

The ECB is an Emacs Class Browser. This module speech-enables ECB

12.52.1 Emacspeak-Ecb Commands

12.52.1.1 emacspeak-ecb-speak-window-directories

`emacspeak-ecb-speak-window-directories` [Command]

Speak contents of directories window.

12.52.1.2 emacspeak-ecb-speak-window-history

`emacspeak-ecb-speak-window-history` [Command]

Speak contents of history window.

12.52.1.3 emacspeak-ecb-speak-window-methods

`emacspeak-ecb-speak-window-methods` [Command]

Speak contents of methods window.

12.52.1.4 emacspeak-ecb-speak-window-sources

`emacspeak-ecb-speak-window-sources` [Command]

Speak contents of sources window.

12.52.1.5 emacspeak-ecb-tree-backspace

`emacspeak-ecb-tree-backspace` [Command]

Back up during incremental search in tree buffers.

12.52.1.6 emacspeak-ecb-tree-clear

`emacspeak-ecb-tree-clear` [Command]

Clear search pattern during incremental search in tree buffers.

12.53 emacspeak-eclim

ECLIM == Eclipse/Vim integration. http://www.eclim.org turns Eclipse into a headless server that can be called from other programs. Package Emacs-Eclim connects Emacs to Eclim. Package emacspeak-eclim speech-enables emacs-eclim.

12.54 emacspeak-ediff

Ediff provides a nice visual interface to diff. ;;;Comparing and patching files is easy with ediff when you can see the screen. This module provides Emacspeak extensions to work fluently with ediff. Try it out, it's an excellent example of why Emacspeak is better than a traditional screenreader. This module was originally written to interface to the old ediff.el bundled with GNU Emacs 19.28 and earlier. It has been updated to work with the newer and much larger ediff system found in Emacs 19.29 and later.

When using under modern versions of Emacs, I recommend setting (setq ediff-window-setup-function 'ediff-setup-windows-plain) so that Emacs always displays Ediff windows in a single frame.

12.54.1 Emacspeak-Ediff Commands

12.54.1.1 emacspeak-ediff-speak-current-difference

`emacspeak-ediff-speak-current-difference` [Command]

Speak the current difference

12.54.2 emacspeak-ediff Options

`User Option` *voice-ediff-current-diff-A* [Variable]

Personality used for ediff-current-diff-A This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

`User Option` *voice-ediff-current-diff-Ancestor* [Variable]

Personality used for ediff-current-diff-Ancestor This personality uses voice-lighten-extra whose effect can be changed globally by customizing voice-lighten-extra-settings.

`User Option` *voice-ediff-current-diff-B* [Variable]

Personality used for ediff-current-diff-B This personality uses voice-brighten whose effect can be changed globally by customizing voice-brighten-settings.

`User Option` *voice-ediff-current-diff-C* [Variable]

Personality used for ediff-current-diff-C This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-ediff-even-diff-A* [Variable]

Personality used for ediff-even-diff-A This personality uses voice-brighten whose effect can be changed globally by customizing voice-brighten-settings.

`User Option` *voice-ediff-even-diff-Ancestor* [Variable]

Personality used for ediff-even-diff-Ancestor This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-ediff-even-diff-B* [Variable]

Personality used for ediff-even-diff-B This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

`User Option` *voice-ediff-even-diff-C* [Variable]

Personality used for ediff-even-diff-C This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-ediff-fine-diff-A* [Variable]

Personality used for ediff-fine-diff-A This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

`User Option` *voice-ediff-fine-diff-Ancestor* [Variable]

Personality used for ediff-fine-diff-Ancestor This personality uses voice-lighten-extra whose effect can be changed globally by customizing voice-lighten-extra-settings.

`User Option` *voice-ediff-fine-diff-B* [Variable]

Personality used for ediff-fine-diff-B This personality uses voice-brighten whose effect can be changed globally by customizing voice-brighten-settings.

`User Option` *voice-ediff-fine-diff-C* [Variable]

Personality used for ediff-fine-diff-C This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-ediff-odd-diff-A* [Variable]

Personality used for ediff-odd-diff-A This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

`User Option` *voice-ediff-odd-diff-Ancestor* [Variable]

Personality used for ediff-odd-diff-Ancestor This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-ediff-odd-diff-B* [Variable]

Personality used for ediff-odd-diff-B This personality uses voice-brighten whose effect can be changed globally by customizing voice-brighten-settings.

`User Option` *voice-ediff-odd-diff-C* [Variable]

Personality used for ediff-odd-diff-C This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

12.55 emacspeak-eglot

EGLOT == LSP Support for emacs

12.55.1 emacspeak-eglot Options

`User Option` *voice-eglot-mode-line* [Variable]

Personality used for eglot-mode-line This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

12.56 emacspeak-ein

EIN == Emacs IPython Notebook You can install package EIN via mELPA This module speech-enables EIN

12.56.1 Emacspeak-Ein Commands

12.56.1.1 emacspeak-ein-speak-current-cell

`emacspeak-ein-speak-current-cell` [Command]

Speak current cell.

12.56.2 emacspeak-ein Options

`User Option` *voice-ein:cell-input-area* [Variable]

Personality used for ein:cell-input-area This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-ein:cell-input-prompt* [Variable]

Personality used for ein:cell-input-prompt This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

`User Option` *voice-ein:cell-output-area* [Variable]

Personality used for ein:cell-output-area This personality uses voice-brighten whose effect can be changed globally by customizing voice-brighten-settings.

`User Option` *voice-ein:cell-output-area-error* [Variable]

Personality used for ein:cell-output-area-error This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-ein:cell-output-prompt* [Variable]

Personality used for ein:cell-output-prompt This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-ein:cell-output-stderr* [Variable]

Personality used for ein:cell-output-stderr This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-ein:markdown-header-face-1* [Variable]

Personality used for ein:markdown-header-face-1 This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-ein:markdown-header-face-2* [Variable]

Personality used for ein:markdown-header-face-2 This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

`User Option` *voice-ein:markdown-header-face-3* [Variable]

Personality used for ein:markdown-header-face-3 This personality uses voice-annotate whose effect can be changed globally by customizing voice-annotate-settings.

`User Option` *voice-ein:markdown-header-face-4* [Variable]

Personality used for ein:markdown-header-face-4 This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-ein:markdown-header-face-5* [Variable]
Personality used for ein:markdown-header-face-5 This personality uses voice-monotone-light whose effect can be changed globally by customizing voice-monotone-light-settings.

`User Option` *voice-ein:markdown-header-face-6* [Variable]
Personality used for ein:markdown-header-face-6 This personality uses voice-monotone-medium whose effect can be changed globally by customizing voice-monotone-medium-settings.

`User Option` *voice-ein:notification-tab-normal* [Variable]
Personality used for ein:notification-tab-normal This personality uses voice-smoothen-extra whose effect can be changed globally by customizing voice-smoothen-extra-settings.

`User Option` *voice-ein:notification-tab-selected* [Variable]
Personality used for ein:notification-tab-selected This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

12.57 emacspeak-elfeed

ELFEED == Feed Reader for Emacs. Install from elpa M-x package-install elfeed

12.57.1 Emacspeak-Elfeed Commands

12.57.1.1 emacspeak-elfeed-eww-entry-at-point

`emacspeak-elfeed-eww-entry-at-point` [Command]
Display current article in EWW.

12.57.1.2 emacspeak-elfeed-filter-entry-at-point

`emacspeak-elfeed-filter-entry-at-point` [Command]
Display current article after filtering.

12.57.1.3 emacspeak-elfeed-next-entry

`emacspeak-elfeed-next-entry` [Command]
Move to next entry and speak it.

12.57.1.4 emacspeak-elfeed-previous-entry

`emacspeak-elfeed-previous-entry` [Command]
Move to previous entry and speak it.

12.57.1.5 emacspeak-elfeed-speak-entry-at-point

`emacspeak-elfeed-speak-entry-at-point` [Command]
Speak entry at point.

12.58 emacspeak-elisp-refs

Speech-enable package elisp-refs. ELISP-REFS ==

12.59 emacspeak-elpy

ELPY == Emacs Lisp Python IDE Speech-enables all aspects of elpy.

12.60 emacspeak-elscreen

ELSCREEN == Emacs Window Session Manager Speech-enable interactive commands.

12.61 emacspeak-emms

Speech-enables EMMS — the Emacs equivalent of XMMS available from the Emacs package archive. http://savannah.gnu.org/project/emms EMMS is under active development, to get the current CVS version, use Emacspeak command M-x emacspeak-cvs-gnu-get-project-snapshot RET emms RET

12.61.1 Emacspeak-Emms Commands

12.61.1.1 emacspeak-emms-speak-current-track

`emacspeak-emms-speak-current-track` [Command]

Speak current track.

12.62 emacspeak-enriched

emacspeak extensions to voiceify rich text.

12.62.1 Emacspeak-Enriched Commands

12.62.1.1 emacspeak-enriched-voiceify-faces

`emacspeak-enriched-voiceify-faces` (*start end*) [Command]

Map base fonts to voices.
Useful in voiceifying rich text.

(fn START END)

12.63 emacspeak-entertain

Auditory interface to misc games

12.63.1 Emacspeak-Entertain Commands

12.63.1.1 emacspeak-hangman-speak-guess

`emacspeak-hangman-speak-guess` [Command]

Speak current guessed string.

12.63.1.2 emacspeak-hangman-speak-statistics

`emacspeak-hangman-speak-statistics` [Command]

Speak statistics.

12.63.2 emacspeak-entertain Options

`User Option` *voice-mpuz-solved* [Variable]

Personality used for mpuz-solved This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

`User Option` *voice-mpuz-trivial* [Variable]

Personality used for mpuz-trivial This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-mpuz-unsolved* [Variable]

Personality used for mpuz-unsolved This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

12.64 emacspeak-epa

EPA == EasyPG Assistant Integrate GPG functionality into Emacs. Speech-enable all interactive commands.

12.64.1 emacspeak-epa Options

`User Option` *voice-epa-field-body* [Variable]

Personality used for epa-field-body This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

`User Option` *voice-epa-field-name* [Variable]

Personality used for epa-field-name This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

`User Option` *voice-epa-mark* [Variable]

Personality used for epa-mark This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-epa-string* [Variable]

Personality used for epa-string This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-epa-validity-disabled* [Variable]

Personality used for epa-validity-disabled This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-epa-validity-high* [Variable]

Personality used for epa-validity-high This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

`User Option` *voice-epa-validity-low* [Variable]

Personality used for epa-validity-low This personality uses voice-smoothen-extra whose effect can be changed globally by customizing voice-smoothen-extra-settings.

`User Option` *voice-epa-validity-medium* [Variable]

Personality used for epa-validity-medium This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

12.65 emacspeak-eperiodic

eperiodic produces an interactive periodic table of elements and can be found at http://vegemite.chem.nottingham.ac.uk/~matt/emacs/eperiodic.el

12.65.1 Emacspeak-Eperiodic Commands

12.65.1.1 emacspeak-eperiodic-goto-property-section

`emacspeak-eperiodic-goto-property-section` [Command]

Mark position and jump to properties section.

12.65.1.2 emacspeak-eperiodic-next-line

`emacspeak-eperiodic-next-line` [Command]

Move to next row and speak element.

12.65.1.3 emacspeak-eperiodic-play-description

`emacspeak-eperiodic-play-description` [Command]

Play audio description from WebElements.

12.65.1.4 emacspeak-eperiodic-previous-line

`emacspeak-eperiodic-previous-line` [Command]

Move to next row and speak element.

12.65.1.5 emacspeak-eperiodic-speak-current-element

`emacspeak-eperiodic-speak-current-element` [Command]

Speak element at point.

12.66 emacspeak-epub

12.66.1 Introduction

This module implements the Emacspeak EPub Bookshelf — a unified interface for organizing, locating and reading EPub EBooks on the emacspeak Audio Desktop. The epub reader is built using the Emacs Web Browser (EWW), and all of emacspeak's EWW conveniences are available when reading EBooks — see See Section 12.74 [emacspeak-eww], page 124, for useful tools including bookmarking and structured navigation. For now it supports epub2 — it will support epub3 some time in the future.

The main entry point is command `emacspeak-epub` bound to *C-e g*. This command opens a new bookshelf buffer unless the user has previously opened a specific bookshelf. A *bookshelf* is a buffer that lists books placed on a given bookshelf — these are listed by *title* and *author*. The bookshelf buffer is in a special mode that provides single-key commands for adding, removing and finding books, as well as for opening the selected book using Emacs' built-in Web browser (`eww`).

The next few sections give a high-level overview of the emacspeak Bookshelf and EPub interaction, followed by detailed documentation on the various commands and user options.

12.66.2 Organizing EBooks On The Emacspeak Desktop

In the simplest case, EBooks can be placed under a specific directory (with sub-directories as needed). Customize user option `emacspeak-epub-library-directory` to point to this location. Here is a quick summary of commands for organizing, saving and opening a bookshelf:

a	emacspeak-epub-bookshelf-add-epub
b	emacspeak-epub-bookshelf-open
c	emacspeak-epub-bookshelf-clear
d	emacspeak-epub-bookshelf-remove-this-book
r	emacspeak-epub-bookshelf-rename
l	emacspeak-epub-locate-epubs
C-a	emacspeak-epub-bookshelf-add-directory
C-d	emacspeak-epub-bookshelf-remove-directory
C-l	emacspeak-epub-bookshelf-redraw
C-o	emacspeak-epub-bookshelf-open-epub
M-s	emacspeak-epub-bookshelf-save
C-x C-q	emacspeak-epub-bookshelf-refresh
C-x C-s	emacspeak-epub-bookshelf-save

12.66.3 Integrating With Project Gutenberg

Gutenberg integration provides one-shot commands for downloading the latest copy of the Gutenberg catalog and finding and downloading the desired epub for offline reading.

C	emacspeak-epub-gutenberg-catalog
g	emacspeak-epub-gutenberg-download

Once downloaded, these EBooks can be organized under `emacspeak-epub-library-directory` For more advanced usage, see the next section on integrating with Calibre catalogs.

12.66.4 Calibre Integration

Project Calibre enables the indexing and searching of large EBook collections. Read the Calibre documentation for organizing and indexing your EBook library. See user options named `emacspeak-epub-calibre-*` for customizing emacspeak to work with Calibre. Once set up, Calibre integration provides the following commands from the *bookshelf* buffer:

`/`	emacspeak-epub-calibre-results
`A`	emacspeak-epub-bookshelf-calibre-author
`S`	emacspeak-epub-bookshelf-calibre-search
`T`	emacspeak-epub-bookshelf-calibre-title

12.66.5 Reading EBooks From The Bookshelf

The most efficient means to read an EBook is to have EWW render the entire book — this works well even for very large EBooks given that EWW is efficient at rendering HTML. Rendering the entire book means that all of the contents are available for searching. To view an EBook in its entirety, use command `emacspeak-epub-eww`. You can open the EPub table of contents with command `emacspeak-epub-open`; for a well-constructed epub, this TOC should provide hyperlinks to each section listed in the table of contents.

`RET`	emacspeak-epub-eww
`e`	emacspeak-epub-eww
`f`	emacspeak-epub-browse-files
`o`	emacspeak-epub-open

12.66.6 Emacspeak-Epub Commands

12.66.6.1 emacspeak-calibre-mode

`emacspeak-calibre-mode` [Command]

A Calibre Front-end.
Letters do not insert themselves; instead, they are commands.

key binding
— ——-

RET emacspeak-epub-calibre-dired-at-point

In addition to any hooks its parent mode 'special-mode' might have run,
this mode runs the hook 'emacspeak-calibre-mode-hook', as the final or penultimate step
during initialization.

12.66.6.2 emacspeak-epub

`emacspeak-epub` [Command]

C-e g

`<fn> g`

EPub Interaction.
When opened, displays a bookshelf consisting of epubs found at the root directory.
For detailed documentation, see M-x emacspeak-epub-mode

12.66.6.3 emacspeak-epub-bookshelf-add-directory

`emacspeak-epub-bookshelf-add-directory` (*directory* **&optional** *recursive*) [Command]

Add EPubs found in specified directory to the bookshelf.
Interactive prefix arg searches recursively in directory.

(fn DIRECTORY &optional RECURSIVE)

12.66.6.4 emacspeak-epub-bookshelf-add-epub

`emacspeak-epub-bookshelf-add-epub` (*epub-file*) [Command]

Add epub file to current bookshelf.

(fn EPUB-FILE)

12.66.6.5 emacspeak-epub-bookshelf-calibre-author

`emacspeak-epub-bookshelf-calibre-author` (*pattern*) [Command]

Add results of an author search to current bookshelf.

(fn PATTERN)

12.66.6.6 emacspeak-epub-bookshelf-calibre-search

`emacspeak-epub-bookshelf-calibre-search` (*pattern*) [Command]

Add results of an title/author search to current bookshelf.

(fn PATTERN)

12.66.6.7 emacspeak-epub-bookshelf-calibre-title

`emacspeak-epub-bookshelf-calibre-title` (*pattern*) [Command]

Add results of an title search to current bookshelf.

(fn PATTERN)

12.66.6.8 emacspeak-epub-bookshelf-clear

`emacspeak-epub-bookshelf-clear` [Command]

Clear all books from bookshelf.

12.66.6.9 emacspeak-epub-bookshelf-load

`emacspeak-epub-bookshelf-load` [Command]

Load bookshelf metadata from disk.

12.66.6.10 emacspeak-epub-bookshelf-open

`emacspeak-epub-bookshelf-open` (*bookshelf*) [Command]

Load bookshelf metadata from specified bookshelf.

(fn BOOKSHELF)

12.66.6.11 emacspeak-epub-bookshelf-open-epub

`emacspeak-epub-bookshelf-open-epub` (*epub-file*) [Command]

Open epub file and add it to current bookshelf.

(fn EPUB-FILE)

12.66.6.12 emacspeak-epub-bookshelf-redraw

`emacspeak-epub-bookshelf-redraw` (**&optional** *author-first*) [Command]

Redraw Bookshelf.
Optional interactive prefix arg author-first prints author at the left.

(fn &optional AUTHOR-FIRST)

12.66.6.13 emacspeak-epub-bookshelf-refresh

`emacspeak-epub-bookshelf-refresh` [Command]

Refresh and redraw bookshelf.

12.66.6.14 emacspeak-epub-bookshelf-remove-directory

`emacspeak-epub-bookshelf-remove-directory` (*directory* **&optional** *recursive*) [Command]

Remove EPubs found in specified directory from the bookshelf.
Interactive prefix arg searches recursively in directory.

(fn DIRECTORY &optional RECURSIVE)

12.66.6.15 emacspeak-epub-bookshelf-remove-this-book

`emacspeak-epub-bookshelf-remove-this-book` [Command]

Remove the book on current line from this bookshelf.
No book files are deleted.

12.66.6.16 emacspeak-epub-bookshelf-rename

`emacspeak-epub-bookshelf-rename` (*name* **&optional** *overwrite*) [Command]

Saves current bookshelf to specified name.
Interactive prefix arg 'overwrite' will overwrite existing file.

(fn NAME &optional OVERWRITE)

12.66.6.17 emacspeak-epub-bookshelf-save

`emacspeak-epub-bookshelf-save` [Command]

Save bookshelf metadata.

12.66.6.18 emacspeak-epub-browse-files

`emacspeak-epub-browse-files` (*epub*) [Command]

Browse list of HTML files in EPub.
Useful if table of contents in toc.ncx is empty.

(fn EPUB)

12.66.6.19 emacspeak-epub-calibre-dired-at-point

`emacspeak-epub-calibre-dired-at-point` [Command]

Open directory containing current result.

12.66.6.20 emacspeak-epub-calibre-results

`emacspeak-epub-calibre-results` [Command]

Show most recent Calibre search results.

12.66.6.21 emacspeak-epub-delete

`emacspeak-epub-delete` [Command]

Delete EPub under point.

12.66.6.22 emacspeak-epub-eww

`emacspeak-epub-eww` (*epub-file* **&optional** *broken-ncx*) [Command]

Display entire book using EWW from EPub.
Uses content listed in toc.ncx or equivalent by default.
Interactive prefix arg broken-ncx asks to use the sorted list of html files in the epub file instead.

(fn EPUB-FILE &optional BROKEN-NCX)

12.66.6.23 emacspeak-epub-google

`emacspeak-epub-google` (*query*) [Command]

Search for Epubs from Google Books.

(fn QUERY)

12.66.6.24 emacspeak-epub-gutenberg-catalog

`emacspeak-epub-gutenberg-catalog` (**&optional** *refresh*) [Command]

Open Gutenberg catalog.
Fetch if needed, or if refresh is T.

(fn &optional REFRESH)

12.66.6.25 emacspeak-epub-gutenberg-download

`emacspeak-epub-gutenberg-download` (*book-id* **&optional** *download*) [Command]

Open web page for specified book.
Place download url for epub in kill ring.
With interactive prefix arg ‘download’, download the epub.

(fn BOOK-ID &optional DOWNLOAD)

12.66.6.26 emacspeak-epub-locate-epubs

`emacspeak-epub-locate-epubs` (*pattern*) [Command]

Locate epub files using locate.

(fn PATTERN)

12.66.6.27 emacspeak-epub-mode

`emacspeak-epub-mode` [Command]

An EPub Front-end.
Letters do not insert themselves; instead, they are commands.
key binding
— ——-

C-a emacspeak-epub-bookshelf-add-directory
C-d emacspeak-epub-bookshelf-remove-directory
C-l emacspeak-epub-bookshelf-redraw
RET emacspeak-epub-eww
C-o emacspeak-epub-bookshelf-open-epub
C-x Prefix Command
ESC Prefix Command
/ emacspeak-epub-calibre-results
A emacspeak-epub-bookshelf-calibre-author
C emacspeak-epub-gutenberg-catalog
G emacspeak-epub-google
O emacspeak-epub-open-with-nov
S emacspeak-epub-bookshelf-calibre-search
T emacspeak-epub-bookshelf-calibre-title
a emacspeak-epub-bookshelf-add-epub

b emacspeak-epub-bookshelf-open
c emacspeak-epub-bookshelf-clear
d emacspeak-epub-bookshelf-remove-this-book
e emacspeak-epub-eww
f emacspeak-epub-browse-files
g emacspeak-epub-gutenberg-download
l emacspeak-epub-locate-epubs
n next-line
o emacspeak-epub-open
p previous-line
r emacspeak-epub-bookshelf-rename

M-s emacspeak-epub-bookshelf-save

C-x C-q emacspeak-epub-bookshelf-refresh
C-x C-s emacspeak-epub-bookshelf-save

In addition to any hooks its parent mode 'special-mode' might have run,
this mode runs the hook 'emacspeak-epub-mode-hook', as the final or penultimate step during initialization.

12.66.6.28 emacspeak-epub-next

`emacspeak-epub-next` [Command]

Move to next book.

12.66.6.29 emacspeak-epub-open

`emacspeak-epub-open` (*epub-file*) [Command]

Open specified Epub.
Filename may need to be shell-quoted when called from Lisp.

(fn EPUB-FILE)

12.66.6.30 emacspeak-epub-open-with-nov

`emacspeak-epub-open-with-nov` [Command]

Open ebook at point in nov-mode.

12.66.6.31 emacspeak-epub-previous

`emacspeak-epub-previous` [Command]

Move to previous book.

12.66.6.32 emacspeak-epub-url-executor

`emacspeak-epub-url-executor` (*url*) [Command]

Custom URL executor for use in EPub Mode.

(fn URL)

12.66.7 emacspeak-epub Options

`User Option` *emacspeak-epub-bookshelf-directory* [Variable]
Directory where we keep .bsf files defining various bookshelves.

`User Option` *emacspeak-epub-calibre-root-dir* [Variable]
Root of Calibre library.

`User Option` *emacspeak-epub-gutenberg-mirror* [Variable]
Base URL for Gutenberg mirror.

`User Option` *emacspeak-epub-gutenberg-suffix* [Variable]
Suffix of book type we retrieve.

`User Option` *emacspeak-epub-library-directory* [Variable]
Directory under which we store Epubs.

12.67 emacspeak-erc

erc.el is a modern Emacs client for IRC including color and font locking support. erc.el - an Emacs IRC client (by Alexander L. Belikoff) http://www.cs.cmu.edu/~berez/irc/erc.el

12.67.1 Emacspeak-Erc Commands

12.67.1.1 emacspeak-erc-add-name-to-monitor

`emacspeak-erc-add-name-to-monitor` (*name* **&optional** *quiten-pronunciation*) [Command]
Add people to monitor in this room.
Optional interactive prefix arg defines a pronunciation that
  silences speaking of this perso's name.

(fn NAME &optional QUITEN-PRONUNCIATION)

12.67.1.2 emacspeak-erc-delete-name-from-monitor

`emacspeak-erc-delete-name-from-monitor` (*name*) [Command]
Remove name to monitor in this room.

(fn NAME)

12.67.1.3 emacspeak-erc-setup-cricket-rules

`emacspeak-erc-setup-cricket-rules` [Command]
Set up #cricket channels.

12.67.1.4 emacspeak-erc-toggle-my-monitor

`emacspeak-erc-toggle-my-monitor` (**&optional** *prefix*) [Command]
Toggle state of ERC monitor of my messages.
Interactive PREFIX arg means toggle the global default value, and then
set the current local value to the result.

12.67.1.5 emacspeak-erc-toggle-room-monitor

`emacspeak-erc-toggle-room-monitor` (**&optional** *prefix*) [Command]

Toggle state of ERC room monitor.
Interactive
PREFIX arg means toggle the global default value, and then
set the current local value to the result.

12.67.1.6 emacspeak-erc-toggle-speak-all-participants

`emacspeak-erc-toggle-speak-all-participants` (**&optional** *prefix*) [Command]

Toggle state of ERC speak all participants..
Interactive
PREFIX arg means toggle the global default value, and then
set the current local value to the result.

12.67.2 emacspeak-erc Options

`User Option` *emacspeak-erc-ignore-notices* [Variable]

Set to T if you dont want to see notification messages from the server.

`User Option` *emacspeak-erc-my-nick* [Variable]

My IRC nick.

`User Option` *emacspeak-erc-speak-all-participants* [Variable]

Speak all things said if t.

12.68 emacspeak-eshell

EShell is a shell implemented entirely in Emacs Lisp. It is part of emacs 21 –and can also be used under Emacs 20. This module speech-enables EShell

12.68.1 emacspeak-eshell Options

`User Option` *voice-eshell-ls-archive* [Variable]

Personality used for eshell-ls-archive This personality uses voice-lighten-extra whose effect can be changed globally by customizing voice-lighten-extra-settings.

`User Option` *voice-eshell-ls-backup* [Variable]

Personality used for eshell-ls-backup This personality uses voice-monotone-medium whose effect can be changed globally by customizing voice-monotone-medium-settings.

`User Option` *voice-eshell-ls-clutter* [Variable]

Personality used for eshell-ls-clutter This personality uses voice-smoothen-extra whose effect can be changed globally by customizing voice-smoothen-extra-settings.

`User Option` *voice-eshell-ls-directory* [Variable]

Personality used for eshell-ls-directory This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

User Option *voice-eshell-ls-executable* [Variable]
Personality used for eshell-ls-executable This personality uses voice-animate-extra whose effect can be changed globally by customizing voice-animate-extra-settings.

User Option *voice-eshell-ls-missing* [Variable]
Personality used for eshell-ls-missing This personality uses voice-brighten whose effect can be changed globally by customizing voice-brighten-settings.

User Option *voice-eshell-ls-product* [Variable]
Personality used for eshell-ls-product This personality uses voice-lighten-medium whose effect can be changed globally by customizing voice-lighten-medium-settings.

User Option *voice-eshell-ls-readonly* [Variable]
Personality used for eshell-ls-readonly This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

User Option *voice-eshell-ls-special* [Variable]
Personality used for eshell-ls-special This personality uses voice-lighten-extra whose effect can be changed globally by customizing voice-lighten-extra-settings.

User Option *voice-eshell-ls-symlink* [Variable]
Personality used for eshell-ls-symlink This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

User Option *voice-eshell-ls-unreadable* [Variable]
Personality used for eshell-ls-unreadable This personality uses voice-monotone-medium whose effect can be changed globally by customizing voice-monotone-medium-settings.

User Option *voice-eshell-prompt* [Variable]
Personality used for eshell-prompt This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

12.69 emacspeak-ess

ESS == Emacs Speaks Statistics This module makes ESS speak.

12.70 emacspeak-etable

table.el provides rich table editing for emacs. this module speech-enables table.el

12.70.1 Emacspeak-Etable Commands

12.70.1.1 emacspeak-etable-speak-cell

`emacspeak-etable-speak-cell` [Command]
Speak current cell.

12.71 emacspeak-eterm

This module makes eterm talk. Eterm is the new terminal emulator for Emacs. Use of emacspeak with eterm really needs an info page. At present, the only documentation is the source level documentation. This module uses Control-t as an additional prefix key to allow the user To move around the terminal and have different parts spoken.

12.71.1 Emacspeak-Eterm Commands

12.71.1.1 emacspeak-eterm-copy-region-to-register

`emacspeak-eterm-copy-region-to-register` (*register*) [Command]

Copy text from terminal to an Emacs REGISTER.
This copies region delimited by the emacspeak eterm marker
set by command M-x emacspeak-eterm-set-marker and the
emacspeak eterm pointer to a register.

(fn REGISTER)

12.71.1.2 emacspeak-eterm-define-window

`emacspeak-eterm-define-window` (*id*) [Command]

Prompt for a window ID.
The window is then define to be
the rectangle delimited by point and eterm mark. This is to
be used when emacspeak is set to review mode inside an
eterm.

(fn ID)

12.71.1.3 emacspeak-eterm-describe-window

`emacspeak-eterm-describe-window` (*id*) [Command]

Describe an eterm window.
Description indicates eterm window coordinates and whether it is stretchable

(fn ID)

12.71.1.4 emacspeak-eterm-goto-line

`emacspeak-eterm-goto-line` (*line*) [Command]

Move emacspeak eterm pointer to a specified LINE.

(fn LINE)

12.71.1.5 emacspeak-eterm-kill-ring-save-region

`emacspeak-eterm-kill-ring-save-region` [Command]

Copy text from terminal to kill ring.
This copies region delimited by the emacspeak eterm marker

set by command M-x emacspeak-eterm-set-marker and the
emacspeak eterm pointer.

12.71.1.6 emacspeak-eterm-maybe-send-raw

`emacspeak-eterm-maybe-send-raw` [Command]

Send a raw character through if in the terminal buffer.
Execute end of line if
in a non eterm buffer if executed via C-e C-e

12.71.1.7 emacspeak-eterm-paste-register

`emacspeak-eterm-paste-register` (*register*) [Command]

Paste contents of REGISTER at current location.
If the specified register contains text, then that text is
sent to the terminal as if it were typed by the user.

(fn REGISTER)

12.71.1.8 emacspeak-eterm-pointer-backward-word

`emacspeak-eterm-pointer-backward-word` (*count*) [Command]

Move the pointer backward by words.
Interactive numeric prefix arg specifies number of words to move.
Argument COUNT specifies number of words by which to move.

(fn COUNT)

12.71.1.9 emacspeak-eterm-pointer-down

`emacspeak-eterm-pointer-down` (*count*) [Command]

Move the pointer down a line.
Argument COUNT specifies number of lines by which to move.

(fn COUNT)

12.71.1.10 emacspeak-eterm-pointer-forward-word

`emacspeak-eterm-pointer-forward-word` (*count*) [Command]

Move the pointer forward by words.
Interactive numeric prefix arg specifies number of words to move.
Argument COUNT specifies number of words by which to move.

(fn COUNT)

12.71.1.11 emacspeak-eterm-pointer-left

`emacspeak-eterm-pointer-left` (*count*) [Command]

Move the pointer left.

Argument COUNT specifies number of columns by which to move.

(fn COUNT)

12.71.1.12 emacspeak-eterm-pointer-right

`emacspeak-eterm-pointer-right` (*count*) [Command]

Move the pointer right.
Argument COUNT specifies number of columns by which to move.

(fn COUNT)

12.71.1.13 emacspeak-eterm-pointer-to-bottom

`emacspeak-eterm-pointer-to-bottom` [Command]

Move the pointer to the bottom of the screen.

12.71.1.14 emacspeak-eterm-pointer-to-cursor

`emacspeak-eterm-pointer-to-cursor` [Command]

Move the pointer to the cursor.

12.71.1.15 emacspeak-eterm-pointer-to-left-edge

`emacspeak-eterm-pointer-to-left-edge` [Command]

Move the pointer to the right edge.

12.71.1.16 emacspeak-eterm-pointer-to-next-color-change

`emacspeak-eterm-pointer-to-next-color-change` (**&optional** *count*) [Command]

Move the eterm pointer to the next color change.
This allows you to move between highlighted regions of the screen.
Optional argument COUNT specifies how many changes to skip.

(fn &optional COUNT)

12.71.1.17 emacspeak-eterm-pointer-to-previous-color-change

`emacspeak-eterm-pointer-to-previous-color-change` (**&optional** *count*) [Command]

Move the eterm pointer to the next color change.
This allows you to move between highlighted regions of the screen.
Optional argument COUNT specifies how many changes to skip.

(fn &optional COUNT)

12.71.1.18 emacspeak-eterm-pointer-to-right-edge

`emacspeak-eterm-pointer-to-right-edge` [Command]

Move the pointer to the right edge.

12.71.1.19 emacspeak-eterm-pointer-to-top

`emacspeak-eterm-pointer-to-top` [Command]

Move the pointer to the top of the screen.

12.71.1.20 emacspeak-eterm-pointer-up

`emacspeak-eterm-pointer-up` (*count*) [Command]

Move the pointer up a line.
Argument COUNT .specifies number of lines by which to move.

(fn COUNT)

12.71.1.21 emacspeak-eterm-search-backward

`emacspeak-eterm-search-backward` [Command]

Search backward on the terminal.

12.71.1.22 emacspeak-eterm-search-forward

`emacspeak-eterm-search-forward` [Command]

Search forward on the terminal.

12.71.1.23 emacspeak-eterm-set-filter-window

`emacspeak-eterm-set-filter-window` (*flag*) [Command]

Prompt for the id of a predefined window,
and set the 'filter' window to it.
Non-nil interactive prefix arg 'unsets' the filter window;
this is equivalent to having the entire terminal as the filter window (this is what eterm starts up with).
Setting the filter window results in emacspeak only monitoring screen
activity within the filter window.

(fn FLAG)

12.71.1.24 emacspeak-eterm-set-focus-window

`emacspeak-eterm-set-focus-window` (*flag*) [Command]

Prompt for the id of a predefined window,
and set the 'focus' window to it.
Non-nil interactive prefix arg 'unsets' the focus window;
this is equivalent to having the entire terminal as the focus window (this is what eterm starts up with).
Setting the focus window results in emacspeak monitoring screen
and speaking that window upon seeing screen activity.

(fn FLAG)

12.71.1.25 emacspeak-eterm-set-marker

`emacspeak-eterm-set-marker` [Command]

Set Emacspeak eterm marker.
This sets the emacspeak eterm marker to the position pointed to by the emacspeak eterm pointer.

12.71.1.26 emacspeak-eterm-speak-cursor

`emacspeak-eterm-speak-cursor` [Command]

Speak cursor position.

12.71.1.27 emacspeak-eterm-speak-pointer

`emacspeak-eterm-speak-pointer` [Command]

Speak current pointer position.

12.71.1.28 emacspeak-eterm-speak-pointer-char

`emacspeak-eterm-speak-pointer-char` (**&optional** *prefix*) [Command]

Speak char under eterm pointer.
Pronounces character phonetically unless called with a PREFIX arg.

(fn &optional PREFIX)

12.71.1.29 emacspeak-eterm-speak-pointer-line

`emacspeak-eterm-speak-pointer-line` [Command]

Speak the line the pointer is on.

12.71.1.30 emacspeak-eterm-speak-pointer-word

`emacspeak-eterm-speak-pointer-word` [Command]

Speak the word the pointer is on.

12.71.1.31 emacspeak-eterm-speak-predefined-window

`emacspeak-eterm-speak-predefined-window` [Command]

Speak a predefined eterm window between 1 and 10.

12.71.1.32 emacspeak-eterm-speak-screen

`emacspeak-eterm-speak-screen` (**&optional** *flag*) [Command]

Speak the screen. Default is to speak from the emacspeak pointer to point.
Optional prefix arg FLAG causes region above
the Emacspeak pointer to be spoken.

(fn &optional FLAG)

12.71.1.33 emacspeak-eterm-speak-window

`emacspeak-eterm-speak-window` (*id*) [Command]

Speak an eterm window.
Argument ID specifies the window.

(fn ID)

12.71.1.34 emacspeak-eterm-toggle-filter-window

`emacspeak-eterm-toggle-filter-window` [Command]

Toggle active state of filter window.

12.71.1.35 emacspeak-eterm-toggle-focus-window

`emacspeak-eterm-toggle-focus-window` [Command]

Toggle active state of focus window.

12.71.1.36 emacspeak-eterm-toggle-pointer-mode

`emacspeak-eterm-toggle-pointer-mode` (**&optional** *prefix*) [Command]

Toggle emacspeak eterm pointer mode.
With optional interactive prefix arg, turn it on.
When emacspeak eterm is in pointer mode, the eterm read pointer
stays where it is rather than automatically moving to the terminal cursor when
there is terminal activity.

12.71.1.37 emacspeak-eterm-toggle-review

`emacspeak-eterm-toggle-review` [Command]

Toggle state of eterm review.
In review mode, you can move around the terminal and listen to the contents
without sending input to the terminal itself.

12.71.1.38 emacspeak-eterm-yank-window

`emacspeak-eterm-yank-window` (*id*) [Command]

Yank contents of an eterm window at point.

(fn ID)

12.71.1.39 emacspeak-toggle-eterm-autospeak

`emacspeak-toggle-eterm-autospeak` (**&optional** *prefix*) [Command]

Toggle state of eterm autospeak.
When eterm autospeak is turned on and the terminal is in line mode,
all output to the terminal is automatically spoken.
Interactive prefix arg means toggle the global default value, and then set the
current local value to the result.

12.71.2 emacspeak-eterm Options

`User Option` *voice-term-underline* [Variable]

Personality used for term-underline This personality uses voice-brighten-medium whose effect can be changed globally by customizing voice-brighten-medium-settings.

12.72 emacspeak-eudc

EUDC –Emacs Universal Directory Client provides a unified interface to directory servers e.g. ldap servers this module speech enables eudc

12.72.1 Emacspeak-Eudc Commands

12.72.1.1 emacspeak-eudc-send-mail

`emacspeak-eudc-send-mail` [Command]

Send email to the address given by the current record.

12.73 emacspeak-evil

EVIL == VIM In Emacs This is work-in-progress and is not complete.

12.73.1 Emacspeak-Evil Commands

12.73.1.1 emacspeak-evil-toggle-evil

`emacspeak-evil-toggle-evil` [Command]

Interactively toggle evil-mode.

12.73.2 emacspeak-evil Options

`User Option` *voice-evil-ex-commands* [Variable]

Personality used for evil-ex-commands This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-evil-ex-info* [Variable]

Personality used for evil-ex-info This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-evil-ex-lazy-highlight* [Variable]

Personality used for evil-ex-lazy-highlight This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

`User Option` *voice-evil-ex-search* [Variable]

Personality used for evil-ex-search This personality uses voice-bolden-and-animate whose effect can be changed globally by customizing voice-bolden-and-animate-settings.

`User Option` *voice-evil-ex-substitute-matches* [Variable]

Personality used for evil-ex-substitute-matches This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-evil-ex-substitute-replacement* [Variable]

Personality used for evil-ex-substitute-replacement This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

12.74 emacspeak-eww

EWW == Emacs Web Browser

EWW is a light-weight Web browser built into Emacs starting with Emacs-24.4 . This module speech-enables EWW.

It implements additional interactive commands for navigating the DOM. It also provides a set of filters for interactively filtering the DOM by various attributes such as id, class and role. Finally, this module updates EWW's built-in key-bindings with Emacspeak conveniences.

12.74.1 Structured Navigation

These commands move through section headers as defined in HTML.

1 `emacspeak-eww-next-h1` Move to next `H1` heading.

2 `emacspeak-eww-next-h2` Move to next `H2` heading.

3 `emacspeak-eww-next-h3` Move to next `H3` heading.

4 `emacspeak-eww-next-h4` Move to next `H4` heading.

. `emacspeak-eww-next-h` Move to next heading. (`H1`...`H4`).

M-1 `emacspeak-eww-previous-h1` Move to previous `H1` heading.

M-2 `emacspeak-eww-previous-h2` Move to previous `H2` heading.

M-3 `emacspeak-eww-previous-h3` Move to previous `H3` heading.

M-4 `emacspeak-eww-previous-h4` Move to previous `H4` heading.

, `emacspeak-eww-previous-h` Move to previous heading (`H1`...`H4`).

This next set of DOM commands enable navigating by HTML elements.

M-SPC `emacspeak-eww-speak-this-element` Speak contents of current element.

J `emacspeak-eww-next-element-like-this` Jump to next element that is the same as the one under point. If there are multiple HTML elements under point, prompts for element-name using completion.

K `emacspeak-eww-previous-element-like-this` Jump to previous element that is the same as the one under point. If there are multiple HTML elements under point, prompts for element-name using completion.

N `emacspeak-eww-next-element-from-history` Jump to next element based on previous J/K command history.

P `emacspeak-eww-previous-element-from-history` Jump to previous element based on previous J/K history.

O `emacspeak-eww-previous-li` Jump to previous list item.

o `emacspeak-eww-next-li` Jump to next list item.

T `emacspeak-eww-previous-table` Jump to previous table in page.

t `emacspeak-eww-next-table` Jump to next table.

[`emacspeak-eww-previous-p` Jump to previous paragraph.

] `emacspeak-eww-next-p` Jump to next paragraph.

b `shr-previous-link` Jump to previous link.

f `shr-next-link` Jump to next link.

n `emacspeak-eww-next-element` Jump to next element.

p `emacspeak-eww-previous-element` Jump to previous element.

s `eww-readable` Use EWW's built-in readable tool.

: `emacspeak-eww-tags-at-point` Display currently active HTML tags at point.

12.74.2 Filtering Content Using The DOM

These commands use EWW's HTML DOM to display different filtered views of the Web page. With an interactive prefix argument, these commands prompt for a list of filters. Command `emacspeak-eww-restore` bound to *DEL* can be used to restore the previous view.

A `eww-view-dom-having-attribute` Display DOM nodes having specified attribute. Valid attributes are available via completion.

C `eww-view-dom-having-class` Display DOM nodes having specified class. Valid classes are available via completion.

E `eww-view-dom-having-elements` Display specified elements from the Dom. Valid element names are available via completion.

I `eww-view-dom-having-id` Display DOM nodes having specified ID. Valid id values are available via completion.

R `eww-view-dom-having-role` Display DOM nodes having specified role. Valid roles are available via completion.

M-a `eww-view-dom-not-having-attribute` Filter out DOM nodes having specified attribute. Valid attribute values are available via completion.

M-c `eww-view-dom-not-having-class` Filter out DOM nodes having specified class. Valid class values are available via completion.

M-e `eww-view-dom-not-having-elements` Filter out specified element DOM nodes. Valid element names are available via completion.

M-i `eww-view-dom-not-having-id` Dfilter out Display DOM nodes having specified ID. Valid id values are available via completion.

M-r `eww-view-dom-not-having-role` Filter out DOM nodes having specified role. Valid role values are available via completion.

12.74.3 Diving Into (Focusing) On Specific Content

Contrast this with filtering described in the previous section. There, we discussed commands that *filter* the DOM to render specific types of elements. For HTML as spoken on the Web, there is a separate use-case that is helpful as a dual to filtering, namely, displaying a specific portion of a page, typically the contents of a `div` element. These elements often appear many times on a page, and can be deeply nested, making it difficult to focus on the relevant content on the page, e.g. news sites. Commands `emacspeak-eww-dive-into-div` help in such cases, *`C-d`* renders the `div` containing point in a separate buffer As with the filtering commands, *`l`* returns to the buffer where these commands were executed. Long-term users of Emacspeak who still remember Emacs-W3 will recognize this as the *focus* command implemented by Emacspeak for W3.

12.74.4 Updated Commands For Following Links

These key-bindings are available when point is on a link. They enable context-specific actions for following links, e.g., to play media streams, or to open various feed-types such as `ATOM`, `RSS`, or `OPML`.

`k` `shr-copy-url` Copy URL under point to the kill-ring.

`;` `emacspeak-eww-play-media-at-point` Play media URL under point using `emacs-m-player`.

`U` `emacspeak-eww-curl-play-media-at-point` Play media url under point by first downloading the URL using CURL. This is useful for sites that do multiple redirects before returning the actual media stream URL.

`C-o` `emacspeak-feeds-opml-display` Display link under point as an `OPML` feed .

`C-r` `emacspeak-feeds-rss-display` Display link under point as an `RSS` feed.

`C-a` `emacspeak-feeds-atom-display` Display link under point as an `ATOM` feed.

`y` `emacspeak-m-player-youtube-player` Play link under point as a Youtube stream.

12.74.5 Table Browsing

Summary Of Keyboard Commands:

- *`M-<left>`* emacspeak-eww-table-previous-cell— Speak previous cell.
- *`M-<right>`* emacspeak-eww-table-next-cell — Speak previous cell.
- *`M-<up>`* emacspeak-eww-table-previous-row — Speak cell above.
- *`M-<down>`* emacspeak-eww-table-next-row — Speak cell below.
- *`M-.`* emacspeak-eww-table-speak-cell — Speak current cell.
- *`M-,`* emacspeak-eww-table-speak-dimensions — Speak number of rows and columns.
- *`C-t`* emacspeak-eww-table-data — Browse this table in Emacspeak's Table UI — See Section 12.181 [emacspeak-table-ui], page 268.

Emacspeak EWW supports table navigation via keys *`M-.`*, *`M-LEFT`* and *`M-RIGHT`*, to speak the current, previous and next table cell respectively. The latter commands also move to the cell being spoken. You can get a sense of the table's size via *M-,* which speaks the number of rows and cells in the table. This works for plain tables, not nested tables; for nested tables, first have then *unnested* using one of the XSLT transforms like `sort-tables`.

12.74.6 Miscellaneous Commands

C-RET `emacspeak-eww-fillin-form-field` When on an input field, insert username/password information accessed via auth-source.

' `emacspeak-speak-rest-of-buffer` Speak rest of current Web page starting from point.

*** `eww-add-bookmark` Bookmark current Web page.

= `dtk-toggle-punctuation-mode`
Toggle punctuation mode.

? `emacspeak-google-similar-to-this-page` Google similarity search.

C-t

G `emacspeak-google-command`
Prefix key to invoke Google-specific commands.

L `emacspeak-eww-links-rel` Display any related links discovered via the document's `meta` tag.

Q `emacspeak-kill-buffer-quietly` Delete this buffer.

V `eww-view-source` Display Web page source.

e `emacspeak-we-xsl-map` Prefix key for invoking XSLT-based filters.

k `eww-copy-page-url` Copy page URL to kill-ring.

In addition, see commands in See Section 12.90 [emacspeak-google], page 154, for Google-Search specific commands, many of which are available via prefix-key *G*.

12.74.7 Filtering Content Using XSLT And XPath

C-c `emacspeak-we-junk-by-class-list` Prompts for list of class-names with completion, and filters out matching elements.

C-f `emacspeak-we-count-matches` Prompts for XPath expression, and returns count of matching elements.

C-p `emacspeak-we-xpath-junk-and-follow` Follows link under point, and displays that page after filtering by a specified XPath expression.

C-t `emacspeak-we-count-tables` Display a count of tables in the page.

C-x `emacspeak-we-count-nested-tables` Counts nested tables.

C `emacspeak-we-extract-by-class-list` Prompts for a list of class-names, and displays matching elements.

D `emacspeak-we-junk-by-class-list` Filters out elements having specified class attributes.

I `emacspeak-we-extract-by-id-list` Extracts elements by specified list of ID values.

M `emacspeak-we-extract-tables-by-match-list` Extracts tables that match specified selection pattern.

P `emacspeak-we-follow-and-extract-main` Follows link under point, and extracts readable content, by default, this is all paragraphs and headings.

S `emacspeak-we-style-filter` Filters content by style attribute.

T `emacspeak-we-extract-tables-by-position-list` Extracts tables by their position on the page.

X `emacspeak-we-extract-nested-table-list` Extracts nested tables.

a `emacspeak-we-xslt-apply` Prompt for and apply specified XSLT transform to current page.

b `emacspeak-we-follow-and-filter-by-id` Follow link under point, and filter by specified id value.

c `emacspeak-we-extract-by-class` Extracts elements by class.

d `emacspeak-we-junk-by-class` Filters out elements having specified class value.

e `emacspeak-we-url-expand-and-execute` Follow link under point, but pass the result to a custom executor.

f `emacspeak-we-xslt-filter` Apply a specified XSLT filter (XPath) to current page.

i `emacspeak-we-extract-by-id` Extract elements by id value.

j `emacspeak-we-xslt-junk` Filter out elements matching specified pattern.

k `emacspeak-we-toggle-xsl-keep-result` Debugging tool — retains the HTML source after XSLT.

m `emacspeak-we-extract-table-by-match` Extract matching table.

p `emacspeak-we-xpath-filter-and-follow` Follow link under point, and filter results by a specified XPath filter.

r `emacspeak-we-extract-by-role` Extract elements by specified role value.

s `emacspeak-we-xslt-select` Select default XSLT transform that is applied before rendering the page.

t `emacspeak-we-extract-table-by-position` Extracts tables by their position on the page.

u `emacspeak-we-extract-matching-urls` Display matching links on the page.

v `emacspeak-we-class-filter-and-follow-link` Follow link under point, and filter by specified class value.

w `emacspeak-we-extract-by-property` Extract element using a combination of DOM attributes.

x `emacspeak-we-extract-nested-table` Extract a nested table using a match-list.

y `emacspeak-we-class-filter-and-follow` Follow link under point and filter by class values.

12.74.8 EWW And EBooks On The Emacspeak Audio Desktop

Modules emacspeak-epub and emacspeak-bookshare provide EBook front-ends to EPub-2 and Daisy EBooks. Both modules now use EWW to render these EBooks. Module emacspeak-eww provides a simple bookmarking facility — called eww-marks (to avoid confusion with EWW's Web Bookmarks). When reading an EBook, you can use `m` to create an EWW-mark at that position. These marks are automatically saved across Emacs sessions. To open a previously created eww-mark, use command `emacspeak-eww-open-mark` bound to `C-x r e`. This command reads a eww-mark name with completion. Use this command with an interactive prefix arg to delete a previously created eww-mark.

12.74.9 Emacspeak-Eww Commands

12.74.9.1 emacspeak-eww-add-mark

`emacspeak-eww-add-mark` (*name*) [Command]

Interactively add a mark with name title+'name' at current position.

(fn NAME)

12.74.9.2 emacspeak-eww-curl-play-media-at-point

`emacspeak-eww-curl-play-media-at-point` [Command]

Use Curl to pull a URL, then pass
the first line to MPlayer as a playlist.
Useful in handling double-redirect from TuneIn.

12.74.9.3 emacspeak-eww-delete-mark

`emacspeak-eww-delete-mark` (*name*) [Command]

Interactively delete a mark with name 'name' at current position.

(fn NAME)

12.74.9.4 emacspeak-eww-dive-into-div

`emacspeak-eww-dive-into-div` [Command]

Focus on current div by rendering it in a new buffer.

12.74.9.5 emacspeak-eww-fillin-form-field

`emacspeak-eww-fillin-form-field` [Command]

Fill in user or passwd field using auth-source backend.

12.74.9.6 emacspeak-eww-google-knowledge-card

`emacspeak-eww-google-knowledge-card` [Command]

Show just the knowledge card.

Warning, this is fragile, and depends on a stable id/class for the knowledge card.

12.74.9.7 emacspeak-eww-links-rel

`emacspeak-eww-links-rel` [Command]

Display Link tags of type rel. Web pages for which alternate links are available are cued by an auditory icon on the header line.

12.74.9.8 emacspeak-eww-marks-load

`emacspeak-eww-marks-load` [Command]

Load saved marks.

12.74.9.9 emacspeak-eww-marks-save

`emacspeak-eww-marks-save` [Command]

Save Emacspeak EWW marks.

12.74.9.10 emacspeak-eww-masquerade

`emacspeak-eww-masquerade` [Command]

Toggle masquerade state.

12.74.9.11 emacspeak-eww-next-dd

`emacspeak-eww-next-dd` (**&optional** *speak*) [Command]

Move forward to the next dd.

Optional interactive prefix arg speaks the dd.

The dd is automatically spoken if there is no user activity.

12.74.9.12 emacspeak-eww-next-dl

`emacspeak-eww-next-dl` (**&optional** *speak*) [Command]

Move forward to the next dl.

Optional interactive prefix arg speaks the dl.

The dl is automatically spoken if there is no user activity.

12.74.9.13 emacspeak-eww-next-dt

`emacspeak-eww-next-dt` (**&optional** *speak*) [Command]

Move forward to the next dt.

Optional interactive prefix arg speaks the dt.

The dt is automatically spoken if there is no user activity.

12.74.9.14 emacspeak-eww-next-element

`emacspeak-eww-next-element` (*el*) [Command]

Move forward to the next specified element.

(fn EL)

12.74.9.15 emacspeak-eww-next-element-from-history

`emacspeak-eww-next-element-from-history` [Command]

Uses element navigation history to decide where we jump.

12.74.9.16 emacspeak-eww-next-element-like-this

`emacspeak-eww-next-element-like-this` (*element*) [Command]

Moves to next element like current.
Prompts if content at point is enclosed by multiple elements.

(fn ELEMENT)

12.74.9.17 emacspeak-eww-next-h

`emacspeak-eww-next-h` (**&optional** *speak*) [Command]

Move forward to the next h.
Optional interactive prefix arg speaks the h.
The h is automatically spoken if there is no user activity.

12.74.9.18 emacspeak-eww-next-h1

`emacspeak-eww-next-h1` (**&optional** *speak*) [Command]

Move forward to the next h1.
Optional interactive prefix arg speaks the h1.
The h1 is automatically spoken if there is no user activity.

12.74.9.19 emacspeak-eww-next-h2

`emacspeak-eww-next-h2` (**&optional** *speak*) [Command]

Move forward to the next h2.
Optional interactive prefix arg speaks the h2.
The h2 is automatically spoken if there is no user activity.

12.74.9.20 emacspeak-eww-next-h3

`emacspeak-eww-next-h3` (**&optional** *speak*) [Command]

Move forward to the next h3.
Optional interactive prefix arg speaks the h3.
The h3 is automatically spoken if there is no user activity.

12.74.9.21 emacspeak-eww-next-h4

`emacspeak-eww-next-h4` (**&optional** *speak*) [Command]

Move forward to the next h4.
Optional interactive prefix arg speaks the h4.
The h4 is automatically spoken if there is no user activity.

12.74.9.22 emacspeak-eww-next-h5

`emacspeak-eww-next-h5` (**&optional** *speak*) [Command]

Move forward to the next h5.
Optional interactive prefix arg speaks the h5.
The h5 is automatically spoken if there is no user activity.

12.74.9.23 emacspeak-eww-next-h6

`emacspeak-eww-next-h6` (**&optional** *speak*) [Command]

Move forward to the next h6.
Optional interactive prefix arg speaks the h6.
The h6 is automatically spoken if there is no user activity.

12.74.9.24 emacspeak-eww-next-li

`emacspeak-eww-next-li` (**&optional** *speak*) [Command]

Move forward to the next li.
Optional interactive prefix arg speaks the li.
The li is automatically spoken if there is no user activity.

12.74.9.25 emacspeak-eww-next-ol

`emacspeak-eww-next-ol` (**&optional** *speak*) [Command]

Move forward to the next ol.
Optional interactive prefix arg speaks the ol.
The ol is automatically spoken if there is no user activity.

12.74.9.26 emacspeak-eww-next-p

`emacspeak-eww-next-p` (**&optional** *speak*) [Command]

Move forward to the next p.
Optional interactive prefix arg speaks the p.
The p is automatically spoken if there is no user activity.

12.74.9.27 emacspeak-eww-next-table

`emacspeak-eww-next-table` (**&optional** *speak*) [Command]

Move forward to the next table.
Optional interactive prefix arg speaks the table.
The table is automatically spoken if there is no user activity.

12.74.9.28 emacspeak-eww-next-ul

`emacspeak-eww-next-ul` (**&optional** *speak*) [Command]

Move forward to the next ul.
Optional interactive prefix arg speaks the ul.
The ul is automatically spoken if there is no user activity.

12.74.9.29 emacspeak-eww-open-mark

`emacspeak-eww-open-mark` (*name* **&optional** *delete*) [Command]

`C-x r e`

Open specified EWW marked location. If the content is already being displayed in this Emacs session, jump to it directly. With optional interactive prefix arg 'delete', delete that mark instead.

(fn NAME &optional DELETE)

12.74.9.30 emacspeak-eww-play-media-at-point

`emacspeak-eww-play-media-at-point` (**&optional** *playlist-p*) [Command]

`C-e M-;`

`<fn> M-;`

Play media url under point.
Optional interactive prefix arg 'playlist-p' says to treat the
link as a playlist. A second interactive prefix arg adds
mplayer option -allow-dangerous-playlist-parsing

(fn &optional PLAYLIST-P)

12.74.9.31 emacspeak-eww-previous-dd

`emacspeak-eww-previous-dd` (**&optional** *speak*) [Command]

Move backward to the next dd.
Optional interactive prefix arg speaks the dd.
The dd is automatically spoken if there is no user activity.

12.74.9.32 emacspeak-eww-previous-dl

`emacspeak-eww-previous-dl` (**&optional** *speak*) [Command]

Move backward to the next dl.
Optional interactive prefix arg speaks the dl.
The dl is automatically spoken if there is no user activity.

12.74.9.33 emacspeak-eww-previous-dt

`emacspeak-eww-previous-dt` (**&optional** *speak*) [Command]

Move backward to the next dt.
Optional interactive prefix arg speaks the dt.
The dt is automatically spoken if there is no user activity.

12.74.9.34 emacspeak-eww-previous-element

`emacspeak-eww-previous-element` (*el*) [Command]

Move backward to the previous specified element.

(fn EL)

12.74.9.35 emacspeak-eww-previous-element-from-history

`emacspeak-eww-previous-element-from-history` [Command]

Uses element navigation history to decide where we jump.

12.74.9.36 emacspeak-eww-previous-element-like-this

`emacspeak-eww-previous-element-like-this` (*element*) [Command]

Moves to next element like current.
Prompts if content at point is enclosed by multiple elements.

(fn ELEMENT)

12.74.9.37 emacspeak-eww-previous-h

`emacspeak-eww-previous-h` (**&optional** *speak*) [Command]

Move backward to the next h.
Optional interactive prefix arg speaks the h.
The h is automatically spoken if there is no user activity.

12.74.9.38 emacspeak-eww-previous-h1

`emacspeak-eww-previous-h1` (**&optional** *speak*) [Command]

Move backward to the next h1.
Optional interactive prefix arg speaks the h1.
The h1 is automatically spoken if there is no user activity.

12.74.9.39 emacspeak-eww-previous-h2

`emacspeak-eww-previous-h2` (**&optional** *speak*) [Command]

Move backward to the next h2.
Optional interactive prefix arg speaks the h2.
The h2 is automatically spoken if there is no user activity.

12.74.9.40 emacspeak-eww-previous-h3

`emacspeak-eww-previous-h3` (**&optional** *speak*) [Command]

Move backward to the next h3.
Optional interactive prefix arg speaks the h3.
The h3 is automatically spoken if there is no user activity.

12.74.9.41 emacspeak-eww-previous-h4

`emacspeak-eww-previous-h4` (**&optional** *speak*) [Command]

Move backward to the next h4.
Optional interactive prefix arg speaks the h4.
The h4 is automatically spoken if there is no user activity.

12.74.9.42 emacspeak-eww-previous-h5

`emacspeak-eww-previous-h5` (**&optional** *speak*) [Command]

Move backward to the next h5.
Optional interactive prefix arg speaks the h5.
The h5 is automatically spoken if there is no user activity.

12.74.9.43 emacspeak-eww-previous-h6

`emacspeak-eww-previous-h6` (**&optional** *speak*) [Command]

Move backward to the next h6.
Optional interactive prefix arg speaks the h6.
The h6 is automatically spoken if there is no user activity.

12.74.9.44 emacspeak-eww-previous-li

`emacspeak-eww-previous-li` (**&optional** *speak*) [Command]

Move backward to the next li.
Optional interactive prefix arg speaks the li.
The li is automatically spoken if there is no user activity.

12.74.9.45 emacspeak-eww-previous-ol

`emacspeak-eww-previous-ol` (**&optional** *speak*) [Command]

Move backward to the next ol.
Optional interactive prefix arg speaks the ol.
The ol is automatically spoken if there is no user activity.

12.74.9.46 emacspeak-eww-previous-p

`emacspeak-eww-previous-p` (**&optional** *speak*) [Command]

Move backward to the next p.
Optional interactive prefix arg speaks the p.
The p is automatically spoken if there is no user activity.

12.74.9.47 emacspeak-eww-previous-table

`emacspeak-eww-previous-table` (**&optional** *speak*) [Command]

Move backward to the next table.
Optional interactive prefix arg speaks the table.
The table is automatically spoken if there is no user activity.

12.74.9.48 emacspeak-eww-previous-ul

`emacspeak-eww-previous-ul` (**&optional** *speak*) [Command]

Move backward to the next ul.
Optional interactive prefix arg speaks the ul.
The ul is automatically spoken if there is no user activity.

12.74.9.49 emacspeak-eww-reading-settings

`emacspeak-eww-reading-settings` [Command]

Setup speech-rate, punctuation and split-caps for reading prose.

12.74.9.50 emacspeak-eww-restore

`emacspeak-eww-restore` [Command]

Restore buffer to pre-filtered canonical state.

12.74.9.51 emacspeak-eww-shell-command-on-url-at-point

`emacspeak-eww-shell-command-on-url-at-point` (**&optional** *prefix*) [Command]

Run specified shell command on URL at point.
Warning: Running shell script cbox through this fails mysteriously.

(fn &optional PREFIX)

12.74.9.52 emacspeak-eww-smart-tabs

`emacspeak-eww-smart-tabs` (*char* **&optional** *define*) [Command]

C-, SPC

C-x @ a *SPC*
Open URL in EWW keyed by 'char'.
To associate a URL with a char, use this command
with an interactive prefix arg.

(fn CHAR &optional DEFINE)

12.74.9.53 emacspeak-eww-smart-tabs-add

`emacspeak-eww-smart-tabs-add` (*char url*) [Command]

Add a URL to the specified location in smart tabs.

(fn CHAR URL)

12.74.9.54 emacspeak-eww-smart-tabs-load

`emacspeak-eww-smart-tabs-load` [Command]

Load our smart tabsfrom a file.

12.74.9.55 emacspeak-eww-smart-tabs-save

`emacspeak-eww-smart-tabs-save` [Command]

Save our smart tabs to a file for reloading.

12.74.9.56 emacspeak-eww-speak-this-element

`emacspeak-eww-speak-this-element` (*element*) [Command]

Speaks to next element like current.
Uses most recently navigated structural unit.
Otherwise, prompts if content at point is enclosed by multiple elements.

(fn ELEMENT)

12.74.9.57 emacspeak-eww-table-data

`emacspeak-eww-table-data` [Command]

View table at point as a data table using Emacspeak Table UI.

12.74.9.58 emacspeak-eww-table-next-cell

`emacspeak-eww-table-next-cell` (**&optional** *prefix*) [Command]

Speak next cell after making it current.
Interactive prefix arg moves to the last cell in the table.

(fn &optional PREFIX)

12.74.9.59 emacspeak-eww-table-next-row

`emacspeak-eww-table-next-row` (**&optional** *prefix*) [Command]

Speak cell after moving to next row.
Optional interactive prefix arg moves to end of table.

(fn &optional PREFIX)

12.74.9.60 emacspeak-eww-table-previous-cell

`emacspeak-eww-table-previous-cell` (**&optional** *prefix*) [Command]

Speak previous cell after making it current.
With interactive prefix arg, move to the start of the table.

(fn &optional PREFIX)

12.74.9.61 emacspeak-eww-table-previous-row

`emacspeak-eww-table-previous-row` (**&optional** *prefix*) [Command]

Speak cell after moving to previous row.
Optional interactive prefix arg moves to start of table.

(fn &optional PREFIX)

12.74.9.62 emacspeak-eww-table-speak-cell

`emacspeak-eww-table-speak-cell` [Command]

Speak current cell.

12.74.9.63 emacspeak-eww-table-speak-dimensions

`emacspeak-eww-table-speak-dimensions` [Command]

Speak number of rows and cells.

12.74.9.64 emacspeak-eww-tags-at-point

`emacspeak-eww-tags-at-point` [Command]

Display tags at point.

12.74.9.65 emacspeak-eww-update-title

`emacspeak-eww-update-title` (*title*) [Command]

Interactively set title — renames buffer, and sets header-line.

(fn TITLE)

12.74.10 emacspeak-eww Options

`User Option` *emacspeak-eww-inhibit-images* [Variable]

Turn this on to avoid rendering images.

`User Option` *voice-eww-form-checkbox* [Variable]

Personality used for eww-form-checkbox This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-eww-form-file* [Variable]

Personality used for eww-form-file This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-eww-form-select* [Variable]

Personality used for eww-form-select This personality uses voice-annotate whose effect can be changed globally by customizing voice-annotate-settings.

`User Option` *voice-eww-form-submit* [Variable]

Personality used for eww-form-submit This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

`User Option` *voice-eww-form-text* [Variable]

Personality used for eww-form-text This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-eww-form-textarea* [Variable]

Personality used for eww-form-textarea This personality uses voice-brighten whose effect can be changed globally by customizing voice-brighten-settings.

`User Option` *voice-eww-invalid-certificate* [Variable]

Personality used for eww-invalid-certificate This personality uses voice-bolden-and-animate whose effect can be changed globally by customizing voice-bolden-and-animate-settings.

`User Option` *voice-eww-valid-certificate* [Variable]

Personality used for eww-valid-certificate This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-shr-selected-link* [Variable]

Personality used for shr-selected-link This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

`User Option` *voice-shr-strike-through* [Variable]

Personality used for shr-strike-through This personality uses voice-annotate whose effect can be changed globally by customizing voice-annotate-settings.

12.75 emacspeak-feeds

This module provides Feeds support for Emacspeak

12.75.1 Emacspeak-Feeds Commands

12.75.1.1 emacspeak-feeds-add-feed

`emacspeak-feeds-add-feed` (*title url type*) [Command]

C-e M-u

<fn> M-u

Add specified feed to our feed store.

(fn TITLE URL TYPE)

12.75.1.2 emacspeak-feeds-archive-feeds

`emacspeak-feeds-archive-feeds` [Command]

Archive list of subscribed fees to personal resource directory.
Archiving is useful when synchronizing feeds across multiple machines.

12.75.1.3 emacspeak-feeds-atom-display

`emacspeak-feeds-atom-display` (*feed-url*) [Command]

C-, a

C-x @ a a

Display ATOM feed.

(fn FEED-URL)

12.75.1.4 emacspeak-feeds-browse

`emacspeak-feeds-browse` (*feed*) [Command]

C-e C-u

C-; C-u

<fn> C-u

C-x @ h C-u
Browse specified feed.

(fn FEED)

12.75.1.5 emacspeak-feeds-fastload-feeds

`emacspeak-feeds-fastload-feeds` [Command]
Fast load list of feeds from archive.
This directly updates emacspeak-feeds from the archive, rather than adding those entries to the current set of subscribed feeds.

12.75.1.6 emacspeak-feeds-opml-display

`emacspeak-feeds-opml-display` (*feed-url*) [Command]

C-, o

C-x @ a o

Display OPML feed.

(fn FEED-URL)

12.75.1.7 emacspeak-feeds-restore-feeds

`emacspeak-feeds-restore-feeds` [Command]
Restore list of subscribed fees from personal resource directory.
Archiving is useful when synchronizing feeds across multiple machines.

12.75.1.8 emacspeak-feeds-rss-display

`emacspeak-feeds-rss-display` (*feed-url*) [Command]

C-, r

C-x @ a r

Display RSS feed.

(fn FEED-URL)

12.75.2 emacspeak-feeds Options

`User Option` *emacspeak-feeds* [Variable]
Table of RSS/Atom feeds. The feed list is persisted to file saved-feeds on exit.

12.76 emacspeak-filtertext

It is often useful to view the results of filtering large amounts of text.;;; Typically you do this with various combinations of grep and friends. When doing so it requires explicit effort to not destroy the original text being filtered. This module provides a textfilter utility that: A) Copies over the selected text to a special filtertext buffer B) Implements a filtertext mode for that buffer that allows easy application of filters C: Provides commands for reverting to the original unfiltered text D: Provides commands for saving results from intermediate filters.

12.76.1 Emacspeak-Filtertext Commands

12.76.1.1 emacspeak-filtertext

`emacspeak-filtertext` (*start end*) [Command]

> *C-e* ^
>
> *<fn>* ^
>
> Copy over text in region to special filtertext buffer in
> preparation for interactively filtering text.
>
> (fn START END)

12.76.1.2 emacspeak-filtertext-mode

`emacspeak-filtertext-mode` [Command]

> Major mode for FilterText interaction.
>
> key binding
> — ——-
>
> = keep-lines
> ^ flush-lines
> r emacspeak-filtertext-revert
>
> In addition to any hooks its parent mode ‘text-mode’ might have run,
> this mode runs the hook ‘emacspeak-filtertext-mode-hook’, as the final or penultimate step
> during initialization.

12.76.1.3 emacspeak-filtertext-revert

`emacspeak-filtertext-revert` [Command]

> Revert to original text.

12.77 emacspeak-fix-interactive

Emacs commands that use the ’interactive spec to read interactive arguments are a problem for Emacspeak. This is because the prompting for the arguments is done from C See (callint.c) in the Emacs sources. Advising the various input functions, e.g. read-file-name therefore will not help. This module defines a function that solves this problem. emacspeak-fix-commands-that-use-interactive needs to be called To speech enable such functions.

12.77.1 Emacspeak-Fix-Interactive Commands

12.77.1.1 emacspeak-fix-all-recent-commands

`emacspeak-fix-all-recent-commands` [Command]

Fix recently loaded interactive commands.
This command looks through 'load-history' and fixes commands if necessary.
Memoizes call in emacspeak-load-history-pointer to memoize this call.

12.77.1.2 emacspeak-fix-commands-loaded-from

`emacspeak-fix-commands-loaded-from` (*module*) [Command]

Fix all commands loaded from a specified module.

(fn MODULE)

12.78 emacspeak-flycheck

FLYCHECK == On-the-fly checking.

12.78.1 emacspeak-flycheck Options

`User Option` *voice-flycheck-error* [Variable]

Personality used for flycheck-error This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-flycheck-error-list-error* [Variable]

Personality used for flycheck-error-list-error This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-flycheck-error-list-highlight* [Variable]

Personality used for flycheck-error-list-highlight This personality uses voice-bolden-medium whose effect can be changed globally by customizing voice-bolden-medium-settings.

`User Option` *voice-flycheck-error-list-highlight-at-point* [Variable]

Personality used for flycheck-error-list-highlight-at-point This personality uses voice-bolden-extra whose effect can be changed globally by customizing voice-bolden-extra-settings.

`User Option` [Variable]
voice-flycheck-error-list-highlightflycheck-error-list-column-number

Personality used for flycheck-error-list-highlightflycheck-error-list-column-number This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-flycheck-error-list-info* [Variable]
Personality used for flycheck-error-list-info This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-flycheck-error-list-line-number* [Variable]
Personality used for flycheck-error-list-line-number This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-flycheck-error-list-warning* [Variable]
Personality used for flycheck-error-list-warning This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

`User Option` *voice-flycheck-info* [Variable]
Personality used for flycheck-info This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-flycheck-warning* [Variable]
Personality used for flycheck-warning This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

12.79 emacspeak-flymake

Speech-enable flymake

12.79.1 emacspeak-flymake Options

`User Option` *voice-flymake-error* [Variable]
Personality used for flymake-error This personality uses voice-highlight whose effect can be changed globally by customizing voice-highlight-settings.

`User Option` *voice-flymake-note* [Variable]
Personality used for flymake-note This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

`User Option` *voice-flymake-warning* [Variable]
Personality used for flymake-warning This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

12.80 emacspeak-flyspell

This module speech enables flyspell. it loads flyspell-correct if available, And when loading flyspell-correct sets up that module to use one of three supported correction styles:

- ido: IDO-like completion with C-s and C-r moving through choices.
- popup:Use up and down arrows to move through corrections.
- helm: A helm interface for picking amongst corrections.

See documentation for package flyspell-correct for additional details.

Use Customization emacspeak-flyspell-correct to pick between ido, popup and helm.

12.80.1 emacspeak-flyspell Options

`User Option` *emacspeak-flyspell-correct* [Variable]
Correction style to use with flyspell.

`User Option` *voice-flyspell-duplicate* [Variable]
Personality used for flyspell-duplicate This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-flyspell-incorrect* [Variable]
Personality used for flyspell-incorrect This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

12.81 emacspeak-folding

Folding mode turns emacs into a folding editor. Folding mode is what I use: emacs 19 comes with similar packages, e.g. allout.el This module defines some advice forms that make folding mode a pleasure to use. Think of a fold as a container.

12.82 emacspeak-forge

FORGE == Work with Github, Gitlab etc from inside magit. This module speech-enables magit/forge.

12.82.1 emacspeak-forge Options

`User Option` *voice-forge-post-author* [Variable]
Personality used for forge-post-author This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-forge-post-date* [Variable]
Personality used for forge-post-date This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-forge-topic-closed* [Variable]
Personality used for forge-topic-closed This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

`User Option` *voice-forge-topic-merged* [Variable]
Personality used for forge-topic-merged This personality uses voice-brighten whose effect can be changed globally by customizing voice-brighten-settings.

`User Option` *voice-forge-topic-open* [Variable]
Personality used for forge-topic-open This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-forge-topic-unmerged* [Variable]
Personality used for forge-topic-unmerged This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

`User Option` *voice-forge-topic-unread* [Variable]
Personality used for forge-topic-unread This personality uses voice-highlight whose effect can be changed globally by customizing voice-highlight-settings.

12.83 emacspeak-forms

Provide additional advice to forms-mode

12.83.1 Emacspeak-Forms Commands

12.83.1.1 emacspeak-forms-find-file

`emacspeak-forms-find-file` (*filename*) [Command]

Visit a forms file

(fn FILENAME)

12.83.1.2 emacspeak-forms-flush-unwanted-records

`emacspeak-forms-flush-unwanted-records` [Command]

Prompt for pattern and flush matching lines

12.83.1.3 emacspeak-forms-rerun-filter

`emacspeak-forms-rerun-filter` [Command]

Rerun filter –allows us to nuke more matching records

12.83.1.4 emacspeak-forms-speak-field

`emacspeak-forms-speak-field` [Command]

Speak current form field name and value.
Assumes that point is at the front of a field value.

12.83.1.5 emacspeak-forms-summarize-current-position

`emacspeak-forms-summarize-current-position` [Command]

Summarize current position in list of records

12.83.1.6 emacspeak-forms-summarize-current-record

`emacspeak-forms-summarize-current-record` [Command]

Summarize current record

12.84 emacspeak-geiser

geiser.el — GNU Emacs and Scheme talk to each other This module speech-enables all interactive aspects of geiser, including the geiser->scheme REPL. This is used by racket-mode for racket interaction, And also for interacting with Guile.

12.85 emacspeak-gh-explorer

GH-EXPLORER == GitHub Explorer This module speech-enables Github Explorer.

12.85.1 Emacspeak-Gh-Explorer Commands

12.85.1.1 emacspeak-gh-explorer-next

`emacspeak-gh-explorer-next` [Command]

Move forward and speak current entry.

12.85.1.2 emacspeak-gh-explorer-previous

`emacspeak-gh-explorer-previous` [Command]

Moveback and speak current entry.

12.86 emacspeak-gnuplot

This module speech-enables gnuplot-mode an Emacs add-on that enables fluent interaction with gnuplot. Use gnuplot to generate plots of mathematical functions for inclusion in documents.

12.87 emacspeak-gnus

This module advises gnus to speak. Updating support in 2014 (Emacspeak is nearly 20 years old) Updating in 2018 as I switch to gnus as my primary mail interface. These customizations to gnus make it convenient to listen to news: You can read news mostly by using the four arrow keys. By default all article headers are hidden, so you hear the real news.

12.87.1 Emacspeak-Gnus Commands

12.87.1.1 emacspeak-gnus-personal-gmail-last-week

`emacspeak-gnus-personal-gmail-last-week` [Command]

Look for mail addressed personally in the last week.

12.87.1.2 emacspeak-gnus-personal-gmail-recent

`emacspeak-gnus-personal-gmail-recent` [Command]

Look for mail addressed personally in the last day.

12.87.1.3 emacspeak-gnus-summary-catchup-quietly-and-exit

`emacspeak-gnus-summary-catchup-quietly-and-exit` [Command]

Catch up on all articles in current group.

12.87.2 emacspeak-gnus Options

`User Option` *emacspeak-gnus-large-article* [Variable]

Articles having more than emacspeak-gnus-large-article lines will be considered to be a large article. A large article is not spoken all at once; instead you hear only the first screenful.

`User Option` *emacspeak-gnus-punctuation-mode* [Variable]

Pronunciation mode to use for gnus buffers.

`User Option` *voice-gnus-button* [Variable]
Personality used for gnus-button This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-gnus-cite-1* [Variable]
Personality used for gnus-cite-1 This personality uses voice-bolden-medium whose effect can be changed globally by customizing voice-bolden-medium-settings.

`User Option` *voice-gnus-cite-10* [Variable]
Personality used for gnus-cite-10 This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-gnus-cite-11* [Variable]
Personality used for gnus-cite-11 This personality uses voice-lighten-extra whose effect can be changed globally by customizing voice-lighten-extra-settings.

`User Option` *voice-gnus-cite-2* [Variable]
Personality used for gnus-cite-2 This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-gnus-cite-3* [Variable]
Personality used for gnus-cite-3 This personality uses voice-lighten-extra whose effect can be changed globally by customizing voice-lighten-extra-settings.

`User Option` *voice-gnus-cite-4* [Variable]
Personality used for gnus-cite-4 This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

`User Option` *voice-gnus-cite-5* [Variable]
Personality used for gnus-cite-5 This personality uses voice-smoothen-extra whose effect can be changed globally by customizing voice-smoothen-extra-settings.

`User Option` *voice-gnus-cite-6* [Variable]
Personality used for gnus-cite-6 This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-gnus-cite-7* [Variable]
Personality used for gnus-cite-7 This personality uses voice-lighten-extra whose effect can be changed globally by customizing voice-lighten-extra-settings.

`User Option` *voice-gnus-cite-8* [Variable]
Personality used for gnus-cite-8 This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-gnus-cite-9* [Variable]
Personality used for gnus-cite-9 This personality uses voice-bolden-medium whose effect can be changed globally by customizing voice-bolden-medium-settings.

`User Option` *voice-gnus-cite-attribution* [Variable]
Personality used for gnus-cite-attribution This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

`User Option` *voice-gnus-emphasis-bold* [Variable]
Personality used for gnus-emphasis-bold This personality uses voice-bolden-and-animate whose effect can be changed globally by customizing voice-bolden-and-animate-settings.

`User Option` *voice-gnus-emphasis-bold-italic* [Variable]
Personality used for gnus-emphasis-bold-italic This personality uses voice-bolden-medium whose effect can be changed globally by customizing voice-bolden-medium-settings.

`User Option` *voice-gnus-emphasis-highlight-words* [Variable]
Personality used for gnus-emphasis-highlight-words This personality uses voice-lighten-extra whose effect can be changed globally by customizing voice-lighten-extra-settings.

`User Option` *voice-gnus-emphasis-italic* [Variable]
Personality used for gnus-emphasis-italic This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-gnus-emphasis-strikethru* [Variable]
Personality used for gnus-emphasis-strikethru This personality uses voice-bolden-extra whose effect can be changed globally by customizing voice-bolden-extra-settings.

`User Option` *voice-gnus-emphasis-underline* [Variable]
Personality used for gnus-emphasis-underline This personality uses voice-brighten-extra whose effect can be changed globally by customizing voice-brighten-extra-settings.

`User Option` *voice-gnus-emphasis-underline-bold* [Variable]
Personality used for gnus-emphasis-underline-bold This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-gnus-emphasis-underline-bold-italic* [Variable]
Personality used for gnus-emphasis-underline-bold-italic This personality uses voice-lighten-extra whose effect can be changed globally by customizing voice-lighten-extra-settings.

`User Option` *voice-gnus-emphasis-underline-italic* [Variable]
Personality used for gnus-emphasis-underline-italic This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-gnus-group-mail-1* [Variable]
Personality used for gnus-group-mail-1 This personality uses voice-brighten whose effect can be changed globally by customizing voice-brighten-settings.

`User Option` *voice-gnus-group-mail-1-empty* [Variable]
Personality used for gnus-group-mail-1-empty This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-gnus-group-mail-2* [Variable]
Personality used for gnus-group-mail-2 This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

User Option *voice-gnus-group-mail-2-empty* [Variable]
Personality used for gnus-group-mail-2-empty This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

User Option *voice-gnus-group-mail-3* [Variable]
Personality used for gnus-group-mail-3 This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

User Option *voice-gnus-group-mail-3-empty* [Variable]
Personality used for gnus-group-mail-3-empty This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

User Option *voice-gnus-group-mail-low* [Variable]
Personality used for gnus-group-mail-low This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

User Option *voice-gnus-group-mail-low-empty* [Variable]
Personality used for gnus-group-mail-low-empty This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

User Option *voice-gnus-group-news-1* [Variable]
Personality used for gnus-group-news-1 This personality uses voice-highlight whose effect can be changed globally by customizing voice-highlight-settings.

User Option *voice-gnus-group-news-1-empty* [Variable]
Personality used for gnus-group-news-1-empty This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

User Option *voice-gnus-group-news-2* [Variable]
Personality used for gnus-group-news-2 This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

User Option *voice-gnus-group-news-2-empty* [Variable]
Personality used for gnus-group-news-2-empty This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

User Option *voice-gnus-group-news-3* [Variable]
Personality used for gnus-group-news-3 This personality uses voice-brighten whose effect can be changed globally by customizing voice-brighten-settings.

User Option *voice-gnus-group-news-3-empty* [Variable]
Personality used for gnus-group-news-3-empty This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

User Option *voice-gnus-group-news-4* [Variable]
Personality used for gnus-group-news-4 This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

User Option *voice-gnus-group-news-4-empty* [Variable]
Personality used for gnus-group-news-4-empty This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-gnus-group-news-5* [Variable]

Personality used for gnus-group-news-5 This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

`User Option` *voice-gnus-group-news-5-empty* [Variable]

Personality used for gnus-group-news-5-empty This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-gnus-group-news-6* [Variable]

Personality used for gnus-group-news-6 This personality uses voice-annotate whose effect can be changed globally by customizing voice-annotate-settings.

`User Option` *voice-gnus-group-news-6-empty* [Variable]

Personality used for gnus-group-news-6-empty This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-gnus-group-news-low* [Variable]

Personality used for gnus-group-news-low This personality uses voice-smoothen-extra whose effect can be changed globally by customizing voice-smoothen-extra-settings.

`User Option` *voice-gnus-group-news-low-empty* [Variable]

Personality used for gnus-group-news-low-empty This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-gnus-header-content* [Variable]

Personality used for gnus-header-content This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-gnus-header-from* [Variable]

Personality used for gnus-header-from This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-gnus-header-name* [Variable]

Personality used for gnus-header-name This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

`User Option` *voice-gnus-header-newsgroups* [Variable]

Personality used for gnus-header-newsgroups This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-gnus-header-subject* [Variable]

Personality used for gnus-header-subject This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-gnus-server-agent* [Variable]

Personality used for gnus-server-agent This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-gnus-server-closed* [Variable]

Personality used for gnus-server-closed This personality uses voice-bolden-medium whose effect can be changed globally by customizing voice-bolden-medium-settings.

`User Option` *voice-gnus-server-cloud* [Variable]

Personality used for gnus-server-cloud This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

`User Option` *voice-gnus-server-cloud-host* [Variable]

Personality used for gnus-server-cloud-host This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-gnus-server-denied* [Variable]

Personality used for gnus-server-denied This personality uses voice-bolden-extra whose effect can be changed globally by customizing voice-bolden-extra-settings.

`User Option` *voice-gnus-server-offline* [Variable]

Personality used for gnus-server-offline This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

`User Option` *voice-gnus-server-opened* [Variable]

Personality used for gnus-server-opened This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-gnus-signature* [Variable]

Personality used for gnus-signature This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

`User Option` *voice-gnus-splash* [Variable]

Personality used for gnus-splash This personality uses voice-brighten whose effect can be changed globally by customizing voice-brighten-settings.

`User Option` *voice-gnus-summary-cancelled* [Variable]

Personality used for gnus-summary-cancelled This personality uses voice-bolden-extra whose effect can be changed globally by customizing voice-bolden-extra-settings.

`User Option` *voice-gnus-summary-high-ancient* [Variable]

Personality used for gnus-summary-high-ancient This personality uses voice-smoothen-extra whose effect can be changed globally by customizing voice-smoothen-extra-settings.

`User Option` *voice-gnus-summary-high-read* [Variable]

Personality used for gnus-summary-high-read This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-gnus-summary-high-ticked* [Variable]

Personality used for gnus-summary-high-ticked This personality uses voice-brighten-extra whose effect can be changed globally by customizing voice-brighten-extra-settings.

`User Option` *voice-gnus-summary-high-undownloaded* [Variable]

Personality used for gnus-summary-high-undownloaded This personality uses voice-highlight whose effect can be changed globally by customizing voice-highlight-settings.

`User Option` *voice-gnus-summary-high-unread* [Variable]
Personality used for gnus-summary-high-unread This personality uses voice-brighten-extra whose effect can be changed globally by customizing voice-brighten-extra-settings.

`User Option` *voice-gnus-summary-low-ancient* [Variable]
Personality used for gnus-summary-low-ancient This personality uses voice-smoothen-extra whose effect can be changed globally by customizing voice-smoothen-extra-settings.

`User Option` *voice-gnus-summary-low-read* [Variable]
Personality used for gnus-summary-low-read This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-gnus-summary-low-ticked* [Variable]
Personality used for gnus-summary-low-ticked This personality uses voice-brighten-extra whose effect can be changed globally by customizing voice-brighten-extra-settings.

`User Option` *voice-gnus-summary-low-undownloaded* [Variable]
Personality used for gnus-summary-low-undownloaded This personality uses voice-bolden-and-animate whose effect can be changed globally by customizing voice-bolden-and-animate-settings.

`User Option` *voice-gnus-summary-low-unread* [Variable]
Personality used for gnus-summary-low-unread This personality uses voice-bolden-medium whose effect can be changed globally by customizing voice-bolden-medium-settings.

`User Option` *voice-gnus-summary-normal-ancient* [Variable]
Personality used for gnus-summary-normal-ancient This personality uses voice-smoothen-extra whose effect can be changed globally by customizing voice-smoothen-extra-settings.

`User Option` *voice-gnus-summary-normal-read* [Variable]
Personality used for gnus-summary-normal-read This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

`User Option` *voice-gnus-summary-normal-ticked* [Variable]
Personality used for gnus-summary-normal-ticked This personality uses voice-brighten-extra whose effect can be changed globally by customizing voice-brighten-extra-settings.

`User Option` *voice-gnus-summary-normal-undownloaded* [Variable]
Personality used for gnus-summary-normal-undownloaded This personality uses voice-bolden-and-animate whose effect can be changed globally by customizing voice-bolden-and-animate-settings.

`User Option` *voice-gnus-summary-normal-unread* [Variable]
Personality used for gnus-summary-normal-unread This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-gnus-summary-selected* [Variable]

Personality used for gnus-summary-selected This personality uses voice-animate-extra whose effect can be changed globally by customizing voice-animate-extra-settings.

12.88 emacspeak-go-mode

GO-MODE == Go Language support in emacs

12.89 emacspeak-gomoku

Auditory interface to gomoku

12.89.1 Emacspeak-Gomoku Commands

12.89.1.1 emacspeak-gomoku-display-statistics

`emacspeak-gomoku-display-statistics` [Command]

Display statistics from previous games

12.89.1.2 emacspeak-gomoku-goto-x-y

`emacspeak-gomoku-goto-x-y` (*x y*) [Command]

Prompt for and go to that square.

(fn X Y)

12.89.1.3 emacspeak-gomoku-show-current-column

`emacspeak-gomoku-show-current-column` [Command]

Aurally display current column

12.89.1.4 emacspeak-gomoku-show-current-negative-diagonal

`emacspeak-gomoku-show-current-negative-diagonal` [Command]

Aurally display current negative sloped diagonal

12.89.1.5 emacspeak-gomoku-show-current-positive-diagonal

`emacspeak-gomoku-show-current-positive-diagonal` [Command]

Aurally display current positively sloped diagonal

12.89.1.6 emacspeak-gomoku-show-current-row

`emacspeak-gomoku-show-current-row` [Command]

Aurally display current row

12.89.1.7 emacspeak-gomoku-speak-emacs-previous-move

`emacspeak-gomoku-speak-emacs-previous-move` [Command]

Speak emacs' previous move

12.89.1.8 emacspeak-gomoku-speak-humans-previous-move

`emacspeak-gomoku-speak-humans-previous-move` [Command]

Speak human' previous move

12.89.1.9 emacspeak-gomoku-speak-number-of-moves

`emacspeak-gomoku-speak-number-of-moves` [Command]

Speak number of moves so far

12.89.1.10 emacspeak-gomoku-speak-square

`emacspeak-gomoku-speak-square` [Command]

Speak coordinates and state of square at point

12.90 emacspeak-google

There are a number of search tools that can be implemented on the Google search page — in a JS-powered browser, these show up as the Google tool-belt. This module implements a minor mode for use in Google result pages that enables these tools via single keyboard commands. Originally all options were available as tbs=p:v Now, some specialized searches, e.g. blog search are tbm=

12.90.1 Emacspeak-Google Commands

12.90.1.1 emacspeak-google-extract-from-cache

`emacspeak-google-extract-from-cache` [Command]

Extract current page from the Google cache.

12.90.1.2 emacspeak-google-knowledge-search

`emacspeak-google-knowledge-search` (*query* **&optional** *limit*) [Command]

C-; k

C-x @ h k

Perform a Google Knowledge Graph search.
Optional interactive prefix arg 'limit' prompts for number of results, default is 1.

(fn QUERY &optional LIMIT)

12.90.1.3 emacspeak-google-on-this-site

`emacspeak-google-on-this-site` [Command]

Perform a google search restricted to the current WWW site.

12.90.1.4 emacspeak-google-open-link

`emacspeak-google-open-link` [Command]

Open Google link under point.

12.90.1.5 emacspeak-google-show-toolbelt

`emacspeak-google-show-toolbelt` [Command]

Reload search page with toolbelt showing.

12.90.1.6 emacspeak-google-sign-in

`emacspeak-google-sign-in` [Command]

Sign in to Google.

12.90.1.7 emacspeak-google-sign-out

`emacspeak-google-sign-out` [Command]

Sign out to Google.

12.90.1.8 emacspeak-google-similar-to-this-page

`emacspeak-google-similar-to-this-page` (*url*) [Command]

Ask Google to find documents similar to this one.

(fn URL)

12.90.1.9 emacspeak-google-toolbelt-change

`emacspeak-google-toolbelt-change` [Command]

Command to change values in the toolbelt and execute the query.

12.90.1.10 emacspeak-google-toolbelt-change-Shopping

`emacspeak-google-toolbelt-change-Shopping` [Command]

Change Shopping in the currently active toolbelt.

12.90.1.11 emacspeak-google-toolbelt-change-blog

`emacspeak-google-toolbelt-change-blog` [Command]

Change blog in the currently active toolbelt.

12.90.1.12 emacspeak-google-toolbelt-change-books

`emacspeak-google-toolbelt-change-books` [Command]

Change books in the currently active toolbelt.

12.90.1.13 emacspeak-google-toolbelt-change-books-format

`emacspeak-google-toolbelt-change-books-format` [Command]

Change books-format in the currently active toolbelt.

12.90.1.14 emacspeak-google-toolbelt-change-books-type

`emacspeak-google-toolbelt-change-books-type` [Command]

Change books-type in the currently active toolbelt.

12.90.1.15 emacspeak-google-toolbelt-change-books-viewability

`emacspeak-google-toolbelt-change-books-viewability` [Command]

Change books-viewability in the currently active toolbelt.

12.90.1.16 emacspeak-google-toolbelt-change-commercial

`emacspeak-google-toolbelt-change-commercial` [Command]

Change commercial in the currently active toolbelt.

12.90.1.17 emacspeak-google-toolbelt-change-commercial-prices

`emacspeak-google-toolbelt-change-commercial-prices` [Command]

Change commercial-prices in the currently active toolbelt.

12.90.1.18 emacspeak-google-toolbelt-change-date-filter

`emacspeak-google-toolbelt-change-date-filter` [Command]

Change date-filter in the currently active toolbelt.

12.90.1.19 emacspeak-google-toolbelt-change-forums

`emacspeak-google-toolbelt-change-forums` [Command]

Change forums in the currently active toolbelt.

12.90.1.20 emacspeak-google-toolbelt-change-group-discussions

`emacspeak-google-toolbelt-change-group-discussions` [Command]

Change group-discussions in the currently active toolbelt.

12.90.1.21 emacspeak-google-toolbelt-change-images

`emacspeak-google-toolbelt-change-images` [Command]

Change images in the currently active toolbelt.

12.90.1.22 emacspeak-google-toolbelt-change-in-depth

`emacspeak-google-toolbelt-change-in-depth` [Command]

Change in-depth in the currently active toolbelt.

12.90.1.23 emacspeak-google-toolbelt-change-literal

`emacspeak-google-toolbelt-change-literal` [Command]

Change literal in the currently active toolbelt.

12.90.1.24 emacspeak-google-toolbelt-change-news

`emacspeak-google-toolbelt-change-news` [Command]

Change news in the currently active toolbelt.

12.90.1.25 emacspeak-google-toolbelt-change-non-commercial

`emacspeak-google-toolbelt-change-non-commercial` [Command]

Change non-commercial in the currently active toolbelt.

12.90.1.26 emacspeak-google-toolbelt-change-patents

`emacspeak-google-toolbelt-change-patents` [Command]

Change patents in the currently active toolbelt.

12.90.1.27 emacspeak-google-toolbelt-change-places

`emacspeak-google-toolbelt-change-places` [Command]

Change places in the currently active toolbelt.

12.90.1.28 emacspeak-google-toolbelt-change-recent

`emacspeak-google-toolbelt-change-recent` [Command]

Change recent in the currently active toolbelt.

12.90.1.29 emacspeak-google-toolbelt-change-recipes

`emacspeak-google-toolbelt-change-recipes` [Command]

Change recipes in the currently active toolbelt.

12.90.1.30 emacspeak-google-toolbelt-change-reviews

`emacspeak-google-toolbelt-change-reviews` [Command]

Change reviews in the currently active toolbelt.

12.90.1.31 emacspeak-google-toolbelt-change-social

`emacspeak-google-toolbelt-change-social` [Command]

Change social in the currently active toolbelt.

12.90.1.32 emacspeak-google-toolbelt-change-sort-by-date

`emacspeak-google-toolbelt-change-sort-by-date` [Command]

Change sort-by-date in the currently active toolbelt.

12.90.1.33 emacspeak-google-toolbelt-change-structured-snippets

`emacspeak-google-toolbelt-change-structured-snippets` [Command]

Change structured-snippets in the currently active toolbelt.

12.90.1.34 emacspeak-google-toolbelt-change-timeline

`emacspeak-google-toolbelt-change-timeline` [Command]

Change timeline in the currently active toolbelt.

12.90.1.35 emacspeak-google-toolbelt-change-timeline-high

`emacspeak-google-toolbelt-change-timeline-high` [Command]

Change timeline-high in the currently active toolbelt.

12.90.1.36 emacspeak-google-toolbelt-change-timeline-low

`emacspeak-google-toolbelt-change-timeline-low` [Command]

Change timeline-low in the currently active toolbelt.

12.90.1.37 emacspeak-google-toolbelt-change-video

`emacspeak-google-toolbelt-change-video` [Command]

Change video in the currently active toolbelt.

12.90.1.38 emacspeak-google-toolbelt-change-video-duration

`emacspeak-google-toolbelt-change-video-duration` [Command]

Change video-duration in the currently active toolbelt.

12.90.1.39 emacspeak-google-toolbelt-change-web-history-not-visited

`emacspeak-google-toolbelt-change-web-history-not-visited` [Command]

Change web-history-not-visited in the currently active toolbelt.

12.90.1.40 emacspeak-google-toolbelt-change-web-history-visited

`emacspeak-google-toolbelt-change-web-history-visited` [Command]

Change web-history-visited in the currently active toolbelt.

12.90.1.41 emacspeak-google-tts

`emacspeak-google-tts` (*text* **&optional** *lang*) [Command]

Speak text using Google Network TTS.
Optional interactive prefix arg ‘lang’ specifies language identifier.

(fn TEXT &optional LANG)

12.90.1.42 emacspeak-google-tts-region

`emacspeak-google-tts-region` (*start end* **&optional** *ask-lang*) [Command]

C-. g

C-' g

C-x @ s g

Speak region using Google Network TTS.

(fn START END &optional ASK-LANG)

12.90.1.43 emacspeak-google-what-is-my-ip

`emacspeak-google-what-is-my-ip` [Command]

Show my public IP

12.90.1.44 emacspeak-google-who-links-to-this-page

`emacspeak-google-who-links-to-this-page` [Command]

Perform a google search to locate documents that link to the current page.

12.90.2 emacspeak-google Options

`User Option` *emacspeak-google-kg-key* [Variable]

API Key for Google Knowledge Graph.

`User Option` *emacspeak-google-tts-default-language* [Variable]

Default language used for Google TTS.

12.91 emacspeak-gridtext

Emacspeak's table browsing mode allows one to efficiently access content that is tabular in nature. That module also provides functions for inferring table structure where possible. Often, such structure is hard to infer automatically –but might be known to the user e.g. treat columns 1 through 30 as one column of a table and so on. This module allows the user to specify a conceptual grid that is "overlaid" on the region of text to turn it into a table for tabular browsing. For now, elements of the grid are "one line" high –but that may change in the future if necessary. This module is useful for browsing structured text files and the output from programs that tabulate their output. It's also useful for handling multicolumn text. The "grid" is specified as a list of (start end) tuples..

12.91.1 Emacspeak-Gridtext Commands

12.91.1.1 emacspeak-gridtext-apply

`emacspeak-gridtext-apply` (*start end grid*) [Command]

C-e # a

<fn> # a

Apply grid to region.

(fn START END GRID)

12.91.1.2 emacspeak-gridtext-load

`emacspeak-gridtext-load` (*file*) [Command]

C-e # l

<fn> # l

Load saved grid settings.

(fn FILE)

12.91.1.3 emacspeak-gridtext-save

`emacspeak-gridtext-save` (*file*) [Command]

C-e # s

<fn> # s

Save out grid settings.

(fn FILE)

12.92 emacspeak-gtags

GTAGS == Emacs support for GNU global. GNU global implements a modern tags solution Package gtags interfaces Emacs to this tool.

12.93 emacspeak-gud

Provide additional advice to ease debugger interaction with gud

12.94 emacspeak-haskell

Speech-enable package haskell-mode

12.94.1 emacspeak-haskell Options

`User Option` *voice-haskell-interactive-face-compile-error* [Variable]

Personality used for haskell-interactive-face-compile-error This personality uses voice-warning whose effect can be changed globally by customizing voice-warning-settings.

`User Option` *voice-haskell-interactive-face-compile-warning* [Variable]

Personality used for haskell-interactive-face-compile-warning This personality uses voice-warning whose effect can be changed globally by customizing voice-warning-settings.

`User Option` *voice-haskell-interactive-face-garbage* [Variable]

Personality used for haskell-interactive-face-garbage This personality uses voice-monotone-light whose effect can be changed globally by customizing voice-monotone-light-settings.

`User Option` *voice-haskell-interactive-face-prompt* [Variable]

Personality used for haskell-interactive-face-prompt This personality uses voice-brighten whose effect can be changed globally by customizing voice-brighten-settings.

`User Option` *voice-haskell-interactive-face-prompt-cont* [Variable]

Personality used for haskell-interactive-face-prompt-cont This personality uses voice-brighten-extra whose effect can be changed globally by customizing voice-brighten-extra-settings.

`User Option` *voice-haskell-interactive-face-result* [Variable]

Personality used for haskell-interactive-face-result This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

12.95 emacspeak-helm

HELM == Smart narrowing/selection in emacs This module speech-enables Helm interaction. See tvr/helm-prepare.el in the GitHub repository for my helm setup. that file provides convenient emacspeak-centric keybindings for Helm interaction.

12.96 emacspeak-hide

Flexible hide and show for emacspeak. This module allows one to easily hide or expose blocks of lines starting with a common prefix. It is motivated by the need to flexibly hide quoted text in email but is designed to be more general. the prefix parsing is inspired by filladapt.el

12.96.1 Emacspeak-Hide Commands

12.96.1.1 emacspeak-hide-or-expose-all-blocks

`emacspeak-hide-or-expose-all-blocks` [Command]

Hide or expose all blocks in buffer.

12.96.1.2 emacspeak-hide-or-expose-block

`emacspeak-hide-or-expose-block` (**&optional** *prefix*) [Command]

`C-e j`

`<fn> j`

Hide or expose a block of text.
This command either hides or exposes a block of text
starting on the current line. A block of text is defined as
a portion of the buffer in which all lines start with a
common PREFIX. Optional interactive prefix arg causes all
blocks in current buffer to be hidden or exposed.

(fn &optional PREFIX)

12.96.1.3 emacspeak-hide-speak-block-sans-prefix

`emacspeak-hide-speak-block-sans-prefix` [Command]

`C-e C-j`

`<fn> C-j`

Speaks current block after stripping its prefix.
If the current block is not hidden, it first hides it.
This is useful because as you locate blocks, you can invoke this
command to listen to the block,
and when you have heard enough navigate easily to move past the block.

12.97 emacspeak-hide-lines

HIDE-LINES == hide/show lines from melpa

12.98 emacspeak-hideshow

speech-enable hideshow.el

12.99 emacspeak-hydra

Speech-enable package hydra: For uses of hydra see module See Section 12.126 [emacspeak-muggles], page 203.

12.99.1 Emacspeak-Hydra Commands

12.99.1.1 emacspeak-hydra-toggle-talkative

`emacspeak-hydra-toggle-talkative` [Command]

> Toggle state of hydra-is-helpful

12.100 emacspeak-ibuffer

speech-enable ibuffer.el this is an alternative to buffer-menu

12.100.1 Emacspeak-Ibuffer Commands

12.100.1.1 emacspeak-ibuffer-speak-buffer-line

`emacspeak-ibuffer-speak-buffer-line` [Command]

> Speak information about this buffer

12.101 emacspeak-ido

speech-enable ido.el This is an interesting task since most of the value-add provided by package ido.el is visual feedback. Speech UI Challenge: What is the most efficient means of conveying a dynamically updating set of choices? current strategy is to walk the list using c-s and c-r as provided by ido Set number matches shown (ido-max-prospects) to 3 using Custom so you dont hear the entire list.

12.101.1 emacspeak-ido Options

`User Option` *emacspeak-ido-typing-delay* [Variable]

> How long we wait before speaking completions.

`User Option` *voice-ido-first-match* [Variable]

> Personality used for ido-first-match This personality uses voice-brighten whose effect can be changed globally by customizing voice-brighten-settings.

`User Option` *voice-ido-incomplete-regexp* [Variable]

> Personality used for ido-incomplete-regexp This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-ido-indicator* [Variable]

Personality used for ido-indicator This personality uses voice-brighten whose effect can be changed globally by customizing voice-brighten-settings.

`User Option` *voice-ido-only-match* [Variable]

Personality used for ido-only-match This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-ido-subdir* [Variable]

Personality used for ido-subdir This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-ido-virtual* [Variable]

Personality used for ido-virtual This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

12.102 emacspeak-iedit

IEDIT == Edit multiple regions This module speech-enables iedit.

12.102.1 emacspeak-iedit Options

`User Option` *voice-iedit-occurrence* [Variable]

Personality used for iedit-occurrence This personality uses voice-lock-overlay-1 whose effect can be changed globally by customizing voice-lock-overlay-1-settings.

`User Option` *voice-iedit-read-only-occurrence* [Variable]

Personality used for iedit-read-only-occurrence This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

12.103 emacspeak-indium

INDIUM == Javascript IDE This module speech-enables Indium.

12.104 emacspeak-info

This module speech-enables the Emacs Info Reader.

12.104.1 Emacspeak-Info Commands

12.104.1.1 emacspeak-info-next-section

`emacspeak-info-next-section` [Command]

Move forward to next section in this node.

12.104.1.2 emacspeak-info-previous-section

`emacspeak-info-previous-section` [Command]

Move backward to previous section in this node.

12.104.1.3 emacspeak-info-speak-header

`emacspeak-info-speak-header` [Command]

Speak info header line.

12.104.1.4 emacspeak-info-wizard

`emacspeak-info-wizard` (*node-spec*) [Command]

C-h TAB

<f1> TAB

<help> TAB
Read a node spec from the minibuffer and launch
Info-goto-node.
See documentation for command 'Info-goto-node' for details on
node-spec.

(fn NODE-SPEC)

12.104.2 emacspeak-info Options

`User Option` *emacspeak-info-select-node-speak-chunk* [Variable]

Specifies how much of the selected node gets spoken. Possible values are: screenfull – speak the displayed screen node – speak the entire node.

`User Option` *voice-Info-quoted* [Variable]

Personality used for Info-quoted This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-info-header-node* [Variable]

Personality used for info-header-node This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

`User Option` *voice-info-header-xref* [Variable]

Personality used for info-header-xref This personality uses voice-brighten whose effect can be changed globally by customizing voice-brighten-settings.

`User Option` *voice-info-index-match* [Variable]

Personality used for info-index-match This personality uses 'voice-bolden-medium whose effect can be changed globally by customizing 'voice-bolden-medium-settings.

`User Option` *voice-info-menu-header* [Variable]

Personality used for info-menu-header This personality uses voice-bolden-medium whose effect can be changed globally by customizing voice-bolden-medium-settings.

`User Option` *voice-info-menu-star* [Variable]

Personality used for info-menu-star This personality uses voice-brighten whose effect can be changed globally by customizing voice-brighten-settings.

`User Option` *voice-info-node* [Variable]

Personality used for info-node This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-info-title-1* [Variable]

Personality used for info-title-1 This personality uses voice-bolden-extra whose effect can be changed globally by customizing voice-bolden-extra-settings.

`User Option` *voice-info-title-2* [Variable]

Personality used for info-title-2 This personality uses voice-bolden-medium whose effect can be changed globally by customizing voice-bolden-medium-settings.

`User Option` *voice-info-title-3* [Variable]

Personality used for info-title-3 This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-info-title-4* [Variable]

Personality used for info-title-4 This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-info-xref* [Variable]

Personality used for info-xref This personality uses voice-animate-extra whose effect can be changed globally by customizing voice-animate-extra-settings.

`User Option` *voice-info-xref-visited* [Variable]

Personality used for info-xref-visited This personality uses voice-animate-medium whose effect can be changed globally by customizing voice-animate-medium-settings.

12.105 emacspeak-ispell

This module speech enables ispell. Implementation note: This is hard because of how ispell.el is written Namely, all of the work is done by one huge hairy function. This makes advising it hard. The ispell commands work well with Emacspeak as long as the list of correction choices are few. For interactively moving through corrections, install package flyspell-correct from MELPA (package-install "flyspell-correct") Then use M-x flyspell-mode. Package flyspell is speech-enabled by Emacspeak module emacspeak-flyspell And that module sets up flyspell-correct to use IDO-style completion, i.e. you can move through corrections with C-r and C-s.

12.105.1 emacspeak-ispell Options

`User Option` *emacspeak-ispell-max-choices* [Variable]

Emacspeak will not speak the choices if there are more than this many available corrections.

12.106 emacspeak-ivy

IVY == One More Smart Completion Technique Speech-enable ivy-style completion. This is still experimental and preliminary.

12.106.1 emacspeak-ivy Options

`User Option` *voice-ivy-action* [Variable]
Personality used for ivy-action This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

`User Option` *voice-ivy-current-match* [Variable]
Personality used for ivy-current-match This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-ivy-cursor* [Variable]
Personality used for ivy-cursor This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

`User Option` *voice-ivy-minibuffer-match-face-1* [Variable]
Personality used for ivy-minibuffer-match-face-1 This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-ivy-minibuffer-match-face-2* [Variable]
Personality used for ivy-minibuffer-match-face-2 This personality uses voice-monotone-medium whose effect can be changed globally by customizing voice-monotone-medium-settings.

`User Option` *voice-ivy-minibuffer-match-face-3* [Variable]
Personality used for ivy-minibuffer-match-face-3 This personality uses voice-monotone-light whose effect can be changed globally by customizing voice-monotone-light-settings.

`User Option` *voice-ivy-minibuffer-match-face-4* [Variable]
Personality used for ivy-minibuffer-match-face-4 This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-ivy-modified-buffer* [Variable]
Personality used for ivy-modified-buffer This personality uses voice-bolden-and-animate whose effect can be changed globally by customizing voice-bolden-and-animate-settings.

`User Option` *voice-ivy-remote* [Variable]
Personality used for ivy-remote This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-ivy-subdir* [Variable]
Personality used for ivy-subdir This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

`User Option` *voice-ivy-virtual* [Variable]
Personality used for ivy-virtual This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

12.107 emacspeak-jabber

emacs-jabber.el implements a jabber client for emacs emacs-jabber is hosted at sourceforge. I use emacs-jabber with my gmail.com account

12.107.1 Emacspeak-Jabber Commands

12.107.1.1 emacspeak-jabber-chat-next-message

`emacspeak-jabber-chat-next-message` [Command]

Move forward to and speak the next message in this chat session.

12.107.1.2 emacspeak-jabber-chat-previous-message

`emacspeak-jabber-chat-previous-message` [Command]

Move backward to and speak the previous message in this chat session.

12.107.1.3 emacspeak-jabber-chat-speak-this-message

`emacspeak-jabber-chat-speak-this-message` (**&optional** *copy-as-kill*) [Command]

Speak chat message under point.
With optional interactive prefix arg 'copy-as-kill', copy it to
the kill ring as well.

(fn &optional COPY-AS-KILL)

12.107.1.4 emacspeak-jabber-popup-roster

`emacspeak-jabber-popup-roster` [Command]

C-e x j

<fn> x j

Pop to Jabber roster.

12.107.1.5 emacspeak-jabber-speak-recent-message

`emacspeak-jabber-speak-recent-message` [Command]

C-e x SPC

<fn> x SPC
Speak most recent message if one exists.

12.107.2 emacspeak-jabber Options

`User Option` *emacspeak-jabber-speak-presence-alerts* [Variable]

Set to T if you want to hear presence alerts.

`User Option` *voice-jabber-chat-error* [Variable]

Personality used for jabber-chat-error This personality uses voice-bolden-and-animate whose effect can be changed globally by customizing voice-bolden-and-animate-settings.

`User Option` *voice-jabber-chat-prompt-foreign* [Variable]

Personality used for jabber-chat-prompt-foreign This personality uses voice-brighten-medium whose effect can be changed globally by customizing voice-brighten-medium-settings.

`User Option` *voice-jabber-chat-prompt-local* [Variable]

Personality used for jabber-chat-prompt-local This personality uses voice-smoothen-medium whose effect can be changed globally by customizing voice-smoothen-medium-settings.

`User Option` *voice-jabber-chat-prompt-system* [Variable]

Personality used for jabber-chat-prompt-system This personality uses voice-brighten-extra whose effect can be changed globally by customizing voice-brighten-extra-settings.

`User Option` *voice-jabber-chat-text-local* [Variable]

Personality used for jabber-chat-text-local This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

`User Option` *voice-jabber-roster-user-away* [Variable]

Personality used for jabber-roster-user-away This personality uses voice-smoothen-extra whose effect can be changed globally by customizing voice-smoothen-extra-settings.

`User Option` *voice-jabber-roster-user-chatty* [Variable]

Personality used for jabber-roster-user-chatty This personality uses voice-brighten whose effect can be changed globally by customizing voice-brighten-settings.

`User Option` *voice-jabber-roster-user-dnd* [Variable]

Personality used for jabber-roster-user-dnd This personality uses voice-lighten-medium whose effect can be changed globally by customizing voice-lighten-medium-settings.

`User Option` *voice-jabber-roster-user-error* [Variable]

Personality used for jabber-roster-user-error This personality uses voice-bolden-and-animate whose effect can be changed globally by customizing voice-bolden-and-animate-settings.

`User Option` *voice-jabber-roster-user-offline* [Variable]

Personality used for jabber-roster-user-offline This personality uses voice-smoothen-extra whose effect can be changed globally by customizing voice-smoothen-extra-settings.

`User Option` *voice-jabber-roster-user-online* [Variable]

Personality used for jabber-roster-user-online This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-jabber-roster-user-xa* [Variable]

Personality used for jabber-roster-user-xa This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-jabber-title-large* [Variable]
Personality used for jabber-title-large This personality uses voice-bolden-extra whose effect can be changed globally by customizing voice-bolden-extra-settings.

`User Option` *voice-jabber-title-medium* [Variable]
Personality used for jabber-title-medium This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-jabber-title-small* [Variable]
Personality used for jabber-title-small This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

12.108 emacspeak-jdee

Speech enable Java IDE. The Java IDE –JDEE– can be found at http://sunsite.auc.dk/jdee/

12.109 emacspeak-js2

JS2-mode http://js2-mode.googlecode.com/svn/trunk is a new, powerful Emacs mode for working with JavaScript. This module speech-enables js2.

12.109.1 emacspeak-js2 Options

`User Option` *voice-js2-error* [Variable]
Personality used for js2-error This personality uses voice-bolden-extra whose effect can be changed globally by customizing voice-bolden-extra-settings.

`User Option` *voice-js2-external-variable* [Variable]
Personality used for js2-external-variable This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

`User Option` *voice-js2-function-call* [Variable]
Personality used for js2-function-call This personality uses voice-annotate whose effect can be changed globally by customizing voice-annotate-settings.

`User Option` *voice-js2-function-param* [Variable]
Personality used for js2-function-param This personality uses voice-lighten-extra whose effect can be changed globally by customizing voice-lighten-extra-settings.

`User Option` *voice-js2-instance-member* [Variable]
Personality used for js2-instance-member This personality uses voice-lighten-medium whose effect can be changed globally by customizing voice-lighten-medium-settings.

`User Option` *voice-js2-jsdoc-html-tag-delimiter* [Variable]
Personality used for js2-jsdoc-html-tag-delimiter This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

`User Option` *voice-js2-jsdoc-html-tag-name* [Variable]
Personality used for js2-jsdoc-html-tag-name This personality uses voice-bolden-medium whose effect can be changed globally by customizing voice-bolden-medium-settings.

User Option *voice-js2-jsdoc-tag* [Variable]
Personality used for js2-jsdoc-tag This personality uses voice-bolden-medium whose effect can be changed globally by customizing voice-bolden-medium-settings.

User Option *voice-js2-jsdoc-type* [Variable]
Personality used for js2-jsdoc-type This personality uses voice-smoothen-medium whose effect can be changed globally by customizing voice-smoothen-medium-settings.

User Option *voice-js2-jsdoc-value* [Variable]
Personality used for js2-jsdoc-value This personality uses voice-lighten-medium whose effect can be changed globally by customizing voice-lighten-medium-settings.

User Option *voice-js2-magic-paren* [Variable]
Personality used for js2-magic-paren This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

User Option *voice-js2-object-property* [Variable]
Personality used for js2-object-property This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

User Option *voice-js2-object-property-access* [Variable]
Personality used for js2-object-property-access This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

User Option *voice-js2-private-function-call* [Variable]
Personality used for js2-private-function-call This personality uses voice-smoothen-extra whose effect can be changed globally by customizing voice-smoothen-extra-settings.

User Option *voice-js2-private-member* [Variable]
Personality used for js2-private-member This personality uses voice-lighten-extra whose effect can be changed globally by customizing voice-lighten-extra-settings.

User Option *voice-js2-warning* [Variable]
Personality used for js2-warning This personality uses voice-bolden-and-animate whose effect can be changed globally by customizing voice-bolden-and-animate-settings.

12.110 emacspeak-keymap

This module defines the emacspeak keybindings. Note that the <fn> key found on laptops is denoted <fn>

12.110.1 emacspeak-keymap Options

User Option *emacspeak-alt-keys* [Variable]
Specifies alt key bindings for the audio desktop. You can turn the ‘Pause’ key on your Linux PC keyboard into a ‘alt’ key on Linux by having it emit the sequence ‘C-x@a’.

Bindings specified here are available on prefix key ‘alt’ (not to be confused with alt==meta) for example, if you bind ‘s’ to command emacspeak-emergency-tts-restart then that command will be available on key ‘ALT s’

KEYS should be a string constant in the format used for saving keyboard macros (see ‘edmacro-mode’).

Command is an interactive command or a prefix-command that can be bound to a key.

The value of this variable is an association list. The car of each element specifies a key sequence. The cdr specifies an interactive command that the key sequence executes.

`User Option` *emacspeak-ctl-z-keys* [Variable]

Specifies ctl-z key bindings for the audio desktop.

Bindings specified here are available on prefix key ‘C-z’ for example, if you bind ‘zb’ to command ‘suspend-frame ’ then that command will be available on key ‘C-z z’.

KEYS should be a string constant in the format used for saving keyboard macros (see ‘edmacro-mode’).

Command is an interactive command or a prefix-command that can be bound to a key.

The value of this variable is an association list. The car of each element specifies a key sequence. The cdr specifies an interactive command that the key sequence executes.

`User Option` *emacspeak-hyper-keys* [Variable]

Specifies hyper key bindings for the audio desktop. Emacs can use the ‘hyper’ key as a modifier key. You can turn the ‘windows’ keys on your Linux PC keyboard into a ‘hyper’ key on Linux by having it emit the sequence ‘C-x@h’.

Bindings specified here are available on prefix key ‘hyper’ for example, if you bind ‘b’ to command ‘bbdb ’ then that command will be available on key ‘hyper b’.

KEYS should be a string constant in the format used for saving keyboard macros (see ‘edmacro-mode’).

Command is an interactive command or a prefix-command that can be bound to a key.

The value of this variable is an association list. The car of each element specifies a key sequence. The cdr specifies an interactive command that the key sequence executes.

`User Option` *emacspeak-personal-ctlx-keys* [Variable]

Specifies personal-ctlx key bindings for use with C-e C-x for the audio desktop. Bindings specified here are available on prefix key C-e C-x for example, if you bind ‘C-s’ to command emacspeak-emergency-tts-restart then that command will be available on key C-e C-x C-s. KEYS should be a string constant in the format used for saving keyboard macros (see ‘edmacro-mode’).

Command is an interactive command or a prefix-command that can be bound to a key.

The value of this variable is an association list. The car of each element specifies a key sequence. The cdr specifies an interactive command that the key sequence executes.

`User Option` *emacspeak-personal-keys* [Variable]

Specifies personal key bindings for the audio desktop. Bindings specified here are available on prefix key C-e x for example, if you bind ‘s’ to command emacspeak-emergency-tts-restart then that command will be available on key C-e x s.

KEYS should be a string constant in the format used for saving keyboard macros (see 'edmacro-mode').

Command is an interactive command or a prefix-command that can be bound to a key.

The value of this variable is an association list. The car of each element specifies a key sequence. The cdr specifies an interactive command that the key sequence executes.

`User Option` *emacspeak-super-keys* [Variable]

Specifies super key bindings for the audio desktop. You can turn the right 'windows menu' keys on your Linux PC keyboard into a 'super' key on Linux by having it emit the sequence 'C-x@s'.

Bindings specified here are available on prefix key 'super' for example, if you bind 's' to command emacspeak-emergency-tts-restart then that command will be available on key 'super s'. KEYS should be a string constant in the format used for saving keyboard macros (see 'edmacro-mode').

Command is an interactive command or a prefix-command that can be bound to a key.

The value of this variable is an association list. The car of each element specifies a key sequence. The cdr specifies an interactive command that the key sequence executes.

12.111 emacspeak-kmacro

speech-enables kmacro — a kbd macro interface

12.112 emacspeak-librivox

LIBRIVOX == `http://www.librivox.org` — Free Audio Books. API Info: `https://librivox.org/api/info` It provides a simple Web API This module implements an Emacspeak Librivox client.

12.112.1 Usage

main entry point is command `emacspeak-librivox` bound to *`C-; l`*. This prompts with the following choices:

- *`a`* Author — Search by Author.
- *`t`* Title — Search by Title.
- *`g`* Genre — Search by Genre — with minibuffer completion.
- *`p`* Play — Play a book.
- *`d`* Directory — Browse local cache.

Search results are displayed in a Web page that provides controls for accessing the book.

12.112.2 Emacspeak-Librivox Commands

12.112.2.1 emacspeak-librivox

`emacspeak-librivox` (*search-type*) [Command]

`C-; l`

C-x @ h l

Launch a Librivox Search.

(fn SEARCH-TYPE)

12.112.2.2 emacspeak-librivox-play

`emacspeak-librivox-play` (*rss-url*) [Command]

Play book stream

(fn RSS-URL)

12.112.2.3 emacspeak-librivox-search-by-author

`emacspeak-librivox-search-by-author` (*author* **&optional** *offset*) [Command]

Search by author. Both exact and partial matches for
'author'. Optional interactive prefix arg 'offset' prompts for offset
— use this for retrieving next set of results.

(fn AUTHOR &optional OFFSET)

12.112.2.4 emacspeak-librivox-search-by-genre

`emacspeak-librivox-search-by-genre` (*genre* **&optional** *offset*) [Command]

Search by genre.
Optional prefix arg 'offset' prompts for offset.

(fn GENRE &optional OFFSET)

12.112.2.5 emacspeak-librivox-search-by-title

`emacspeak-librivox-search-by-title` (*title* **&optional** *offset*) [Command]

Search by title. Both exact and partial matches for 'title'. Optional
prefix arg 'offset' prompts for offset — use this for retrieving
more results.

(fn TITLE &optional OFFSET)

12.112.3 emacspeak-librivox Options

`User Option` *emacspeak-librivox-local-cache* [Variable]

Location where we cache LIBRIVOX playlists.

12.113 emacspeak-lispy

LISPY == smart Navigation Of Lisp code This module speech-enables lispy.

12.113.1 Overview

Lispy editing keeps delimiters balanced and Lispy navigators reliably place point on either the opening or closing delimiter of the current s-expression. Emacspeak leverages this fact in the type of spoken feedback that is produced. All navigation commands produce the following:

- Speak the current s-expression when at the front of a sexp.
- Speak the current line with option `emacspeak-show-point` turned on when at the end of an s-expression.
- Produce auditory icon `left` or `right` to indicate point being at the beginning or end of current line.
- Indicate with an auditory icon if point did not move.

12.113.2 emacspeak-lispy Options

`User Option` *voice-lispy-face-hint* [Variable]

Personality used for lispy-face-hint This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

`User Option` *voice-lispy-face-key-nosel* [Variable]

Personality used for lispy-face-key-nosel This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-lispy-face-key-sel* [Variable]

Personality used for lispy-face-key-sel This personality uses voice-brighten whose effect can be changed globally by customizing voice-brighten-settings.

`User Option` *voice-lispy-face-opt-nosel* [Variable]

Personality used for lispy-face-opt-nosel This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-lispy-face-opt-sel* [Variable]

Personality used for lispy-face-opt-sel This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-lispy-face-req-nosel* [Variable]

Personality used for lispy-face-req-nosel This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-lispy-face-req-sel* [Variable]

Personality used for lispy-face-req-sel This personality uses voice-brighten-extra whose effect can be changed globally by customizing voice-brighten-extra-settings.

`User Option` *voice-lispy-face-rst-nosel* [Variable]

Personality used for lispy-face-rst-nosel This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-lispy-face-rst-sel* [Variable]

Personality used for lispy-face-rst-sel This personality uses voice-lighten-extra whose effect can be changed globally by customizing voice-lighten-extra-settings.

12.114 emacspeak-lua

LUA == lua-mode Speech-enable lua-mode.

12.115 emacspeak-m-player

Defines an Emacspeak front-end for interacting with `mplayer`. Program `mplayer` is a versatile media player capable of playing many streaming media formats. This module provides complete access to all `mplayer` functionality from a convenient Emacs interface.

12.115.1 Usage

The main entry-point is command `emacspeak-multimedia` bound to *`C-e ;`*. This prompts for and launches the desired media stream. Once a stream is playing, you can control it with single-letter keystrokes in the `*M-Player*` buffer. Alternatively, you can switch away from that buffer to do real work, And invoke `m-player` commands by first pressing *`C-e ;`*. As an example, pressing *`v`* in the `*M-Player*` buffer prompts for and sets the volume; When not in the `*M-Player*` buffer, you can achieve the same by pressing *`C-e ; v`*. Press *`C-h b`* in the `*M-Player*` buffer to list `m-player` keybindings.

12.115.2 Emacspeak-M-Player Commands

12.115.2.1 emacspeak-m-player

`emacspeak-m-player` (*resource* **&optional** *play-list*) [Command]

> Play specified resource using m-player. Optional prefix argument play-list interprets resource as a play-list. Second interactive prefix arg adds option -allow-dangerous-playlist-parsing to mplayer. Resource is a media resource or playlist containing media resources. The player is placed in a buffer in emacspeak-m-player-mode.
>
> (fn RESOURCE &optional PLAY-LIST)

12.115.2.2 emacspeak-m-player-add-autopan

`emacspeak-m-player-add-autopan` [Command]

> Add predefined autopan effect.

12.115.2.3 emacspeak-m-player-add-autosat

`emacspeak-m-player-add-autosat` [Command]

> Add predefined ZamAutoSat (auto saturation) effect.

12.115.2.4 emacspeak-m-player-add-equalizer

`emacspeak-m-player-add-equalizer` (**&optional** *reset*) [Command]

> Add equalizer to playing stream. Equalizer is updated as each change is made, and the final effect set by pressing RET. Interactive prefix arg ‘reset’ starts with all filters set to 0.
>
> (fn &optional RESET)

12.115.2.5 emacspeak-m-player-add-filter

`emacspeak-m-player-add-filter` (*filter-name* **&optional** *edit*) [Command]

Adds specified filter. Prompts with one of several pre-defined filters. Optional interactive prefix arg 'edit' enables editing the selected filter before it is applied.

(fn FILTER-NAME &optional EDIT)

12.115.2.6 emacspeak-m-player-add-ladspa

`emacspeak-m-player-add-ladspa` [Command]

Apply plugin to running MPlayer.
Copies invocation string to kill-ring so it can be added easily to our pre-defined filters if appropriate.

12.115.2.7 emacspeak-m-player-alt-src-step

`emacspeak-m-player-alt-src-step` (*step*) [Command]

Move within an ASF playlist.

(fn STEP)

12.115.2.8 emacspeak-m-player-amark-add

`emacspeak-m-player-amark-add` (*name* **&optional** *prompt-position*) [Command]

Set AMark 'name' at current position in current audio stream.
Interactive prefix arg prompts for position.
As the default, use current position.

(fn NAME &optional PROMPT-POSITION)

12.115.2.9 emacspeak-m-player-amark-jump

`emacspeak-m-player-amark-jump` [Command]

Jump to specified AMark.

12.115.2.10 emacspeak-m-player-amark-save

`emacspeak-m-player-amark-save` [Command]

Ensure that amarks are saved in the directory of the resource .

12.115.2.11 emacspeak-m-player-apply-reverb-preset

`emacspeak-m-player-apply-reverb-preset` (*preset*) [Command]

Prompt for a predefined reverb preset and apply it.
You need to use mplayer built with ladspa support, and have package tap-reverb already installed.

(fn PRESET)

12.115.2.12 emacspeak-m-player-backward-10min

`emacspeak-m-player-backward-10min` [Command]

Move backward by ten minutes.

12.115.2.13 emacspeak-m-player-backward-10s

`emacspeak-m-player-backward-10s` [Command]

Move back by 10 seconds.

12.115.2.14 emacspeak-m-player-backward-1min

`emacspeak-m-player-backward-1min` [Command]

Move back by 1 minute.

12.115.2.15 emacspeak-m-player-balance

`emacspeak-m-player-balance` [Command]

Set left/right balance.

12.115.2.16 emacspeak-m-player-beginning-of-track

`emacspeak-m-player-beginning-of-track` [Command]

Move to beginning of track.

12.115.2.17 emacspeak-m-player-bind-accelerator

`emacspeak-m-player-bind-accelerator` (*directory key*) [Command]

Binds key to invoke m-player on specified directory.

(fn DIRECTORY KEY)

12.115.2.18 emacspeak-m-player-clear-filters

`emacspeak-m-player-clear-filters` [Command]

Clear all active filters

12.115.2.19 emacspeak-m-player-command

`emacspeak-m-player-command` (*key*) [Command]

Invoke MPlayer commands.

(fn KEY)

12.115.2.20 emacspeak-m-player-customize-options

`emacspeak-m-player-customize-options` [Command]

Use Customize to manipulate MPlayer options.

12.115.2.21 emacspeak-m-player-delete-filter

`emacspeak-m-player-delete-filter` (*filter*) [Command]

Delete filter.

(fn FILTER)

12.115.2.22 emacspeak-m-player-delete-ladspa

`emacspeak-m-player-delete-ladspa` [Command]

Delete plugin from running MPlayer.

12.115.2.23 emacspeak-m-player-display-metadata

`emacspeak-m-player-display-metadata` [Command]

Display metadata after refreshing it if needed.

12.115.2.24 emacspeak-m-player-display-percent

`emacspeak-m-player-display-percent` [Command]

Display current percentage.

12.115.2.25 emacspeak-m-player-display-position

`emacspeak-m-player-display-position` [Command]

Display current position in track and its length.

12.115.2.26 emacspeak-m-player-double-speed

`emacspeak-m-player-double-speed` [Command]

Scale speed by 2.0

12.115.2.27 emacspeak-m-player-edit-reverb

`emacspeak-m-player-edit-reverb` [Command]

Edit current ladspa reverb filter.
See option emacspeak-m-player-reverb-filter to customize reverb filter values.
You need to use mplayer built with ladspa support, and have package
tap-reverb already installed.

12.115.2.28 emacspeak-m-player-end-of-track

`emacspeak-m-player-end-of-track` [Command]

Move to beginning of track.

12.115.2.29 emacspeak-m-player-equalizer-control

`emacspeak-m-player-equalizer-control` (*v*) [Command]

Manipulate values in specified vector using minibuffer.
Applies the resulting value at each step.

(fn V)

12.115.2.30 emacspeak-m-player-equalizer-preset

`emacspeak-m-player-equalizer-preset` (*name*) [Command]

Prompts for equalizer preset and applies it to current stream.

The following presets are available:

flat classical club dance full-bass full-bass-and-treble
full-treble headphones large-hall live party pop reggae rock
ska soft soft-rock techno

(fn NAME)

12.115.2.31 emacspeak-m-player-faster

`emacspeak-m-player-faster` [Command]

Speed up playback.
This affects pitch.

12.115.2.32 emacspeak-m-player-forward-10min

`emacspeak-m-player-forward-10min` [Command]

Move forward by ten minutes.

12.115.2.33 emacspeak-m-player-forward-10s

`emacspeak-m-player-forward-10s` [Command]

Move forward by 10 seconds.

12.115.2.34 emacspeak-m-player-forward-1min

`emacspeak-m-player-forward-1min` [Command]

Move forward by 1 minute.

12.115.2.35 emacspeak-m-player-from-media-history

`emacspeak-m-player-from-media-history` (*posn*) [Command]

C-. h

C-' h

C-x @ s h

Play media from media-history.
Numeric arg ‘posn’ specifies position in history.

(fn POSN)

12.115.2.36 emacspeak-m-player-get-length

`emacspeak-m-player-get-length` [Command]

Display length of track in seconds.

12.115.2.37 emacspeak-m-player-half-speed

`emacspeak-m-player-half-speed` [Command]

Scale speed by 0.5.

12.115.2.38 emacspeak-m-player-left-channel

`emacspeak-m-player-left-channel` [Command]

Play both channels on left channel.

12.115.2.39 emacspeak-m-player-load

`emacspeak-m-player-load` (*resource* **&optional** *append*) [Command]

Load specified resource into a running m-player.
Interactive prefix arg appends the new resource to what is playing.

(fn RESOURCE &optional APPEND)

12.115.2.40 emacspeak-m-player-load-file

`emacspeak-m-player-load-file` (*f*) [Command]

Load specified file.

(fn F)

12.115.2.41 emacspeak-m-player-load-playlist

`emacspeak-m-player-load-playlist` (*f*) [Command]

Load specified playlist file.

(fn F)

12.115.2.42 emacspeak-m-player-locate-media

`emacspeak-m-player-locate-media` (*pattern*) [Command]

`C-. l`

`C-' l`

`C-x @ s l`

Locate media matching specified pattern. The results can be
played as a play-list by pressing [RET] on the first line.
Pattern is first converted to a regexp that accepts common
punctuation separators (-,._'") in place of white-space.
Results are placed in a Locate buffer and can be played using
M-Player — use M-x emacspeak-dired-open-this-file locally bound to C-RET
to play individual tracks.

(fn PATTERN)

12.115.2.43 emacspeak-m-player-mode

`emacspeak-m-player-mode` [Command]

Major mode for m-player interaction.

key binding
— ——-

C-l ladspa
RET emacspeak-m-player-load
ESC Prefix Command
SPC emacspeak-m-player-pause
% emacspeak-m-player-display-percent
(emacspeak-m-player-left-channel
) emacspeak-m-player-right-channel
+ emacspeak-m-player-volume-up
, emacspeak-m-player-backward-10s
- emacspeak-m-player-volume-down
. emacspeak-m-player-forward-10s
/ emacspeak-m-player-restore-process
1 .. 9 emacspeak-m-player-volume-set
; emacspeak-m-player-pop-to-player
< emacspeak-m-player-backward-1min
= emacspeak-m-player-volume-up
> emacspeak-m-player-forward-1min
? emacspeak-m-player-display-position
A emacspeak-m-player-amark-add
C emacspeak-m-player-clear-filters
E emacspeak-m-player-add-equalizer
G emacspeak-m-player-seek-percentage
L emacspeak-m-player-locate-media
M emacspeak-m-player-display-metadata
O emacspeak-m-player-reset-options
P emacspeak-m-player-apply-reverb-preset
Q emacspeak-m-player-quit
R emacspeak-m-player-edit-reverb
S emacspeak-m-player-amark-save
[emacspeak-m-player-slower
\ emacspeak-m-player-persist-process
] emacspeak-m-player-faster
a emacspeak-m-player-add-autopan
b emacspeak-wizards-view-buffers-filtered-by-m-player-mode
c emacspeak-m-player-slave-command
d emacspeak-m-player-delete-filter
e emacspeak-m-player-equalizer-preset
f emacspeak-m-player-add-filter
g emacspeak-m-player-seek-absolute

i emacspeak-m-player-stream-info
j emacspeak-m-player-amark-jump
k emacspeak-m-player-quit
l emacspeak-m-player-get-length
m emacspeak-m-player-mode-line
n emacspeak-m-player-next-track
o emacspeak-m-player-customize-options
p emacspeak-m-player-previous-track
q bury-buffer
r emacspeak-m-player-seek-relative
s emacspeak-m-player-scale-speed
t emacspeak-m-player-play-tracks-jump
u emacspeak-m-player-url
v emacspeak-m-player-volume-change
w emacspeak-m-player-write-clip
x emacspeak-m-player-pan
z emacspeak-m-player-add-autosat
{ emacspeak-m-player-half-speed
} emacspeak-m-player-double-speed
DEL emacspeak-m-player-reset-speed
<down> emacspeak-m-player-forward-1min
<end> emacspeak-m-player-end-of-track
<home> emacspeak-m-player-beginning-of-track
<left> emacspeak-m-player-backward-10s
<next> emacspeak-m-player-forward-10min
<prior> emacspeak-m-player-backward-10min
<right> emacspeak-m-player-forward-10s
<up> emacspeak-m-player-backward-1min

M-, emacspeak-m-player-set-clip-start
M-. emacspeak-m-player-set-clip-end
M-l emacspeak-m-player-load-playlist

In addition to any hooks its parent mode 'comint-mode' might have run,
this mode runs the hook 'emacspeak-m-player-mode-hook', as the final or penultimate step
during initialization.

12.115.2.44 emacspeak-m-player-mode-line

`emacspeak-m-player-mode-line` [Command]

Meaningful mode-line for *M-Player* buffers.

12.115.2.45 emacspeak-m-player-next-track

`emacspeak-m-player-next-track` [Command]

Move to next track.

12.115.2.46 emacspeak-m-player-pan

`emacspeak-m-player-pan` [Command]

Pan from left to right and back from right to left one step at a time.

12.115.2.47 emacspeak-m-player-pause

`emacspeak-m-player-pause` [Command]

Pause or unpause media player.

12.115.2.48 emacspeak-m-player-persist-process

`emacspeak-m-player-persist-process` (**&optional** *name*) [Command]

Persists current m-player process instance by renaming its buffer.
Optional interactive prefix arg prompts for name to use for player.

(fn &optional NAME)

12.115.2.49 emacspeak-m-player-play-rss

`emacspeak-m-player-play-rss` (*rss-url*) [Command]

Play an RSS stream by converting to an M3U playlist.

(fn RSS-URL)

12.115.2.50 emacspeak-m-player-play-tracks-jump

`emacspeak-m-player-play-tracks-jump` (*step*) [Command]

Move within the play tree.

(fn STEP)

12.115.2.51 emacspeak-m-player-play-tree-up

`emacspeak-m-player-play-tree-up` (*step*) [Command]

Move within the play tree.

(fn STEP)

12.115.2.52 emacspeak-m-player-pop-to-player

`emacspeak-m-player-pop-to-player` [Command]

Pop to m-player buffer.

12.115.2.53 emacspeak-m-player-previous-track

`emacspeak-m-player-previous-track` [Command]

Move to previous track.

12.115.2.54 emacspeak-m-player-quit

`emacspeak-m-player-quit` [Command]

Quit media player.

12.115.2.55 emacspeak-m-player-reset-options

`emacspeak-m-player-reset-options` [Command]

Reset MPlayer options to initial defaults.

12.115.2.56 emacspeak-m-player-reset-speed

`emacspeak-m-player-reset-speed` [Command]

Reset playing speed to normal.

12.115.2.57 emacspeak-m-player-restore-process

`emacspeak-m-player-restore-process` [Command]

Restore emacspeak-m-player-process from current buffer.
Check first if current buffer is in emacspeak-m-player-mode.

12.115.2.58 emacspeak-m-player-right-channel

`emacspeak-m-player-right-channel` [Command]

Play on right channel.

12.115.2.59 emacspeak-m-player-scale-speed

`emacspeak-m-player-scale-speed` (*factor*) [Command]

Scale speed by specified factor.

(fn FACTOR)

12.115.2.60 emacspeak-m-player-seek-absolute

`emacspeak-m-player-seek-absolute` (*pos*) [Command]

Seek to absolute specified pos in seconds.
The time position can also be specified as HH:MM:SS.

(fn POS)

12.115.2.61 emacspeak-m-player-seek-percentage

`emacspeak-m-player-seek-percentage` (*pos*) [Command]

Seek to absolute specified pos in percent.

(fn POS)

12.115.2.62 emacspeak-m-player-seek-relative

`emacspeak-m-player-seek-relative` (*offset*) [Command]

Seek by offset into stream from current position.
Time offset can be specified as a number of seconds, or as HH:MM:SS.

(fn OFFSET)

12.115.2.63 emacspeak-m-player-set-clip-end

`emacspeak-m-player-set-clip-end` [Command]

Set end of clip marker.

12.115.2.64 emacspeak-m-player-set-clip-start

`emacspeak-m-player-set-clip-start` [Command]

Set start of clip marker.

12.115.2.65 emacspeak-m-player-shuffle

`emacspeak-m-player-shuffle` [Command]

`C-e` :

`<fn>` :

Launch M-Player with shuffle turned on.

12.115.2.66 emacspeak-m-player-slave-command

`emacspeak-m-player-slave-command` (*command*) [Command]

Dispatch slave command read from minibuffer.

(fn COMMAND)

12.115.2.67 emacspeak-m-player-slower

`emacspeak-m-player-slower` [Command]

Slow down playback.
This affects pitch.

12.115.2.68 emacspeak-m-player-stream-info

`emacspeak-m-player-stream-info` (**&optional** *toggle-cue*) [Command]

Speak and display metadata if available.
Interactive prefix arg toggles automatic cueing of ICY info updates.

(fn &optional TOGGLE-CUE)

12.115.2.69 emacspeak-m-player-url

`emacspeak-m-player-url` (*url* **&optional** *playlist-p*) [Command]

C-, u

C-x @ a u

Call emacspeak-m-player with specified URL.

(fn URL &optional PLAYLIST-P)

12.115.2.70 emacspeak-m-player-using-hrtf

`emacspeak-m-player-using-hrtf` [Command]

C-; '

C-x @ h '

Add af resample=48000,hrtf to startup options.
This will work if the soundcard is set to 48000.

12.115.2.71 emacspeak-m-player-using-openal

`emacspeak-m-player-using-openal` (*resource* **&optional** *play-list*) [Command]

C-; ;

C-x @ h ;

Use openal as the audio output driver. Adding hrtf=true to ~/.alsoftrc gives HRTF. You need to have openal installed and have an mplayer that has been compiled with openal support to use this feature. Calling spec is like 'emacspeak-m-player'.

(fn RESOURCE &optional PLAY-LIST)

12.115.2.72 emacspeak-m-player-volume-change

`emacspeak-m-player-volume-change` (*value*) [Command]

Change volume to specified absolute value.

(fn VALUE)

12.115.2.73 emacspeak-m-player-volume-down

`emacspeak-m-player-volume-down` [Command]

Decrease volume.

12.115.2.74 emacspeak-m-player-volume-set

`emacspeak-m-player-volume-set` (**&optional** *arg*) [Command]

Set Volume in steps from 1 to 9.

(fn &optional ARG)

12.115.2.75 emacspeak-m-player-volume-up

`emacspeak-m-player-volume-up` [Command]

Increase volume.

12.115.2.76 emacspeak-m-player-write-clip

`emacspeak-m-player-write-clip` [Command]

Invoke mp3splt to clip selected range in current file.

12.115.2.77 emacspeak-m-player-youtube-player

`emacspeak-m-player-youtube-player` (*url* **&optional** *best*) [Command]

C-, y

C-x @ a y

Use youtube-dl and mplayer to stream the audio for YouTube content.
Default is to pick smallest (lowest quality) audio format.
Optional prefix arg 'best' chooses highest quality.

(fn URL &optional BEST)

12.115.2.78 emacspeak-multimedia

`emacspeak-multimedia` [Command]

C-e ;

<fn> ;

Start or control Emacspeak multimedia player.

Uses current context to prompt for media to play.
Controls media playback when already playing a stream.

key binding
— ——-

C-l ladspa
RET emacspeak-m-player-load
ESC Prefix Command
SPC emacspeak-m-player-pause
% emacspeak-m-player-display-percent
(emacspeak-m-player-left-channel
) emacspeak-m-player-right-channel
+ emacspeak-m-player-volume-up
, emacspeak-m-player-backward-10s
- emacspeak-m-player-volume-down
. emacspeak-m-player-forward-10s
/ emacspeak-m-player-restore-process
1 .. 9 emacspeak-m-player-volume-set

; emacspeak-m-player-pop-to-player
< emacspeak-m-player-backward-1min
= emacspeak-m-player-volume-up
> emacspeak-m-player-forward-1min
? emacspeak-m-player-display-position
A emacspeak-m-player-amark-add
C emacspeak-m-player-clear-filters
E emacspeak-m-player-add-equalizer
G emacspeak-m-player-seek-percentage
L emacspeak-m-player-locate-media
M emacspeak-m-player-display-metadata
O emacspeak-m-player-reset-options
P emacspeak-m-player-apply-reverb-preset
Q emacspeak-m-player-quit
R emacspeak-m-player-edit-reverb
S emacspeak-m-player-amark-save
[emacspeak-m-player-slower
\ emacspeak-m-player-persist-process
] emacspeak-m-player-faster
a emacspeak-m-player-add-autopan
b emacspeak-wizards-view-buffers-filtered-by-m-player-mode
c emacspeak-m-player-slave-command
d emacspeak-m-player-delete-filter
e emacspeak-m-player-equalizer-preset
f emacspeak-m-player-add-filter
g emacspeak-m-player-seek-absolute
i emacspeak-m-player-stream-info
j emacspeak-m-player-amark-jump
k emacspeak-m-player-quit
l emacspeak-m-player-get-length
m emacspeak-m-player-mode-line
n emacspeak-m-player-next-track
o emacspeak-m-player-customize-options
p emacspeak-m-player-previous-track
q bury-buffer
r emacspeak-m-player-seek-relative
s emacspeak-m-player-scale-speed
t emacspeak-m-player-play-tracks-jump
u emacspeak-m-player-url
v emacspeak-m-player-volume-change
w emacspeak-m-player-write-clip
x emacspeak-m-player-pan
z emacspeak-m-player-add-autosat
{ emacspeak-m-player-half-speed
} emacspeak-m-player-double-speed
DEL emacspeak-m-player-reset-speed
<down> emacspeak-m-player-forward-1min

<end> emacspeak-m-player-end-of-track
<home> emacspeak-m-player-beginning-of-track
<left> emacspeak-m-player-backward-10s
<next> emacspeak-m-player-forward-10min
<prior> emacspeak-m-player-backward-10min
<right> emacspeak-m-player-forward-10s
<up> emacspeak-m-player-backward-1min

M-, emacspeak-m-player-set-clip-start
M-. emacspeak-m-player-set-clip-end
M-l emacspeak-m-player-load-playlist

.

12.115.3 emacspeak-m-player Options

`User Option` *emacspeak-m-player-program* [Variable]
Media player program.

`User Option` *emacspeak-m-player-reverb-filter* [Variable]
Tap Reverb Settings.

`User Option` *emacspeak-m-player-youtube-dl* [Variable]
YouTube download tool

`User Option` *emacspeak-media-location-bindings* [Variable]
Map specific key sequences to launching MPlayer accelerators on a specific directory.

12.116 emacspeak-magit

MAGIT == Git interface in Emacs git clone git://github.com/magit/magit.git

12.116.1 emacspeak-magit Options

`User Option` *voice-magit-bisect-bad* [Variable]
Personality used for magit-bisect-bad This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

`User Option` *voice-magit-bisect-good* [Variable]
Personality used for magit-bisect-good This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-magit-bisect-skip* [Variable]
Personality used for magit-bisect-skip This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-magit-blame-date* [Variable]
Personality used for magit-blame-date This personality uses voice-bolden-and-animate whose effect can be changed globally by customizing voice-bolden-and-animate-settings.

`User Option` *voice-magit-blame-dimmed* [Variable]
Personality used for magit-blame-dimmed This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

`User Option` *voice-magit-blame-hash* [Variable]
Personality used for magit-blame-hash This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-magit-blame-heading* [Variable]
Personality used for magit-blame-heading This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-magit-blame-highlight* [Variable]
Personality used for magit-blame-highlight This personality uses voice-brighten whose effect can be changed globally by customizing voice-brighten-settings.

`User Option` *voice-magit-blame-name* [Variable]
Personality used for magit-blame-name This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

`User Option` *voice-magit-blame-summary* [Variable]
Personality used for magit-blame-summary This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-magit-branch-current* [Variable]
Personality used for magit-branch-current This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-magit-branch-local* [Variable]
Personality used for magit-branch-local This personality uses voice-brighten whose effect can be changed globally by customizing voice-brighten-settings.

`User Option` *voice-magit-branch-remote* [Variable]
Personality used for magit-branch-remote This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-magit-branch-remote-head* [Variable]
Personality used for magit-branch-remote-head This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-magit-branch-upstream* [Variable]
Personality used for magit-branch-upstream This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

`User Option` *voice-magit-cherry-equivalent* [Variable]
Personality used for magit-cherry-equivalent This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-magit-cherry-unmatched* [Variable]
Personality used for magit-cherry-unmatched This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

User Option *voice-magit-diff-added* [Variable]

Personality used for magit-diff-added This personality uses voice-animate-extra whose effect can be changed globally by customizing voice-animate-extra-settings.

User Option *voice-magit-diff-added-highlight* [Variable]

Personality used for magit-diff-added-highlight This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

User Option *voice-magit-diff-base* [Variable]

Personality used for magit-diff-base This personality uses voice-annotate whose effect can be changed globally by customizing voice-annotate-settings.

User Option *voice-magit-diff-base-highlight* [Variable]

Personality used for magit-diff-base-highlight This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

User Option *voice-magit-diff-conflict-heading* [Variable]

Personality used for magit-diff-conflict-heading This personality uses voice-bolden-extra whose effect can be changed globally by customizing voice-bolden-extra-settings.

User Option *voice-magit-diff-context* [Variable]

Personality used for magit-diff-context This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

User Option *voice-magit-diff-context-highlight* [Variable]

Personality used for magit-diff-context-highlight This personality uses voice-brighten whose effect can be changed globally by customizing voice-brighten-settings.

User Option *voice-magit-diff-file-heading* [Variable]

Personality used for magit-diff-file-heading This personality uses voice-brighten whose effect can be changed globally by customizing voice-brighten-settings.

User Option *voice-magit-diff-file-heading-highlight* [Variable]

Personality used for magit-diff-file-heading-highlight This personality uses voice-bolden-extra whose effect can be changed globally by customizing voice-bolden-extra-settings.

User Option *voice-magit-diff-file-heading-selection* [Variable]

Personality used for magit-diff-file-heading-selection This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

User Option *voice-magit-diff-hunk-heading* [Variable]

Personality used for magit-diff-hunk-heading This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

User Option *voice-magit-diff-hunk-heading-highlight* [Variable]

Personality used for magit-diff-hunk-heading-highlight This personality uses voice-brighten whose effect can be changed globally by customizing voice-brighten-settings.

User Option *voice-magit-diff-hunk-heading-selection* [Variable]

Personality used for magit-diff-hunk-heading-selection This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-magit-diff-hunk-region* [Variable]
Personality used for magit-diff-hunk-region This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

`User Option` *voice-magit-diff-lines-boundary* [Variable]
Personality used for magit-diff-lines-boundary This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-magit-diff-lines-heading* [Variable]
Personality used for magit-diff-lines-heading This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-magit-diff-our* [Variable]
Personality used for magit-diff-our This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

`User Option` *voice-magit-diff-our-highlight* [Variable]
Personality used for magit-diff-our-highlight This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-magit-diff-removed* [Variable]
Personality used for magit-diff-removed This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-magit-diff-removed-highlight* [Variable]
Personality used for magit-diff-removed-highlight This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

`User Option` *voice-magit-diff-revision-summary* [Variable]
Personality used for magit-diff-revision-summary This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-magit-diff-revision-summary-highlight* [Variable]
Personality used for magit-diff-revision-summary-highlight This personality uses voice-brighten whose effect can be changed globally by customizing voice-brighten-settings.

`User Option` *voice-magit-diff-their* [Variable]
Personality used for magit-diff-their This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

`User Option` *voice-magit-diff-their-highlight* [Variable]
Personality used for magit-diff-their-highlight This personality uses voice-bolden-and-animate whose effect can be changed globally by customizing voice-bolden-and-animate-settings.

`User Option` *voice-magit-diff-whitespace-warning* [Variable]
Personality used for magit-diff-whitespace-warning This personality uses voice-monotone-light whose effect can be changed globally by customizing voice-monotone-light-settings.

User Option *voice-magit-diffstat-added* [Variable]
Personality used for magit-diffstat-added This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

User Option *voice-magit-diffstat-removed* [Variable]
Personality used for magit-diffstat-removed This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

User Option *voice-magit-dimmed* [Variable]
Personality used for magit-dimmed This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

User Option *voice-magit-filename* [Variable]
Personality used for magit-filename This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

User Option *voice-magit-hash* [Variable]
Personality used for magit-hash This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

User Option *voice-magit-head* [Variable]
Personality used for magit-head This personality uses voice-bolden-medium whose effect can be changed globally by customizing voice-bolden-medium-settings.

User Option *voice-magit-header-line* [Variable]
Personality used for magit-header-line This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

User Option *voice-magit-header-line-key* [Variable]
Personality used for magit-header-line-key This personality uses voice-bolden-extra whose effect can be changed globally by customizing voice-bolden-extra-settings.

User Option *voice-magit-header-line-log-select* [Variable]
Personality used for magit-header-line-log-select This personality uses voice-highlight whose effect can be changed globally by customizing voice-highlight-settings.

User Option *voice-magit-keyword* [Variable]
Personality used for magit-keyword This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

User Option *voice-magit-keyword-squash* [Variable]
Personality used for magit-keyword-squash This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

User Option *voice-magit-log-author* [Variable]
Personality used for magit-log-author This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

User Option *voice-magit-log-date* [Variable]
Personality used for magit-log-date This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-magit-log-graph* [Variable]
Personality used for magit-log-graph This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-magit-reflog-amend* [Variable]
Personality used for magit-reflog-amend This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

`User Option` *voice-magit-reflog-checkout* [Variable]
Personality used for magit-reflog-checkout This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

`User Option` *voice-magit-reflog-cherry-pick* [Variable]
Personality used for magit-reflog-cherry-pick This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-magit-reflog-commit* [Variable]
Personality used for magit-reflog-commit This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-magit-reflog-merge* [Variable]
Personality used for magit-reflog-merge This personality uses voice-annotate whose effect can be changed globally by customizing voice-annotate-settings.

`User Option` *voice-magit-reflog-other* [Variable]
Personality used for magit-reflog-other This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-magit-reflog-rebase* [Variable]
Personality used for magit-reflog-rebase This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-magit-reflog-remote* [Variable]
Personality used for magit-reflog-remote This personality uses voice-annotate whose effect can be changed globally by customizing voice-annotate-settings.

`User Option` *voice-magit-reflog-reset* [Variable]
Personality used for magit-reflog-reset This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-magit-refname* [Variable]
Personality used for magit-refname This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-magit-refname-stash* [Variable]
Personality used for magit-refname-stash This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-magit-refname-wip* [Variable]
Personality used for magit-refname-wip This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-magit-section-heading* [Variable]
Personality used for magit-section-heading This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-magit-section-heading-selection* [Variable]
Personality used for magit-section-heading-selection This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-magit-section-highlight* [Variable]
Personality used for magit-section-highlight This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

`User Option` *voice-magit-section-secondary-heading* [Variable]
Personality used for magit-section-secondary-heading This personality uses voice-bolden-medium whose effect can be changed globally by customizing voice-bolden-medium-settings.

`User Option` *voice-magit-sequence-done* [Variable]
Personality used for magit-sequence-done This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-magit-sequence-drop* [Variable]
Personality used for magit-sequence-drop This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-magit-sequence-head* [Variable]
Personality used for magit-sequence-head This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-magit-sequence-onto* [Variable]
Personality used for magit-sequence-onto This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-magit-sequence-part* [Variable]
Personality used for magit-sequence-part This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-magit-sequence-pick* [Variable]
Personality used for magit-sequence-pick This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

`User Option` *voice-magit-sequence-stop* [Variable]
Personality used for magit-sequence-stop This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

`User Option` *voice-magit-signature-bad* [Variable]
Personality used for magit-signature-bad This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

User Option *voice-magit-signature-error* [Variable]
Personality used for magit-signature-error This personality uses voice-bolden-and-animate whose effect can be changed globally by customizing voice-bolden-and-animate-settings.

User Option *voice-magit-signature-expired* [Variable]
Personality used for magit-signature-expired This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

User Option *voice-magit-signature-expired-key* [Variable]
Personality used for magit-signature-expired-key This personality uses voice-monotone-medium whose effect can be changed globally by customizing voice-monotone-medium-settings.

User Option *voice-magit-signature-good* [Variable]
Personality used for magit-signature-good This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

User Option *voice-magit-signature-revoked* [Variable]
Personality used for magit-signature-revoked This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

User Option *voice-magit-signature-untrusted* [Variable]
Personality used for magit-signature-untrusted This personality uses voice-bolden-and-animate whose effect can be changed globally by customizing voice-bolden-and-animate-settings.

User Option *voice-magit-tag* [Variable]
Personality used for magit-tag This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

12.117 emacspeak-make-mode

This module speech enables make-mode

12.117.1 emacspeak-make-mode Options

User Option *voice-makefile-makepp-perl* [Variable]
Personality used for makefile-makepp-perl This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

User Option *voice-makefile-shell* [Variable]
Personality used for makefile-shell This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

User Option *voice-makefile-space* [Variable]
Personality used for makefile-space This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

User Option *voice-makefile-targets* [Variable]
Personality used for makefile-targets This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

12.118 emacspeak-man

Provide additional advice to man-mode

12.118.1 Emacspeak-Man Commands

12.118.1.1 emacspeak-man-browse-man-page

`emacspeak-man-browse-man-page` [Command]

> Browse the man page –read it a paragraph at a time

12.118.1.2 emacspeak-man-speak-this-section

`emacspeak-man-speak-this-section` [Command]

> Speak current section

12.118.2 emacspeak-man Options

`User Option` *voice-Man-overstrike* [Variable]

> Personality used for Man-overstrike This personality uses voice-bolden-medium whose effect can be changed globally by customizing voice-bolden-medium-settings.

`User Option` *voice-Man-reverse* [Variable]

> Personality used for Man-reverse This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

`User Option` *voice-Man-underline* [Variable]

> Personality used for Man-underline This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

12.119 emacspeak-markdown

MARKDOWN == Light-weight markup. This module speech-enables markdown.el

12.119.1 emacspeak-markdown Options

`User Option` *voice-markdown-header-face-1* [Variable]

> Personality used for markdown-header-face-1 This personality uses voice-bolden-medium whose effect can be changed globally by customizing voice-bolden-medium-settings.

`User Option` *voice-markdown-header-face-2* [Variable]

> Personality used for markdown-header-face-2 This personality uses voice-bolden-and-animate whose effect can be changed globally by customizing voice-bolden-and-animate-settings.

`User Option` *voice-markdown-header-face-3* [Variable]

> Personality used for markdown-header-face-3 This personality uses voice-bolden-extra whose effect can be changed globally by customizing voice-bolden-extra-settings.

`User Option` *voice-markdown-header-face-4* [Variable]

Personality used for markdown-header-face-4 This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

`User Option` *voice-markdown-header-face-5* [Variable]

Personality used for markdown-header-face-5 This personality uses voice-lighten-extra whose effect can be changed globally by customizing voice-lighten-extra-settings.

`User Option` *voice-markdown-header-face-6* [Variable]

Personality used for markdown-header-face-6 This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

12.120 emacspeak-maths

12.120.1 Setup

Do not try what follows until you have read js/node/README.org and successfully set up nvm (Node Version Manager) as described there.

12.120.2 Technical Overview

Spoken mathematics on the emacspeak audio desktop. Use a NodeJS based speech-rule-engine for Mathematics as the backend for processing mathematical markup. The result of this processing is an annotated S-expression that is rendered via Emacspeak's speech facilities. Annotations follow Aural CSS as implemented in Emacspeak, This allows us to map these expressions to aural properties supported by specific TTS engines.

12.120.3 Basic Usage:

Start the server/client: M-x emacspeak-maths-start. Once the server and client are started, you can browse any number of math expressions using the emacspeak-maths-navigator described below. Note: In general, once everything is configured correctly, using the maths navigator automatically starts the server and client. Invoke the Navigator using s-spc — this is the <windows> key on Linux. Now you can use these keys:

- Show Output <o> Switch to output buffer and quit Maths Navigator
- Enter: <SPC> Enter a LaTeX expression.
- Smart-Enter: <enter> Enter the guessed expression with no prompting.
- Alt-Text <a> Process alt-text under point as LaTeX.
- Down <down> Move down a level.
- Up <up> Move up a level.
- Left <left> Move left.
- Right <right> Move right.
- Exit <any other key> Exit navigator.

The current expression is spoken after each of the above commands. It is also displayed in a special buffer *Spoken Math*. That buffer holds all previously generated output, And Emacs commands forward-page and backward-page can be used to move through each chunk of output.

12.120.4 Emacspeak-Maths Commands

12.120.4.1 emacspeak-maths-depth

`emacspeak-maths-depth` [Command]

Move depth in current Math expression. (auto-generated)

12.120.4.2 emacspeak-maths-down

`emacspeak-maths-down` [Command]

Move down in current Math expression. (auto-generated)

12.120.4.3 emacspeak-maths-enter

`emacspeak-maths-enter` (*latex*) [Command]

Send a LaTeX expression to Maths server.
Tries to guess default based on context.

(fn LATEX)

12.120.4.4 emacspeak-maths-enter-guess

`emacspeak-maths-enter-guess` [Command]

Send the guessed LaTeX expression to Maths server.
Guess is based on context.

12.120.4.5 emacspeak-maths-flush-output

`emacspeak-maths-flush-output` [Command]

Flush client buffer if things go out of sync.

12.120.4.6 emacspeak-maths-left

`emacspeak-maths-left` [Command]

Move left in current Math expression. (auto-generated)

12.120.4.7 emacspeak-maths-restart

`emacspeak-maths-restart` [Command]

Restart Node math-server if running. Otherwise starts a new one.

12.120.4.8 emacspeak-maths-right

`emacspeak-maths-right` [Command]

Move right in current Math expression. (auto-generated)

12.120.4.9 emacspeak-maths-root

`emacspeak-maths-root` [Command]

Move root in current Math expression. (auto-generated)

12.120.4.10 emacspeak-maths-shutdown

`emacspeak-maths-shutdown` [Command]

Shutdown client and server processes.

12.120.4.11 emacspeak-maths-speak-alt

`emacspeak-maths-speak-alt` [Command]

Speak alt text as Maths.
For use on Wikipedia pages for example.

12.120.4.12 emacspeak-maths-spoken-mode

`emacspeak-maths-spoken-mode` [Command]

Special mode for interacting with Spoken Math.

This mode is used by the special buffer that displays spoken math
returned from the Node server.
This mode is similar to Emacs' 'view-mode'.
see the key-binding list at the end of this description.
Emacs online help facility to look up help on these commands.

key binding
— ——-

[backward-page
] forward-page
h emacspeak-maths-left
j emacspeak-maths-down
k emacspeak-maths-up
l emacspeak-maths-right

In addition to any hooks its parent mode 'special-mode' might have run,
this mode runs the hook 'emacspeak-maths-spoken-mode-hook', as the final or penultimate step
during initialization.

12.120.4.13 emacspeak-maths-start

`emacspeak-maths-start` [Command]

Start Maths server bridge.

12.120.4.14 emacspeak-maths-switch-to-output

`emacspeak-maths-switch-to-output` [Command]

Switch to output buffer.

12.120.4.15 emacspeak-maths-up

`emacspeak-maths-up` [Command]
Move up in current Math expression. (auto-generated)

12.121 emacspeak-message

advice for posting message commands

12.121.1 emacspeak-message Options

`User Option` *emacspeak-message-punctuation-mode* [Variable]
Pronunciation mode to use for message buffers.

`User Option` *voice-message-cited-text* [Variable]
Personality used for message-cited-text This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-message-header-cc* [Variable]
Personality used for message-header-cc This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-message-header-name* [Variable]
Personality used for message-header-name This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

`User Option` *voice-message-header-newsgroups* [Variable]
Personality used for message-header-newsgroups This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-message-header-other* [Variable]
Personality used for message-header-other This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-message-header-subject* [Variable]
Personality used for message-header-subject This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-message-header-to* [Variable]
Personality used for message-header-to This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-message-header-xheader* [Variable]
Personality used for message-header-xheader This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-message-mml* [Variable]
Personality used for message-mml This personality uses voice-brighten whose effect can be changed globally by customizing voice-brighten-settings.

`User Option` *voice-message-separator* [Variable]
Personality used for message-separator This personality uses voice-bolden-extra whose effect can be changed globally by customizing voice-bolden-extra-settings.

12.122 emacspeak-metapost

Speech-enables metapost mode. metapost is a powerful drawing package typically installed as mpost by modern TeX installations.

12.123 emacspeak-midge

This module speech enables midge. Midge is a MIDI composer/editor tool. From the package README file: ; Midge, for midi generator, is a text to midi translator. ; It creates type 1 (ie multitrack) midi files from text ; descriptions of music. It is a single perl script, which ; does not require any additional modules. The package also provides a convenient emacs mode for editing and playing midge files. Midge's homepage is at: http://www.dmriley.demon.co.uk/code/midge/

12.124 emacspeak-mines

MINES == Minesweeper game in emacs. The game itself provides a fully keyboard driven interface. In addition, Emacspeak provides these additional interactive commands:

- *SPC* Speak current cell.
- `.` Speak neighbors of current cell.
- `,` Speak number of marks
- `a` Move to beginning of row.
- `e` Move to end of row.
- `g` Move to specified cell
- `s` Move to next uncovered cell.
- `/` Speak number of remaining uncovered cells.
- `'` Speaks entire board.

Speaking cell neighbors uses appropriate clause boundaries to group related cells — neighbors are read left-to-right, top-to-bottom. Moving to the left/right edge of the board produces an appropriate auditory icon.

12.124.1 Emacspeak-Mines Commands

12.124.1.1 emacspeak-mines-beginning-of-row

`emacspeak-mines-beginning-of-row` [Command]

Move to beginning of row

12.124.1.2 emacspeak-mines-end-of-row

`emacspeak-mines-end-of-row` [Command]

Move to end of row

12.124.1.3 emacspeak-mines-goto

`emacspeak-mines-goto` (*index*) [Command]

Move to specified cell.

(fn INDEX)

12.124.1.4 emacspeak-mines-jump-to-uncovered-cell

`emacspeak-mines-jump-to-uncovered-cell` (*from-beginning*) [Command]

Jump to next uncovered cell. With interactive prefix-arg, jump to beginning of board before searching.

(fn FROM-BEGINNING)

12.124.1.5 emacspeak-mines-speak-board

`emacspeak-mines-speak-board` [Command]

Speak the board.

12.124.1.6 emacspeak-mines-speak-cell

`emacspeak-mines-speak-cell` [Command]

Speak current cell.

12.124.1.7 emacspeak-mines-speak-mark-count

`emacspeak-mines-speak-mark-count` [Command]

Count and speak number of marks.

12.124.1.8 emacspeak-mines-speak-neighbors

`emacspeak-mines-speak-neighbors` [Command]

Speak neighboring cells in sorted order.

12.124.1.9 emacspeak-mines-speak-uncovered-count

`emacspeak-mines-speak-uncovered-count` [Command]

Speak number of uncovered cells.

12.125 emacspeak-mspools

Speech-enable mspools –a package that lets you monitor multiple maildrops

12.126 emacspeak-muggles

MUGGLES == Emacspeak spells for power-users. This module implements no new functionality — contrast with emacspeak-wizards. Instead, it uses package hydra to provide convenience key-bindings that access existing Emacspeak functionality. You need to install package Hydra first: '`M-x package-install hydra`'.

Note that on newer versions of Emacs, loading this module will attempt to automatically install package hydra if it is not found.

12.126.1 Using Hydras

See the high-level documentation for the Hydra package by executing *`C-h C-j`* hydra *`RET`*. The documentation in this section uses the same terminology as shown in the hydra package documentation.

12.126.2 Implemented Muggles

- Brightness: *print* Control display brightness using xbacklight.
- Navigate: *s-n* Navigate with ease.
- org-mode structure nav: *C-c C-SPC* Structure navigation for org-mode.
- org-mode tables: *C-c t* Table UI for org-mode tables.
- hideshow: C-, h Provide HideShow bindings.
- toggle-option: *C-c o* Single binding for toggling options.
- Repeatable-Yank: *C-y* Smart yank
- Vuiet Explorer: *C-; v* Vuiet Music Explorer and Player
- undo-only/undo-redo: *C-/* Undo-only on `/` and undo-redo on `\`
- emacspeak-maths *s-SPC* Speak and browse math.

Emacspeak automatically speaks Hydra hints when displayed. To silence all Hydra hints, set hydra-is-helpful to nil. To temporarily silence speaking of Hydra hints, Muggles can bind command `emacspeak-hydra-toggle-talkative`. As an example, Muggle '`ViewMode`' binds `s` to this command.

12.126.3 Emacspeak-Muggles Commands

12.126.3.1 emacspeak-muggles-brightness/body

`emacspeak-muggles-brightness/body` [Command]

<print>

Call the body in the "emacspeak-muggles-brightness" hydra.

The heads for the associated hydra are:

"?": '(emacspeak-hydra-self-help "emacspeak-muggles-brightness")',
"s": 'xbacklight-set',
"g": 'xbacklight-get',
"t": 'emacspeak-hydra-toggle-talkative',
"<print>": 'xbacklight-black',
"0": 'xbacklight-black',
"1": 'xbacklight-white',
"d": 'xbacklight-decrement',
"i": 'xbacklight-increment',
"SPC": 'xbacklight-increment'

The body can be accessed via 'emacspeak-muggles-brightness/body'.

12.126.3.2 emacspeak-muggles-hideshow/body

`emacspeak-muggles-hideshow/body` [Command]

C-, h

C-x @ a h

Call the body in the "emacspeak-muggles-hideshow" hydra.

The heads for the associated hydra are:

"?": '(emacspeak-hydra-self-help "emacspeak-muggles-hideshow")',
"h": 'hs-hide-block',
"s": 'hs-show-block',
"H": 'hs-hide-all',
"S": 'hs-show-all',
"a": 'hs-show-all',
"l": 'hs-hide-level',
"i": 'hs-hide-initial-comment-block'

The body can be accessed via 'emacspeak-muggles-hideshow/body'.

12.126.3.3 emacspeak-muggles-ido-yank

`emacspeak-muggles-ido-yank` [Command]

C-M-y

Pick what to yank using ido completion.

12.126.3.4 emacspeak-muggles-lispy-or-sp

`emacspeak-muggles-lispy-or-sp` [Command]

Toggle between lispy and smartparens.

12.126.3.5 emacspeak-muggles-maths-navigator/body

`emacspeak-muggles-maths-navigator/body` [Command]

s-SPC

Call the body in the "emacspeak-muggles-maths-navigator" hydra.

The heads for the associated hydra are:

"o": 'emacspeak-maths-switch-to-output',
"RET": 'emacspeak-maths-enter-guess',
"SPC": 'emacspeak-maths-enter',
"a": 'emacspeak-maths-speak-alt',
"d": 'emacspeak-maths-depth',
"r": 'emacspeak-maths-root',
"<up>": 'emacspeak-maths-up',
"<down>": 'emacspeak-maths-down',
"<left>": 'emacspeak-maths-left',
"<right>": 'emacspeak-maths-right'

The body can be accessed via 'emacspeak-muggles-maths-navigator/body'.

12.126.3.6 emacspeak-muggles-navigate/body

`emacspeak-muggles-navigate/body` [Command]

s-n

Call the body in the "emacspeak-muggles-navigate" hydra.

The heads for the associated hydra are:

"?": '(emacspeak-hydra-self-help "emacspeak-muggles-navigate")',
"s": 'emacspeak-hydra-toggle-talkative',
"n": 'next-line',
"p": 'previous-line',
"f": 'forward-char',
"b": 'backward-char',
"a": 'beginning-of-line',
"e": 'move-end-of-line',
"j": 'next-line',
"k": 'previous-line',
"v": 'scroll-up-command',
"V": 'scroll-down-command',
"l": 'recenter-top-bottom',
"<": 'beginning-of-buffer',
">": 'end-of-buffer'

The body can be accessed via 'emacspeak-muggles-navigate/body'.

12.126.3.7 emacspeak-muggles-org-nav/body

`emacspeak-muggles-org-nav/body` [Command]

Call the body in the "emacspeak-muggles-org-nav" hydra.

The heads for the associated hydra are:

"?": '(emacspeak-hydra-self-help "emacspeak-muggles-org-nav")',
"SPC": 'emacspeak-outline-speak-this-heading',
"n": 'emacspeak-outline-speak-next-heading',
"p": 'emacspeak-outline-speak-previous-heading',
"f": 'org-forward-heading-same-level',
"b": 'org-backward-heading-same-level',
"u": 'outline-up-heading',
"g": 'org-goto'

The body can be accessed via 'emacspeak-muggles-org-nav/body'.

12.126.3.8 emacspeak-muggles-org-table/body

`emacspeak-muggles-org-table/body` [Command]

Call the body in the "emacspeak-muggles-org-table" hydra.

The heads for the associated hydra are:

"?": ‘(emacspeak-hydra-self-help "emacspeak-muggles-org-table")’,
"j": ‘org-table-next-row’,
"k": ‘org-table-previous-row’,
"h": ‘org-table-previous-field’,
"l": ‘org-table-next-field’,
"SPC": ‘emacspeak-org-table-speak-current-element’,
".": ‘emacspeak-org-table-speak-coordinates’,
"b": ‘emacspeak-org-table-speak-both-headers-and-element’,
"r": ‘emacspeak-org-table-speak-row-header-and-element’,
"c": ‘emacspeak-org-table-speak-column-header-and-element’

The body can be accessed via ‘emacspeak-muggles-org-table/body’.

12.126.3.9 emacspeak-muggles-toggle-option/body

`emacspeak-muggles-toggle-option/body` [Command]

C-c o

Call the body in the "emacspeak-muggles-toggle-option" hydra.

The heads for the associated hydra are:

"?": ‘(emacspeak-hydra-self-help "emacspeak-muggles-toggle-option")’,
"C-f": ‘(call-interactively #’folding-mode)’,
"C": ‘(call-interactively #’flycheck-mode)’,
"F": ‘(call-interactively #’flyspell-mode)’,
"a": ‘(call-interactively #’abbrev-mode)’,
"d": ‘(call-interactively #’toggle-debug-on-error)’,
"f": ‘(call-interactively #’auto-fill-mode)’,
"g": ‘(call-interactively #’toggle-debug-on-quit)’,
"h": ‘(setq hydra-is-helpful (not hydra-is-helpful))’,
"i": ‘(call-interactively #’ido-everywhere)’,
"I": ‘(call-interactively #’flx-ido-mode)’,
"p": ‘emacspeak-muggles-lispy-or-sp’,
"t": ‘(call-interactively #’toggle-truncate-lines)’,
"u": ‘(call-interactively #’ido-ubiquitous-mode)’,
"q": ‘nil’

The body can be accessed via ‘emacspeak-muggles-toggle-option/body’.

12.126.3.10 emacspeak-muggles-undo-only/undo-redo/body

`emacspeak-muggles-undo-only/undo-redo/body` [Command]

C-/

Call the body in the "emacspeak-muggles-undo-only/undo-redo" hydra.

The heads for the associated hydra are:

"?": ‘(emacspeak-hydra-self-help "emacspeak-muggles-undo-only/undo-redo")’,
"/": ‘undo-only’,
"\": ‘undo-redo’

The body can be accessed via ‘emacspeak-muggles-undo-only/undo-redo/body’.

12.126.3.11 emacspeak-muggles-vuiet/body

`emacspeak-muggles-vuiet/body` [Command]

C-; v

C-x @ h v

Call the body in the "emacspeak-muggles-vuiet" hydra.

The heads for the associated hydra are:

";": ‘vuiet-playing-track-lyrics’,
"=": ‘vuiet-player-volume-inc’,
"-": ‘vuiet-player-volume-dec’,
"A": ‘vuiet-play-artist-loved-tracks’,
"’": ‘vuiet-play-loved-tracks’,
",": ‘vuiet-seek-backward’,
".": ‘vuiet-seek-forward’,
"C-s": ‘vuiet-artist-info-search’,
"L": ‘vuiet-playing-artist-lastfm-page’,
"SPC": ‘vuiet-play-pause’,
"a": ‘vuiet-artist-info’,
"i": ‘emacspeak-vuiet-track-info’,
"l": ‘vuiet-love-track’,
"n": ‘vuiet-next’,
"p": ‘vuiet-play-artist’,
"r": ‘vuiet-replay’,
"s": ‘vuiet-stop’,
"t": ‘vuiet-play-track’,
"u": ‘vuiet-unlove-track’

The body can be accessed via ‘emacspeak-muggles-vuiet/body’.

12.126.3.12 emacspeak-muggles-yank-pop/body

`emacspeak-muggles-yank-pop/body` [Command]

Call the body in the "emacspeak-muggles-yank-pop" hydra.

The heads for the associated hydra are:

"?": ‘(emacspeak-hydra-self-help "emacspeak-muggles-yank-pop")’,
"C-y": ‘yank’,
"M-y": ‘yank-pop’,
"y": ‘(funcall-interactively #’yank-pop 1)’,
"Y": ‘(funcall-interactively #’yank-pop -1)’,
"i": ‘emacspeak-muggles-ido-yank’,
"l": ‘browse-kill-ring’

The body can be accessed via ‘emacspeak-muggles-yank-pop/body’.

12.127 emacspeak-muse

Speech enable Muse

12.127.1 emacspeak-muse Options

`User Option` *voice-muse-emphasis-1* [Variable]
Personality used for muse-emphasis-1 This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-muse-emphasis-2* [Variable]
Personality used for muse-emphasis-2 This personality uses voice-lighten-medium whose effect can be changed globally by customizing voice-lighten-medium-settings.

`User Option` *voice-muse-emphasis-3* [Variable]
Personality used for muse-emphasis-3 This personality uses voice-lighten-extra whose effect can be changed globally by customizing voice-lighten-extra-settings.

`User Option` *voice-muse-header-1* [Variable]
Personality used for muse-header-1 This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-muse-header-2* [Variable]
Personality used for muse-header-2 This personality uses voice-bolden-medium whose effect can be changed globally by customizing voice-bolden-medium-settings.

`User Option` *voice-muse-header-3* [Variable]
Personality used for muse-header-3 This personality uses voice-bolden-extra whose effect can be changed globally by customizing voice-bolden-extra-settings.

`User Option` *voice-muse-header-4* [Variable]
Personality used for muse-header-4 This personality uses voice-bolden-extra whose effect can be changed globally by customizing voice-bolden-extra-settings.

`User Option` *voice-muse-header-5* [Variable]
Personality used for muse-header-5 This personality uses voice-bolden-extra whose effect can be changed globally by customizing voice-bolden-extra-settings.

12.128 emacspeak-navi-mode

NAVI-MODE == Remote control for buffer navigation

12.129 emacspeak-net-utils

This module speech enables net-utils

12.130 emacspeak-newsticker

Newsticker provides a continuously updating newsticker using RSS Provides functionality similar to amphetadesk –but in pure elisp

12.131 emacspeak-nov

NOV == Yet Another EPub Reader Package nov.el is an alternative to Emacspeak's built-in EPub reader. This module speech-enables nov.el In addition, opening an epub using nov results in directory-specific settings being loaded from file *emacspeak-speak-directory-settings* — That file can set book-specific settings such as speech-rate and punctuation-mode among others.

12.132 emacspeak-nxml

nxml-mode is a new XML mode for emacs by James Clark. Package nxml is available from the Emacs package archive.

12.132.1 Emacspeak-Nxml Commands

12.132.1.1 emacspeak-nxml-summarize-outline

`emacspeak-nxml-summarize-outline` [Command]

Intelligent spoken display of current outline entry.

12.132.2 emacspeak-nxml Options

`User Option` *voice-nxml-attribute-colon* [Variable]

Personality used for nxml-attribute-colon This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-nxml-attribute-local-name* [Variable]

Personality used for nxml-attribute-local-name This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

`User Option` *voice-nxml-attribute-prefix* [Variable]

Personality used for nxml-attribute-prefix This personality uses voice-monotone-medium whose effect can be changed globally by customizing voice-monotone-medium-settings.

`User Option` *voice-nxml-attribute-value* [Variable]

Personality used for nxml-attribute-value This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-nxml-attribute-value-delimiter* [Variable]

Personality used for nxml-attribute-value-delimiter This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

`User Option` *voice-nxml-cdata-section-CDATA* [Variable]
Personality used for nxml-cdata-section-CDATA This personality uses voice-animate-extra whose effect can be changed globally by customizing voice-animate-extra-settings.

`User Option` *voice-nxml-cdata-section-content* [Variable]
Personality used for nxml-cdata-section-content This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-nxml-cdata-section-delimiter* [Variable]
Personality used for nxml-cdata-section-delimiter This personality uses voice-monotone-medium whose effect can be changed globally by customizing voice-monotone-medium-settings.

`User Option` *voice-nxml-char-ref-delimiter* [Variable]
Personality used for nxml-char-ref-delimiter This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

`User Option` *voice-nxml-char-ref-number* [Variable]
Personality used for nxml-char-ref-number This personality uses voice-animate-medium whose effect can be changed globally by customizing voice-animate-medium-settings.

`User Option` *voice-nxml-comment-content* [Variable]
Personality used for nxml-comment-content This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-nxml-comment-delimiter* [Variable]
Personality used for nxml-comment-delimiter This personality uses voice-smoothen-medium whose effect can be changed globally by customizing voice-smoothen-medium-settings.

`User Option` *voice-nxml-delimited-data* [Variable]
Personality used for nxml-delimited-data This personality uses voice-animate-medium whose effect can be changed globally by customizing voice-animate-medium-settings.

`User Option` *voice-nxml-delimiter* [Variable]
Personality used for nxml-delimiter This personality uses voice-bolden-medium whose effect can be changed globally by customizing voice-bolden-medium-settings.

`User Option` *voice-nxml-element-colon* [Variable]
Personality used for nxml-element-colon This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-nxml-element-local-name* [Variable]
Personality used for nxml-element-local-name This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-nxml-element-prefix* [Variable]
Personality used for nxml-element-prefix This personality uses voice-monotone-medium whose effect can be changed globally by customizing voice-monotone-medium-settings.

`User Option` *voice-nxml-entity-ref-delimiter* [Variable]

Personality used for nxml-entity-ref-delimiter This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

`User Option` *voice-nxml-entity-ref-name* [Variable]

Personality used for nxml-entity-ref-name This personality uses voice-lighten-medium whose effect can be changed globally by customizing voice-lighten-medium-settings.

`User Option` *voice-nxml-hash* [Variable]

Personality used for nxml-hash This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-nxml-markup-declaration-delimiter* [Variable]

Personality used for nxml-markup-declaration-delimiter This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

`User Option` *voice-nxml-name* [Variable]

Personality used for nxml-name This personality uses voice-animate-extra whose effect can be changed globally by customizing voice-animate-extra-settings.

`User Option` *voice-nxml-namespace-attribute-colon* [Variable]

Personality used for nxml-namespace-attribute-colon This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-nxml-namespace-attribute-prefix* [Variable]

Personality used for nxml-namespace-attribute-prefix This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

`User Option` *voice-nxml-namespace-attribute-value* [Variable]

Personality used for nxml-namespace-attribute-value This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-nxml-namespace-attribute-value-delimiter* [Variable]

Personality used for nxml-namespace-attribute-value-delimiter This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

`User Option` *voice-nxml-namespace-attribute-xmlns* [Variable]

Personality used for nxml-namespace-attribute-xmlns This personality uses voice-smoothen-extra whose effect can be changed globally by customizing voice-smoothen-extra-settings.

`User Option` *voice-nxml-processing-instruction-content* [Variable]

Personality used for nxml-processing-instruction-content This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

`User Option` *voice-nxml-processing-instruction-delimiter* [Variable]

Personality used for nxml-processing-instruction-delimiter This personality uses voice-lighten-extra whose effect can be changed globally by customizing voice-lighten-extra-settings.

`User Option` *voice-nxml-processing-instruction-target* [Variable]

Personality used for nxml-processing-instruction-target This personality uses voice-animate-extra whose effect can be changed globally by customizing voice-animate-extra-settings.

`User Option` *voice-nxml-prolog-keyword* [Variable]

Personality used for nxml-prolog-keyword This personality uses voice-animate-extra whose effect can be changed globally by customizing voice-animate-extra-settings.

`User Option` *voice-nxml-prolog-literal-content* [Variable]

Personality used for nxml-prolog-literal-content This personality uses voice-monotone-medium whose effect can be changed globally by customizing voice-monotone-medium-settings.

`User Option` *voice-nxml-prolog-literal-delimiter* [Variable]

Personality used for nxml-prolog-literal-delimiter This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-nxml-ref* [Variable]

Personality used for nxml-ref This personality uses voice-animate-medium whose effect can be changed globally by customizing voice-animate-medium-settings.

`User Option` *voice-nxml-tag-delimiter* [Variable]

Personality used for nxml-tag-delimiter This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

`User Option` *voice-nxml-tag-slash* [Variable]

Personality used for nxml-tag-slash This personality uses voice-smoothen-medium whose effect can be changed globally by customizing voice-smoothen-medium-settings.

`User Option` *voice-rng-error* [Variable]

Personality used for rng-error This personality uses voice-bolden-and-animate whose effect can be changed globally by customizing voice-bolden-and-animate-settings.

12.133 emacspeak-ocr

This module defines Emacspeak front-end to OCR. This module assumes that sane is installed and working for image acquisition, and that there is an OCR engine that can take acquired images and produce text. Prerequisites: Sane installed and working. scanimage to generate tiff files from scanner. tiffcp to compress the tiff file. working ocr executable by default this module assumes that the OCR executable is named `"ocr"`

12.133.1 Emacspeak-Ocr Commands

12.133.1.1 emacspeak-ocr

`emacspeak-ocr` [Command]

`C-e C-o`

`<fn> C-o`

An OCR front-end for the Emacspeak desktop.

Page image is acquired using tools from the SANE package.
The acquired image is run through the OCR engine if one is
available, and the results placed in a buffer that is
suitable for browsing the results.

For detailed help, invoke command emacspeak-ocr bound to
C-e C-o to launch emacspeak-ocr-mode, and press
'?' to display mode-specific help for emacspeak-ocr-mode.

12.133.1.2 emacspeak-ocr-backward-page

`emacspeak-ocr-backward-page` (**&optional** *count-ignored*) [Command]

Like backward page, but tracks page number of current document.

(fn &optional COUNT-IGNORED)

12.133.1.3 emacspeak-ocr-customize

`emacspeak-ocr-customize` [Command]

Customize OCR settings.

12.133.1.4 emacspeak-ocr-flipflop-and-recognize-image

`emacspeak-ocr-flipflop-and-recognize-image` [Command]

Run OCR engine on current image after flip-flopping it.
Useful if you've scanned a page upside down and are using an engine that does not automatically flip the image for you.
You need the imagemagik family of tools — we use mogrify to transform the image.
Prompts for image file if file corresponding to the expected
'current page' is not found.

12.133.1.5 emacspeak-ocr-forward-page

`emacspeak-ocr-forward-page` (**&optional** *count-ignored*) [Command]

Like forward page, but tracks page number of current document.

(fn &optional COUNT-IGNORED)

12.133.1.6 emacspeak-ocr-mode

`emacspeak-ocr-mode` [Command]

An OCR front-end for the Emacspeak desktop.

Pre-requisites:

1) A working scanner back-end like SANE on Linux.

2) An OCR engine.

1: Make sure your scanner back-end works, and that you have the utilities to scan a document and acquire an image as a tiff file. Then set variable emacspeak-ocr-scan-image-program to point at this utility. By default, this is set to 'scanimage' which is the image scanning utility provided by SANE.

By default, this front-end attempts to compress the acquired tiff image; make sure you have a utility like tiffcp. Variable emacspeak-ocr-compress-image is set to 'tiffcp' by default; if you use something else, you should customize this variable.

2: Next, make sure you have an OCR engine installed and working. By default this front-end assumes that OCR is available as /usr/bin/ocr.

Once you have ensured that acquiring an image and applying OCR to it work independently of Emacs, you can use this Emacspeak front-end to enable easy OCR access from within Emacspeak.

The Emacspeak OCR front-end is launched by command emacspeak-ocr bound to C-e C-o.

This command switches to a special buffer that has OCR commands bounds to single keystrokes– see the key-binding list at the end of this description. Use Emacs online help facility to look up help on these commands.

emacspeak-ocr-mode provides the necessary functionality to scan, OCR, read and save documents. By default, scanned images and the resulting text are saved under directory ~/ocr; see variable emacspeak-ocr-working-directory. Invoking command emacspeak-ocr-open-working-directory bound to M-x emacspeak-ocr-open-working-directory will open this directory.

By default, the document being scanned is named 'untitled'. You can name the document by using command emacspeak-ocr-name-document bound to M-x emacspeak-ocr-name-document. The document name is used in constructing the name of the image and text files.

Key Bindings:

See key binding
— ——-

RET emacspeak-ocr-scan-and-recognize
SPC emacspeak-ocr-read-current-page
1 .. 9 emacspeak-ocr-page
? describe-mode
C emacspeak-ocr-set-compress-image-options
I emacspeak-ocr-set-scan-image-options
[emacspeak-ocr-backward-page
] emacspeak-ocr-forward-page
c emacspeak-ocr-customize
d emacspeak-ocr-open-working-directory
f emacspeak-ocr-flipflop-and-recognize-image
i emacspeak-ocr-scan-image
j emacspeak-ocr-scan-photo
n emacspeak-ocr-name-document
o emacspeak-ocr-recognize-image
p emacspeak-ocr-page
q bury-buffer
s emacspeak-ocr-save-current-page
w emacspeak-ocr-write-document

.

In addition to any hooks its parent mode 'text-mode' might have run,
this mode runs the hook 'emacspeak-ocr-mode-hook', as the final or penultimate step during initialization.

12.133.1.7 emacspeak-ocr-name-document

`emacspeak-ocr-name-document` (*name*) [Command]

Name document being scanned in the current OCR buffer.
Pick a short but meaningful name.

(fn NAME)

12.133.1.8 emacspeak-ocr-open-working-directory

`emacspeak-ocr-open-working-directory` [Command]

Launch dired on OCR working directory.

12.133.1.9 emacspeak-ocr-page

`emacspeak-ocr-page` [Command]

Move to specified page.

12.133.1.10 emacspeak-ocr-read-current-page

`emacspeak-ocr-read-current-page` [Command]

Speaks current page.

12.133.1.11 emacspeak-ocr-recognize-image

`emacspeak-ocr-recognize-image` [Command]

Run OCR engine on current image.
Prompts for image file if file corresponding to the expected
'current page' is not found.

12.133.1.12 emacspeak-ocr-save-current-page

`emacspeak-ocr-save-current-page` [Command]

Writes out recognized text from current page
to an appropriately named file.

12.133.1.13 emacspeak-ocr-scan-and-recognize

`emacspeak-ocr-scan-and-recognize` [Command]

Scan in a page and run OCR engine on it.
Use this command once you've verified that the separate
steps of acquiring an image and running the OCR engine work
correctly by themselves.

12.133.1.14 emacspeak-ocr-scan-image

`emacspeak-ocr-scan-image` [Command]

Acquire page image.

12.133.1.15 emacspeak-ocr-scan-photo

`emacspeak-ocr-scan-photo` (**&optional** *metadata*) [Command]

Scan in a photograph.
The scanned image is converted to JPEG.

(fn &optional METADATA)

12.133.1.16 emacspeak-ocr-set-compress-image-options

`emacspeak-ocr-set-compress-image-options` (*setting*) [Command]

Interactively update image compression options.
Prompts with current setting in the minibuffer.
Setting persists for current Emacs session.

(fn SETTING)

12.133.1.17 emacspeak-ocr-set-scan-image-options

`emacspeak-ocr-set-scan-image-options` (*setting*) [Command]

Interactively update scan image options.
Prompts with current setting in the minibuffer.
Setting persists for current Emacs session.

(fn SETTING)

12.133.1.18 emacspeak-ocr-write-document

`emacspeak-ocr-write-document` [Command]

Writes out recognized text from all pages in current document.

12.133.2 emacspeak-ocr Options

`User Option` *emacspeak-ocr-compress-image* [Variable]

Command used to compress the scanned tiff file.

`User Option` *emacspeak-ocr-compress-image-options* [Variable]

Options used for compressing tiff image.

`User Option` *emacspeak-ocr-compress-photo-options* [Variable]

Options used when created JPEG from scanned photographs.

`User Option` *emacspeak-ocr-engine* [Variable]

OCR engine to process acquired image.

`User Option` *emacspeak-ocr-engine-options* [Variable]

Command line options to pass to OCR engine.

`User Option` *emacspeak-ocr-image-extension* [Variable]

Filename extension used for acquired image.

`User Option` *emacspeak-ocr-jpeg-metadata-writer* [Variable]

Program to add metadata to JPEG files.

`User Option` *emacspeak-ocr-keep-uncompressed-image* [Variable]

If set to T, uncompressed image is not removed.

`User Option` *emacspeak-ocr-photo-compress* [Variable]

Program to create JPEG compressed images.

`User Option` *emacspeak-ocr-scan-image* [Variable]

Name of image acquisition program.

`User Option` *emacspeak-ocr-scan-image-options* [Variable]

Command line options to pass to image acquisition program.

`User Option` *emacspeak-ocr-scan-photo-options* [Variable]

Options used when scanning in photographs.

`User Option` *emacspeak-ocr-working-directory* [Variable]

Directory where images and OCR results will be placed.

12.134 emacspeak-org

Speech-enable org — Org allows you to keep organized notes and todo lists. Homepage: http://www.astro.uva.nl/~dominik/Tools/org/ or http://orgmode.org/

12.134.1 Emacspeak-Org Commands

12.134.1.1 emacspeak-org-capture-link

`emacspeak-org-capture-link` [Command]

C-; h

C-x @ h h

Capture hyperlink to current context.
To use this command, first do 'customize-variable' 'org-capture-template'
and assign letter 'h' to a template that creates the hyperlink on capture.

12.134.1.2 emacspeak-org-popup-input

`emacspeak-org-popup-input` [Command]

Pops up an org input area.

12.134.1.3 emacspeak-org-popup-input-buffer

`emacspeak-org-popup-input-buffer` (*mode*) [Command]

Provide an input buffer in a specified mode.

(fn MODE)

12.134.1.4 emacspeak-org-table-speak-both-headers-and-element

`emacspeak-org-table-speak-both-headers-and-element` [Command]

echoes both row and col headers.

12.134.1.5 emacspeak-org-table-speak-column-header

`emacspeak-org-table-speak-column-header` [Command]

echoes column header

12.134.1.6 emacspeak-org-table-speak-column-header-and-element

`emacspeak-org-table-speak-column-header-and-element` [Command]

echoes col header and element

12.134.1.7 emacspeak-org-table-speak-coordinates

`emacspeak-org-table-speak-coordinates` [Command]

echoes coordinates

12.134.1.8 emacspeak-org-table-speak-current-element

`emacspeak-org-table-speak-current-element` [Command]

echoes current table element

12.134.1.9 emacspeak-org-table-speak-row-header

`emacspeak-org-table-speak-row-header` [Command]
echoes row header

12.134.1.10 emacspeak-org-table-speak-row-header-and-element

`emacspeak-org-table-speak-row-header-and-element` [Command]
echoes row header and element

12.134.2 emacspeak-org Options

`User Option` *emacspeak-org-table-after-movement-function* [Variable]
The function to call after moving in a table

`User Option` *voice-org-agenda-calendar-event* [Variable]
Personality used for org-agenda-calendar-event This personality uses voice-animate-extra whose effect can be changed globally by customizing voice-animate-extra-settings.

`User Option` *voice-org-agenda-calendar-sexp* [Variable]
Personality used for org-agenda-calendar-sexp This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

`User Option` *voice-org-agenda-column-dateline* [Variable]
Personality used for org-agenda-column-dateline This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-org-agenda-diary* [Variable]
Personality used for org-agenda-diary This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

`User Option` *voice-org-agenda-done* [Variable]
Personality used for org-agenda-done This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-org-agenda-filter-category* [Variable]
Personality used for org-agenda-filter-category This personality uses voice-lighten-extra whose effect can be changed globally by customizing voice-lighten-extra-settings.

`User Option` *voice-org-agenda-filter-tags* [Variable]
Personality used for org-agenda-filter-tags This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-org-agenda-restriction-lock* [Variable]
Personality used for org-agenda-restriction-lock This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-org-agenda-structure* [Variable]
Personality used for org-agenda-structure This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-org-archived* [Variable]
Personality used for org-archived This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-org-beamer-tag* [Variable]
Personality used for org-beamer-tag This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-org-block* [Variable]
Personality used for org-block This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-org-checkbox* [Variable]
Personality used for org-checkbox This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

`User Option` *voice-org-clock-overlay* [Variable]
Personality used for org-clock-overlay This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

`User Option` *voice-org-code* [Variable]
Personality used for org-code This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-org-column* [Variable]
Personality used for org-column This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-org-column-title* [Variable]
Personality used for org-column-title This personality uses voice-lighten-extra whose effect can be changed globally by customizing voice-lighten-extra-settings.

`User Option` *voice-org-date* [Variable]
Personality used for org-date This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

`User Option` *voice-org-date-selected* [Variable]
Personality used for org-date-selected This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-org-default* [Variable]
Personality used for org-default This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

`User Option` *voice-org-document-info* [Variable]
Personality used for org-document-info This personality uses voice-bolden-medium whose effect can be changed globally by customizing voice-bolden-medium-settings.

`User Option` *voice-org-document-info-keyword* [Variable]
Personality used for org-document-info-keyword This personality uses voice-bolden-extra whose effect can be changed globally by customizing voice-bolden-extra-settings.

`User Option` *voice-org-document-title* [Variable]
: Personality used for org-document-title This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-org-done* [Variable]
: Personality used for org-done This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-org-drawer* [Variable]
: Personality used for org-drawer This personality uses voice-smoothen-medium whose effect can be changed globally by customizing voice-smoothen-medium-settings.

`User Option` *voice-org-ellipsis* [Variable]
: Personality used for org-ellipsis This personality uses voice-smoothen-extra whose effect can be changed globally by customizing voice-smoothen-extra-settings.

`User Option` *voice-org-footnote* [Variable]
: Personality used for org-footnote This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

`User Option` *voice-org-formula* [Variable]
: Personality used for org-formula This personality uses voice-animate-extra whose effect can be changed globally by customizing voice-animate-extra-settings.

`User Option` *voice-org-headline-done* [Variable]
: Personality used for org-headline-done This personality uses voice-monotone-medium whose effect can be changed globally by customizing voice-monotone-medium-settings.

`User Option` *voice-org-hide* [Variable]
: Personality used for org-hide This personality uses voice-smoothen-extra whose effect can be changed globally by customizing voice-smoothen-extra-settings.

`User Option` *voice-org-indent* [Variable]
: Personality used for org-indent This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

`User Option` *voice-org-inlinetask* [Variable]
: Personality used for org-inlinetask This personality uses voice-smoothen-extra whose effect can be changed globally by customizing voice-smoothen-extra-settings.

`User Option` *voice-org-level-1* [Variable]
: Personality used for org-level-1 This personality uses voice-bolden-medium whose effect can be changed globally by customizing voice-bolden-medium-settings.

`User Option` *voice-org-level-2* [Variable]
: Personality used for org-level-2 This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-org-level-3* [Variable]
: Personality used for org-level-3 This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

`User Option` *voice-org-level-4* [Variable]

Personality used for org-level-4 This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-org-level-5* [Variable]

Personality used for org-level-5 This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

`User Option` *voice-org-level-6* [Variable]

Personality used for org-level-6 This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-org-level-7* [Variable]

Personality used for org-level-7 This personality uses voice-lighten-medium whose effect can be changed globally by customizing voice-lighten-medium-settings.

`User Option` *voice-org-level-8* [Variable]

Personality used for org-level-8 This personality uses voice-lighten-extra whose effect can be changed globally by customizing voice-lighten-extra-settings.

`User Option` *voice-org-link* [Variable]

Personality used for org-link This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-org-meta-line* [Variable]

Personality used for org-meta-line This personality uses voice-smoothen-medium whose effect can be changed globally by customizing voice-smoothen-medium-settings.

`User Option` *voice-org-property-value* [Variable]

Personality used for org-property-value This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

`User Option` *voice-org-scheduled* [Variable]

Personality used for org-scheduled This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

`User Option` *voice-org-scheduled-previously* [Variable]

Personality used for org-scheduled-previously This personality uses voice-lighten-medium whose effect can be changed globally by customizing voice-lighten-medium-settings.

`User Option` *voice-org-scheduled-today* [Variable]

Personality used for org-scheduled-today This personality uses voice-bolden-extra whose effect can be changed globally by customizing voice-bolden-extra-settings.

`User Option` *voice-org-sexp-date* [Variable]

Personality used for org-sexp-date This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-org-special-keyword* [Variable]

Personality used for org-special-keyword This personality uses voice-lighten-extra whose effect can be changed globally by customizing voice-lighten-extra-settings.

User Option *voice-org-table* [Variable]

Personality used for org-table This personality uses voice-bolden-medium whose effect can be changed globally by customizing voice-bolden-medium-settings.

User Option *voice-org-tag* [Variable]

Personality used for org-tag This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

User Option *voice-org-target* [Variable]

Personality used for org-target This personality uses voice-bolden-medium whose effect can be changed globally by customizing voice-bolden-medium-settings.

User Option *voice-org-time-grid* [Variable]

Personality used for org-time-grid This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

User Option *voice-org-todo* [Variable]

Personality used for org-todo This personality uses voice-bolden-and-animate whose effect can be changed globally by customizing voice-bolden-and-animate-settings.

User Option *voice-org-upcoming-deadline* [Variable]

Personality used for org-upcoming-deadline This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

User Option *voice-org-verbatim* [Variable]

Personality used for org-verbatim This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

User Option *voice-org-warning* [Variable]

Personality used for org-warning This personality uses voice-bolden-and-animate whose effect can be changed globally by customizing voice-bolden-and-animate-settings.

12.135 emacspeak-orgalist

Speech-enable orgalist — create org-like lists everywhere.

12.136 emacspeak-origami

ORIGAMI == One More Flexible Folding Mechanism This module speech-enables origami-mode.

12.137 emacspeak-outline

Provide additional advice to outline-mode

12.137.1 Emacspeak-Outline Commands

12.137.1.1 emacspeak-outline-speak-backward-heading

`emacspeak-outline-speak-backward-heading` [Command]

Analogous to outline-backward-same-level
except that the outline section is spoken

12.137.1.2 emacspeak-outline-speak-forward-heading

`emacspeak-outline-speak-forward-heading` [Command]

Analogous to outline-forward-same-level,
except that the outline section is spoken

12.137.1.3 emacspeak-outline-speak-next-heading

`emacspeak-outline-speak-next-heading` [Command]

Analogous to outline-next-visible-heading,
except that the outline section is spoken

12.137.1.4 emacspeak-outline-speak-previous-heading

`emacspeak-outline-speak-previous-heading` [Command]

Analogous to outline-previous-visible-heading,
except that the outline section is spoken

12.137.1.5 emacspeak-outline-speak-this-heading

`emacspeak-outline-speak-this-heading` [Command]

Speak current outline section starting from point

12.138 emacspeak-package

PACKAGE == package.el Manage Emacs packages. This module speech-enables package.el with a few convenience commands.

12.138.1 Emacspeak-Package Commands

12.138.1.1 emacspeak-package-next-line

`emacspeak-package-next-line` [Command]

Move to next line and speak it.

12.138.1.2 emacspeak-package-previous-line

`emacspeak-package-previous-line` [Command]

Move to next line and speak it.

12.138.1.3 emacspeak-package-summarize-line

`emacspeak-package-summarize-line` [Command]

Succinct Summary.

12.138.2 emacspeak-package Options

`User Option` *voice-package-description* [Variable]
Personality used for package-description This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-package-help-section-name* [Variable]
Personality used for package-help-section-name This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-package-name* [Variable]
Personality used for package-name This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-package-status-avail-obso* [Variable]
Personality used for package-status-avail-obso This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-package-status-available* [Variable]
Personality used for package-status-available This personality uses voice-annotate whose effect can be changed globally by customizing voice-annotate-settings.

`User Option` *voice-package-status-built-in* [Variable]
Personality used for package-status-built-in This personality uses voice-monotone-medium whose effect can be changed globally by customizing voice-monotone-medium-settings.

`User Option` *voice-package-status-dependency* [Variable]
Personality used for package-status-dependency This personality uses voice-monotone-medium whose effect can be changed globally by customizing voice-monotone-medium-settings.

`User Option` *voice-package-status-disabled* [Variable]
Personality used for package-status-disabled This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

`User Option` *voice-package-status-external* [Variable]
Personality used for package-status-external This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

`User Option` *voice-package-status-held* [Variable]
Personality used for package-status-held This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-package-status-incompat* [Variable]
Personality used for package-status-incompat This personality uses voice-animate-extra whose effect can be changed globally by customizing voice-animate-extra-settings.

`User Option` *voice-package-status-installed* [Variable]
Personality used for package-status-installed This personality uses voice-lighten-extra whose effect can be changed globally by customizing voice-lighten-extra-settings.

`User Option` *voice-package-status-new* [Variable]
Personality used for package-status-new This personality uses voice-brighten whose effect can be changed globally by customizing voice-brighten-settings.

`User Option` *voice-package-status-unsigned* [Variable]
Personality used for package-status-unsigned This personality uses voice-animate-extra whose effect can be changed globally by customizing voice-animate-extra-settings.

12.139 emacspeak-paradox

PARADOX == paradox.el Improved package management interface Manage Emacs packages. This module speech-enables paradox.el with a few convenience commands.

12.139.1 Emacspeak-Paradox Commands

12.139.1.1 emacspeak-paradox-summarize-line

`emacspeak-paradox-summarize-line` [Command]
Succinct Summary.

12.139.2 emacspeak-paradox Options

`User Option` *voice-paradox-description-face-multiline* [Variable]
Personality used for paradox-description-face-multiline This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

12.140 emacspeak-perl

Provide additional advice to perl-mode

12.141 emacspeak-pianobar

12.141.1 PIANOBAR == Pandora Client for Emacs

pianobar git://github.com/PromyLOPh/pianobar.git Ubuntu/Debian: sudo apt-get install pianobar

Pianobar Is a stand-alone client for Pandora Radio. pianobar.el available on the Emacs Wiki at http://www.emacswiki.org/emacs/pianobar.el Provides access to Pandora Radio via pianobar from the comfort of Emacs. This module speech-enables Pianobar and enhances it for the Complete Audio Desktop.

12.141.2 Emacspeak Usage:

Emacspeak implements command emacspeak-pianobar, a light-weight wrapper on top of pianobar. Emacspeak binds this command to `C-e '`. Command emacspeak-pianobar is designed to let you launch Pandora channels and switch tracks/channels without moving away from your primary tasks such as editting code or reading/composing email. Toward this end, launching command emacspeak-pianobar the first time initializes the `*pianobar*` buffer and launches command `pianobar`; but focus remains in your current buffer. Pianobar can be controlled with single keystrokes while in the pianobar buffer — switch to using `C-e ''`. The most useful keys are `right` for skipping tracks, `up` and `down` for switching channels etc.; see the keys bound in `pianobar-key-map` for a complete list. Pressing `C-e '` in the `*pianobar*` buffer buries the `*pianobar*`. From here on, Pianobar can be controlled by pressing the Pianobar prefix key (`C-e '`) followed by keys from `pianobar-key-map`.

12.141.3 Emacspeak-Pianobar Commands

12.141.3.1 emacspeak-pianobar

`emacspeak-pianobar` [Command]

`C-e '`

`<fn> '`

Start or control Emacspeak Pianobar player.

12.141.3.2 emacspeak-pianobar-command

`emacspeak-pianobar-command` (*key*) [Command]

Invoke Pianobar commands.

(fn KEY)

12.141.3.3 emacspeak-pianobar-electric-mode-toggle

`emacspeak-pianobar-electric-mode-toggle` [Command]

Toggle electric mode in pianobar buffer.
If electric mode is on, keystrokes invoke pianobar commands directly.

12.141.3.4 emacspeak-pianobar-next-preset

`emacspeak-pianobar-next-preset` [Command]

Switch to next preset.

12.141.3.5 emacspeak-pianobar-previous-preset

`emacspeak-pianobar-previous-preset` [Command]

Switch to previous preset.

12.141.3.6 emacspeak-pianobar-send-raw

`emacspeak-pianobar-send-raw` (*string*) [Command]

Send raw string with newline added to pianobar.

(fn STRING)

12.141.3.7 emacspeak-pianobar-switch-to-preset

`emacspeak-pianobar-switch-to-preset` [Command]

Switch to one of the presets.

12.141.3.8 emacspeak-pianobar-volume-down

`emacspeak-pianobar-volume-down` [Command]

Decrease volume

12.141.3.9 emacspeak-pianobar-volume-up

`emacspeak-pianobar-volume-up` [Command]

Increase volume

12.142 emacspeak-popup

POPUP == popup.el from MELPA

12.142.1 emacspeak-popup Options

`User Option` *voice-popup-isearch-match* [Variable]

Personality used for popup-isearch-match This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

12.143 emacspeak-proced

PROCED == Process Editor A new Task Manager for Emacs. Proced is part of emacs 23.

12.143.1 Emacspeak-Proced Commands

12.143.1.1 emacspeak-proced-jump-to-process

`emacspeak-proced-jump-to-process` (*name*) [Command]

Jump to process by name.

(fn NAME)

12.143.1.2 emacspeak-proced-next-field

`emacspeak-proced-next-field` [Command]

Navigate to next field.

12.143.1.3 emacspeak-proced-next-line

`emacspeak-proced-next-line` [Command]

Move to next line and speak a summary.

12.143.1.4 emacspeak-proced-previous-field

`emacspeak-proced-previous-field` [Command]

Navigate to previous field.

12.143.1.5 emacspeak-proced-previous-line

`emacspeak-proced-previous-line` [Command]

Move to next line and speak a summary.

12.143.1.6 emacspeak-proced-speak-args

`emacspeak-proced-speak-args` [Command]

Speak command invocation for this process.

12.143.1.7 emacspeak-proced-speak-field

`emacspeak-proced-speak-field` (*field-name*) [Command]

Speak value of specified field in current line.

(fn FIELD-NAME)

12.143.1.8 emacspeak-proced-speak-that-field

`emacspeak-proced-speak-that-field` [Command]

Speak desired field via single keystroke.

12.143.1.9 emacspeak-proced-speak-this-field

`emacspeak-proced-speak-this-field` (**&optional** *position*) [Command]

Speak field at specified column — defaults to current column.

(fn &optional POSITION)

12.144 emacspeak-project

Speech-enable project.el

12.145 emacspeak-projectile

PROJECTILE == '`M-x package-install projectile`'. Project management in Emacs.

12.146 emacspeak-pronounce

This module implements user customizable pronunciation dictionaries for emacspeak. Custom pronunciations can be defined per file, per directory and/or per major mode. Emacspeak maintains a persistent user dictionary upon request and loads these in new emacspeak sessions. This module implements the user interface to the custom dictionary as well as providing the internal API used by the rest of emacspeak in using the dictionary. Algorithm:

The persistent dictionary is a hash table where the hash keys are filenames, directory names, or major-mode names. The hash values are association lists defining the dictionary. Users of this module can retrieve a dictionary made up of all applicable association lists for a given file.

12.146.1 Emacspeak-Pronounce Commands

12.146.1.1 emacspeak-pronounce-clear-dictionaries

`emacspeak-pronounce-clear-dictionaries` [Command]

Clear all current pronunciation dictionaries.

12.146.1.2 emacspeak-pronounce-define-local-pronunciation

`emacspeak-pronounce-define-local-pronunciation` (*word pron*) [Command]

Define buffer local pronunciation.
Argument 'word' specified the word to be pronounced.
Argument 'pron' specifies the new pronunciation.

(fn WORD PRON)

12.146.1.3 emacspeak-pronounce-define-pronunciation

`emacspeak-pronounce-define-pronunciation` [Command]

Interactively define entries in the pronunciation dictionaries.
Default term to define is delimited by region.
First loads any persistent dictionaries if not already loaded.

12.146.1.4 emacspeak-pronounce-define-template-pronunciation

`emacspeak-pronounce-define-template-pronunciation` [Command]

Interactively define template entries in the pronunciation dictionaries.
Default term to define is delimited by region.
First loads any persistent dictionaries if not already loaded.

12.146.1.5 emacspeak-pronounce-dispatch

`emacspeak-pronounce-dispatch` [Command]

C-e M-d

<fn> M-d

Provides the user interface front-end to Emacspeak's pronunciation dictionaries.

12.146.1.6 emacspeak-pronounce-edit-pronunciations

`emacspeak-pronounce-edit-pronunciations` (*key*) [Command]

Prompt for and launch a pronunciation editor on the
specified pronunciation dictionary key.

(fn KEY)

12.146.1.7 emacspeak-pronounce-load-dictionaries

`emacspeak-pronounce-load-dictionaries` (**&optional** *filename*) [Command]

Load pronunciation dictionaries.
Optional argument FILENAME specifies the dictionary file,
Default is emacspeak-pronounce-dictionaries-file.

(fn &optional FILENAME)

12.146.1.8 emacspeak-pronounce-refresh-pronunciations

`emacspeak-pronounce-refresh-pronunciations` [Command]

Refresh pronunciation table for current buffer.
Activates pronunciation dictionaries if not already active.

12.146.1.9 emacspeak-pronounce-save-dictionaries

`emacspeak-pronounce-save-dictionaries` [Command]

Saves pronunciation dictionaries.

12.146.1.10 emacspeak-pronounce-toggle-use-of-dictionaries

`emacspeak-pronounce-toggle-use-of-dictionaries` (**&optional** *state*) [Command]

Toggle use of pronunciation dictionaries in current buffer.
Pronunciations can be defined on a per file, per directory and/or
per mode basis. Pronunciations are activated on a per buffer
basis. Turning on the use of pronunciation dictionaries results
in emacspeak composing a pronunciation table based on the
currently defined pronunciation dictionaries. After this, the
pronunciations will be applied whenever text in the buffer is
spoken. Optional argument state can be used from Lisp programs
to explicitly turn pronunciations on or off.

(fn &optional STATE)

12.146.2 emacspeak-pronounce Options

`User Option` *emacspeak-pronounce-common-xml-namespace-uri-pronunciations* [Variable]

Pronunciations for well known namespace URIs.

`User Option` *emacspeak-pronounce-dictionaries-file* [Variable]

File that holds emacspeak pronunciations.

`User Option` *emacspeak-pronounce-internet-smileys-pronunciations* [Variable]

Pronunciation dictionary used in all instant messenger and IRC chat modes. See http://www.charm.net/~kmarsh/smiley.html.

12.147 emacspeak-py

This speech-enables python-mode available on sourceforge and ELPA

12.148 emacspeak-pydoc

PYDOC == Python Documentation Viewer

12.149 emacspeak-python

This speech-enables python-mode bundled with Emacs

12.150 emacspeak-racer

RACER == Rust documentation and completion

12.150.1 emacspeak-racer Options

User Option *voice-racer-tooltip* [Variable]

Personality used for racer-tooltip This personality uses voice-brighten whose effect can be changed globally by customizing voice-brighten-settings.

12.151 emacspeak-racket

racket-mode implements an IDE for racket, a dialect of scheme.

12.152 emacspeak-re-builder

Speech-enable re-builder. Will be used to advantage in efficiently setting up outline regexp wizards

12.152.1 emacspeak-re-builder Options

User Option *voice-reb-match-0* [Variable]

Personality used for reb-match-0 This personality uses voice-lock-overlay-0 whose effect can be changed globally by customizing voice-lock-overlay-0-settings.

User Option *voice-reb-match-1* [Variable]

Personality used for reb-match-1 This personality uses voice-lock-overlay-1 whose effect can be changed globally by customizing voice-lock-overlay-1-settings.

User Option *voice-reb-match-2* [Variable]

Personality used for reb-match-2 This personality uses voice-lock-overlay-2 whose effect can be changed globally by customizing voice-lock-overlay-2-settings.

User Option *voice-reb-match-3* [Variable]

Personality used for reb-match-3 This personality uses voice-lock-overlay-3 whose effect can be changed globally by customizing voice-lock-overlay-3-settings.

12.153 emacspeak-reftex

This module speech-enables refteex – reftex is a minor mode that makes navigation of TeX documents possible via a table of contents buffer.

12.154 emacspeak-related

RELATED == Switch among related buffers (melpa). Speech-enable interactive commands.

12.155 emacspeak-rg

RG == Emacs front-end to ripgrep (rg).

12.156 emacspeak-rmail

emacspeak extensions to rmail

12.156.1 Emacspeak-Rmail Commands

12.156.1.1 emacspeak-rmail-speak-current-message-labels

`emacspeak-rmail-speak-current-message-labels` [Command]

Speak labels of current message

12.156.1.2 emacspeak-rmail-summarize-current-message

`emacspeak-rmail-summarize-current-message` [Command]

Summarize current message

12.157 emacspeak-rpm-spec

speech-enable rpm-spec-mode –part of Emacs 21 on RH 7.3

12.158 emacspeak-rst

RST == rst-mode for editing rst text files. This module speech-enables rst-mode.

12.158.1 emacspeak-rst Options

`User Option` *voice-rst-adornment* [Variable]

Personality used for rst-adornment This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

`User Option` *voice-rst-block* [Variable]

Personality used for rst-block This personality uses voice-annotate whose effect can be changed globally by customizing voice-annotate-settings.

`User Option` *voice-rst-comment* [Variable]

Personality used for rst-comment This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-rst-definition* [Variable]

Personality used for rst-definition This personality uses voice-bolden-medium whose effect can be changed globally by customizing voice-bolden-medium-settings.

`User Option` *voice-rst-directive* [Variable]

Personality used for rst-directive This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

`User Option` *voice-rst-emphasis1* [Variable]

Personality used for rst-emphasis1 This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

`User Option` *voice-rst-emphasis2* [Variable]

Personality used for rst-emphasis2 This personality uses voice-animate-extra whose effect can be changed globally by customizing voice-animate-extra-settings.

`User Option` *voice-rst-external* [Variable]

Personality used for rst-external This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

`User Option` *voice-rst-level-1* [Variable]

Personality used for rst-level-1 This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-rst-level-2* [Variable]

Personality used for rst-level-2 This personality uses voice-bolden-medium whose effect can be changed globally by customizing voice-bolden-medium-settings.

`User Option` *voice-rst-level-3* [Variable]

Personality used for rst-level-3 This personality uses voice-lighten-medium whose effect can be changed globally by customizing voice-lighten-medium-settings.

`User Option` *voice-rst-level-4* [Variable]

Personality used for rst-level-4 This personality uses voice-lighten-extra whose effect can be changed globally by customizing voice-lighten-extra-settings.

`User Option` *voice-rst-literal* [Variable]

Personality used for rst-literal This personality uses voice-monotone-medium whose effect can be changed globally by customizing voice-monotone-medium-settings.

`User Option` *voice-rst-reference* [Variable]

Personality used for rst-reference This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-rst-transition* [Variable]

Personality used for rst-transition This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

12.159 emacspeak-ruby

Provide additional advice to Ruby mode

12.160 emacspeak-rust-mode

Speech-enable rust-mode

12.161 emacspeak-sage

Speech-enable `sage-shell-mode`. This is a major mode for interacting with `sage`, `http://www.sagemath.org/` An Open-source Mathematical Software System.

12.161.1 Emacspeak-Sage Commands

12.161.1.1 emacspeak-sage-describe-symbol

`emacspeak-sage-describe-symbol` (*s*) [Command]

Describe Sage symbol at point.

(fn S)

12.161.1.2 emacspeak-sage-get-output

`emacspeak-sage-get-output` [Command]

Return most recent Sage output

12.161.1.3 emacspeak-sage-get-output-as-latex

`emacspeak-sage-get-output-as-latex` [Command]

Return most recent Sage output as LaTeX markup.

12.161.1.4 emacspeak-sage-speak-output

`emacspeak-sage-speak-output` [Command]

Speak last output from Sage.

12.162 emacspeak-sdcv

SDCV == Stardict Dictionary Interface This module sets up Emacspeak for use with sdcv. You need to have command-line sdcv installed. You can install additional stardict dictionaries, see https://wiki.archlinux.org/index.php/sdcv This module sets up Emacs module sdcv to use all the installed dictionaries found on the system.

12.163 emacspeak-ses

ses implements a simple spread sheet and is part of Emacs This module speech-enables ses

12.163.1 Emacspeak-Ses Commands

12.163.1.1 emacspeak-ses-backward-column-and-summarize

`emacspeak-ses-backward-column-and-summarize` [Command]

Move to previous column and summarize.

12.163.1.2 emacspeak-ses-backward-row-and-summarize

`emacspeak-ses-backward-row-and-summarize` [Command]

Move to previous row and summarize.

12.163.1.3 emacspeak-ses-forward-column-and-summarize

`emacspeak-ses-forward-column-and-summarize` [Command]

Move to next column and summarize.

12.163.1.4 emacspeak-ses-forward-row-and-summarize

`emacspeak-ses-forward-row-and-summarize` [Command]

Move to next row and summarize.

12.163.1.5 emacspeak-ses-summarize-cell

`emacspeak-ses-summarize-cell` (*cell-name*) [Command]

Summarize specified cell.

(fn CELL-NAME)

12.163.1.6 emacspeak-ses-summarize-current-cell

`emacspeak-ses-summarize-current-cell` (**&rest** *ignore*) [Command]

Summarize current cell.

(fn &rest IGNORE)

12.164 emacspeak-setup

Entry point for Emacspeak. The simplest and most basic way to start emacspeak is: emacs -q -l <emacspeak-dir>/lisp/emacspeak-setup.el The above starts a vanilla Emacs with just Emacspeak loaded. Once the above has been verified to work, You can add (load-library "emacspeak-setup") To your .emacs file. See tvr/emacs-startup.el in the Emacspeak Git repository for my setup.

12.165 emacspeak-sgml-mode

emacspeak extensions to sgml mode

12.166 emacspeak-sh-script

This module speech-enables sh-script.el

12.167 emacspeak-shx

SHX == Shell Extras For emacs

12.168 emacspeak-slime

SLIME == Superior Lisp Interaction Mode For Emacs Slime is a powerful IDE for developing in Common Lisp and Clojure. It's similar but more modern than package ILisp that I used as a graduate student when developing AsTeR.

12.168.1 emacspeak-slime Options

`User Option` *voice-slime-apropos-label* [Variable]

Personality used for slime-apropos-label This personality uses voice-monotone-light whose effect can be changed globally by customizing voice-monotone-light-settings.

`User Option` *voice-slime-apropos-symbol* [Variable]

Personality used for slime-apropos-symbol This personality uses voice-monotone-light whose effect can be changed globally by customizing voice-monotone-light-settings.

12.169 emacspeak-smart-window

SMART-WINDOW == Smart Window switching for Emacs

12.170 emacspeak-smartparens

SMARTPARENS == Automatic insertion, wrapping and paredit-like navigation with user defined pairs this module speech-enables smartparens. Insertion of a matching delimiter is indicated by a short auditory icon. Structured navigation speaks the current line with the position of point aurally highlighted.

12.170.1 emacspeak-smartparens Options

`User Option` *voice-sp-show-pair-enclosing* [Variable]

Personality used for sp-show-pair-enclosing This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-sp-wrap-overlay-closing-pair* [Variable]

Personality used for sp-wrap-overlay-closing-pair This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

`User Option` *voice-sp-wrap-overlay-opening-pair* [Variable]

Personality used for sp-wrap-overlay-opening-pair This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

12.171 emacspeak-solitaire

Auditory interface to solitaire

12.171.1 Emacspeak-Solitaire Commands

12.171.1.1 emacspeak-solitaire-show-column

`emacspeak-solitaire-show-column` [Command]

Audio format current column.

12.171.1.2 emacspeak-solitaire-show-row

`emacspeak-solitaire-show-row` [Command]

Audio format current row.

12.171.1.3 emacspeak-solitaire-speak-coordinates

`emacspeak-solitaire-speak-coordinates` [Command]

Speak coordinates of current position

12.171.1.4 emacspeak-solitaire-speak-row

`emacspeak-solitaire-speak-row` [Command]

Speak current row.

12.171.1.5 emacspeak-solitaire-speak-stones

`emacspeak-solitaire-speak-stones` [Command]

Speak number of stones remaining.

12.172 emacspeak-sounds

This module provides the interface for generating auditory icons in emacspeak. Design goal: 1) Auditory icons should be used to provide additional feedback, not as a gimmick. 2) The interface should be usable at all times without the icons: e.g. when on a machine without a sound card. 3) General principle for when to use an icon: Convey information about events taking place in parallel. For instance, if making a selection automatically moves the current focus to the next choice, We speak the next choice, while indicating the fact that something was selected with a sound cue. This interface will assume the availability of a shell command "play" that can take one or more sound files and play them. This module will also provide a mapping between names in the elisp world and actual sound files. Modules that wish to use auditory icons should use these names, instead of actual file names. As of Emacspeak 13.0, this module defines a themes architecture for auditory icons. Sound files corresponding to a given theme are found in appropriate subdirectories of emacspeak-sounds-directory

12.172.1 Emacspeak-Sounds Commands

12.172.1.1 emacspeak-audio-setup

`emacspeak-audio-setup` (**&optional** *prefix*) [Command]

C-e (

<fn> (

Call amixer command.

(fn &optional PREFIX)

12.172.1.2 emacspeak-sounds-select-theme

`emacspeak-sounds-select-theme` (*theme*) [Command]

C-e)

<fn>)

Select theme for auditory icons.

(fn THEME)

12.172.1.3 emacspeak-toggle-auditory-icons

`emacspeak-toggle-auditory-icons` (**&optional** *prefix*) [Command]

C-e C-a

<fn> C-a

Toggle use of auditory icons.
Optional interactive PREFIX arg toggles global value.

(fn &optional PREFIX)

12.172.2 emacspeak-sounds Options

`User Option` *emacspeak-auditory-icon-function* [Variable]

Function that plays auditory icons. play : Launches play-program to play. Serve: Send a command to the speech-server to play. Queue : Add auditory icon to speech queue. Native : Use Emacs' builtin sound support. Use Serve when working with remote speech servers.

`User Option` *emacspeak-play-args* [Variable]

Set this to nil if using paplay from pulseaudio.

`User Option` *emacspeak-play-program* [Variable]

Executable that plays sound files.

`User Option` *emacspeak-sounds-default-theme* [Variable]

Default theme for auditory icons.

12.173 emacspeak-speak

This module defines the core speech services used by emacspeak. It depends on the speech server interface modules It protects other parts of emacspeak from becoming dependent on the speech server modules

12.173.1 Emacspeak-Speak Commands

12.173.1.1 emacspeak–choose-completion

`emacspeak--choose-completion` [Command]

Choose the completion at point.

12.173.1.2 emacspeak–persist-variable

`emacspeak--persist-variable` (*var file*) [Command]

Persist variable 'var' to file 'FILE'.
Arranges for 'VAR' to be restored when 'file' is loaded.

(fn VAR FILE)

12.173.1.3 emacspeak-blink-matching-open

`emacspeak-blink-matching-open` [Command]

Move cursor momentarily to the beginning of the sexp before point.
Also display match context in minibuffer.

12.173.1.4 emacspeak-completions-move-to-completion-group

`emacspeak-completions-move-to-completion-group` [Command]

Move to group of choices beginning with character last
typed. If no such group exists, then we try to search for that
char, or dont move.

12.173.1.5 emacspeak-dial-dtk

`emacspeak-dial-dtk` (*number*) [Command]

Prompt for and dial a phone NUMBER with the Dectalk.

(fn NUMBER)

12.173.1.6 emacspeak-execute-repeatedly

`emacspeak-execute-repeatedly` (*command*) [Command]

Execute COMMAND repeatedly.

(fn COMMAND)

12.173.1.7 emacspeak-goto-percent

`emacspeak-goto-percent` (*percent*) [Command]

`C-e M-%`

`<fn> M-%`

Move to end PERCENT of buffer like in View mode.
Display is centered at point.
Also set the mark at the position where point was.

(fn PERCENT)

12.173.1.8 emacspeak-mark-backward-mark

`emacspeak-mark-backward-mark` [Command]

`<C-up>`

Cycle backward through the mark ring.

12.173.1.9 emacspeak-minibuffer-choose-completion

`emacspeak-minibuffer-choose-completion` [Command]

Choose current completion.

12.173.1.10 emacspeak-minibuffer-next-completion

`emacspeak-minibuffer-next-completion` [Command]

Move to next available minibuffer completion.

12.173.1.11 emacspeak-minibuffer-previous-completion

`emacspeak-minibuffer-previous-completion` [Command]

Move to previous available minibuffer completion.

12.173.1.12 emacspeak-open-info

`emacspeak-open-info` [Command]

`C-e TAB`

`<fn> TAB`

Open Emacspeak Info Manual.

12.173.1.13 emacspeak-owindow-next-line

`emacspeak-owindow-next-line` (*count*) [Command]

`ESC <down>`

Move to the next line in the other window and speak it.
Numeric prefix arg COUNT can specify number of lines to move.

(fn COUNT)

12.173.1.14 emacspeak-owindow-previous-line

`emacspeak-owindow-previous-line` (*count*) [Command]

`ESC <up>`

Move to the next line in the other window and speak it.
Numeric prefix arg COUNT specifies number of lines to move.

(fn COUNT)

12.173.1.15 emacspeak-owindow-scroll-down

`emacspeak-owindow-scroll-down` [Command]

`ESC <prior>`

Scroll down the window that command 'other-window' would move to.
Speak the window contents after scrolling.

12.173.1.16 emacspeak-owindow-scroll-up

`emacspeak-owindow-scroll-up` [Command]

`ESC <next>`

Scroll up the window that command 'other-window' would move to.
Speak the window contents after scrolling.

12.173.1.17 emacspeak-owindow-speak-line

`emacspeak-owindow-speak-line` [Command]

ESC <select>
Speak the current line in the other window.

12.173.1.18 emacspeak-read-next-line

`emacspeak-read-next-line` (**&optional** *arg*) [Command]

C-e <down>
<fn> <down>
Read next line, specified by an offset, without moving.
Default is to read the next line.

(fn &optional ARG)

12.173.1.19 emacspeak-read-next-word

`emacspeak-read-next-word` (**&optional** *arg*) [Command]

Read next word, specified as a numeric arg, without moving.
Default is to read the next word.

(fn &optional ARG)

12.173.1.20 emacspeak-read-previous-line

`emacspeak-read-previous-line` (**&optional** *arg*) [Command]

C-e <up>

<fn> <up>

Read previous line, specified by an offset, without moving.
Default is to read the previous line.

(fn &optional ARG)

12.173.1.21 emacspeak-read-previous-word

`emacspeak-read-previous-word` (**&optional** *arg*) [Command]

Read previous word, specified as a prefix arg, without moving.
Default is to read the previous word.

(fn &optional ARG)

12.173.1.22 emacspeak-shell-command

`emacspeak-shell-command` (*command*) [Command]

C-e $

<fn> $

Run shell command COMMANDAND speak its output.

(fn COMMAND)

12.173.1.23 emacspeak-silence

`emacspeak-silence` [Command]

<silence>
Silence is golden. Stop speech, and pause/resume any media
streams. Runs ‘emacspeak-silence-hook’ which can be used to
configure which media players get silenced or paused/resumed.

12.173.1.24 emacspeak-speak-and-skip-extent-upto-char

`emacspeak-speak-and-skip-extent-upto-char` (*char*) [Command]

Search forward from point until we hit char.
Speak text between point and the char we hit.

(fn CHAR)

12.173.1.25 emacspeak-speak-and-skip-extent-upto-this-char

`emacspeak-speak-and-skip-extent-upto-this-char` [Command]

Speak extent delimited by point and last character typed.

12.173.1.26 emacspeak-speak-browse-buffer-by-style

`emacspeak-speak-browse-buffer-by-style` (**&optional** *browse*) [Command]

Browse current buffer by style.
Default is to speak chunk having current style.
Interactive prefix arg ‘browse’ repeatedly browses through
chunks having same style as the current text chunk.

(fn &optional BROWSE)

12.173.1.27 emacspeak-speak-buffer

`emacspeak-speak-buffer` (**&optional** *arg*) [Command]

C-e b

<fn> b

Speak current buffer contents.
With prefix ARG, speaks the rest of the buffer from point.
Negative prefix arg speaks from start of buffer to point.
If voice lock mode is on, the paragraphs in the buffer are
voice annotated first, see command ‘emacspeak-speak-voice-annotate-paragraphs’.

(fn &optional ARG)

12.173.1.28 emacspeak-speak-buffer-filename

`emacspeak-speak-buffer-filename` (**&optional** *filename*) [Command]

C-e C-f

<fn> C-f

Speak name of file being visited in current buffer.
Speak default directory if invoked in a dired buffer,
or when the buffer is not visiting any file.
Interactive prefix arg ‘filename’ speaks only the final path
component.
The result is put in the kill ring for convenience.

(fn &optional FILENAME)

12.173.1.29 emacspeak-speak-buffer-interactively

`emacspeak-speak-buffer-interactively` [Command]

C-e B

<fn> B

Speak the start of, rest of, or the entire buffer.
’s’ to speak the start.
’r’ to speak the rest.
any other key to speak entire buffer.

12.173.1.30 emacspeak-speak-char

`emacspeak-speak-char` (**&optional** *prefix*) [Command]

C-e c

<fn> c

Speak character under point.
Pronounces character phonetically unless called with a PREFIX arg.

(fn &optional PREFIX)

12.173.1.31 emacspeak-speak-char-name

`emacspeak-speak-char-name` (*char*) [Command]

tell me what this is

(fn CHAR)

12.173.1.32 emacspeak-speak-completions-if-available

`emacspeak-speak-completions-if-available` [Command]

Speak completions if available.

12.173.1.33 emacspeak-speak-continuously

`emacspeak-speak-continuously` [Command]

C-e RET

<fn> RET

Speak a buffer continuously.
First prompts using the minibuffer for the kind of action to
perform after speaking each chunk. E.G. speak a line at a time
etc. Speaking commences at current buffer position. Pressing
C-g breaks out, leaving point on last chunk that
was spoken. Any other key continues to speak the buffer.

12.173.1.34 emacspeak-speak-current-column

`emacspeak-speak-current-column` [Command]

C-e =

<fn> =

Speak the current column.

12.173.1.35 emacspeak-speak-current-field

`emacspeak-speak-current-field` [Command]

C-e .

<fn> .

Speak current field.

12.173.1.36 emacspeak-speak-current-kill

`emacspeak-speak-current-kill` (**&optional** *count*) [Command]

C-e k

<fn> k

Speak the current kill entry.
This is the text that will be yanked in
by the next <S-insertchar>. Prefix numeric arg, COUNT, specifies that the
text that will be yanked as a result of a <S-insertchar> followed by count-1
M-x yank-pop be spoken. The kill number that is spoken says what
numeric prefix arg to give to command yank.

(fn &optional COUNT)

12.173.1.37 emacspeak-speak-current-mark

`emacspeak-speak-current-mark` (*count*) [Command]

C-e C-SPC

C-e C-@

<fn> C-SPC
<fn> C-@

Speak the line containing the mark.
With no argument, speaks the line containing the mark–this is
where 'exchange-point-and-mark' C-x C-x would
jump. Numeric prefix arg 'COUNT' speaks line containing mark 'n'
where 'n' is one less than the number of times one has to jump
using 'set-mark-command' to get to this marked position. The
location of the mark is indicated by an aural highlight achieved
by a change in voice personality.

(fn COUNT)

12.173.1.38 emacspeak-speak-current-percentage

`emacspeak-speak-current-percentage` [Command]

C-e %

<fn> %

Announce the percentage into the current buffer.

12.173.1.39 emacspeak-speak-current-window

`emacspeak-speak-current-window` [Command]

Speak contents of current window.
Speaks entire window irrespective of point.

12.173.1.40 emacspeak-speak-date-as-seconds

`emacspeak-speak-date-as-seconds` (*time*) [Command]

Read time value as a human-readable string, return seconds.
Seconds value is also placed in the kill-ring.

(fn TIME)

12.173.1.41 emacspeak-speak-display-char

`emacspeak-speak-display-char` (**&optional** *prefix*) [Command]

Display char under point using current speech display table.
Behavior is the same as command 'emacspeak-speak-char'
bound to C-e c
for characters in the range 0–127.
Optional argument PREFIX specifies that the character should be spoken phonetically.

(fn &optional PREFIX)

12.173.1.42 emacspeak-speak-face-interval-and-move

`emacspeak-speak-face-interval-and-move` [Command]

Speaks region delimited by text in current face, and moves past the chunk.

12.173.1.43 emacspeak-speak-front-of-buffer

`emacspeak-speak-front-of-buffer` [Command]

Speak the buffer from start to point

12.173.1.44 emacspeak-speak-header-line

`emacspeak-speak-header-line` [Command]

C-e SPC

<fn> SPC

Speak header line if set.

12.173.1.45 emacspeak-speak-help

`emacspeak-speak-help` (**&optional** *arg*) [Command]

C-e h

<fn> h

Speak help buffer if one present.
With prefix arg, speaks the rest of the buffer from point.
Negative prefix arg speaks from start of buffer to point.

(fn &optional ARG)

12.173.1.46 emacspeak-speak-help-interactively

`emacspeak-speak-help-interactively` [Command]

Speak the start of, rest of, or the entire help.
's' to speak the start.
'r' to speak the rest.
any other key to speak entire help.

12.173.1.47 emacspeak-speak-line

`emacspeak-speak-line` (**&optional** *arg*) [Command]

C-e l

<fn> l

Speaks current line. With prefix ARG, speaks the rest of the line from point. Negative prefix optional arg speaks from start of line to point. Voicifies if option 'voice-lock-mode' is on. Indicates indentation with a tone or a spoken message if audio indentation is in use see 'emacspeak-toggle-audio-indentation' bound to
C-e d i. Indicates position of point
with an aural highlight if option 'emacspeak-show-point' is turned on
–see command 'emacspeak-toggle-show-point' bound to
C-e C-d. Lines that start hidden blocks of text,
e.g. outline header lines, or header lines of blocks created by

command ‘emacspeak-hide-or-expose-block’ are indicated with auditory
icon ellipses. Presence of additional presentational overlays (created
via property display, before-string, or after-string) is indicated
with auditory icon ‘more’. These can then be spoken using command
C-e C-M-l.

(fn &optional ARG)

12.173.1.48 emacspeak-speak-line-interactively

`emacspeak-speak-line-interactively` [Command]

`C-e L`

`<fn> L`

Speak the start of, rest of, or the entire line.
’s’ to speak the start.
’r’ to speak the rest.
any other key to speak entire line.

12.173.1.49 emacspeak-speak-line-number

`emacspeak-speak-line-number` [Command]

`C-e C-l`

`<fn> C-l`

Print the current buffer line number and narrowed line number of point.

12.173.1.50 emacspeak-speak-line-set-column-filter

`emacspeak-speak-line-set-column-filter` (*filter*) [Command]

`C-e |`

`<fn> |`

Set up filter for selectively speaking or ignoring portions of lines.
The filter is specified as a list of pairs.
For example, to filter columns 1 – 10 and 20 – 25,
specify filter as
((0 9) (20 25)). Filter settings are persisted across sessions. A
persisted filter is used as the default when prompting for a filter.
This allows one to accumulate a set of filters for specific files like
/var/adm/messages and /var/adm/maillog over time.
Option emacspeak-speak-line-invert-filter determines
the sense of the filter.

(fn FILTER)

12.173.1.51 emacspeak-speak-message-again

`emacspeak-speak-message-again` (**&optional** *from-message-cache*) [Command]

`C-e a`

<fn> a

Speak the last message from Emacs once again.
The message is also placed in the kill ring for convenient yanking
if 'emacspeak-speak-message-again-should-copy-to-kill-ring' is set.

(fn &optional FROM-MESSAGE-CACHE)

12.173.1.52 emacspeak-speak-message-at-time

`emacspeak-speak-message-at-time` (*time message*) [Command]

C-e @

<fn> @

Set up ring-at-time to speak message at specified time.
Provides simple stop watch functionality in addition to other things.
See documentation for command run-at-time for details on time-spec.

(fn TIME MESSAGE)

12.173.1.53 emacspeak-speak-microseconds-since-epoch

`emacspeak-speak-microseconds-since-epoch` (*ms*) [Command]

Speaks time value specified as microseconds since epoch, e.g. as from float-time.

(fn MS)

12.173.1.54 emacspeak-speak-milliseconds-since-epoch

`emacspeak-speak-milliseconds-since-epoch` (*ms*) [Command]

Speaks time value specified as milliseconds since epoch, e.g. as from float-time.

(fn MS)

12.173.1.55 emacspeak-speak-minibuffer

`emacspeak-speak-minibuffer` (**&optional** *arg*) [Command]

Speak the minibuffer contents
With prefix arg, speaks the rest of the buffer from point.
Negative prefix arg speaks from start of buffer to point.

(fn &optional ARG)

12.173.1.56 emacspeak-speak-minor-mode-line

`emacspeak-speak-minor-mode-line` (**&optional** *log-msg*) [Command]

C-e M

<fn> M

Speak the minor mode-information.
Optional interactive prefix arg 'log-msg' logs spoken info to *Messages*.

(fn &optional LOG-MSG)

12.173.1.57 emacspeak-speak-mode-line

`emacspeak-speak-mode-line` (**&optional** *buffer-info*) [Command]

`C-e m`

`<fn> m`

Speak the mode-line.
Speaks header-line if that is set when called non-interactively.
Interactive prefix arg speaks buffer info.

(fn &optional BUFFER-INFO)

12.173.1.58 emacspeak-speak-next-face-chunk

`emacspeak-speak-next-face-chunk` [Command]

Moves to the front of next chunk having current style.
Speak that chunk after moving.

12.173.1.59 emacspeak-speak-next-field

`emacspeak-speak-next-field` [Command]

`C-e >`

`<fn> >`

Move to and speak next field.

12.173.1.60 emacspeak-speak-next-personality-chunk

`emacspeak-speak-next-personality-chunk` [Command]

Moves to the front of next chunk having current personality.
Speak that chunk after moving.

12.173.1.61 emacspeak-speak-next-window

`emacspeak-speak-next-window` [Command]

`C-e C-n`

`<fn> C-n`

Speak the next window.

12.173.1.62 emacspeak-speak-other-buffer

`emacspeak-speak-other-buffer` (*buffer*) [Command]

`C-e M-b`

<fn> M-b

Speak specified buffer.
Useful to listen to a buffer without switching contexts.

(fn BUFFER)

12.173.1.63 emacspeak-speak-other-window

`emacspeak-speak-other-window` (**&optional** *arg*) [Command]

Speak contents of 'other' window.
Speaks entire window irrespective of point.
Semantics of 'other' is the same as for the builtin Emacs command 'other-window'.
Optional argument ARG specifies 'other' window to speak.

(fn &optional ARG)

12.173.1.64 emacspeak-speak-overlay-properties

`emacspeak-speak-overlay-properties` [Command]

C-e C-M-l

<fn> C-M-l
Speak display, before-string or after-string property if any.

12.173.1.65 emacspeak-speak-page

`emacspeak-speak-page` (**&optional** *arg*) [Command]

C-e [

<fn> [

Speak a page.
With prefix ARG, speaks rest of current page.
Negative prefix arg will read from start of current page to point.
If option 'voice-lock-mode' is on, then it will use any defined personality.

(fn &optional ARG)

12.173.1.66 emacspeak-speak-page-interactively

`emacspeak-speak-page-interactively` [Command]

C-e]

<fn>]

Speak the start of, rest of, or the entire page.
's' to speak the start.
'r' to speak the rest.
any other key to speak entire page.

12.173.1.67 emacspeak-speak-paragraph

`emacspeak-speak-paragraph` (**&optional** *arg*) [Command]

`C-e p`

`<fn> p`

Speak paragraph.
With prefix arg, speaks rest of current paragraph.
Negative prefix arg will read from start of current paragraph to point.
If voice-lock-mode is on, then it will use any defined personality.

(fn &optional ARG)

12.173.1.68 emacspeak-speak-paragraph-interactively

`emacspeak-speak-paragraph-interactively` [Command]

`C-e P`

`<fn> P`

Speak the start of, rest of, or the entire paragraph.
's' to speak the start.
'r' to speak the rest.
any other key to speak entire paragraph.

12.173.1.69 emacspeak-speak-preceding-char

`emacspeak-speak-preceding-char` [Command]

Speak character before point.

12.173.1.70 emacspeak-speak-predefined-window

`emacspeak-speak-predefined-window` (**&optional** *arg*) [Command]

`C-e 9`

`C-e 8`

`C-e 7`

`C-e 6`

`C-e 5`

`C-e 4`

`C-e 3`

`C-e 2`

`C-e 1`

`C-e 0`

`<fn> 9`

`<fn> 8`

<fn> 7

<fn> 6

<fn> 5

<fn> 4

<fn> 3

<fn> 2

<fn> 1

<fn> 0

Speak one of the first 10 windows on the screen.
Speaks entire window irrespective of point.
In general, you'll never have Emacs split the screen into more than two or three.
Argument ARG determines the 'other' window to speak.
Semantics of 'other' is the same as for the builtin Emacs command 'other-window'.

(fn &optional ARG)

12.173.1.71 emacspeak-speak-previous-face-chunk

`emacspeak-speak-previous-face-chunk` [Command]

Moves to the front of previous chunk having current face.
Speak that chunk after moving.

12.173.1.72 emacspeak-speak-previous-field

`emacspeak-speak-previous-field` [Command]

C-e <

<fn> <

Move to previous field and speak it.

12.173.1.73 emacspeak-speak-previous-personality-chunk

`emacspeak-speak-previous-personality-chunk` [Command]

Moves to the front of previous chunk having current personality.
Speak that chunk after moving.

12.173.1.74 emacspeak-speak-previous-window

`emacspeak-speak-previous-window` [Command]

C-e C-p

<fn> C-p

Speak the previous window.

12.173.1.75 emacspeak-speak-rectangle

`emacspeak-speak-rectangle` (*start end*) [Command]

C-e R

<fn> R

Speak a rectangle of text.
Rectangle is delimited by point and mark. When call from a program, arguments specify the START and END of the rectangle.

(fn START END)

12.173.1.76 emacspeak-speak-region

`emacspeak-speak-region` (*start end*) [Command]

C-e r

<fn> r

Speak region.
Argument START and END specify region to speak.

(fn START END)

12.173.1.77 emacspeak-speak-rest-of-buffer

`emacspeak-speak-rest-of-buffer` [Command]

C-e n

<fn> n

Speak remainder of the buffer starting at point

12.173.1.78 emacspeak-speak-run-shell-command

`emacspeak-speak-run-shell-command` (*command* **&optional** *read-as-csv*) [Command]

C-e !

<fn> !

Invoke shell COMMAND and display its output as a table. The results are placed in a buffer in Emacspeak's table browsing mode. Optional interactive prefix arg read-as-csv interprets the result as csv. . Use this for running shell commands that produce tabulated output. This command should be used for shell commands that produce tabulated output that works with Emacspeak's table recognizer. Verify this first by running the command in a shell and executing command 'emacspeak-table-display-table-in-region' normally bound to C-e M-i.

(fn COMMAND &optional READ-AS-CSV)

12.173.1.79 emacspeak-speak-seconds-since-epoch

`emacspeak-speak-seconds-since-epoch` (*seconds*) [Command]

Speaks time value specified as seconds since epoch, e.g. as from float-time.

(fn SECONDS)

12.173.1.80 emacspeak-speak-sentence

`emacspeak-speak-sentence` (**&optional** *arg*) [Command]

Speak current sentence.
With prefix ARG, speaks the rest of the sentence from point.
Negative prefix arg speaks from start of sentence to point.

(fn &optional ARG)

12.173.1.81 emacspeak-speak-set-display-table

`emacspeak-speak-set-display-table` (**&optional** *prefix*) [Command]

Sets up buffer specific speech display table that controls how
special characters are spoken. Interactive prefix argument causes
setting to be global.

(fn &optional PREFIX)

12.173.1.82 emacspeak-speak-sexp

`emacspeak-speak-sexp` (**&optional** *arg*) [Command]

C-e "

<fn> "

Speak current sexp.
With prefix ARG, speaks the rest of the sexp from point.
Negative prefix arg speaks from start of sexp to point.

(fn &optional ARG)

12.173.1.83 emacspeak-speak-sexp-interactively

`emacspeak-speak-sexp-interactively` [Command]

Speak the start of, rest of, or the entire sexp.
's' to speak the start.
'r' to speak the rest.
any other key to speak entire sexp.

12.173.1.84 emacspeak-speak-skim-buffer

`emacspeak-speak-skim-buffer` [Command]

C-e ,

<fn> ,

Skim the current buffer a paragraph at a time.

12.173.1.85 emacspeak-speak-spaces-at-point

`emacspeak-speak-spaces-at-point` [Command]

C-e C-M-SPC
C-e C-M-@

<fn> C-M-SPC
<fn> C-M-@

Speak the white space at point.

12.173.1.86 emacspeak-speak-spell-current-word

`emacspeak-speak-spell-current-word` [Command]

Spell word at point.

12.173.1.87 emacspeak-speak-text-range

`emacspeak-speak-text-range` (*property*) [Command]

Speak text range identified by this PROPERTY.

(fn PROPERTY)

12.173.1.88 emacspeak-speak-this-face-chunk

`emacspeak-speak-this-face-chunk` [Command]

Speak chunk of text around point that has current face.

12.173.1.89 emacspeak-speak-this-personality-chunk

`emacspeak-speak-this-personality-chunk` [Command]

Speak chunk of text around point that has current
personality.

12.173.1.90 emacspeak-speak-time

`emacspeak-speak-time` (**&optional** *world*) [Command]

C-e t

<fn> t

Speak the time.
Optional interactive prefix arg 'C-u'invokes world clock.
Timezone is specified using minibuffer completion.
Copies time to kill-ring.
Second interactive prefix sets clock to new timezone.

(fn &optional WORLD)

12.173.1.91 emacspeak-speak-version

`emacspeak-speak-version` (**&optional** *speak-rev*) [Command]

C-e V

<fn> V

Announce version information for running emacspeak.
Optional interactive prefix arg 'speak-rev' speaks only the Git revision number.

(fn &optional SPEAK-REV)

12.173.1.92 emacspeak-speak-visual-line

`emacspeak-speak-visual-line` [Command]

Speaks current visual line.
Cues the start of a physical line with auditory icon 'left'.

12.173.1.93 emacspeak-speak-voice-annotate-paragraphs

`emacspeak-speak-voice-annotate-paragraphs` [Command]

Locate paragraphs and voice annotate the first word.
Here, paragraph is taken to mean a chunk of text preceded by a blank line.
Useful to do this before you listen to an entire buffer.

12.173.1.94 emacspeak-speak-which-function

`emacspeak-speak-which-function` [Command]

C-e M-w

<fn> M-w

Speak which function we are on. Uses which-function from
which-func without turning that mode on. We actually use
semantic to do the work.

12.173.1.95 emacspeak-speak-window-information

`emacspeak-speak-window-information` [Command]

C-e C-w

<fn> C-w

Speaks information about current window.

12.173.1.96 emacspeak-speak-word

`emacspeak-speak-word` (**&optional** *arg*) [Command]

C-e w

<fn> w

Speak current word.
With prefix ARG, speaks the rest of the word from point.
Negative prefix arg speaks from start of word to point.
If executed on the same buffer position a second time, the word is spelled out instead of being spoken.

(fn &optional ARG)

12.173.1.97 emacspeak-speak-word-interactively

`emacspeak-speak-word-interactively` [Command]

Speak the start of, rest of, or the entire word.
's' to speak the start.
'r' to speak the rest.
any other key to speak entire word.

12.173.1.98 emacspeak-speak-world-clock

`emacspeak-speak-world-clock` (*zone* **&optional** *set*) [Command]

Display current date and time for specified zone.
Optional second arg 'set' sets the TZ environment variable as well.

(fn ZONE &optional SET)

12.173.1.99 emacspeak-switch-to-reference-buffer

`emacspeak-switch-to-reference-buffer` [Command]

Switch back to buffer that generated completions.

12.173.1.100 emacspeak-toggle-action-mode

`emacspeak-toggle-action-mode` (**&optional** *prefix*) [Command]

Toggle state of Emacspeak action mode.
Interactive PREFIX arg means toggle the global default value, and then set the current local value to the result.

12.173.1.101 emacspeak-toggle-audio-indentation

`emacspeak-toggle-audio-indentation` (**&optional** *prefix*) [Command]

`C-e d i`

`<fn> d i`

Toggle state of Emacspeak audio indentation.
Interactive PREFIX arg means toggle the global default value, and then set the current local value to the result.
Specifying the method of indentation as 'tones'
results in the Dectalk producing a tone whose length is a function of the line's indentation. Specifying 'speak'
results in the number of initial spaces being spoken.

12.173.1.102 emacspeak-toggle-character-echo

`emacspeak-toggle-character-echo` (**&optional** *prefix*) [Command]

C-e d k

<fn> d k

Toggle state of Emacspeak character echo.
Interactive PREFIX arg means toggle the global default value, and then set the current local value to the result.

12.173.1.103 emacspeak-toggle-header-line

`emacspeak-toggle-header-line` [Command]

Toggle Emacspeak's default header line.

12.173.1.104 emacspeak-toggle-line-echo

`emacspeak-toggle-line-echo` (**&optional** *prefix*) [Command]

C-e d l

<fn> d l

Toggle state of Emacspeak line echo.
Interactive PREFIX arg means toggle the global default value, and then set the current local value to the result.

12.173.1.105 emacspeak-toggle-mail-alert

`emacspeak-toggle-mail-alert` (**&optional** *prefix*) [Command]

C-e M-m

<fn> M-m

Toggle state of Emacspeak mail alert.
Interactive PREFIX arg means toggle the global default value, and then set the current local value to the result.
Turning on this option results in Emacspeak producing an auditory icon indicating the arrival of new mail when displaying the mode line.

12.173.1.106 emacspeak-toggle-show-point

`emacspeak-toggle-show-point` (**&optional** *prefix*) [Command]

C-e C-d

<fn> C-d

Toggle state of Emacspeak-show-point.
Interactive PREFIX arg means toggle the global default value, and then set the current local value to the result.

12.173.1.107 emacspeak-toggle-speak-line-invert-filter

`emacspeak-toggle-speak-line-invert-filter` (**&optional** *prefix*) [Command]

`C-e \`

`<fn> \`

Toggle state of how column filter is interpreted.
Interactive PREFIX arg means toggle the global default value, and then set the current local value to the result.

12.173.1.108 emacspeak-toggle-speak-messages

`emacspeak-toggle-speak-messages` (**&optional** *prefix*) [Command]

`C-e q`

`<fn> q`

Toggle the state of whether emacspeak echoes messages.

12.173.1.109 emacspeak-toggle-word-echo

`emacspeak-toggle-word-echo` (**&optional** *prefix*) [Command]

`C-e d w`

`<fn> d w`

Toggle state of Emacspeak word echo.
Interactive PREFIX arg means toggle the global default value, and then set the current local value to the result.

12.173.1.110 emacspeak-view-notifications

`emacspeak-view-notifications` [Command]

`C-h T`

`<f1> T`

`<help> T`

Display notifications.

12.173.1.111 emacspeak-voiceify-rectangle

`emacspeak-voiceify-rectangle` (*start end* **&optional** *personality*) [Command]

Voicify the current rectangle.
When calling from a program,arguments are
START END personality
Prompts for PERSONALITY with completion when called interactively.

(fn START END &optional PERSONALITY)

12.173.1.112 emacspeak-voiceify-region

`emacspeak-voiceify-region` (*start end* **&optional** *personality*) [Command]

Voicify the current region.
When calling from a program,arguments are
START END personality.
Prompts for PERSONALITY with completion when called interactively.

(fn START END &optional PERSONALITY)

12.173.1.113 emacspeak-zap-tts

`emacspeak-zap-tts` [Command]

C-e d z

<fn> d z

Send this command to the TTS directly.

12.173.2 emacspeak-speak Options

`User Option` *emacspeak-audio-indentation* [Variable]

Option indicating if line indentation is cued. If non-nil , then speaking a line indicates its indentation. You can use command 'emacspeak-toggle-audio-indentation' bound to C-e d i to toggle this setting..

`User Option` *emacspeak-audio-indentation-method* [Variable]

Current technique used to cue indentation. Default is 'speak'. You can specify 'tone' for producing a beep indicating the indentation. Automatically becomes local in any buffer where it is set.

`User Option` *emacspeak-character-echo* [Variable]

If t, then emacspeak echoes characters as you type. You can use C-e d k to toggle this setting.

`User Option` *emacspeak-line-echo* [Variable]

If t, then emacspeak echoes lines as you type. You can use C-e d l to set this option.

`User Option` *emacspeak-mail-alert* [Variable]

Option to indicate cueing of new mail. If t, emacspeak will alert you about newly arrived mail with an auditory icon when displaying the mode line. You can use command 'emacspeak-toggle-mail-alert' bound to C-e M-m to set this option. If you have online access to a voicemail drop, you can have a voice-mail alert set up by specifying the location of the voice-mail drop via custom option emacspeak-voicemail-spool-file.

`User Option` *emacspeak-mail-alert-interval* [Variable]

Interval in seconds between mail alerts for the same pending message.

`User Option` *emacspeak-mail-spool-file* [Variable]

Mail spool file examined to alert you about newly arrived mail.

`User Option` *emacspeak-silence-hook* [Variable]
Functions run after emacspeak-silence is called.

`User Option` *emacspeak-speak-default-os-coding-system* [Variable]
List of coding systems on this platform.

`User Option` *emacspeak-speak-filter-persistent-store* [Variable]
File where emacspeak filters are persisted.

`User Option` *emacspeak-speak-line-column-filter* [Variable]
List that specifies columns to be filtered. The list when set holds pairs of start-col.end-col pairs that specifies the columns that should not be spoken. Each column contains a single character –this is inspired by cut -c on UNIX.

`User Option` *emacspeak-speak-maximum-line-length* [Variable]
Threshold for determining 'long' lines. Emacspeak will ask for confirmation before speaking lines that are longer than this length. This is to avoid accidentally opening a binary file and torturing the speech synthesizer with a long string of gibberish.

`User Option` *emacspeak-speak-messages* [Variable]
Option indicating if messages are spoken. If nil, emacspeak will not speak messages as they are echoed to the message area. You can use command 'emacspeak-toggle-speak-messages' bound to C-e q.

`User Option` *emacspeak-speak-time-format-string* [Variable]
Format string that specifies how the time should be spoken. See the documentation for function 'format-time-string'

`User Option` *emacspeak-speak-zoneinfo-directory* [Variable]
Directory containing timezone data.

`User Option` *emacspeak-use-header-line* [Variable]
Use default header line defined by Emacspeak for buffers that dont customize the header.

`User Option` *emacspeak-voicemail-spool-file* [Variable]
Mail spool file examined to alert you about newly arrived voicemail.

`User Option` *emacspeak-word-echo* [Variable]
If t, then emacspeak echoes words as you type. You can use C-e d w to toggle this option.

12.174 emacspeak-speedbar

This module advises speedbar.el for use with Emacs. The latest speedbar can be obtained from ftp://ftp.ultranet.com/pub/zappo/ This module ensures that speedbar works smoothly outside a windowing system in addition to speech enabling all interactive commands. Emacspeak also adds an Emacspeak environment specific entry point to speedbar –emacspeak-speedbar-goto-speedbar– and binds this

12.174.1 Emacspeak-Speedbar Commands

12.174.1.1 emacspeak-speedbar-click

`emacspeak-speedbar-click` [Command]

Does the equivalent of the mouse click from the keyboard

12.174.1.2 emacspeak-speedbar-goto-speedbar

`emacspeak-speedbar-goto-speedbar` [Command]

Switch to the speedbar

12.175 emacspeak-sql

This module speech enables sql-mode– available from the Emacs package archive. sql-mode.el implemented by the above package sets up an Emacs to SQL interface where you can interactively evaluate SQL expressions.

12.176 emacspeak-sudoku

Playing SuDoku using speech output. Written to discover what type of feedback one needs for this task. See http://emacspeak.blogspot.com/2006/02/playing-sudoku-using-auditory-feedback.html

12.176.1 Emacspeak-Sudoku Commands

12.176.1.1 emacspeak-sudoku-board-columns-summarize

`emacspeak-sudoku-board-columns-summarize` [Command]

Summarize columns — speaks number of remaining cells.

12.176.1.2 emacspeak-sudoku-board-distribution-summarize

`emacspeak-sudoku-board-distribution-summarize` [Command]

Shows distribution of filled numbers.

12.176.1.3 emacspeak-sudoku-board-rows-summarize

`emacspeak-sudoku-board-rows-summarize` [Command]

Summarize rows — speaks number of remaining cells.

12.176.1.4 emacspeak-sudoku-board-sub-squares-summarize

`emacspeak-sudoku-board-sub-squares-summarize` [Command]

Summarize sub-squares — speaks number of remaining cells.

12.176.1.5 emacspeak-sudoku-board-summarizer

`emacspeak-sudoku-board-summarizer` [Command]

Dispatch to appropriate summarizer.

d Number Distribution

r	Row Distribution
c	Column Distribution
s	Sub-square Distribution.

12.176.1.6 emacspeak-sudoku-down-sub-square

`emacspeak-sudoku-down-sub-square` [Command]

Move to top-left corner of sub-square below current one.

12.176.1.7 emacspeak-sudoku-erase-current-column

`emacspeak-sudoku-erase-current-column` [Command]

Erase current column.

12.176.1.8 emacspeak-sudoku-erase-current-row

`emacspeak-sudoku-erase-current-row` [Command]

Erase current row.

12.176.1.9 emacspeak-sudoku-erase-current-sub-square

`emacspeak-sudoku-erase-current-sub-square` [Command]

Erase current sub-square.

12.176.1.10 emacspeak-sudoku-hint

`emacspeak-sudoku-hint` [Command]

Provide hint for current cell.

12.176.1.11 emacspeak-sudoku-history-pop

`emacspeak-sudoku-history-pop` [Command]

Pop saved state off stack and redraw board.

12.176.1.12 emacspeak-sudoku-history-push

`emacspeak-sudoku-history-push` [Command]

Push current state on to history stack.

12.176.1.13 emacspeak-sudoku-how-many-remaining

`emacspeak-sudoku-how-many-remaining` [Command]

Speak number of remaining squares to fill.

12.176.1.14 emacspeak-sudoku-next-sub-square

`emacspeak-sudoku-next-sub-square` [Command]

Move to top-left corner of next sub-square.

12.176.1.15 emacspeak-sudoku-previous-sub-square

`emacspeak-sudoku-previous-sub-square` [Command]

Move to top-left corner of previous sub-square.

12.176.1.16 emacspeak-sudoku-speak-current-cell-coordinates

`emacspeak-sudoku-speak-current-cell-coordinates` [Command]

speak current cell coordinates.

12.176.1.17 emacspeak-sudoku-speak-current-cell-value

`emacspeak-sudoku-speak-current-cell-value` [Command]

Speak value in current cell.

12.176.1.18 emacspeak-sudoku-speak-current-column

`emacspeak-sudoku-speak-current-column` [Command]

Speak current column.

12.176.1.19 emacspeak-sudoku-speak-current-row

`emacspeak-sudoku-speak-current-row` [Command]

Speak current row.

12.176.1.20 emacspeak-sudoku-speak-current-sub-square

`emacspeak-sudoku-speak-current-sub-square` [Command]

Speak current sub-square.

12.176.1.21 emacspeak-sudoku-speak-remaining-in-column

`emacspeak-sudoku-speak-remaining-in-column` [Command]

Speaks number of remaining cells in current column.

12.176.1.22 emacspeak-sudoku-speak-remaining-in-row

`emacspeak-sudoku-speak-remaining-in-row` [Command]

Speaks number of remaining cells in current row.

12.176.1.23 emacspeak-sudoku-speak-remaining-in-sub-square

`emacspeak-sudoku-speak-remaining-in-sub-square` [Command]

Speaks number of remaining cells in current sub-square.

12.176.1.24 emacspeak-sudoku-up-sub-square

`emacspeak-sudoku-up-sub-square` [Command]

Move to top-left corner of sub-square above current one.

12.177 emacspeak-supercite

Speech-enable supercite.

12.178 emacspeak-syslog

SYSLOG-MODE == Working with various log files. Install package syslog-mode from melpa.

12.178.1 emacspeak-syslog Options

`User Option` *voice-syslog-debug* [Variable]

Personality used for syslog-debug This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

`User Option` *voice-syslog-error* [Variable]

Personality used for syslog-error This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-syslog-file* [Variable]

Personality used for syslog-file This personality uses voice-smoothen-extra whose effect can be changed globally by customizing voice-smoothen-extra-settings.

`User Option` *voice-syslog-hide* [Variable]

Personality used for syslog-hide This personality uses voice-annotate whose effect can be changed globally by customizing voice-annotate-settings.

`User Option` *voice-syslog-hour* [Variable]

Personality used for syslog-hour This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-syslog-info* [Variable]

Personality used for syslog-info This personality uses voice-highlight whose effect can be changed globally by customizing voice-highlight-settings.

`User Option` *voice-syslog-ip* [Variable]

Personality used for syslog-ip This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-syslog-su* [Variable]

Personality used for syslog-su This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-syslog-warn* [Variable]

Personality used for syslog-warn This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

12.179 emacspeak-tab-bar

tab-bar == tabs for window configuration. Speech-enable tab-bar interaction. If you have *browse-url-new-window-flag* set to T to have EWW open Web pages in a new buffer, then set *eww-browse-url-new-window-is-tab* to nil to avoid leaking tabs.

12.179.1 emacspeak-tab-bar Options

`User Option` *voice-tab-bar* [Variable]

Personality used for tab-bar This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-tab-bar-tab* [Variable]

Personality used for tab-bar-tab This personality uses voice-highlight whose effect can be changed globally by customizing voice-highlight-settings.

`User Option` *voice-tab-bar-tab-inactive* [Variable]

Personality used for tab-bar-tab-inactive This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

`User Option` *voice-tab-line* [Variable]

Personality used for tab-line This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

12.180 emacspeak-table

Implements a module that provides a high level interface to tabulated information.

12.181 emacspeak-table-ui

User interface to tables

12.181.1 Emacspeak-Table-Ui Commands

12.181.1.1 emacspeak-table-copy-current-element-to-kill-ring

`emacspeak-table-copy-current-element-to-kill-ring` [Command]

`C-e C-t w`

`<fn> C-t w`

Copy current table element to kill ring.

12.181.1.2 emacspeak-table-copy-current-element-to-register

`emacspeak-table-copy-current-element-to-register` (*register*) [Command]

`C-e C-t x`

`<fn> C-t x`

Copy current table element to specified register.

(fn REGISTER)

12.181.1.3 emacspeak-table-copy-to-clipboard

`emacspeak-table-copy-to-clipboard` [Command]

C-e C-t k

<fn> C-t k
Copy table in current buffer to the table clipboard.
Current buffer must be in emacspeak-table mode.

12.181.1.4 emacspeak-table-display-table-in-region

`emacspeak-table-display-table-in-region` (*start end*) [Command]

C-e M-i

<fn> M-i

Recognize tabular data in current region and display it in table browsing mode in a a separate buffer.
emacspeak table mode is designed to let you browse tabular data using all the power of the two-dimensional spatial layout while giving you sufficient contextual information. The tables subdirectory of the emacspeak distribution contains some sample tables –these are the CalTrain schedules. Execute command 'describe-mode' bound to C-h m in a buffer that is in emacspeak table mode to read the documentation on the table browser.

(fn START END)

12.181.1.5 emacspeak-table-find-csv-file

`emacspeak-table-find-csv-file` (*filename*) [Command]

C-e C-t ,

<fn> C-t ,
Process a csv (comma separated values) file.
The processed data is presented using emacspeak table navigation.

(fn FILENAME)

12.181.1.6 emacspeak-table-find-file

`emacspeak-table-find-file` (*filename*) [Command]

Open a file containing table data and display it in table mode.
emacspeak table mode is designed to let you browse tabular data using all the power of the two-dimensional spatial layout while giving you sufficient contextual information. The etc/tables subdirectory of the emacspeak distribution contains some sample tables –these are the CalTrain schedules. Execute command 'describe-mode' bound to C-h m in a buffer that is in emacspeak table mode to read the documentation on the table browser.

(fn FILENAME)

12.181.1.7 emacspeak-table-goto

`emacspeak-table-goto` (*row column*) [Command]

C-e C-t j

<fn> C-t j
Prompt for a table cell coordinates and jump to it.

(fn ROW COLUMN)

12.181.1.8 emacspeak-table-goto-bottom

`emacspeak-table-goto-bottom` [Command]

C-e C-t B

<fn> C-t B
C-e C-t M->
<fn> C-t M->
Goes to the bottom of the current column.

12.181.1.9 emacspeak-table-goto-left

`emacspeak-table-goto-left` [Command]

C-e C-t A

C-e C-t <

<fn> C-t A
<fn> C-t <
Goes to the left of the current row.

12.181.1.10 emacspeak-table-goto-right

`emacspeak-table-goto-right` [Command]

C-e C-t E

C-e C-t >

<fn> C-t E
<fn> C-t >
Goes to the right of the current row.

12.181.1.11 emacspeak-table-goto-top

`emacspeak-table-goto-top` [Command]

C-e C-t T

<fn> C-t T
C-e C-t M-<
<fn> C-t M-<
Goes to the top of the current column.

12.181.1.12 emacspeak-table-mode

`emacspeak-table-mode` [Command]

Major mode for browsing tables.
Table mode is designed to allow speech users to browse tabular data with full contextual feedback while retaining all the power of the two-dimensional spatial layout of tables.

In table mode, the arrow keys move between cells of the table. Emacspeak speaks the cell contents in a user-customizable way. The visual display is kept in sync with the speech you hear; however Emacspeak is examining the entire table in order to speak the current cell content intelligently.

You can interactively specify that emacspeak should speak either the row or column header (or both) while speaking each cell. You can also specify a row or column filter that should be applied when speaking entire rows or columns –this lets you view slices of a table. You can move to a specific row or column by searching the cell contents or by searching the row or column headers to locate items of interest.

Here is a short description of the special commands provided in this mode.

The next four commands help you move to the edges of the table:

E	emacspeak-table-goto-right
A	emacspeak-table-goto-left
B	emacspeak-table-goto-bottom
T	emacspeak-table-goto-top

The next two commands let you search the table.
The commands ask you if you want to search rows or columns.
When searching headers remember that row 0 is the column header, and that column 0 is the row header.

h	emacspeak-table-search-headers
s	emacspeak-table-search

The next command lets you specify how cell contents should be spoken. Specify one of: 'b' for both, 'c' for column, 'r' for row, 'f' for row filtering and 'g' for column filtering. –table cells with then be spoken with both (or either)row and column headers, or with the filter applied.

a	emacspeak-table-select-automatic-speaking-method

The next set of commands speak the current table cell:

.	emacspeak-table-speak-coordinates
b	emacspeak-table-speak-both-headers-and-element
SPC	emacspeak-table-speak-current-element
c	emacspeak-table-speak-column-header-and-element
r	e macspeak-table-speak-row-header-and-element

The next set of commands navigate the table:

right	emacspeak-table-next-column
left	emacspeak-table-previous-column
down	emacspeak-table-next-row
up	emacspeak-table-previous-row
j	emacspeak-table-goto
S-tab	emacspeak-table-previous-column
TAB	emacspeak-table-next-column

Row and Column Filtering

Filtering is designed to let you view slices of a table.
They are specified as lists of numbers and strings.
The concept is best explained with an example.

A row filter specifies which of the entries in the current row should be spoken.Entries are numbered starting with 0. Thus, when working with a table having 8 columns, a row filter of (1 2 3) will speak only entries 1 2 and 3. Use the sample tables in etc/tables to familiarize yourself with this feature. Note that you can intersperse meaningful strings in the list that specifies the filter.

Full List Of Keybindings:
key binding
— ——-

C-b emacspeak-table-previous-column
C-f emacspeak-table-next-column
TAB emacspeak-table-next-column
C-n emacspeak-table-next-row
C-p emacspeak-table-previous-row
ESC Prefix Command
SPC emacspeak-table-speak-current-element
# emacspeak-table-sort-on-current-column
, emacspeak-table-find-csv-file
. emacspeak-table-speak-coordinates
< emacspeak-table-goto-left
= emacspeak-table-speak-dimensions
> emacspeak-table-goto-right
A emacspeak-table-goto-left

B emacspeak-table-goto-bottom
C emacspeak-table-search-column
E emacspeak-table-goto-right
Q emacspeak-kill-buffer-quietly
R emacspeak-table-search-row
T emacspeak-table-goto-top
a emacspeak-table-select-automatic-speaking-method
b emacspeak-table-speak-both-headers-and-element
c emacspeak-table-speak-column-header-and-element
f emacspeak-table-speak-row-filtered
g emacspeak-table-speak-column-filtered
h emacspeak-table-search-headers
j emacspeak-table-goto
k emacspeak-table-copy-to-clipboard
n emacspeak-table-next-row
p emacspeak-table-previous-row
q quit-window
r emacspeak-table-speak-row-header-and-element
s emacspeak-table-search
v emacspeak-table-view-csv-buffer
w emacspeak-table-copy-current-element-to-kill-ring
x emacspeak-table-copy-current-element-to-register
<S-tab> emacspeak-table-previous-column
<down> emacspeak-table-next-row
<left> emacspeak-table-previous-column
<right> emacspeak-table-next-column
<up> emacspeak-table-previous-row

M-< emacspeak-table-goto-top
M-> emacspeak-table-goto-bottom
M-l emacspeak-table-ui-filter-load
M-s emacspeak-table-ui-filter-save

In addition to any hooks its parent mode 'special-mode' might have run, this mode runs the hook 'emacspeak-table-mode-hook', as the final or penultimate step during initialization.

12.181.1.13 emacspeak-table-next-column

`emacspeak-table-next-column` (**&optional** *count*) [Command]

C-e C-t TAB
C-e C-t C-f
C-e C-t <right>
<fn> C-t TAB
<fn> C-t C-f
<fn> C-t <right>
Move to the next column if possible

(fn &optional COUNT)

12.181.1.14 emacspeak-table-next-row

`emacspeak-table-next-row` (**&optional** *count*) [Command]

C-e C-t n

C-e C-t C-n
C-e C-t <down>
<fn> C-t n
<fn> C-t C-n
<fn> C-t <down>
Move to the next row if possible

(fn &optional COUNT)

12.181.1.15 emacspeak-table-paste-from-clipboard

`emacspeak-table-paste-from-clipboard` [Command]

Paste the emacspeak table clipboard into the current buffer.
Use the major mode of this buffer to decide what kind of table
markup to use.

12.181.1.16 emacspeak-table-previous-column

`emacspeak-table-previous-column` (**&optional** *count*) [Command]

C-e C-t C-b
C-e C-t <left>
C-e C-t <S-tab>
<fn> C-t C-b
<fn> C-t <left>
<fn> C-t <S-tab>
Move to the previous column if possible

(fn &optional COUNT)

12.181.1.17 emacspeak-table-previous-row

`emacspeak-table-previous-row` (**&optional** *count*) [Command]

C-e C-t p

C-e C-t C-p
C-e C-t <up>
<fn> C-t p
<fn> C-t C-p
<fn> C-t <up>
Move to the previous row if possible

(fn &optional COUNT)

12.181.1.18 emacspeak-table-search

`emacspeak-table-search` (**&optional** *what*) [Command]

C-e C-t s

<fn> C-t s
Search the table for matching elements. Interactively prompts for row or column to search and pattern to look for. If there is a match, makes the matching cell current. When called from a program, 'what' can be either 'row' or 'column'.

(fn &optional WHAT)

12.181.1.19 emacspeak-table-search-column

`emacspeak-table-search-column` [Command]

C-e C-t C

<fn> C-t C
Search in current table column.

12.181.1.20 emacspeak-table-search-headers

`emacspeak-table-search-headers` [Command]

C-e C-t h

<fn> C-t h
Search the table row or column headers. Interactively prompts for row or column to search and pattern to look for. If there is a match, makes the matching row or column current.

12.181.1.21 emacspeak-table-search-row

`emacspeak-table-search-row` [Command]

C-e C-t R

<fn> C-t R
Search in current table row.

12.181.1.22 emacspeak-table-select-automatic-speaking-method

`emacspeak-table-select-automatic-speaking-method` [Command]

C-e C-t a

<fn> C-t a
Interactively select the kind of automatic speech to produce when browsing table elements

12.181.1.23 emacspeak-table-sort-on-current-column

`emacspeak-table-sort-on-current-column` [Command]

C-e C-t #

<fn> C-t #
Sort table on current column.

12.181.1.24 emacspeak-table-speak-both-headers-and-element

`emacspeak-table-speak-both-headers-and-element` [Command]

C-e C-t b

<fn> C-t b
Speak both row and column header and table element

12.181.1.25 emacspeak-table-speak-column-filtered

`emacspeak-table-speak-column-filtered` (**&optional** *prefix*) [Command]

C-e C-t g

<fn> C-t g
Speaks a table column after applying a specified column filter.
Optional prefix arg prompts for a new filter.

(fn &optional PREFIX)

12.181.1.26 emacspeak-table-speak-column-header-and-element

`emacspeak-table-speak-column-header-and-element` [Command]

C-e C-t c

<fn> C-t c
Speak column header and table element

12.181.1.27 emacspeak-table-speak-coordinates

`emacspeak-table-speak-coordinates` [Command]

C-e C-t .

<fn> C-t .
Speak current table coordinates.

12.181.1.28 emacspeak-table-speak-current-element

`emacspeak-table-speak-current-element` [Command]

C-e C-t SPC
<fn> C-t SPC
Speak current table element

12.181.1.29 emacspeak-table-speak-dimensions

`emacspeak-table-speak-dimensions` [Command]

C-e C-t =

<fn> C-t =
Speak current table dimensions.

12.181.1.30 emacspeak-table-speak-row-filtered

`emacspeak-table-speak-row-filtered` (**&optional** *prefix*) [Command]

C-e C-t f

<fn> C-t f
Speaks a table row after applying a specified row filter.
Optional prefix arg prompts for a new filter.

(fn &optional PREFIX)

12.181.1.31 emacspeak-table-speak-row-header-and-element

`emacspeak-table-speak-row-header-and-element` [Command]

C-e C-t r

<fn> C-t r
Speak row header and table element

12.181.1.32 emacspeak-table-ui-filter-load

`emacspeak-table-ui-filter-load` (*file*) [Command]

C-e C-t M-l
<fn> C-t M-l
Load saved filter settings.

(fn FILE)

12.181.1.33 emacspeak-table-ui-filter-save

`emacspeak-table-ui-filter-save` (*file*) [Command]

C-e C-t M-s
<fn> C-t M-s
Save out filter settings.

(fn FILE)

12.181.1.34 emacspeak-table-view-csv-buffer

`emacspeak-table-view-csv-buffer` (**&optional** *buffer-name*) [Command]

C-e C-t v

<fn> C-t v

Process a csv (comma separated values) data.
The processed data is presented using emacspeak table navigation.

(fn &optional BUFFER-NAME)

12.181.1.35 emacspeak-table-view-csv-url

`emacspeak-table-view-csv-url` (*url* **&optional** *buffer-name*) [Command]

Process a csv (comma separated values) data at 'URL'.
The processed data is presented using emacspeak table navigation.

(fn URL &optional BUFFER-NAME)

12.182 emacspeak-tabulate

This module is a simple table recognizer. Can recognize the columns in tabulated output, e.g. ps, ls output

12.182.1 Emacspeak-Tabulate Commands

12.182.1.1 emacspeak-tabulate-region

`emacspeak-tabulate-region` (*start end* **&optional** *mark-fields*) [Command]

C-e i

<fn> i

Voicifies the white-space of a table if one found. Optional interactive prefix arg mark-fields specifies if the header row information is used to mark fields in the white-space.

(fn START END &optional MARK-FIELDS)

12.183 emacspeak-tapestry

emacspeak extensions to speak window widnow layouts

12.183.1 Emacspeak-Tapestry Commands

12.183.1.1 emacspeak-speak-window-layout

`emacspeak-speak-window-layout` (**&optional** *details*) [Command]

Describe the current layout of visible buffers in current frame.
Use interactive prefix arg to get coordinate positions of the
displayed buffers.

(fn &optional DETAILS)

12.183.1.2 emacspeak-tapestry-describe-tapestry

`emacspeak-tapestry-describe-tapestry` (**&optional** *details*) [Command]

C-e M-t

<fn> M-t

Describe the current layout of visible buffers in current frame.
Use interactive prefix arg to get coordinate positions of the
displayed buffers.

(fn &optional DETAILS)

12.183.1.3 emacspeak-tapestry-select-window-by-name

`emacspeak-tapestry-select-window-by-name` (*buffer-name*) [Command]

C-e W

<fn> W

Select window by the name of the buffer it displays.
This is useful when using modes like ECB or the new GDB UI where
  you want to preserve the window layout
but quickly switch to a window by name.

(fn BUFFER-NAME)

12.184 emacspeak-tar

Auditory interface to tar mode

12.184.1 Emacspeak-Tar Commands

12.184.1.1 emacspeak-tar-speak-file-date

`emacspeak-tar-speak-file-date` [Command]

Speak date of file current entry

12.184.1.2 emacspeak-tar-speak-file-permissions

`emacspeak-tar-speak-file-permissions` [Command]

Speak permissions of file current entry

12.184.1.3 emacspeak-tar-speak-file-size

`emacspeak-tar-speak-file-size` [Command]

Speak size of file current entry

12.185 emacspeak-tcl

Provide additional advice to tcl-mode

12.186 emacspeak-tempo

tempo.el provides the infrastructure for building up templates. This is used by html-helper-mode to allow for easy writing of HTML This module extends Emacspeak to provide fluent spoken feedback

12.187 emacspeak-tetris

Speech-enables tetris.

12.187.1 Emacspeak-Tetris Commands

12.187.1.1 emacspeak-tetris-goto-bottom-row

`emacspeak-tetris-goto-bottom-row` [Command]

Move to and speak bottom row

12.187.1.2 emacspeak-tetris-goto-top-row

`emacspeak-tetris-goto-top-row` [Command]

Move to and speak the top row

12.187.1.3 emacspeak-tetris-speak-column

`emacspeak-tetris-speak-column` (**&optional** *x*) [Command]

Speak column –default is to speak current column

(fn &optional X)

12.187.1.4 emacspeak-tetris-speak-coordinates

`emacspeak-tetris-speak-coordinates` [Command]

Speak current position

12.187.1.5 emacspeak-tetris-speak-current-shape

`emacspeak-tetris-speak-current-shape` [Command]

Speak current shape

12.187.1.6 emacspeak-tetris-speak-current-shape-and-coordinates

`emacspeak-tetris-speak-current-shape-and-coordinates` [Command]

Speak shape orientation and coordinates

12.187.1.7 emacspeak-tetris-speak-next-shape

`emacspeak-tetris-speak-next-shape` [Command]

Speak next shape

12.187.1.8 emacspeak-tetris-speak-row

`emacspeak-tetris-speak-row` [Command]

Speak current tetris row

12.187.1.9 emacspeak-tetris-speak-row-number

`emacspeak-tetris-speak-row-number` [Command]

Speak where on the tetris board we are

12.187.1.10 emacspeak-tetris-speak-score

`emacspeak-tetris-speak-score` [Command]

Speak the score

12.187.1.11 emacspeak-tetris-speak-x-coordinate

`emacspeak-tetris-speak-x-coordinate` [Command]

Speak current position

12.188 emacspeak-texinfo

This module speech enables net-texinfo mode

12.189 emacspeak-threes

THREES == threes game. This module speech-enable the game. `https://en.wikipedia.org/wiki/Threes` for history of the game and details of game play. This module adds additional convenience keybindings to the default arrow-key bindings implemented in threes.el. In addition, this module implements commands that speak the board as well as getting a column-specific view of the board.

f	Move right
b	Move left
n	Move down
p	Move up
SPC	Speak the board
/	Speak board by column.
.	Speak current score.
,	Speak number of zeros on the board.
s	Save current state
u	Pop state from stack
?	Speak next tile

The updated board is spoken after each turn. The next upcoming tile is spoken after the current state of the board. You can use *SPC* and */* to review the board.

12.189.1 Emacspeak-Threes Commands

12.189.1.1 emacspeak-threes-0

`emacspeak-threes-0` [Command]

Set next tile.

12.189.1.2 emacspeak-threes-1

`emacspeak-threes-1` [Command]

Set next tile.

12.189.1.3 emacspeak-threes-2

`emacspeak-threes-2` [Command]

Set next tile.

12.189.1.4 emacspeak-threes-3

`emacspeak-threes-3` [Command]

Set next tile.

12.189.1.5 emacspeak-threes-export

`emacspeak-threes-export` (**&optional** *prompt*) [Command]

Exports game stack to a file.
Optional interactive prefix arg prompts for a file.
Note that the file is overwritten silently.

(fn &optional PROMPT)

12.189.1.6 emacspeak-threes-import

`emacspeak-threes-import` (**&optional** *prompt*) [Command]

Import game.
Optional interactive prefix arg prompts for a filename.

(fn &optional PROMPT)

12.189.1.7 emacspeak-threes-pop-state

`emacspeak-threes-pop-state` [Command]

Reset state from stack.

12.189.1.8 emacspeak-threes-prune-stack

`emacspeak-threes-prune-stack` (*drop*) [Command]

Prune game stack to specified length.

(fn DROP)

12.189.1.9 emacspeak-threes-push-state

`emacspeak-threes-push-state` [Command]

Push current game state on stack.

12.189.1.10 emacspeak-threes-score

`emacspeak-threes-score` [Command]

Speak the score.

12.189.1.11 emacspeak-threes-speak-board

`emacspeak-threes-speak-board` [Command]

Speak the board.

12.189.1.12 emacspeak-threes-speak-empty-count

`emacspeak-threes-speak-empty-count` [Command]

Speak number of cells that are non-empty.

12.189.1.13 emacspeak-threes-speak-next

`emacspeak-threes-speak-next` [Command]

Speak upcoming tile.

12.189.1.14 emacspeak-threes-speak-transposed-board

`emacspeak-threes-speak-transposed-board` [Command]

Speak the board by columns.

12.190 emacspeak-tide

TIDE == Typescript IDE for emacs. This module speech-enables both tide and typescript-mode.

12.190.1 emacspeak-tide Options

`User Option` *voice-tide-file* [Variable]

Personality used for tide-file This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-tide-line-number* [Variable]

Personality used for tide-line-number This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-tide-match* [Variable]

Personality used for tide-match This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

12.191 emacspeak-todo-mode

todo-mode (part of Emacs 21) provides todo-lists that can be integrated with the Emacs calendar. This module speech-enables todo-mode

12.192 emacspeak-transient

TRANSIENT == Transient commands — used by magit and friends. This module speech-enables transient.

12.192.1 Introduction

Package Transient is similar to package Hydra in the sense that it can be used to create a sequence of chained/hierarchical commands that are invoked via a sequence of keys. It is used by Magit for dispatching to the various Git commands. Speech-enabling package Transient results in the various interactive commands producing auditory feedback. Transient shows an ephemeral window with the currently available commands, Emacspeak speech-enables transient–show to speak that content.

Finally, this module defines a new minor mode called transient-emacspeak that enables interactive browsing of the contents displayed temporarily. Note that without this functionality, learning complex packages like Magit would be difficult because the list of available commands (potentially very long) gets spoken in its entirety by the advice on transient–show.

12.192.2 Browsing Contents Of transient–show

When executing a command defined via Transient — e.g. command Magit-dispatch and friends, press `C-z` (transient-suspend) to temporarily suspend the currently active transient. Emacspeak now displays a *transient-emacspeak* buffer that displays the contents of the most recently displayed transient choices. Pressing `C-c` resumes the transient; Pressing `C-q` quits the transient.

12.192.3 Emacspeak-Transient Commands

12.192.3.1 emacspeak-transient-mode

`emacspeak-transient-mode` [Command]

emacspeak integration with Transient.

In addition to any hooks its parent mode ‘special-mode’ might have run,
this mode runs the hook ‘emacspeak-transient-mode-hook’, as the final or penultimate step
during initialization.

key binding
— ——-

12.192.4 emacspeak-transient Options

`User Option` *voice-transient-argument* [Variable]

Personality used for transient-argument This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

`User Option` *voice-transient-disabled-suffix* [Variable]

Personality used for transient-disabled-suffix This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

`User Option` *voice-transient-enabled-suffix* [Variable]

Personality used for transient-enabled-suffix This personality uses voice-brighten whose effect can be changed globally by customizing voice-brighten-settings.

`User Option` *voice-transient-heading* [Variable]

Personality used for transient-heading This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-transient-inactive-argument* [Variable]

Personality used for transient-inactive-argument This personality uses voice-smoothen-extra whose effect can be changed globally by customizing voice-smoothen-extra-settings.

`User Option` *voice-transient-inactive-value* [Variable]

Personality used for transient-inactive-value This personality uses voice-smoothen-extra whose effect can be changed globally by customizing voice-smoothen-extra-settings.

`User Option` *voice-transient-key* [Variable]

Personality used for transient-key This personality uses voice-highlight whose effect can be changed globally by customizing voice-highlight-settings.

`User Option` *voice-transient-mismatched-key* [Variable]

Personality used for transient-mismatched-key This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-transient-nonstandard-key* [Variable]

Personality used for transient-nonstandard-key This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-transient-unreachable* [Variable]

Personality used for transient-unreachable This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-transient-unreachable-key* [Variable]

Personality used for transient-unreachable-key This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-transient-value* [Variable]

Personality used for transient-value This personality uses voice-brighten whose effect can be changed globally by customizing voice-brighten-settings.

12.193 emacspeak-twittering

module twittering-mode.el is found on the emacs wiki This module speech-enables twittering-mode which works using oauth for authentication. Note: As of Thu Sep 2 08:11:25 PDT 2010 twit.el is broken.

Advises interactive functions to speak

12.193.1 Emacspeak-Twittering Commands

12.193.1.1 emacspeak-twittering-jump-to-following-url

`emacspeak-twittering-jump-to-following-url` [Command]

Move to and open closest URI following point.

12.193.1.2 emacspeak-twittering-speak-this-tweet

`emacspeak-twittering-speak-this-tweet` (**&optional** *copy-as-kill*) [Command]

Speak tweet under point.
With interactive prefix arg 'copy-as-kill', copy it to kill ring as well.

(fn &optional COPY-AS-KILL)

12.193.1.3 emacspeak-twittering-twarc

`emacspeak-twittering-twarc` (**&optional** *whose*) [Command]

Download data using credentials for signed-in user.
Interactive prefix arg 'whose' prompts for a username whose
timeline we download.

(fn &optional WHOSE)

12.194 emacspeak-typo

TYPO == Typographical Editing This module speech-enables typo-mode. Typo-mode's magic insertion commands are speech-enabled to speak the inserted char.

12.195 emacspeak-url-template

It is often useful to have "parametrized hot list entries" i.e., hotlist entries that are "templates" for the actual URL. The user provides values for the parametrized portions of the URL e.g. the date. See See Section 12.232 [URL Templates], page 360, for details on the URL templates that are presently defined.

12.195.1 Emacspeak-Url-Template Commands

12.195.1.1 emacspeak-url-template-fetch

`emacspeak-url-template-fetch` (**&optional** *documentation*) [Command]

C-e u

`<fn> u`

Fetch a pre-defined resource.
Use Emacs completion to obtain a list of available resources.
Resources typically prompt for the relevant information
before completing the request.
Optional interactive prefix arg displays documentation for specified resource.

(fn &optional DOCUMENTATION)

12.195.1.2 emacspeak-url-template-help

`emacspeak-url-template-help` [Command]

Display documentation for a URL template.
Use Emacs completion to obtain a list of available
resources.

12.195.1.3 emacspeak-url-template-load

`emacspeak-url-template-load` (*file*) [Command]

Load URL template resources from specified location.

(fn FILE)

12.195.1.4 emacspeak-url-template-nls-add-to-wishlist

`emacspeak-url-template-nls-add-to-wishlist` (*book*) [Command]

Add book under point to wishlist.

(fn BOOK)

12.195.1.5 emacspeak-url-template-save

`emacspeak-url-template-save` (*file*) [Command]

Save out url templates.

(fn FILE)

12.195.2 emacspeak-url-template Options

`User Option` *emacspeak-url-template-currency-base* [Variable]

Currency to use as the base when doing currency conversion.

`User Option` *emacspeak-url-template-currency-list* [Variable]

List of currencies for which we request rates by default.

12.196 emacspeak-vdiff

VDIFF == vimdiff Installable from melpa, vdiff enables synchronized movement through diff buffers without resorting to an extra control-panel as is the case with ediff. In addition to speech-enabling interactive commands and setting up face->voice mappings, this module provides commands that speak the current hunk. These are bound in `vdiff-mode-prefix-map`.

- `emacspeak-vdiff-speak-this-hunk` bound to *`SPC`*.
- `emacspeak-vdiff-speak-other-hunk` bound to *`C-SPC`*.
- `emacspeak-vdiff-speak-other-line` bound to *`l`*.

12.196.1 Emacspeak-Vdiff Commands

12.196.1.1 emacspeak-vdiff-speak-other-hunk

`emacspeak-vdiff-speak-other-hunk` [Command]

Speak corresponding hunk from other buffer.

12.196.1.2 emacspeak-vdiff-speak-other-line

`emacspeak-vdiff-speak-other-line` [Command]

Speak corresponding line from other buffer.

12.196.1.3 emacspeak-vdiff-speak-this-hunk

`emacspeak-vdiff-speak-this-hunk` [Command]

Speak VDiff hunk under point.

12.196.2 emacspeak-vdiff Options

`User Option` *voice-vdiff-refine-added* [Variable]

Personality used for vdiff-refine-added This personality uses voice-lock-overlay-0 whose effect can be changed globally by customizing voice-lock-overlay-0-settings.

`User Option` *voice-vdiff-refine-changed* [Variable]

Personality used for vdiff-refine-changed This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

12.197 emacspeak-view

Provide additional advice to view-mode

12.197.1 Emacspeak-View Commands

12.197.1.1 emacspeak-view-line-to-top

`emacspeak-view-line-to-top` [Command]

Moves current line to top of window

12.198 emacspeak-vm

This module extends the mail reader vm. Uses voice locking for message headers and cited messages

12.198.1 Emacspeak-Vm Commands

12.198.1.1 emacspeak-vm-browse-message

`emacspeak-vm-browse-message` [Command]

Browse an email message –read it paragraph at a time.

12.198.1.2 emacspeak-vm-catch-up-all-messages

`emacspeak-vm-catch-up-all-messages` [Command]

Mark all messages in folder to be deleted. Use with caution.

12.198.1.3 emacspeak-vm-locate-subject-line

`emacspeak-vm-locate-subject-line` [Command]

Locates the subject line in a message being read.
Useful when you're reading a message
that has been forwarded multiple times.

12.198.1.4 emacspeak-vm-mode-line

`emacspeak-vm-mode-line` [Command]

VM mode line information.

12.198.1.5 emacspeak-vm-speak-labels

`emacspeak-vm-speak-labels` [Command]

Speak a message's labels

12.198.1.6 emacspeak-vm-speak-message

`emacspeak-vm-speak-message` [Command]

Move point to the message body.

12.198.1.7 emacspeak-vm-toggle-html-mime-demotion

`emacspeak-vm-toggle-html-mime-demotion` [Command]

Toggle state of HTML Mime promotion/Demotion.

12.198.1.8 emacspeak-vm-yank-header

`emacspeak-vm-yank-header` [Command]

Yank specified header into kill ring.

12.198.2 emacspeak-vm Options

`User Option` *emacspeak-vm-cal2text* [Variable]

Executable that converts calendar invitations on standard input to plain text.

`User Option` *emacspeak-vm-customize-mime-settings* [Variable]

Non-nil will cause Emacspeak to configure VM mime settings to match what the author of Emacspeak uses.

`User Option` *emacspeak-vm-pdf2text* [Variable]
Executable that converts PDF on standard input to plain text using pdftotext.

`User Option` *emacspeak-vm-use-raman-settings* [Variable]
Should VM use the customizations used by the author of Emacspeak.

`User Option` *emacspeak-vm-voice-loc* [Variable]
Set this to T if you want messages automatically voice locked. Note that some badly formed mime messages cause trouble.

12.199 emacspeak-vterm

VTERM == vterm using native vterm library

12.199.1 Usage

- Turn on `emacspeak-comint-autospeak` for using the shell.
- Turn off `emacspeak-comint-autospeak` when using full-screen ncurses apps like `vi`.
- Use `vterm-copy-mode` to review the contents of the terminal — *`C-c C-t`*.

12.199.2 emacspeak-vterm Options

`User Option` *voice-vterm-color-black* [Variable]
Personality used for vterm-color-black This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-vterm-color-blue* [Variable]
Personality used for vterm-color-blue This personality uses voice-brighten whose effect can be changed globally by customizing voice-brighten-settings.

`User Option` *voice-vterm-color-cyan* [Variable]
Personality used for vterm-color-cyan This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

`User Option` *voice-vterm-color-default* [Variable]
Personality used for vterm-color-default This personality uses 'paul whose effect can be changed globally by customizing 'paul-settings.

`User Option` *voice-vterm-color-green* [Variable]
Personality used for vterm-color-green This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-vterm-color-inverse-video* [Variable]
Personality used for vterm-color-inverse-video This personality uses 'betty whose effect can be changed globally by customizing 'betty-settings.

`User Option` *voice-vterm-color-magenta* [Variable]
Personality used for vterm-color-magenta This personality uses voice-annotate whose effect can be changed globally by customizing voice-annotate-settings.

`User Option` *voice-vterm-color-underline* [Variable]
Personality used for vterm-color-underline This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

`User Option` *voice-vterm-color-white* [Variable]
Personality used for vterm-color-white This personality uses 'paul whose effect can be changed globally by customizing 'paul-settings.

`User Option` *voice-vterm-color-yellow* [Variable]
Personality used for vterm-color-yellow This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

12.200 emacspeak-vuiet

VUIET == Emacs Music Explorer And Player with last.fm integration This module speech-enables vuiet.

12.200.1 Emacspeak-Vuiet Commands

12.200.1.1 emacspeak-vuiet-track-info

`emacspeak-vuiet-track-info` [Command]
Speak current playing state.

12.201 emacspeak-wdired

Speech-enable wdired to permit in-place renaming of groups of files.

12.202 emacspeak-we

we is for webedit Invoke XSLT to edit/transform Web pages before they get rendered. we makes emacspeak's webedit layer independent of a given Emacs web browser EWW This module will use the abstraction provided by browse-url to handle Web pages.

12.202.1 Emacspeak-We Commands

12.202.1.1 emacspeak-we-class-filter-and-follow

`emacspeak-we-class-filter-and-follow` (*class url* **&optional** *prompt*) [Command]

`C-e x e y`

`<fn> x e y`

Follow url and point, and filter the result by specified class.
Class can be set locally for a buffer, and overridden with an
interactive prefix arg. If there is a known rewrite url rule, that is
used as well.

(fn CLASS URL &optional PROMPT)

12.202.1.2 emacspeak-we-class-filter-and-follow-link

`emacspeak-we-class-filter-and-follow-link` (**&optional** *prompt*) [Command]

C-e x e v

<fn> x e v
Follow url and point, and filter the result by specified class. Class can be set locally for a buffer, and overridden with an interactive prefix arg. If there is a known rewrite url rule, that is used as well.

(fn &optional PROMPT)

12.202.1.3 emacspeak-we-count-matches

`emacspeak-we-count-matches` (*url locator*) [Command]

C-e x e C-f
<fn> x e C-f
Count matches for locator in Web page.

(fn URL LOCATOR)

12.202.1.4 emacspeak-we-count-nested-tables

`emacspeak-we-count-nested-tables` (*url*) [Command]

C-e x e C-x
<fn> x e C-x
Count nested tables in Web page.

(fn URL)

12.202.1.5 emacspeak-we-count-tables

`emacspeak-we-count-tables` (*url*) [Command]

C-e x e C-t
<fn> x e C-t
Count tables in Web page.

(fn URL)

12.202.1.6 emacspeak-we-extract-by-class

`emacspeak-we-extract-by-class` (*class url* **&optional** *speak*) [Command]

C-e x e c

<fn> x e c
Extract elements having specified class attribute from HTML. Extracts specified elements from current WWW page and displays it in a separate

buffer. Interactive use provides list of class values as completion.

(fn CLASS URL &optional SPEAK)

12.202.1.7 emacspeak-we-extract-by-class-list

`emacspeak-we-extract-by-class-list` (*classes url* **&optional** *speak*) [Command]

`C-e x e C`

`<fn> x e C`
Extract elements having class specified in list 'classes' from HTML.
Extracts specified elements from current WWW page and displays it
in a separate buffer. Interactive use provides list of class
values as completion.

(fn CLASSES URL &optional SPEAK)

12.202.1.8 emacspeak-we-extract-by-id

`emacspeak-we-extract-by-id` (*id url* **&optional** *speak*) [Command]

`C-e x e i`

`<fn> x e i`
Extract elements having specified id attribute from HTML. Extracts
specified elements from current WWW page and displays it in a separate
buffer.
Interactive use provides list of id values as completion.

(fn ID URL &optional SPEAK)

12.202.1.9 emacspeak-we-extract-by-id-list

`emacspeak-we-extract-by-id-list` (*ids url* **&optional** *speak*) [Command]

`C-e x e I`

`<fn> x e I`
Extract elements having id specified in list 'ids' from HTML.
Extracts specified elements from current WWW page and displays it in a
separate buffer. Interactive use provides list of id values as completion.

(fn IDS URL &optional SPEAK)

12.202.1.10 emacspeak-we-extract-by-role

`emacspeak-we-extract-by-role` (*role url* **&optional** *speak*) [Command]

`C-e x e r`

`<fn> x e r`
Extract elements having specified role attribute from HTML. Extracts

specified elements from current WWW page and displays it in a separate buffer. Interactive use provides list of role values as completion.

(fn ROLE URL &optional SPEAK)

12.202.1.11 emacspeak-we-extract-matching-urls

`emacspeak-we-extract-matching-urls` (*pattern url* **&optional** *speak*) [Command]

C-e x e u

<fn> x e u
Extracts links whose URL matches pattern.

(fn PATTERN URL &optional SPEAK)

12.202.1.12 emacspeak-we-extract-nested-table

`emacspeak-we-extract-nested-table` (*index url* **&optional** *speak*) [Command]

C-e x e x

<fn> x e x
Extract nested table specified by ‘table-index’. Default is to operate on current web page when in a browser buffer; otherwise prompt for URL. Optional arg ‘speak’ specifies if the result should be spoken automatically.

(fn INDEX URL &optional SPEAK)

12.202.1.13 emacspeak-we-extract-nested-table-list

`emacspeak-we-extract-nested-table-list` (*tables url* **&optional** *speak*) [Command]

C-e x e X

<fn> x e X
Extract specified list of tables from a Web page.

(fn TABLES URL &optional SPEAK)

12.202.1.14 emacspeak-we-extract-speakable

`emacspeak-we-extract-speakable` (*url* **&optional** *speak*) [Command]

C-e x e z

<fn> x e z
Extract elements having class‘speakable’ from HTML.

(fn URL &optional SPEAK)

12.202.1.15 emacspeak-we-extract-table-by-match

`emacspeak-we-extract-table-by-match` (*match url* **&optional** *speak*) [Command]

C-e x e m

<fn> x e m
Extract table containing specified match.
 Optional arg url specifies the page to extract content from.

(fn MATCH URL &optional SPEAK)

12.202.1.16 emacspeak-we-extract-table-by-position

`emacspeak-we-extract-table-by-position` (*pos url* **&optional** *speak*) [Command]

C-e x e t

<fn> x e t
Extract table at specified pos.
Default is to extract from current page.

(fn POS URL &optional SPEAK)

12.202.1.17 emacspeak-we-extract-tables-by-match-list

`emacspeak-we-extract-tables-by-match-list` (*match-list url* **&optional** *speak*) [Command]

C-e x e M

<fn> x e M
Extract specified tables from a WWW page.
Tables are specified by containing match pattern
 found in the match list.

(fn MATCH-LIST URL &optional SPEAK)

12.202.1.18 emacspeak-we-extract-tables-by-position-list

`emacspeak-we-extract-tables-by-position-list` (*positions url* **&optional** *speak*) [Command]

C-e x e T

<fn> x e T
Extract specified list of nested tables from a WWW page.
Tables are specified by their position in the list
 of nested tables found in the page.

(fn POSITIONS URL &optional SPEAK)

12.202.1.19 emacspeak-we-follow-and-extract-main

`emacspeak-we-follow-and-extract-main` (**&optional** *speak*) [Command]

C-e x e P

<fn> x e P
Follow URL, then extract role=main.

(fn &optional SPEAK)

12.202.1.20 emacspeak-we-follow-and-filter-by-id

`emacspeak-we-follow-and-filter-by-id` (*id prompt*) [Command]

C-e x e b

<fn> x e b
Follow url and point, and filter the result by specified id.
Id can be set locally for a buffer, and overridden with an
interactive prefix arg. If there is a known rewrite url rule, that is
used as well.

(fn ID PROMPT)

12.202.1.21 emacspeak-we-junk-by-class

`emacspeak-we-junk-by-class` (*class url* **&optional** *speak*) [Command]

C-e x e d

<fn> x e d
Extract elements not having specified class attribute from HTML. Extracts
specified elements from current WWW page and displays it in a separate
buffer. Interactive use provides list of class values as completion.

(fn CLASS URL &optional SPEAK)

12.202.1.22 emacspeak-we-junk-by-class-list

`emacspeak-we-junk-by-class-list` (*classes url* **&optional** *speak*) [Command]

C-e x e D

C-e x e C-c
<fn> x e D
<fn> x e C-c
Extract elements not having class specified in list ‘classes’ from HTML.
Extracts specified elements from current WWW page and displays it
in a separate buffer. Interactive use provides list of class
values as completion.

(fn CLASSES URL &optional SPEAK)

12.202.1.23 emacspeak-we-style-filter

`emacspeak-we-style-filter` (*style url* **&optional** *speak*) [Command]

C-e x e S

<fn> x e S
Extract elements matching specified style
from HTML. Extracts specified elements from current WWW
page and displays it in a separate buffer. Optional arg url
specifies the page to extract contents from.

(fn STYLE URL &optional SPEAK)

12.202.1.24 emacspeak-we-toggle-xsl-keep-result

`emacspeak-we-toggle-xsl-keep-result` [Command]

C-e x e k

<fn> x e k
Toggle xsl keep result flag.

12.202.1.25 emacspeak-we-url-expand-and-execute

`emacspeak-we-url-expand-and-execute` (**&optional** *prefix*) [Command]

C-e x e e

<fn> x e e
Applies buffer-specific URL expander/executor function.

(fn &optional PREFIX)

12.202.1.26 emacspeak-we-url-rewrite-and-follow

`emacspeak-we-url-rewrite-and-follow` (**&optional** *prompt*) [Command]

Apply a url rewrite rule as specified in the current buffer
before following link under point. If no rewrite rule is
defined, first prompt for one. Rewrite rules are of the
form '(from to)' where from and to are strings. Typically, the
rewrite rule is automatically set up by Emacspeak tools like
websearch where a rewrite rule is known. Rewrite rules are
useful in jumping directly to the printer friendly version of an
article for example. Optional interactive prefix arg prompts for
a rewrite rule even if one is already defined.

(fn &optional PROMPT)

12.202.1.27 emacspeak-we-xpath-filter-and-follow

`emacspeak-we-xpath-filter-and-follow` (**&optional** *prompt*) [Command]

C-e x e p

<fn> x e p
Follow url and point, and filter the result by specified xpath.
XPath can be set locally for a buffer, and overridden with an interactive prefix arg. If there is a known rewrite url rule, that is used as well.

(fn &optional PROMPT)

12.202.1.28 emacspeak-we-xpath-junk-and-follow

`emacspeak-we-xpath-junk-and-follow` (**&optional** *prompt*) [Command]

C-e x e C-p
<fn> x e C-p
Follow url and point, and filter the result by junking elements specified by xpath.
XPath can be set locally for a buffer, and overridden with an interactive prefix arg. If there is a known rewrite url rule, that is used as well.

(fn &optional PROMPT)

12.202.1.29 emacspeak-we-xsl-toggle

`emacspeak-we-xsl-toggle` [Command]

C-e x e o

<fn> x e o
Toggle application of XSL transformations.

12.202.1.30 emacspeak-we-xslt-apply

`emacspeak-we-xslt-apply` (*xsl*) [Command]

C-e x e a

<fn> x e a
Apply specified transformation to current Web page.

(fn XSL)

12.202.1.31 emacspeak-we-xslt-filter

`emacspeak-we-xslt-filter` (*path url* **&optional** *speak*) [Command]

C-e x e f

<fn> x e f
Extract elements matching specified XPath path locator from Web page – default is the current page being viewed.

(fn PATH URL &optional SPEAK)

12.202.1.32 emacspeak-we-xslt-junk

`emacspeak-we-xslt-junk` (*path url* **&optional** *speak*) [Command]

C-e x e j

<fn> x e j
Junk elements matching specified locator.

(fn PATH URL &optional SPEAK)

12.202.1.33 emacspeak-we-xslt-select

`emacspeak-we-xslt-select` (*xsl*) [Command]

C-e x e s

<fn> x e s
Select XSL transformation applied to Web pages before they are displayed .

(fn XSL)

12.202.2 emacspeak-we Options

`User Option` *emacspeak-we-filters-rename-buffer* [Variable]
Set to T if you want the buffer name to contain the applied filter.

12.203 emacspeak-websearch

This module provides utility functions for searching the WWW

12.203.1 Emacspeak-Websearch Commands

12.203.1.1 emacspeak-websearch-accessible-google

`emacspeak-websearch-accessible-google` (*query* **&optional** *options*) [Command]
Use Google Lite (Experimental).
Optional prefix arg prompts for toolbelt options.

(fn QUERY &optional OPTIONS)

12.203.1.2 emacspeak-websearch-amazon-search

`emacspeak-websearch-amazon-search` [Command]
Amazon search.

12.203.1.3 emacspeak-websearch-ask-jeeves

`emacspeak-websearch-ask-jeeves` (*query*) [Command]
Ask Jeeves for the answer.

(fn QUERY)

12.203.1.4 emacspeak-websearch-biblio-search

`emacspeak-websearch-biblio-search` (*query*) [Command]

Search Computer Science Bibliographies.

(fn QUERY)

12.203.1.5 emacspeak-websearch-citeseer-search

`emacspeak-websearch-citeseer-search` (*term*) [Command]

Perform a CiteSeer search.

(fn TERM)

12.203.1.6 emacspeak-websearch-dispatch

`emacspeak-websearch-dispatch` [Command]

`C-e ?`

`<fn> ?`

Press '?' to list available search engines.

This interface attempts to speak the most relevant information on the result page.

12.203.1.7 emacspeak-websearch-foldoc-search

`emacspeak-websearch-foldoc-search` (*query*) [Command]

Perform a FolDoc search.

(fn QUERY)

12.203.1.8 emacspeak-websearch-google

`emacspeak-websearch-google` (*query* **&optional** *flag*) [Command]

Perform a Google search. First optional interactive prefix arg
'flag' prompts for additional search options. Second interactive
prefix arg is equivalent to hitting the I'm Feeling Lucky button on Google.

(fn QUERY &optional FLAG)

12.203.1.9 emacspeak-websearch-google-feeling-lucky

`emacspeak-websearch-google-feeling-lucky` (*query*) [Command]

Do a I'm Feeling Lucky Google search.

(fn QUERY)

12.203.1.10 emacspeak-websearch-google-news

`emacspeak-websearch-google-news` [Command]

Invoke Google News url template.

12.203.1.11 emacspeak-websearch-google-search-in-date-range

`emacspeak-websearch-google-search-in-date-range` [Command]
Use this from inside the calendar to do Google date-range searches.

12.203.1.12 emacspeak-websearch-google-with-toolbelt

`emacspeak-websearch-google-with-toolbelt` (*query*) [Command]
Launch Google search with toolbelt.

(fn QUERY)

12.203.1.13 emacspeak-websearch-gutenberg

`emacspeak-websearch-gutenberg` (*type query*) [Command]
Perform an Gutenberg search

(fn TYPE QUERY)

12.203.1.14 emacspeak-websearch-help

`emacspeak-websearch-help` [Command]
Displays key mapping used by Emacspeak Websearch.

12.203.1.15 emacspeak-websearch-merriam-webster-search

`emacspeak-websearch-merriam-webster-search` (*query*) [Command]
Search the Merriam Webster Dictionary.

(fn QUERY)

12.203.1.16 emacspeak-websearch-wikipedia-search

`emacspeak-websearch-wikipedia-search` (*query*) [Command]
Search Wikipedia using Google.

(fn QUERY)

12.203.1.17 emacspeak-websearch-youtube-search

`emacspeak-websearch-youtube-search` (*query*) [Command]
YouTube search.

(fn QUERY)

12.204 emacspeak-webspace

WEBSPACE == Smart Web Gadgets For The Emacspeak Desktop

12.204.1 Emacspeak-Webspace Commands

12.204.1.1 emacspeak-webspace-feed-reader

`emacspeak-webspace-feed-reader` (**&optional** *refresh*) [Command]

C-. R

C-' R

C-x @ s R

Display Feed Reader Feed list in a WebSpace buffer.
Optional interactive prefix arg forces a refresh.

(fn &optional REFRESH)

12.204.1.2 emacspeak-webspace-filter

`emacspeak-webspace-filter` [Command]

Open headline at point by following its link property and filter for content.

12.204.1.3 emacspeak-webspace-headlines

`emacspeak-webspace-headlines` [Command]

Startup Headlines ticker using RSS/Atom feeds.

12.204.1.4 emacspeak-webspace-headlines-browse

`emacspeak-webspace-headlines-browse` [Command]

Display buffer of browsable headlines.

12.204.1.5 emacspeak-webspace-headlines-update

`emacspeak-webspace-headlines-update` [Command]

Setup news updates.
Updated headlines found in emacspeak-webspace-headlines.

12.204.1.6 emacspeak-webspace-mode

`emacspeak-webspace-mode` [Command]

Major mode for Webspace interaction.

key binding
— ——-

TAB forward-button
ESC Prefix Command
’ emacspeak-speak-rest-of-buffer
. emacspeak-webspace-filter
/ search-forward
< beginning-of-buffer
> end-of-buffer
? search-backward

b backward-button
f forward-button
n forward-button
p backward-button
q bury-buffer
y emacspeak-webspace-yank-link
<backtab> backward-button

C-M-i backward-button

In addition to any hooks its parent mode 'special-mode' might have run,
this mode runs the hook 'emacspeak-webspace-mode-hook', as the final or penultimate step
during initialization.

12.204.1.7 emacspeak-webspace-open

`emacspeak-webspace-open` [Command]

Open headline at point by following its link property.

12.204.1.8 emacspeak-webspace-yank-link

`emacspeak-webspace-yank-link` [Command]

Yank link under point into kill ring.

12.204.2 emacspeak-webspace Options

`User Option` *emacspeak-webspace-feeds* [Variable]

Feeds to use in Headline Ticker.

12.205 emacspeak-widget

This module implements the necessary extensions to provide talking widgets.

12.205.1 Emacspeak-Widget Commands

12.205.1.1 emacspeak-widget-browse-widget-interactively

`emacspeak-widget-browse-widget-interactively` [Command]

Allows you to browse a widget

12.205.1.2 emacspeak-widget-help

`emacspeak-widget-help` [Command]

Speak help for widget under point.

12.205.1.3 emacspeak-widget-summarize-parent

`emacspeak-widget-summarize-parent` [Command]

Summarize parent of widget at point.

12.205.1.4 emacspeak-widget-summarize-widget-under-point

`emacspeak-widget-summarize-widget-under-point` (**&optional** *level*) [Command]

Summarize a widget if any under point.
Optional interactive prefix specifies how many levels to go up from current widget before summarizing.

(fn &optional LEVEL)

12.205.1.5 emacspeak-widget-update-from-minibuffer

`emacspeak-widget-update-from-minibuffer` (*pos*) [Command]

Sets widget at 'pos' by invoking its prompter.

(fn POS)

12.205.2 emacspeak-widget Options

`User Option` *voice-widget-button* [Variable]

Personality used for widget-button This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

`User Option` *voice-widget-button-pressed* [Variable]

Personality used for widget-button-pressed This personality uses voice-bolden-extra whose effect can be changed globally by customizing voice-bolden-extra-settings.

`User Option` *voice-widget-documentation* [Variable]

Personality used for widget-documentation This personality uses voice-smoothen-extra whose effect can be changed globally by customizing voice-smoothen-extra-settings.

`User Option` *voice-widget-field* [Variable]

Personality used for widget-field This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

`User Option` *voice-widget-inactive* [Variable]

Personality used for widget-inactive This personality uses voice-lighten whose effect can be changed globally by customizing voice-lighten-settings.

`User Option` *voice-widget-single-line-field* [Variable]

Personality used for widget-single-line-field This personality uses voice-smoothen whose effect can be changed globally by customizing voice-smoothen-settings.

12.206 emacspeak-windmove

Package windmove (bundled with Emacs 21) provides commands for navigating to windows based on relative position.

12.207 emacspeak-winring

window configurations in emacs are very useful you can display the same file in different windows, and have different portions of the file displayed. winring allows you to manage window configurations, and this module speech-enables it.

12.208 emacspeak-wizards

Contains various wizards for the Emacspeak desktop.

12.208.1 Emacspeak-Wizards Commands

12.208.1.1 emacspeak-annotate-add-annotation

`emacspeak-annotate-add-annotation` (**&optional** *reset*) [Command]

Add annotation to the annotation working buffer.
Prompt for annotation buffer if not already set.
Interactive prefix arg 'reset' prompts for the annotation
buffer even if one is already set.
Annotation is entered in a temporary buffer and the
annotation is inserted into the working buffer when complete.

(fn &optional RESET)

12.208.1.2 emacspeak-clipboard-copy

`emacspeak-clipboard-copy` (*start end* **&optional** *prompt*) [Command]

C-e C-c

<fn> C-c

Copy contents of the region to the emacspeak clipboard. Previous
contents of the clipboard will be overwritten. The Emacspeak clipboard
is a convenient way of sharing information between independent
Emacspeak sessions running on the same or different machines. Do not
use this for sharing information within an Emacs session –Emacs'
register commands are far more efficient and light-weight. Optional
interactive prefix arg results in Emacspeak prompting for the
clipboard file to use. Argument START and END specifies
region. Optional argument PROMPT specifies whether we prompt for the
name of a clipboard file.

(fn START END &optional PROMPT)

12.208.1.3 emacspeak-clipboard-paste

`emacspeak-clipboard-paste` (**&optional** *paste-table*) [Command]

C-e C-y

<fn> C-y

Yank contents of the Emacspeak clipboard at point.
The Emacspeak clipboard is a convenient way of sharing information between
independent Emacspeak sessions running on the same or different
machines. Do not use this for sharing information within an Emacs
session –Emacs' register commands are far more efficient and
light-weight. Optional interactive prefix arg pastes from
the emacspeak table clipboard instead.

(fn &optional PASTE-TABLE)

12.208.1.4 emacspeak-copy-current-file

`emacspeak-copy-current-file` [Command]

`C-e M-c`

`<fn> M-c`

Copy file visited in current buffer to new location.
Prompts for the new location and preserves modification time
 when copying. If location is a directory, the file is copied
 to that directory under its current name ; if location names
 a file in an existing directory, the specified name is
 used. Asks for confirmation if the copy will result in an
 existing file being overwritten.

12.208.1.5 emacspeak-curl

`emacspeak-curl` (*url*) [Command]

Grab URL using Curl, and preview it with a browser .

(fn URL)

12.208.1.6 emacspeak-customize

`emacspeak-customize` [Command]

`C-e C`

`<fn> C`

Customize Emacspeak.

12.208.1.7 emacspeak-describe-emacspeak

`emacspeak-describe-emacspeak` [Command]

`C-h C-e`

`<f1> C-e`

`<help> C-e`
Give a brief overview of emacspeak.

12.208.1.8 emacspeak-emergency-tts-restart

`emacspeak-emergency-tts-restart` [Command]

C-e <insert>
<fn> <insert>
C-e x s

<fn> x s

For use in an emergency.
Will start TTS engine specified by
emacspeak-emergency-tts-server.

12.208.1.9 emacspeak-frame-label-or-switch-to-labelled-frame

`emacspeak-frame-label-or-switch-to-labelled-frame` [Command]
(**&optional** *prefix*)

C-e M-f

<fn> M-f

Switch to labelled frame.
With optional PREFIX argument, label current frame.

(fn &optional PREFIX)

12.208.1.10 emacspeak-kill-buffer-quietly

`emacspeak-kill-buffer-quietly` [Command]

C-e C-t Q

<fn> C-t Q
Kill current buffer without asking for confirmation.

12.208.1.11 emacspeak-learn-emacs-mode

`emacspeak-learn-emacs-mode` [Command]

C-e <f1>

C-h C-l

<fn> <f1>

<f1> C-l

<help> C-l
Helps you learn the keys. You can press keys and hear what they do.
To leave, press C-g.

12.208.1.12 emacspeak-link-current-file

`emacspeak-link-current-file` [Command]

C-e M-l

<fn> M-l

Link (hard link) file visited in current buffer to new location.
Prompts for the new location and preserves modification time
  when linking. If location is a directory, the file is copied
  to that directory under its current name ; if location names
  a file in an existing directory, the specified name is
  used. Signals an error if target already exists.

12.208.1.13 emacspeak-next-frame-or-buffer

`emacspeak-next-frame-or-buffer` (**&optional** *frame*) [Command]

<C-right>
Move to next buffer.
With optional interactive prefix arg 'frame', move to next frame instead.

(fn &optional FRAME)

12.208.1.14 emacspeak-previous-frame-or-buffer

`emacspeak-previous-frame-or-buffer` (**&optional** *frame*) [Command]

<C-left>

Move to previous buffer.
With optional interactive prefix arg 'frame', move to previous frame instead.

(fn &optional FRAME)

12.208.1.15 emacspeak-select-this-buffer-next-display

`emacspeak-select-this-buffer-next-display` [Command]

C-e <C-right>
<fn> <C-right>
Select this buffer as displayed in a 'next' frame.
See documentation for command
'emacspeak-select-this-buffer-other-window-display' for the
meaning of 'next'.

12.208.1.16 emacspeak-select-this-buffer-other-window-display

`emacspeak-select-this-buffer-other-window-display` [Command]
(**&optional** *arg*)

Switch to this buffer as displayed in a different frame. Emacs
allows you to display the same buffer in multiple windows or
frames. These different windows can display different
portions of the buffer. This is equivalent to leaving a
book open at multiple places at once. This command allows you to
move to the places where you have left the book open. The
number used to invoke this command specifies which of the

displays you wish to select. Typically you will have two or
at most three such displays open. The current display is 0,
the next is 1, and so on. Optional argument ARG specifies
the display to select.

(fn &optional ARG)

12.208.1.17 emacspeak-select-this-buffer-previous-display

`emacspeak-select-this-buffer-previous-display` [Command]

C-e <C-left>
<fn> <C-left>
Select this buffer as displayed in a 'previous' window.
See documentation for command
'emacspeak-select-this-buffer-other-window-display' for the
meaning of 'previous'.

12.208.1.18 emacspeak-show-property-at-point

`emacspeak-show-property-at-point` (**&optional** *property*) [Command]

C-e M-p

<fn> M-p

Show value of PROPERTY at point.
If optional arg property is not supplied, read it interactively.
Provides completion based on properties at point.
If no property is set, show a message and exit.

(fn &optional PROPERTY)

12.208.1.19 emacspeak-show-style-at-point

`emacspeak-show-style-at-point` [Command]

C-e M-v

<fn> M-v

Show value of property personality (and possibly face) at point.

12.208.1.20 emacspeak-skip-blank-lines-backward

`emacspeak-skip-blank-lines-backward` [Command]

<S-up>

Move backward across blank lines.
The line under point is then spoken.
Signals beginning of buffer.

12.208.1.21 emacspeak-skip-blank-lines-forward

`emacspeak-skip-blank-lines-forward` [Command]

<S-down>

Move forward across blank lines.
The line under point is then spoken.
Signals end of buffer.

12.208.1.22 emacspeak-speak-hostname

`emacspeak-speak-hostname` [Command]

C-e M-h

<fn> M-h

Speak host name.

12.208.1.23 emacspeak-speak-popup-messages

`emacspeak-speak-popup-messages` [Command]

C-h M

<f1> M

<help> M

Pop up messages buffer.
If it is already selected then hide it and try to restore
previous window configuration.

12.208.1.24 emacspeak-speak-show-active-network-interfaces

`emacspeak-speak-show-active-network-interfaces` (**&optional** *address*) [Command]

C-e I

<fn> I

Shows all active network interfaces in the echo area.
With interactive prefix argument ADDRESS it prompts for a
specific interface and shows its address. The address is
also copied to the kill ring for convenient yanking.

(fn &optional ADDRESS)

12.208.1.25 emacspeak-speak-telephone-directory

`emacspeak-speak-telephone-directory` (**&optional** *edit*) [Command]

C-e x t

<fn> x t

Lookup and display a phone number.
With prefix arg, opens the phone book for editing.

(fn &optional EDIT)

12.208.1.26 emacspeak-speak-this-buffer-next-display

`emacspeak-speak-this-buffer-next-display` [Command]

C-e <right>
<fn> <right>
Speak this buffer as displayed in a 'previous' window.
See documentation for command
'emacspeak-speak-this-buffer-other-window-display' for the
meaning of 'next'.

12.208.1.27 emacspeak-speak-this-buffer-other-window-display

`emacspeak-speak-this-buffer-other-window-display` (**&optional** *arg*) [Command]

C-e /

<fn> /

Speak this buffer as displayed in a different frame. Emacs allows you to display the same buffer in multiple windows or frames. These different windows can display different portions of the buffer. This is equivalent to leaving a book open at places at once. This command allows you to listen to the places where you have left the book open. The number used to invoke this command specifies which of the displays you wish to speak. Typically you will have two or at most three such displays open. The current display is 0, the next is 1, and so on. Optional argument ARG specifies the display to speak.

(fn &optional ARG)

12.208.1.28 emacspeak-speak-this-buffer-previous-display

`emacspeak-speak-this-buffer-previous-display` [Command]

C-e <left>
<fn> <left>
Speak this buffer as displayed in a 'previous' window.
See documentation for command
'emacspeak-speak-this-buffer-other-window-display' for the
meaning of 'previous'.

12.208.1.29 emacspeak-ssh-tts-restart

`emacspeak-ssh-tts-restart` [Command]

C-e <delete>
C-e <(deletechar>
<fn> <delete>
<fn> <(deletechar>
Restart specified ssh tts server.

12.208.1.30 emacspeak-submit-bug

`emacspeak-submit-bug` [Command]

C-e C-M-b

<fn> C-M-b
Function to submit a bug to the programs maintainer.

12.208.1.31 emacspeak-symlink-current-file

`emacspeak-symlink-current-file` [Command]

C-e M-s

<fn> M-s

Link (symbolic link) file visited in current buffer to new location.
Prompts for the new location and preserves modification time
when linking. If location is a directory, the file is copied
to that directory under its current name ; if location names
a file in an existing directory, the specified name is
used. Signals an error if target already exists.

12.208.1.32 emacspeak-view-emacspeak-news

`emacspeak-view-emacspeak-news` [Command]

C-e N

<fn> N

Display emacspeak News for a given version.

12.208.1.33 emacspeak-view-emacspeak-tips

`emacspeak-view-emacspeak-tips` [Command]

C-e T

<fn> T

Browse Emacspeak productivity tips.

12.208.1.34 emacspeak-wizards-add-autoload-cookies

`emacspeak-wizards-add-autoload-cookies` (**&optional** *f*) [Command]

Add autoload cookies to file f.
Default is to add autoload cookies to current file.

(fn &optional F)

12.208.1.35 emacspeak-wizards-alpha-vantage-quotes

`emacspeak-wizards-alpha-vantage-quotes` (*ticker* **&optional** *custom*) [Command]

`C-e x a`

`<fn> x a`

Retrieve stock quote data from Alpha Vantage. Prompts for 'ticker' — a stock symbol. Optional interactive prefix arg 'custom' provides access to the various functions provided by alpha-vantage.

(fn TICKER &optional CUSTOM)

12.208.1.36 emacspeak-wizards-bindings-from-org

`emacspeak-wizards-bindings-from-org` (*variable filename*) [Command]

Load bindings from a specified file.

(fn VARIABLE FILENAME)

12.208.1.37 emacspeak-wizards-bindings-to-org

`emacspeak-wizards-bindings-to-org` (*variable filename*) [Command]

Persists mapping to org file.

(fn VARIABLE FILENAME)

12.208.1.38 emacspeak-wizards-braille

`emacspeak-wizards-braille` (*s*) [Command]

Insert Braille string at point.

(fn S)

12.208.1.39 emacspeak-wizards-cleanup-shell-path

`emacspeak-wizards-cleanup-shell-path` [Command]

Cleans up duplicates in shell path env variable.

12.208.1.40 emacspeak-wizards-color-at-point

`emacspeak-wizards-color-at-point` [Command]

C-h ,

<f1> ,

<help> ,

Echo foreground/background color at point.

12.208.1.41 emacspeak-wizards-color-diff-at-point

`emacspeak-wizards-color-diff-at-point` (**&optional** *set*) [Command]

C-h \

<f1> \

<help> \

Meaningfully speak difference between background and foreground color at point.
With interactive prefix arg, set foreground and background color first.

(fn &optional SET)

12.208.1.42 emacspeak-wizards-color-wheel

`emacspeak-wizards-color-wheel` (*start*) [Command]

C-e x c

<fn> x c

Interactively manipulate a simple color wheel and display the name
and shade of the resulting color. This makes for a fun color
exploration tool with verbal descriptions of the colors from package
name-this-color. Prompts for a color from which to start exploration.

Keyboard Commands During Interaction:
Up/Down: Increase/Decrement along current axis using specified step-size.
=: Set value on current axis to number read from minibuffer.
Left/Right: Switch color axis along which to move.
b/f: Quit wheel after setting background/foreground color to current value.
n: Read color name from minibuffer.
c: Complement current color.
s: Set stepsize to number read from minibuffer.
q: Quit color wheel, after copying current hex value to kill-ring.

(fn START)

12.208.1.43 emacspeak-wizards-colors

`emacspeak-wizards-colors` [Command]

C-e x C

<fn> x C

Display list of colors and setup a callback to activate color under point as either the foreground or background color.

12.208.1.44 emacspeak-wizards-count-slides-in-region

`emacspeak-wizards-count-slides-in-region` (*start end*) [Command]

Count slides starting from point.

(fn START END)

12.208.1.45 emacspeak-wizards-customize-saved

`emacspeak-wizards-customize-saved` (*pattern*) [Command]

C-h C-s

<f1> C-s

<help> C-s

Customize saved options matching 'pattern'. This command enables updating custom settings for a specific package or group of packages.

(fn PATTERN)

12.208.1.46 emacspeak-wizards-cycle-to-next-buffer

`emacspeak-wizards-cycle-to-next-buffer` [Command]

C-z n

C-, n

C-x @ a n

Cycles to next buffer having same mode.

12.208.1.47 emacspeak-wizards-cycle-to-previous-buffer

`emacspeak-wizards-cycle-to-previous-buffer` [Command]

C-z p

C-, p

C-x @ a p

Cycles to previous buffer having same mode.

12.208.1.48 emacspeak-wizards-display-pod-as-manpage

`emacspeak-wizards-display-pod-as-manpage` (*filename*) [Command]

Create a virtual manpage in Emacs from the Perl Online Documentation.

(fn FILENAME)

12.208.1.49 emacspeak-wizards-enumerate-matching-commands

`emacspeak-wizards-enumerate-matching-commands` (*pattern*) [Command]

Return list of commands whose names match pattern.

(fn PATTERN)

12.208.1.50 emacspeak-wizards-enumerate-matching-faces

`emacspeak-wizards-enumerate-matching-faces` (*pattern*) [Command]

Enumerate faces matching pattern.

(fn PATTERN)

12.208.1.51 emacspeak-wizards-enumerate-obsolete-faces

`emacspeak-wizards-enumerate-obsolete-faces` [Command]

utility function to enumerate old, obsolete maps that we have still
mapped to voices.

12.208.1.52 emacspeak-wizards-enumerate-uncovered-commands

`emacspeak-wizards-enumerate-uncovered-commands` (*pattern* **&optional** *bound-only*) [Command]

Enumerate unadvised commands matching pattern.
Optional interactive prefix arg ‘bound-only’
filters out commands that dont have an active key-binding.

(fn PATTERN &optional BOUND-ONLY)

12.208.1.53 emacspeak-wizards-enumerate-unmapped-faces

`emacspeak-wizards-enumerate-unmapped-faces` (**&optional** *pattern*) [Command]

Enumerate unmapped faces matching pattern.

(fn &optional PATTERN)

12.208.1.54 emacspeak-wizards-espeak-line

`emacspeak-wizards-espeak-line` [Command]

Speak line using espeak polyglot wizard.

12.208.1.55 emacspeak-wizards-espeak-region

`emacspeak-wizards-espeak-region` (*start end*) [Command]

Speak region using ESpeak polyglot wizard.

(fn START END)

12.208.1.56 emacspeak-wizards-espeak-string

`emacspeak-wizards-espeak-string` (*string*) [Command]

Speak string in lang via ESpeak.
Lang is obtained from property 'lang' on string, or via an interactive prompt.

(fn STRING)

12.208.1.57 emacspeak-wizards-eww-buffer-list

`emacspeak-wizards-eww-buffer-list` [Command]

C-z e

Display list of open EWW buffers.

12.208.1.58 emacspeak-wizards-exec-path-from-shell

`emacspeak-wizards-exec-path-from-shell` [Command]

Update exec-path from shell path.

12.208.1.59 emacspeak-wizards-execute-asynchronously

`emacspeak-wizards-execute-asynchronously` (*key*) [Command]

C-. a

C-' a

C-x @ s a

Read key-sequence, then execute its command on a new thread.

(fn KEY)

12.208.1.60 emacspeak-wizards-execute-emacspeak-command

`emacspeak-wizards-execute-emacspeak-command` (*command*) [Command]

C-e M-x

<fn> M-x

Prompt for and execute an Emacspeak command.

(fn COMMAND)

12.208.1.61 emacspeak-wizards-find-file-as-root

`emacspeak-wizards-find-file-as-root` (*file*) [Command]

C-, f

C-x @ a f

Like 'ido-find-file, but automatically edit the file with
root-privileges (using tramp/sudo), if the file is not writable by
user.

(fn FILE)

12.208.1.62 emacspeak-wizards-find-grep

`emacspeak-wizards-find-grep` (*glob pattern*) [Command]

Run compile using find and grep.
Interactive arguments specify filename pattern and search pattern.

(fn GLOB PATTERN)

12.208.1.63 emacspeak-wizards-find-longest-line-in-region

`emacspeak-wizards-find-longest-line-in-region` (*start end*) [Command]

`C-e x =`

`<fn> x =`

Find longest line in region.
Moves to the longest line when called interactively.

(fn START END)

12.208.1.64 emacspeak-wizards-find-longest-paragraph-in-region

`emacspeak-wizards-find-longest-paragraph-in-region` (*start end*) [Command]

Find longest paragraph in region.
Moves to the longest paragraph when called interactively.

(fn START END)

12.208.1.65 emacspeak-wizards-find-shortest-line-in-region

`emacspeak-wizards-find-shortest-line-in-region` (*start end*) [Command]

Find shortest line in region.
Moves to the shortest line when called interactively.

(fn START END)

12.208.1.66 emacspeak-wizards-finder-find

`emacspeak-wizards-finder-find` (*directory*) [Command]

Run find-dired on specified switches after prompting for the
directory to where find is to be launched.

(fn DIRECTORY)

12.208.1.67 emacspeak-wizards-finder-mode

`emacspeak-wizards-finder-mode` [Command]

Emacspeak Finder

In addition to any hooks its parent mode ‘special-mode’ might have run,
this mode runs the hook ‘emacspeak-wizards-finder-mode-hook’, as the final or penultimate step
during initialization.

key	binding
—	——

12.208.1.68 emacspeak-wizards-fix-read-only-text

`emacspeak-wizards-fix-read-only-text` (*start end*) [Command]

Nuke read-only property on text range.

(fn START END)

12.208.1.69 emacspeak-wizards-fix-typo

`emacspeak-wizards-fix-typo` (*ext word correction*) [Command]

Search and replace recursively in all files with extension ‘ext’
for ‘word’ and replace it with correction.
Use with caution.

(fn EXT WORD CORRECTION)

12.208.1.70 emacspeak-wizards-frame-colors

`emacspeak-wizards-frame-colors` [Command]

Display frame’s foreground/background color seetting.

12.208.1.71 emacspeak-wizards-free-geo-ip

`emacspeak-wizards-free-geo-ip` (**&optional** *reverse-geocode*) [Command]

Return list consisting of city and region_name.
Optional interactive prefix arg reverse-geocodes using Google Maps.

(fn &optional REVERSE-GEOCODE)

12.208.1.72 emacspeak-wizards-gen-fn-decl

`emacspeak-wizards-gen-fn-decl` (*f* **&optional** *ext*) [Command]

Generate declare-function call for function ‘f’.
Optional interactive prefix arg ext says this comes from an
external package.

(fn F &optional EXT)

12.208.1.73 emacspeak-wizards-generate-finder

`emacspeak-wizards-generate-finder` [Command]

C-z f

Generate a widget-enabled finder wizard.

12.208.1.74 emacspeak-wizards-generate-voice-sampler

`emacspeak-wizards-generate-voice-sampler` (*step*) [Command]

Generate a buffer that shows a sample line in all the ACSS settings
for the current voice family.

(fn STEP)

12.208.1.75 emacspeak-wizards-google-headlines

`emacspeak-wizards-google-headlines` [Command]

C-. C-n

C-' C-n

C-x @ s C-n

Display just the headlines from Google News for fast loading.

12.208.1.76 emacspeak-wizards-google-news

`emacspeak-wizards-google-news` [Command]

C-. n

C-' n

C-x @ s n

Clean up news.google.com for skimming the news.

12.208.1.77 emacspeak-wizards-how-many-matches

`emacspeak-wizards-how-many-matches` (*start end* **&optional** *prefix*) [Command]

C-e x h

<fn> x h

If you define a file local variable
called ‘emacspeak-occur-pattern’ that holds a regular expression
that matches lines of interest, you can use this command to conveniently
run ‘how-many’ to count matching header lines.
With interactive prefix arg, prompts for and remembers the file local pattern.

(fn START END &optional PREFIX)

12.208.1.78 emacspeak-wizards-iex-show-financials

`emacspeak-wizards-iex-show-financials` (*symbol* **&optional** *refresh*) [Command]

Show financials for specified ticker.
Checks cache, then makes API call if needed.
Optional interactive prefix arg refreshes cache.

(fn SYMBOL &optional REFRESH)

12.208.1.79 emacspeak-wizards-iex-show-metadata

`emacspeak-wizards-iex-show-metadata` [Command]

Account metadata.

12.208.1.80 emacspeak-wizards-iex-show-news

`emacspeak-wizards-iex-show-news` (*symbol* **&optional** *refresh*) [Command]

Show news for specified ticker.
Checks cache, then makes API call if needed.
Optional interactive prefix arg refreshes cache.

(fn SYMBOL &optional REFRESH)

12.208.1.81 emacspeak-wizards-iex-show-price

`emacspeak-wizards-iex-show-price` (*symbol*) [Command]

C-, q

C-x @ a q

Quick Quote: Just stock price from IEXCloud.

(fn SYMBOL)

12.208.1.82 emacspeak-wizards-iex-show-quote

`emacspeak-wizards-iex-show-quote` (**&optional** *refresh*) [Command]

C-. q

C-' q

C-x @ s q

Show portfolio data from cache.
Optional interactive prefix arg forces cache refresh.

The quotes view uses emacspeak's table mode.
In addition, the following keys are available :

F: Show financials for current stock.
N: Show news for current stock.

P: Show live price for current stock.

(fn &optional REFRESH)

12.208.1.83 emacspeak-wizards-iex-show-tops

`emacspeak-wizards-iex-show-tops` [Command]

Uses tops/last end-point to show brief portfolio quotes.

12.208.1.84 emacspeak-wizards-iex-this-financials

`emacspeak-wizards-iex-this-financials` [Command]

Show financials for symbol in current row

12.208.1.85 emacspeak-wizards-iex-this-news

`emacspeak-wizards-iex-this-news` [Command]

Show news for symbol in current row

12.208.1.86 emacspeak-wizards-iex-this-price

`emacspeak-wizards-iex-this-price` [Command]

Show price for symbol in current row

12.208.1.87 emacspeak-wizards-list-voices

`emacspeak-wizards-list-voices` (*pattern*) [Command]

`C-h` "

`<f1>` "

`<help>` "

Show all defined voice-face mappings in a help buffer.
Sample text to use comes from variable
  'ems–wizards-sampler-text

(fn PATTERN)

12.208.1.88 emacspeak-wizards-midi-using-m-score

`emacspeak-wizards-midi-using-m-score` (*midi-file*) [Command]

Play midi file using mscore from musescore package.

(fn MIDI-FILE)

12.208.1.89 emacspeak-wizards-mlb-standings

`emacspeak-wizards-mlb-standings` (**&optional** *raw*) [Command]

Display MLB standings as of today.
Optional interactive prefix arg shows unprocessed results.

(fn &optional RAW)

12.208.1.90 emacspeak-wizards-move-and-speak

`emacspeak-wizards-move-and-speak` (*command count*) [Command]

Speaks a chunk of text bounded by point and a target position.
Target position is specified using a navigation command and a
count that specifies how many times to execute that command
first. Point is left at the target position. Interactively,
command is specified by pressing the key that invokes the
command.

(fn COMMAND COUNT)

12.208.1.91 emacspeak-wizards-nba-standings

`emacspeak-wizards-nba-standings` (**&optional** *raw*) [Command]

Display NBA standings as of today.
Optional interactive prefix arg shows unprocessed results.

(fn &optional RAW)

12.208.1.92 emacspeak-wizards-next-bullet

`emacspeak-wizards-next-bullet` [Command]

Navigate to and speak next ‘bullet’.

12.208.1.93 emacspeak-wizards-next-interactive-defun

`emacspeak-wizards-next-interactive-defun` [Command]

Move point to the next interactive defun

12.208.1.94 emacspeak-wizards-noaa-weather

`emacspeak-wizards-noaa-weather` (**&optional** *ask*) [Command]

`C-; w`

`C-e x w`

`<fn> x w`

`C-x @ h w`

Display weather information using NOAA Weather API.
Data is retrieved only once, subsequent calls switch to previously
displayed results. Kill that buffer or use an interactive prefix
arg (C-u) to get new data. Optional second interactive prefix
arg (C-u C-u) asks for location address; Default is to display
weather for ‘gmaps-my-address’.

(fn &optional ASK)

12.208.1.95 emacspeak-wizards-occur-header-lines

`emacspeak-wizards-occur-header-lines` (**&optional** *prefix*) [Command]

C-e x o

<fn> x o

If you define a file local variable called
'emacspeak-occur-pattern' that holds a regular expression that
matches header lines, you can use this command to conveniently
run 'occur' to find matching header lines. With prefix arg,
prompts for and sets value of the file local pattern.

(fn &optional PREFIX)

12.208.1.96 emacspeak-wizards-pdf-open

`emacspeak-wizards-pdf-open` (*filename* **&optional** *ask-pwd*) [Command]

C-; p

C-x @ h p

Open pdf file as text.
Optional interactive prefix arg ask-pwd prompts for password.

(fn FILENAME &optional ASK-PWD)

12.208.1.97 emacspeak-wizards-previous-bullet

`emacspeak-wizards-previous-bullet` [Command]

Navigate to and speak previous 'bullet'.

12.208.1.98 emacspeak-wizards-quote

`emacspeak-wizards-quote` (**&optional** *refresh*) [Command]

C-e x q

<fn> x q

Top-level dispatch for looking up Stock Market information.

Key : Action
f : Financials
m : Account metadata
n : News
p : Price
q : Quotes
t : tops/last

(fn &optional REFRESH)

12.208.1.99 emacspeak-wizards-scratch

`emacspeak-wizards-scratch` [Command]

C-. SPC

C-' SPC

C-x @ s SPC
Switch to *scratch* buffer, creating it if necessary.

12.208.1.100 emacspeak-wizards-set-colors

`emacspeak-wizards-set-colors` [Command]
Interactively prompt for foreground and background colors.

12.208.1.101 emacspeak-wizards-shell

`emacspeak-wizards-shell` (**&optional** *prefix*) [Command]

C-; s

C-x @ h s

Run Emacs built-in 'shell' command when not in a shell buffer, or
when called with a prefix argument. When called from a shell buffer,
switches to 'next' shell buffer. When called from outside a shell
buffer, find the most 'appropriate shell' and switch to it. Once
switched, set default directory in that target shell to the directory
of the source buffer.

(fn &optional PREFIX)

12.208.1.102 emacspeak-wizards-shell-by-key

`emacspeak-wizards-shell-by-key` (**&optional** *prefix*) [Command]

C-e x 9

C-e x 8

C-e x 7

C-e x 6

C-e x 5

C-e x 4

C-e x 3

C-e x 2

C-e x 1

C-e x 0

<fn> x 9

`<fn> x 8`

`<fn> x 7`

`<fn> x 6`

`<fn> x 5`

`<fn> x 4`

`<fn> x 3`

`<fn> x 2`

`<fn> x 1`

`<fn> x 0`

Switch to shell buffer by key. This provides a predictable means for switching to a specific shell buffer. When invoked from a non-shell-mode buffer that is a dired-buffer or is visiting a file, invokes ‘cd ’ in the shell to change to the value of ‘default-directory’ — if called with a prefix-arg. When already in a shell buffer, interactive prefix arg ‘prefix’ causes this shell to be re-keyed if appropriate — see M-x emacspeak-wizards-shell-re-key for an explanation of how re-keying works.

(fn &optional PREFIX)

12.208.1.103 emacspeak-wizards-shell-command-on-current-file

`emacspeak-wizards-shell-command-on-current-file` (*command*) [Command]

`C-e &`

`<fn> &`

Prompts for and runs shell command on current file.

(fn COMMAND)

12.208.1.104 emacspeak-wizards-shell-directory-reset

`emacspeak-wizards-shell-directory-reset` [Command]

`C-. .`

`C-' .`

`C-e x .`

`<fn> x .`

`C-x @ s .`

Set current directory to this shell’s initial directory if one was defined.

12.208.1.105 emacspeak-wizards-shell-directory-set

`emacspeak-wizards-shell-directory-set` [Command]

C-e x ,

<fn> x ,

Define current directory as this shell's project directory.

12.208.1.106 emacspeak-wizards-shell-toggle

`emacspeak-wizards-shell-toggle` [Command]

C-e <f11>

C-; j

<fn> <f11>
C-x @ h j

Switch to the shell buffer and cd to
the directory of the previously current buffer.

12.208.1.107 emacspeak-wizards-show-defined-voices

`emacspeak-wizards-show-defined-voices` [Command]

Display a buffer with sample text in the defined voices.

12.208.1.108 emacspeak-wizards-show-eval-result

`emacspeak-wizards-show-eval-result` (*form*) [Command]

M-ESC :

Convenience command to pretty-print and view Lisp evaluation results.

(fn FORM)

12.208.1.109 emacspeak-wizards-show-face

`emacspeak-wizards-show-face` (*face*) [Command]

Show salient properties of specified face.

(fn FACE)

12.208.1.110 emacspeak-wizards-show-memory-used

`emacspeak-wizards-show-memory-used` [Command]

Convenience command to view state of memory used in this session so far.

12.208.1.111 emacspeak-wizards-speak-iso-datetime

`emacspeak-wizards-speak-iso-datetime` (*iso*) [Command]

Make ISO date-time speech friendly.

(fn ISO)

12.208.1.112 emacspeak-wizards-spot-words

`emacspeak-wizards-spot-words` (*ext word*) [Command]

Searches recursively in all files with extension 'ext'
for 'word' and displays hits in a compilation buffer.

(fn EXT WORD)

12.208.1.113 emacspeak-wizards-squeeze-blanks

`emacspeak-wizards-squeeze-blanks` (*start end*) [Command]

C-e x |

<fn> x |

Squeeze multiple blank lines in current buffer.

(fn START END)

12.208.1.114 emacspeak-wizards-swap-fg-and-bg

`emacspeak-wizards-swap-fg-and-bg` [Command]

C-h =

<f1> =

<help> =

Swap foreground and background.

12.208.1.115 emacspeak-wizards-term

`emacspeak-wizards-term` (*create*) [Command]

C-; a

C-x @ h a

Switch to an ansi-term buffer or create one.
With prefix arg, always creates a new terminal.
Otherwise cycles through existing terminals, creating the first
term if needed.

(fn CREATE)

12.208.1.116 emacspeak-wizards-terminal

`emacspeak-wizards-terminal` (*program*) [Command]

Launch terminal and rename buffer appropriately.

(fn PROGRAM)

12.208.1.117 emacspeak-wizards-toggle-mm-dd-yyyy-date-pronouncer

`emacspeak-wizards-toggle-mm-dd-yyyy-date-pronouncer` [Command]

Toggle pronunciation of mm-dd-yyyy dates.

12.208.1.118 emacspeak-wizards-toggle-yyyymmdd-date-pronouncer

`emacspeak-wizards-toggle-yyyymmdd-date-pronouncer` [Command]

Toggle pronunciation of yyyymmdd dates.

12.208.1.119 emacspeak-wizards-tramp-open-location

`emacspeak-wizards-tramp-open-location` (*name*) [Command]

Open specified tramp location.
Location is specified by name.

(fn NAME)

12.208.1.120 emacspeak-wizards-tune-in-radio-browse

`emacspeak-wizards-tune-in-radio-browse` (**&optional** *category*) [Command]

C-, t

C-x @ a t

Browse Tune-In Radio.
Optional interactive prefix arg ‘category’ prompts for a category.

(fn &optional CATEGORY)

12.208.1.121 emacspeak-wizards-tune-in-radio-search

`emacspeak-wizards-tune-in-radio-search` [Command]

C-, s

C-x @ a s

Search Tune-In Radio.

12.208.1.122 emacspeak-wizards-units

`emacspeak-wizards-units` [Command]

C-e x u

<fn> x u

Run units in a comint sub-process.

12.208.1.123 emacspeak-wizards-vc-n

`emacspeak-wizards-vc-n` [Command]

Accelerator for VC viewer.

12.208.1.124 emacspeak-wizards-vc-view-mode

`emacspeak-wizards-vc-view-mode` [Command]

Major mode for interactively viewing virtual console contents.

key binding
— ——-

C-l emacspeak-wizards-vc-viewer-refresh

In addition to any hooks its parent mode 'special-mode' might have run,
this mode runs the hook 'emacspeak-wizards-vc-view-mode-hook', as the final or penultimate step
during initialization.

12.208.1.125 emacspeak-wizards-vc-viewer

`emacspeak-wizards-vc-viewer` (*console*) [Command]

C-e x v

<fn> x v

View contents of specified virtual console.

(fn CONSOLE)

12.208.1.126 emacspeak-wizards-vc-viewer-refresh

`emacspeak-wizards-vc-viewer-refresh` [Command]

Refresh view of VC we're viewing.

12.208.1.127 emacspeak-wizards-view-buffers-filtered-by-m-player-mode

`emacspeak-wizards-view-buffers-filtered-by-m-player-mode` [Command]

C-; :

C-x @ h :

Buffer menu filtered by m-player mode.

12.208.1.128 emacspeak-wizards-view-buffers-filtered-by-this-mode

`emacspeak-wizards-view-buffers-filtered-by-this-mode` [Command]

C-z b

C-. m

C-' m

C-, c

C-x @ a c

C-x @ s m

Buffer menu filtered by mode of current-buffer.

12.208.1.129 emacspeak-wizards-voice-sampler

`emacspeak-wizards-voice-sampler` (*personality*) [Command]

Read a personality and apply it to the current line.

(fn PERSONALITY)

12.208.1.130 emacspeak-wizards-ytel-download

`emacspeak-wizards-ytel-download` (*id*) [Command]

Download video at point.

(fn ID)

12.208.1.131 emacspeak-wizards-ytel-play-at-point

`emacspeak-wizards-ytel-play-at-point` (*id* **&optional** *best*) [Command]

Play video. Argument 'id' is the video-id.
Play current video in ytel when called interactively.
Optional interactive prefix arg 'best' picks best audio format.

(fn ID &optional BEST)

12.208.2 emacspeak-wizards Options

`User Option` *emacspeak-clipboard-file* [Variable]

File used to save Emacspeak clipboard. The emacspeak clipboard provides a convenient mechanism for exchanging information between different Emacs sessions.

`User Option` *emacspeak-curl-cookie-store* [Variable]

Cookie store used by Curl.

`User Option` *emacspeak-emergency-tts-server* [Variable]

TTS server to use in an emergency. Set this to a TTS server that is known to work at all times. If you are debugging another speech server and that server gets wedged for some reason, you can use command emacspeak-emergency-tts-restart to get speech back using the reliable TTS server. It's useful to bind the above command to a convenient key.

`User Option` *emacspeak-iex-api-key* [Variable]

Web API key for IEX Finance access. See IEX Login Console at https://iexcloud.io/cloud-login/ for how to get an API key.

`User Option` *emacspeak-speak-telephone-directory* [Variable]

File holding telephone directory. This is just a text file, and we use grep to search it.

`User Option` *emacspeak-ssh-tts-server* [Variable]

SSH TTS server to use by default.

User Option *emacspeak-wizards-alpha-vantage-api-key* [Variable]
API Key used to retrieve stock data from alpha-vantage. Visit https://www.alphavantage.co/support/#api-key to get your key.

User Option *emacspeak-wizards-find-switches-widget* [Variable]
Widget to get find switch.

User Option *emacspeak-wizards-pdf-to-text-options* [Variable]
options to Command for running pdftotext.

User Option *emacspeak-wizards-personal-portfolio* [Variable]
Set this to the stock tickers you want to check. Default is GAFA. Tickers are separated by white-space and are automatically sorted in lexical order with duplicates removed when saving.

User Option *emacspeak-wizards-project-shells* [Variable]
List of shell-name/initial-directory pairs.

User Option *emacspeak-wizards-spot-words-extension* [Variable]
Default file extension used when spotting words.

User Option *emacspeak-wizards-tramp-locations* [Variable]
Tramp locations used by Emacspeak tramp wizard. Locations added here via custom can be opened using command emacspeak-wizards-tramp-open-location bound to M-x emacspeak-wizards-tramp-open-location.

12.209 emacspeak-woman

WOMAN == Man pages implemented in Emacs Lisp

12.209.1 emacspeak-woman Options

User Option *voice-woman-bold* [Variable]
Personality used for woman-bold This personality uses voice-bolden whose effect can be changed globally by customizing voice-bolden-settings.

User Option *voice-woman-edition* [Variable]
Personality used for woman-edition This personality uses voice-bolden-medium whose effect can be changed globally by customizing voice-bolden-medium-settings.

User Option *voice-woman-italic* [Variable]
Personality used for woman-italic This personality uses voice-animate whose effect can be changed globally by customizing voice-animate-settings.

User Option *voice-woman-unknown* [Variable]
Personality used for woman-unknown This personality uses voice-monotone whose effect can be changed globally by customizing voice-monotone-settings.

12.210 emacspeak-xkcd

XKCD == XKCD In Emacs View XKCD comics in Emacs. Speech enables package xkcd Augments it by displaying the alt text and the transcript.

12.210.1 Emacspeak-Xkcd Commands

12.210.1.1 emacspeak-xkcd-open-explanation-browser

`emacspeak-xkcd-open-explanation-browser` [Command]

Open explanation of current xkcd in default browser

12.211 emacspeak-xref

XREF == Cross-references in source code. This is part of Emacs 25. This module speech-enables xref

12.212 emacspeak-xslt

libxml and libxsl are XML libraries for GNOME. xsltproc is a xslt processor using libxsl this module defines routines for applying xsl transformations using xsltproc

12.212.1 Emacspeak-Xslt Commands

12.212.1.1 emacspeak-xslt-view

`emacspeak-xslt-view` (*style url*) [Command]

Browse URL with specified XSL style.

(fn STYLE URL)

12.212.1.2 emacspeak-xslt-view-file

`emacspeak-xslt-view-file` (*style file*) [Command]

Transform 'file' using 'style' and preview via browse-url.

(fn STYLE FILE)

12.212.1.3 emacspeak-xslt-view-region

`emacspeak-xslt-view-region` (*style start end* **&optional** *unescape-charent*) [Command]

Browse XML region with specified XSL style.

(fn STYLE START END &optional UNESCAPE-CHARENT)

12.212.1.4 emacspeak-xslt-view-xml

`emacspeak-xslt-view-xml` (*style url* **&optional** *unescape-charent*) [Command]

Browse XML URL with specified XSL style.

(fn STYLE URL &optional UNESCAPE-CHARENT)

12.213 emacspeak-yaml

YAML == Yet Another Markup Language This module speech-enables yaml-mode.

12.214 emacspeak-yasnippet

YASNIPPET == Template based editing using snippets.

12.215 emacspeak-ytel

YTEL == Youtube From Elisp Install package ytel from Melpa.

12.215.1 Emacspeak-Ytel Commands

12.215.1.1 emacspeak-ytel-download

`emacspeak-ytel-download` (*id*) [Command]

Download video at point.

(fn ID)

12.215.1.2 emacspeak-ytel-play-at-point

`emacspeak-ytel-play-at-point` (*id* **&optional** *best*) [Command]

Play video. Argument ‘id’ is the video-id.
Play current video in ytel when called interactively.
Optional interactive prefix arg ‘best’ picks best audio format.

(fn ID &optional BEST)

12.215.1.3 emacspeak-ytel-yank

`emacspeak-ytel-yank` (*id*) [Command]

Yank youtube URL for video at point.

(fn ID)

12.216 espeak-voices

This module defines the various voices used in voice-lock mode by the ESpeak TTS engine.

12.216.1 Espeak-Voices Commands

12.216.1.1 espeak

`espeak` [Command]

C-e d C-e

<fn> d C-e
Start ESpeak engine.

12.216.2 espeak-voices Options

`User Option` *espeak-default-speech-rate* [Variable]

Default speech rate for eSpeak.

12.217 extra-muggles

MUGGLES == Emacspeak spells for power-users. These are extra hydras that I dont use very often, And are being moved here from emacspeak-muggles to save time at startup.

This module implements no new functionality — contrast with emacspeak-wizards. Instead, it uses package hydra to provide convenience key-bindings that access existing Emacspeak functionality. You need to install package Hydra first: '`M-x package-install hydra`'.

Note that on newer versions of Emacs, loading this module will attempt to automatically install package hydra if it is not found.

12.217.1 Using Hydras

See the high-level documentation for the Hydra package by executing *`C-h C-j`* hydra *`RET`*. The documentation in this section uses the same terminology as shown in the hydra package documentation.

12.217.2 Extra Muggles

- View-Mode: *`C-. v`* Temporarily behave like view-mode.
- SmartParens: *`C-c ,`* Smart Parens
- m-player: *`s-m`* Emacspeak-M-Player Commands
- m-player: *`s-;`* Emacspeak-M-Player muggle
- pianobar: *`s-'`* Emacspeak-M-pianobar Commands
- origami: C-, / Origami bindings.
- outliner: <C-c .> Bindings from outline-minor-mode.
- Info-Summary: <?> in Info Info Summary Muggle

Emacspeak automatically speaks Hydra hints when displayed. To silence all Hydra hints, set hydra-is-helpful to nil. To temporarily silence speaking of Hydra hints, Muggles can bind command `emacspeak-hydra-toggle-talkative`. As an example, Muggle '`ViewMode`' binds `s` to this command.

12.217.3 Extra-Muggles Commands

12.217.3.1 emacspeak-muggles-emacspeak-m-player-mode-map-cmd

`emacspeak-muggles-emacspeak-m-player-mode-map-cmd` [Command]

`s-m`

Temporarily use keymap emacspeak-m-player-mode-map

12.217.3.2 emacspeak-muggles-info-summary/body

`emacspeak-muggles-info-summary/body` [Command]

Call the body in the "emacspeak-muggles-info-summary" hydra.

The heads for the associated hydra are:

"]": ‘Info-forward-node’,
"[": ‘Info-backward-node’,
"n": ‘Info-next’,
"p": ‘Info-prev’,
"s": ‘Info-search’,
"S": ‘Info-search-case-sensitively’,
"l": ‘Info-history-back’,
"r": ‘Info-history-forward’,
"H": ‘Info-history’,
"t": ‘Info-top-node’,
"<": ‘Info-top-node’,
">": ‘Info-final-node’,
"u": ‘Info-up’,
"^": ‘Info-up’,
"m": ‘Info-menu’,
"g": ‘Info-goto-node’,
"b": ‘beginning-of-buffer’,
"e": ‘end-of-buffer’,
"f": ‘Info-follow-reference’,
"i": ‘Info-index’,
",": ‘Info-index-next’,
"I": ‘Info-virtual-index’,
"T": ‘Info-toc’,
"d": ‘Info-directory’,
"c": ‘Info-copy-current-node-name’,
"C": ‘clone-buffer’,
"a": ‘info-apropos’,
"1": ‘Info-nth-menu-item’,
"2": ‘Info-nth-menu-item’,
"3": ‘Info-nth-menu-item’,
"4": ‘Info-nth-menu-item’,
"5": ‘Info-nth-menu-item’,
"6": ‘Info-nth-menu-item’,
"7": ‘Info-nth-menu-item’,
"8": ‘Info-nth-menu-item’,
"9": ‘Info-nth-menu-item’,
"?": ‘Info-summary’,
"h": ‘Info-help’,
"q": ‘quit-window’,
"C-g": ‘nil’

The body can be accessed via ‘emacspeak-muggles-info-summary/body’.

12.217.3.3 emacspeak-muggles-m-player/body

`emacspeak-muggles-m-player/body` [Command]

s-;

Call the body in the "emacspeak-muggles-m-player" hydra.

The heads for the associated hydra are:

";": ‘emacspeak-m-player’,
"+": ‘emacspeak-m-player-volume-up’,
",": ‘emacspeak-m-player-backward-10s’,
"%": ‘emacspeak-m-player-display-percent’,
"-": ‘emacspeak-m-player-volume-down’,
".": ‘emacspeak-m-player-forward-10s’,
"<": ‘emacspeak-m-player-backward-1min’,
"<down>": ‘emacspeak-m-player-forward-1min’,
"<end>": ‘emacspeak-m-player-end-of-track’,
"<home>": ‘emacspeak-m-player-beginning-of-track’,
"<left>": ‘emacspeak-m-player-backward-10s’,
"<next>": ‘emacspeak-m-player-forward-10min’,
"<prior>": ‘emacspeak-m-player-backward-10min’,
"<right>": ‘emacspeak-m-player-forward-10s’,
"<up>": ‘emacspeak-m-player-backward-1min’,
"=": ‘emacspeak-m-player-volume-up’,
">": ‘emacspeak-m-player-forward-1min’,
"?": ‘emacspeak-m-player-display-position’,
"C": ‘emacspeak-m-player-clear-filters’,
"C-m": ‘emacspeak-m-player-load’,
"DEL": ‘emacspeak-m-player-reset-speed’,
"L": ‘emacspeak-m-player-load-file’,
"M-l": ‘emacspeak-m-player-load-playlist’,
"O": ‘emacspeak-m-player-reset-options’,
"P": ‘emacspeak-m-player-apply-reverb-preset’,
"Q": ‘emacspeak-m-player-quit’,
"R": ‘emacspeak-m-player-edit-reverb’,
"S": ‘emacspeak-amark-save’,
"SPC": ‘emacspeak-m-player-pause’,
"[": ‘emacspeak-m-player-slower’,
"]": ‘emacspeak-m-player-faster’,
"a": ‘emacspeak-m-player-amark-add’,
"b": ‘emacspeak-m-player-balance’,
"c": ‘emacspeak-m-player-slave-command’,
"d": ‘emacspeak-m-player-delete-filter’,
"e": ‘emacspeak-m-player-add-equalizer’,

"f": ‘emacspeak-m-player-add-filter’,
"g": ‘emacspeak-m-player-seek-absolute’,
"j": ‘emacspeak-m-player-amark-jump’,
"l": ‘emacspeak-m-player-get-length’,
"m": ‘emacspeak-m-player-mode-line’,
"n": ‘emacspeak-m-player-next-track’,
"o": ‘emacspeak-m-player-customize-options’,
"p": ‘emacspeak-m-player-previous-track’,
"q": ‘bury-buffer’,
"r": ‘emacspeak-m-player-seek-relative’,
"s": ‘emacspeak-m-player-scale-speed’,
"t": ‘emacspeak-m-player-play-tracks-jump’,
"u": ‘emacspeak-m-player-url’,
"v": ‘emacspeak-m-player-volume-change’,
"(": ‘emacspeak-m-player-left-channel’,
")": ‘emacspeak-m-player-right-channel’,
"{": ‘emacspeak-m-player-half-speed’,
"}": ‘emacspeak-m-player-double-speed’

The body can be accessed via ‘emacspeak-muggles-m-player/body’.

12.217.3.4 emacspeak-muggles-outliner/body

`emacspeak-muggles-outliner/body` [Command]

C-. o

C-' o

C-x @ s o

Call the body in the "emacspeak-muggles-outliner" hydra.

The heads for the associated hydra are:

"?": ‘(emacspeak-hydra-self-help "emacspeak-muggles-outliner")’,
"q": ‘outline-hide-sublevels’,
"t": ‘outline-hide-body’,
"o": ‘outline-hide-other’,
"c": ‘outline-hide-entry’,
"l": ‘outline-hide-leaves’,
"d": ‘outline-hide-subtree’,
"a": ‘outline-show-all’,
"e": ‘outline-show-entry’,
"i": ‘outline-show-children’,
"k": ‘outline-show-branches’,
"s": ‘outline-show-subtree’,
"u": ‘outline-up-heading’,
"n": ‘outline-next-visible-heading’,
"p": ‘outline-previous-visible-heading’,

"f": ‘outline-forward-same-level’,
"b": ‘outline-backward-same-level’,
"z": ‘nil’

The body can be accessed via ‘emacspeak-muggles-outliner/body’.

12.217.3.5 emacspeak-muggles-pianobar-key-map-cmd

`emacspeak-muggles-pianobar-key-map-cmd` [Command]

s-'

Temporarily use keymap pianobar-key-map

12.217.3.6 emacspeak-muggles-smartparens/body

`emacspeak-muggles-smartparens/body` [Command]

C-c ,

Call the body in the "emacspeak-muggles-smartparens" hydra.

The heads for the associated hydra are:

"’": ‘(lambda (_) (interactive "P") (sp-wrap-with-pair "’"))’,
"(": ‘(lambda (_) (interactive "P") (sp-wrap-with-pair "("))’,
"<down>": ‘sp-splice-sexp-killing-forward’,
"<left>": ‘sp-forward-barf-sexp’,
"<right>": ‘sp-forward-slurp-sexp’,
"<up>": ‘sp-splice-sexp-killing-backward’,
"?": ‘(emacspeak-hydra-self-help "emacspeak-muggles-smartparens")’,
"C-<left>": ‘sp-backward-barf-sexp’,
"C-<right>": ‘sp-backward-slurp-sexp’,
"R": ‘sp-splice-sexp’,
""": ‘(lambda (_) (interactive "P") (sp-wrap-with-pair "\""))’,
"a": ‘beginning-of-defun’,
"b": ‘sp-backward-sexp’,
"c": ‘sp-convolute-sexp’,
"d": ‘sp-down-sexp’,
"e": ‘end-of-defun’,
"f": ‘sp-forward-sexp’,
"i": ‘sp-indent-defun’,
"j": ‘sp-join-sexp’,
"k": ‘sp-kill-sexp’,
"n": ‘sp-next-sexp’,
"p": ‘sp-previous-sexp’,
"r": ‘sp-splice-sexp-killing-around’,
"s": ‘sp-split-sexp’,
"t": ‘sp-transpose-sexp’,
"u": ‘sp-backward-up-sexp’,
"w": ‘sp-copy-sexp’,

"{": ‘(lambda (_) (interactive "P") (sp-wrap-with-pair "{"))’

The body can be accessed via ‘emacspeak-muggles-smartparens/body’.

12.217.3.7 emacspeak-muggles-view/body

`emacspeak-muggles-view/body` [Command]

C-. v

C-' v

C-x @ s v

Call the body in the "emacspeak-muggles-view" hydra.

The heads for the associated hydra are:

"?": ‘(emacspeak-hydra-self-help "emacspeak-muggles-view")’,
"$": ‘set-selective-display’,
"%": ‘View-goto-percent’,
"’": ‘register-to-point’,
"(": ‘backward-sexp’,
")": ‘forward-sexp’,
".": ‘set-mark-command’,
"/": ‘View-search-regexp-forward’,
"<": ‘beginning-of-buffer’,
"<return>": ‘nil’,
"=": ‘what-line’,
">": ‘end-of-buffer’,
"@": ‘View-back-to-mark’,
"A": ‘beginning-of-defun’,
"DEL": ‘View-scroll-page-backward’,
"E": ‘end-of-defun’,
"J": ‘(emacspeak-hide-or-expose-block ’all)’,
"SPC": ‘View-scroll-page-forward’,
"[": ‘backward-page’,
"\": ‘View-search-regexp-backward’,
"]": ‘forward-page’,
"a": ‘move-beginning-of-line’,
"b": ‘backward-word’,
"c": ‘emacspeak-speak-char’,
"d": ‘View-scroll-half-page-forward’,
"e": ‘move-end-of-line’,
"f": ‘forward-word’,
"g": ‘goto-line’,
"h": ‘backward-char’,
"i": ‘emacspeak-speak-mode-line’,
"j": ‘next-line’,
"k": ‘previous-line’,

"l": ‘forward-char’,
"m": ‘point-to-register’,
"n": ‘View-search-last-regexp-forward’,
"p": ‘View-search-last-regexp-backward’,
"q": ‘nil’,
"r": ‘copy-to-register’,
"s": ‘emacspeak-hydra-toggle-talkative’,
"t": ‘(recenter 0)’,
"u": ‘View-scroll-half-page-backward’,
"w": ‘emacspeak-speak-word’,
"x": ‘exchange-point-and-mark’,
"y": ‘kill-ring-save’,
"{": ‘backward-paragraph’,
"}": ‘forward-paragraph’

The body can be accessed via ‘emacspeak-muggles-view/body’.

12.217.3.8 emacspeak-origami/body

`emacspeak-origami/body` [Command]

C-, /

C-x @ a /

Call the body in the "emacspeak-origami" hydra.

The heads for the associated hydra are:

"o": ‘origami-open-node’,
"c": ‘origami-close-node’,
"n": ‘origami-next-fold’,
"p": ‘origami-previous-fold’,
"f": ‘origami-forward-toggle-node’,
"a": ‘origami-toggle-all-nodes’

The body can be accessed via ‘emacspeak-origami/body’.

12.218 g-utils

Common Code e.g. helper functions. Used by modules like gphoto, gblogger etc.

12.218.1 g-utils Options

`User Option` *g-atom-view-xsl* [Variable]

XSLT transform to convert Atom feed to HTML.

`User Option` *g-cookie-jar* [Variable]

Cookie jar used for Google services. Customize this to live on your local disk.

`User Option` *g-curl-common-options* [Variable]

Common options to pass to all Curl invocations.

`User Option` *g-curl-debug* [Variable]

Set to T to see Curl stderr output.

`User Option` *g-url-under-point* [Variable]

Function used to get URL from current context.

`User Option` *g-xslt-program* [Variable]

XSLT Processor.

12.219 gm-nnir

Makes search GMail more convenient. IMap search operators, GMail search extensions.

12.219.1 Gm-Nnir Commands

12.219.1.1 gm-nnir-group-make-gmail-group

`gm-nnir-group-make-gmail-group` (*query*) [Command]

Use GMail search syntax exclusively.
See https://support.google.com/mail/answer/7190?hl=en for syntax.
note: nnimap-address etc are available as local vars if needed in these functions.

(fn QUERY)

12.219.1.2 gm-nnir-group-make-nnir-group

`gm-nnir-group-make-nnir-group` (*q*) [Command]

GMail equivalent of gnus-group-make-nnir-group.
This uses standard IMap search operators.

(fn Q)

12.220 gmaps

Implements the Google Maps API

12.220.1 Gmaps Commands

12.220.1.1 gmaps

`gmaps` [Command]

`C-; e`

`C-x @ h e`

This function has :around advice: ‘ad-Advice-gmaps’.

Google Maps Interaction.

(fn)

12.220.1.2 gmaps-bicycling-directions

`gmaps-bicycling-directions` (*origin destination*) [Command]

This function has :around advice: ‘ad-Advice-gmaps-bicycling-directions’.

Biking directions from Google Maps.

(fn ORIGIN DESTINATION)

12.220.1.3 gmaps-directions

`gmaps-directions` (*origin destination mode*) [Command]

Display directions obtained from Google Maps.

(fn ORIGIN DESTINATION MODE)

12.220.1.4 gmaps-driving-directions

`gmaps-driving-directions` (*origin destination*) [Command]

This function has :around advice: ‘ad-Advice-gmaps-driving-directions’.

Driving directions from Google Maps.

(fn ORIGIN DESTINATION)

12.220.1.5 gmaps-locations-load

`gmaps-locations-load` [Command]

Load saved GMaps locations.

12.220.1.6 gmaps-locations-save

`gmaps-locations-save` [Command]

Save GMaps Locations.

12.220.1.7 gmaps-mode

`gmaps-mode` [Command]

A Google Maps front-end for the Emacspeak desktop.

In addition to any hooks its parent mode ‘special-mode’ might have run, this mode runs the hook ‘gmaps-mode-hook’, as the final or penultimate step during initialization.

key binding
— ——-

TAB forward-button
ESC Prefix Command

SPC gmaps-place-details
[backward-page
] forward-page
b gmaps-bicycling-directions
c gmaps-set-current-location
d gmaps-driving-directions
f gmaps-set-current-filter
n gmaps-places-nearby
r gmaps-set-current-radius
s gmaps-places-search
t gmaps-transit-directions
w gmaps-walking-directions

M-i backward-button

12.220.1.8 gmaps-place-details

`gmaps-place-details` [Command]

This function has :around advice: 'ad-Advice-gmaps-place-details'.

Display details for place at point.
Insert reviews if already displaying details.

(fn)

12.220.1.9 gmaps-place-reviews

`gmaps-place-reviews` [Command]

Display reviews for place at point.
Place details need to have been expanded first.

12.220.1.10 gmaps-places-nearby

`gmaps-places-nearby` (**&optional** *clear-filter*) [Command]

This function has :around advice: 'ad-Advice-gmaps-places-nearby'.

Find places near current location.
Uses default radius. optional interactive prefix arg clears any active filters.

(fn &optional CLEAR-FILTER)

12.220.1.11 gmaps-places-search

`gmaps-places-search` (*query* **&optional** *clear-filter*) [Command]

This function has :around advice: 'ad-Advice-gmaps-places-search'.

Perform a places search.
Use this only if you dont know the locality of the place you're looking for.

Optional prefix arg clears any active filters.

(fn QUERY &optional CLEAR-FILTER)

12.220.1.12 gmaps-set-current-filter

`gmaps-set-current-filter` (**&optional** *all*) [Command]

Set up filter in current buffer.
Optional interactive prefix arg prompts for all filter fields.

(fn &optional ALL)

12.220.1.13 gmaps-set-current-location

`gmaps-set-current-location` (*address*) [Command]

This function has :around advice: 'ad-Advice-gmaps-set-current-location'.

Set current location.

(fn ADDRESS)

12.220.1.14 gmaps-set-current-radius

`gmaps-set-current-radius` (*radius*) [Command]

This function has :around advice: 'ad-Advice-gmaps-set-current-radius'.

Set current radius

(fn RADIUS)

12.220.1.15 gmaps-transit-directions

`gmaps-transit-directions` (*origin destination*) [Command]

This function has :around advice: 'ad-Advice-gmaps-transit-directions'.

Transit directions from Google Maps.

(fn ORIGIN DESTINATION)

12.220.1.16 gmaps-walking-directions

`gmaps-walking-directions` (*origin destination*) [Command]

This function has :around advice: 'ad-Advice-gmaps-walking-directions'.

Walking directions from Google Maps.

(fn ORIGIN DESTINATION)

12.220.2 gmaps Options

`User Option` *gmaps-api-key* [Variable]

Maps API key — goto https://code.google.com/apis/console to get one.

`User Option` *gmaps-my-address* [Variable]

Location address. Setting this updates gmaps-my-location coordinates via geocoding.

12.221 ladspa

This module uses tools from the Ladspa SDK to expose Ladspa plugins in a consistent way to elisp. The goal is to make it easy to inspect Ladspa Plugins, And invoke them easily from Ladspa host applications such as MPlayer. See `http://emacspeak.blogspot.com/2015/12/a-ladspa-work-bench-for-emacspeak.html` Some Ladspa Packages that provide plugins: sudo apt-get install zam-plugins wah-plugins vco-plugins tap-plugins swh-plugins rev-plugins mcp-plugins liquidsoap-plugin-ladspa ladspa-foo-plugins invada-studio-plugins-ladspa fil-plugins

12.221.1 Ladspa Commands

12.221.1.1 ladspa

`ladspa` (**&optional** *refresh*) [Command]

Launch Ladspa workbench.

(fn &optional REFRESH)

12.221.1.2 ladspa-analyse-plugin-at-point

`ladspa-analyse-plugin-at-point` [Command]

Analyse plugin at point.

12.221.1.3 ladspa-edit-control

`ladspa-edit-control` [Command]

Edit Ladspa control at point by prompting for control values.

12.221.1.4 ladspa-instantiate

`ladspa-instantiate` [Command]

Instantiate plugin at point by prompting for control values.

12.221.1.5 ladspa-mode

`ladspa-mode` [Command]

A Ladspa workbench for the Emacspeak desktop.

In addition to any hooks its parent mode 'special-mode' might have run,
this mode runs the hook 'ladspa-mode-hook', as the final or penultimate step
during initialization.

key binding
— ——-

RET ladspa-instantiate
SPC ladspa-analyse-plugin-at-point
a emacspeak-m-player-add-ladspa
d emacspeak-m-player-delete-ladspa
e ladspa-edit-control
n next-line
p previous-line

12.222 mac-voices

This module defines the various voices used in voice-lock mode by Mac TTS.

12.222.1 mac-voices Options

`User Option` *mac-default-speech-rate* [Variable]

Default speech rate for mac.

12.223 outloud-voices

This module defines the various voices used in voice-lock mode. This module is IBM ViaVoice Outloud specific.

12.223.1 Outloud-Voices Commands

12.223.1.1 outloud

`outloud` (**&optional** *device*) [Command]

`C-e d C-o`

`<fn> d C-o`
Select Outloud server.

(fn &optional DEVICE)

12.223.2 outloud-voices Options

`User Option` *outloud-default-speech-rate* [Variable]

Default speech rate for outloud.

12.224 plain-voices

This module defines the various voices used in voice-lock mode. This module is Plain i.e. suitable for a device for which you haven't yet implemented appropriate voice-locking controls

12.225 soundscape

Soundscapes `https://en.wikipedia.org/wiki/Soundscape` define an acoustic environment. Boodler at `http://boodler.org` is a Python-based SoundScape generator. To use this module, first install boodler. Then install the soundscape packages (*.boop) files available at `http://boodler.org/lib`. Make sure boodler works and produces audio in your environment. finally install the Boodler packages from emacspeak/scapes from the Emacspeak GitHub repository by running `cd emacspeak/scapes; make`

When boodler is set up and all packages installed, copy file emacspeak/scapes/soundscapes to ~/.boodler/Collection. The above file lists all installed SoundScapes. Directory emacspeak/scapes also contains additional Boodler Agents and SoundScapes that I have created for use with Emacspeak.

Module soundscape.el defines Emacs conveniences for running Soundscapes. Main Entry Points:

- M-x soundscape-toggle – Enables or disables automatic SoundScapes.
- M-x soundscape – runs a named SoundScape
- M-x soundscape-stop – Stops a specified running Soundscape.
- M-x soundscape-kill – Kills all running Soundscapes.

When automatic Soundscapes are enabled, SoundScapes are started and stopped based on the current major mode. Active Soundscape are displayed as part of the minor-mode-alist. Command emacspeak-speak-minor-mode-line can be used to have this spoken.

Thus, SoundScapes can be thought of as reflecting the *mood* of the current *mode*. This package defines a single *soundscape-default-theme* that is loaded using `(soundscape-load soundscape-default-theme)`. Emacs modes that provide similar functionality e.g., communication == email, IM, ... map to the same *mood*.

12.225.1 Soundscape Commands

12.225.1.1 soundscape

`soundscape` (*scape*) [Command]

`C-. s`

`C-' s`

`C-x @ s s`

Play soundscape.

(fn SCAPE)

12.225.1.2 soundscape-kill

`soundscape-kill` [Command]

Stop all running soundscapes.

12.225.1.3 soundscape-listener

`soundscape-listener` (**&optional** *restart*) [Command]

Start a Soundscape listener.
Listener is loaded with all Soundscapes defined in 'soundscape-default-theme' .
Optional interactive prefix arg restarts the listener if already running.

(fn &optional RESTART)

12.225.1.4 soundscape-listener-shutdown

`soundscape-listener-shutdown` [Command]

Shutdown listener.

12.225.1.5 soundscape-remote

`soundscape-remote` (*names*) [Command]

Activate scapes named 'names' — a list of strings.

(fn NAMES)

12.225.1.6 soundscape-restart

`soundscape-restart` (**&optional** *device*) [Command]

`C-. r`

`C-' r`

`C-x @ s r`

Restart Soundscape environment.
With prefix arg 'device', prompt for a alsa/ladspa device.
This is then saved to soundscape-device for future use.

(fn &optional DEVICE)

12.225.1.7 soundscape-stop

`soundscape-stop` (*scape*) [Command]

`C-. S`

`C-' S`

`C-x @ s S`

Stop running Soundscape.

(fn SCAPE)

12.225.1.8 soundscape-theme

`soundscape-theme` [Command]

Shows default theme in a special buffer.

12.225.1.9 soundscape-toggle

`soundscape-toggle` [Command]

C-. t

C-' t

C-x @ s t

Toggle automatic SoundScapes.
When turned on, Soundscapes are automatically run based on current major mode.
Run command M-x soundscape-theme to see the default mode->mood mapping.

12.225.1.10 soundscape-update-mood

`soundscape-update-mood` (**&optional** *prompt-mode*) [Command]

C-. u

C-' u

C-x @ s u

Update mood/scape mapping for current mode.
The updated mapping is not persisted.
Optional interactive prefix arg 'prompt-mode' prompts for the mode.

(fn &optional PROMPT-MODE)

12.225.2 soundscape Options

`User Option` *soundscape-data* [Variable]
Soundscape data directory.

`User Option` *soundscape-device* [Variable]
Alsa sound device to use for soundscapes.

`User Option` *soundscape-idle-delay* [Variable]
Number of seconds of idle time before soundscapes are synchronized with current mode.

`User Option` *soundscape-manager-options* [Variable]
User customizable options list passed to boodler. Defaults specify alsa as the output and set master volume to 0.5

12.226 sox

This module defines a convenient speech-enabled interface for editing mp3 and wav files using SoX.

Launching M-x sox creates a special interaction buffer that provides single keystroke commands for editing and applying effects to a selected sound file. For adding mp3 support to sox, do

sudo apt-get libsox-fmt-mp3 install

This module provides support for ladspa effects using module ladspa.el. To use ladspa effects with SoX, you need a relatively new build of Sox; The stock SoX that is package for Debian/Ubuntu does not always work. This module can be used independent of Emacspeak.

12.226.1 Sox Commands

12.226.1.1 sox-add-effect

`sox-add-effect` (*name*) [Command]

Adds effect at the end of the effect list

(fn NAME)

12.226.1.2 sox-delete-effect-at-point

`sox-delete-effect-at-point` [Command]

This function has :around advice: ‘ad-Advice-sox-delete-effect-at-point’.

Delete effect at point.

(fn)

12.226.1.3 sox-edit-effect-at-point

`sox-edit-effect-at-point` [Command]

Edit effect at point.

12.226.1.4 sox-mode

`sox-mode` [Command]

An audio workbench for the Emacspeak desktop.

In addition to any hooks its parent mode ‘special-mode’ might have run, this mode runs the hook ‘sox-mode-hook’, as the final or penultimate step during initialization.

key binding
— ——-

12.226.1.5 sox-open-file

`sox-open-file` (*snd-file*) [Command]

This function has :around advice: ‘ad-Advice-sox-open-file’.

Open specified snd-file on the Audio Workbench.

(fn SND-FILE)

12.226.1.6 sox-play

`sox-play` [Command]

Play sound from current context.

12.226.1.7 sox-refresh

`sox-refresh` [Command]

This function has :around advice: ‘ad-Advice-sox-refresh’.

Redraw Audio Workbench.

(fn)

12.226.1.8 sox-save

`sox-save` (*save-file*) [Command]

Save context to file after prompting.

(fn SAVE-FILE)

12.226.1.9 sox-set-effect

`sox-set-effect` (*name*) [Command]

Set effect.

(fn NAME)

12.226.1.10 sox-show-timestamp

`sox-show-timestamp` [Command]

Show timestamp in stream.

12.226.1.11 sox-stop

`sox-stop` [Command]

Stop currently playing sound from current context.

12.227 sox-gen

Provides binaural audio along with pre-defined themes. This module can be used independent of Emacspeak.

12.227.1 Binaural Beats Using SoX

A binaural beat is an auditory illusion perceived when two different pure-tone sine waves, both with frequencies lower than 1500 Hz, with less than a 40 Hz difference between them, are presented to a listener dichotically (one through each ear). For example, if a 530 Hz pure tone is presented to a subject’s right ear, while a 520 Hz pure tone is presented to the subject’s left ear, the listener will perceive the auditory illusion of a third tone, in addition to the two pure-tones presented to each ear. The third sound is called a binaural beat, and in this example would have a perceived pitch correlating to a frequency of 10 Hz, that being the difference between the 530 Hz and 520 Hz pure tones presented to each ear. For more details, see `https://en.wikipedia.org/wiki/Binaural_beats`.

This module implements a set of user-facing commands for generating binaural beats. The commands are organized from high-level commands that play predefined binaural beats to lower-level commands that can be used to create new effect sequences.

All binaural beat sequences are played with a relatively low gain — they are designed to be heard in the background and when effective blend fully into the background. You can increase the overall volume of all binaural beat sequences by customizing

`User Option` *sox-binaural-gain-offset to a positive value* [Variable]
— default is 0.

12.227.1.1 High-Level Commands For Pre-Defined Binaural Beats

These commands can be called directly to play one of the predefined binaural beats.

- `sox-rev-up`: A set of binaural beats designed for use at the start of the day. Transitions from *Dream* -> *Think* ->*Act* -> *Focus*.
- `sox-wind-down`: A set of binaural beats for winding down at the end of the day. This can be thought of as the reverse of `sox-rev-up` and the sequence transitions from *Act* -> *Think* -> *Dream* ->*Sleep*.
- `sox-turn-down`: Designed for falling asleep. This sequence starts with a short period of *Dream* before moving to *Sleep*.
- `sox-relax`: A variant of the previous sequence, `sox-relax` spends equal time in *Dream* and *Sleep*.
- `sox-binaural`: Provide a completion-based front-end to playing any one of the predefined binaural effects (*Delta*, *Theta*, *Alpha*, *Beta*, or *Gamma*. The previously defined sequences are built up using these effects.
- `sox-beats-binaural`: Plays a collection of binaural beats, prompting for carrier and beat frequencies for each tone. The predefined sequences listed earlier were created after first generating experimental beat-sequences using this command.
- `sox-slide-binaural`: Prompts for two binaural effects (see above) and generates a binaural beat that *slides* from the first effect to the second over a specified duration.
- `sox-chakras`: Pick amongst one of a predefined set of sequences designed for *Chakra* meditation.
- `sox-tone-binaural`: Generate a simple binaural beat with a single carrier frequency.
- `sox-tone-slide-binaural`: Generate a tone that slides from one binaural beat to another.

12.227.2 Sox-Gen Commands

12.227.2.1 sox-beats-binaural

`sox-beats-binaural` (*length beat-spec-list gain*) [Command]
Play binaural audio with beat-spec specifying the various tones.

Param 'beat-spec-list' is a list of '(carrier beat) tupples.

(fn LENGTH BEAT-SPEC-LIST GAIN)

12.227.2.2 sox-binaural

`sox-binaural` (*name duration*) [Command]

C-, b

C-x @ a b

Play specified binaural effect.

(fn NAME DURATION)

12.227.2.3 sox-chakras

`sox-chakras` (*theme duration*) [Command]

Play each chakra for specified duration.
Parameter 'theme' specifies variant.

(fn THEME DURATION)

12.227.2.4 sox-relax

`sox-relax` (*length*) [Command]

Play relax set of binaural beats.
Param 'length' specifies total duration.

(fn LENGTH)

12.227.2.5 sox-rev-up

`sox-rev-up` (*length*) [Command]

Play rev-up set of binaural beats.
Param 'length' specifies total duration.

(fn LENGTH)

12.227.2.6 sox-slide-binaural

`sox-slide-binaural` (*name-1 name-2 duration*) [Command]

Play specified binaural slide from 'name-1' to 'name-2'.

(fn NAME-1 NAME-2 DURATION)

12.227.2.7 sox-tone-binaural

`sox-tone-binaural` (*length freq beat gain*) [Command]

Play binaural audio with carrier frequency 'freq', beat 'beat', and gain ‘gain’.

(fn LENGTH FREQ BEAT GAIN)

12.227.2.8 sox-tone-slide-binaural

`sox-tone-slide-binaural` (*length freq beat-start beat-end gain*) [Command]

Play binaural audio with carrier frequency ‘freq’, beat
‘beat-start’ -> ‘beat-end’, and gain ‘gain’.

(fn LENGTH FREQ BEAT-START BEAT-END GAIN)

12.227.2.9 sox-turn-down

`sox-turn-down` (*length*) [Command]

Play turn-down set of binaural beats.
Param ‘length’ specifies total duration.

(fn LENGTH)

12.227.2.10 sox-wind-down

`sox-wind-down` (*length*) [Command]

Play wind-down set of binaural beats.
Param ‘length’ specifies total duration.

(fn LENGTH)

12.227.3 sox-gen Options

`User Option` *sox-binaural-gain-offset* [Variable]

User specified offset that is added to default gain when generating tones using SoX, e.g., for binaural beats.

`User Option` *sox-binaural-slider-scale* [Variable]

Scale factor used to compute slide duration when moving from one binaural beat to another.

12.228 tetris

12.229 toy-braille

This is a bit of toy code to write in braille. To try this, load this file (‘M-x load-file path/to/toy-braille.el’), then do:

M-: (get-toy-braille-string "just a test")

That’s just a toy, meant as an excuse and maybe a tool to learn a bit of braille, nothing more.

Unicode fonts are needed.

You can try:

M-: (set-default-font "-*-unifont-*-*-*-*-*-*-*-*-*-*-*-*")

or

M-: (set-default-font "-*-clearlyu-*-*-*-*-*-*-*-*-*-*-iso10646-*")

(it will only work if the relative font is installed and properly configured).

References:

http://www.nbp.org/ic/nbp/braille/index.html

http://www.unicode.org/Public/4.0-Update1/UnicodeData-4.0.1.txt

12.230 voice-setup

A voice is to audio as a font is to a visual display. A personality is to audio as a face is to a visual display.

Voice-lock-mode is a minor mode that causes your comments to be spoken in one personality, strings in another, reserved words in another, documentation strings in another, and so on.

Comments will be spoken in 'voice-comment-personality'. Strings will be spoken in 'voice-string-personality'. Function and variable names (in their defining forms) will be spoken in 'voice-function-name-personality'. Reserved words will be spoken in 'voice-keyword-personality'.

To make the text you type be voiceified, use M-x voice-lock-mode. When this minor mode is on, the voices of the current line are updated with every insertion or deletion.

How faces map to voices: TTS engine specific modules e.g., dectalk-voices.el and outloud-voices.el define a standard set of voice names. This module maps standard "personality" names to these pre-defined voices. It does this via special form def-voice-font which takes a personality name, a voice name and a face name to set up the mapping between face and personality, and personality and voice. Newer Emacspeak modules should use voice-setup-add-map when defining face->personality mappings. Older code calls def-voice-font directly, but over time those calls will be changed to the more succinct form provided by voice-setup-add-map. For use from other modules, also see function voice-setup-map-face which is useful when mapping a single face. Both voice-setup-add-map and voice-setup-map-face call special form def-voice-font.

Special form def-voice-font sets up the personality name to be available via custom. new voices can be defined using CSS style specifications see special form defvoice Voices defined via defvoice can be customized via custom see the documentation for defvoice.

12.230.1 Voice-Setup Commands

12.230.1.1 voice-lock-mode

`voice-lock-mode` (**&optional** *arg*) [Command]

C-e d v

<fn> d v

Toggle voice lock mode.

If called interactively, toggle 'Voice-Lock mode'. If the prefix
argument is positive, enable the mode, and if it is zero or
negative, disable the mode.

If called from Lisp, toggle the mode if ARG is 'toggle'. Enable
the mode if ARG is nil, omitted, or is a positive number.
Disable the mode if ARG is a negative number.

The mode's hook is called both when the mode is enabled and when
it is disabled.

(fn &optional ARG)

12.230.1.2 voice-lock-mode–turn-on

`voice-lock-mode--turn-on` [Command]

Turn on Voice Lock mode .

12.230.1.3 voice-setup-describe-personality

`voice-setup-describe-personality` (*personality*) [Command]

C-h C-v

<f1> C-v

<help> C-v

Describe specified voice — analogous to C-h /.
When called interactively, 'personality' defaults to first personality at point.
If there are multiple personalities at point,
these are available via minibuffer history.

(fn PERSONALITY)

12.230.1.4 voice-setup-toggle-silence-personality

`voice-setup-toggle-silence-personality` [Command]

C-e M-q

<fn> M-q

Toggle audibility of personality under point .

12.230.2 voice-setup Options

`User Option` *voice-animate-extra-settings* [Variable]

Adds extra animation to current voice.

`User Option` *voice-animate-medium-settings* [Variable]

Adds medium animation current voice.

`User Option` *voice-animate-settings* [Variable]
Animates current voice.

`User Option` *voice-annotate-settings* [Variable]
Indicate annotation.

`User Option` *voice-bolden-and-animate-settings* [Variable]
Bolden and animate current voice.

`User Option` *voice-bolden-extra-settings* [Variable]
Extra bolden current voice.

`User Option` *voice-bolden-medium-settings* [Variable]
Add medium bolden current voice.

`User Option` *voice-bolden-settings* [Variable]
Bolden current voice.

`User Option` *voice-brighten-extra-settings* [Variable]
Extra brighten current voice.

`User Option` *voice-brighten-medium-settings* [Variable]
Brighten (medium) current voice.

`User Option` *voice-brighten-settings* [Variable]
Brighten current voice.

`User Option` *voice-indent-settings* [Variable]
Indicate indentation .

`User Option` *voice-lighten-extra-settings* [Variable]
Add extra lightness to current voice.

`User Option` *voice-lighten-medium-settings* [Variable]
Add medium lightness to current voice.

`User Option` *voice-lighten-settings* [Variable]
Lighten current voice.

`User Option` *voice-lock-mode-hook* [Variable]
Hook run after entering or leaving ‘voice-lock-mode’. No problems result if this variable is not bound. ‘add-hook’ automatically binds it. (This is true for all hook variables.)

`User Option` *voice-lock-overlay-0-settings* [Variable]
Overlay voice that sets dimension 0 of ACSS structure to 8.

`User Option` *voice-lock-overlay-1-settings* [Variable]
Overlay voice that sets dimension 1 of ACSS structure to 8.

`User Option` *voice-lock-overlay-2-settings* [Variable]
Overlay voice that sets dimension 2 of ACSS structure to 8.

`User Option` *voice-lock-overlay-3-settings* [Variable]
Overlay voice that sets dimension 3 of ACSS structure to 8.

`User Option` *voice-monotone-light-settings* [Variable]
Turns current voice into a light monotone.

`User Option` *voice-monotone-medium-settings* [Variable]
Turns current voice into a medium monotone.

`User Option` *voice-monotone-settings* [Variable]
Turns current voice into a monotone and speaks all punctuations.

`User Option` *voice-punctuations-all-settings* [Variable]
Turns current voice into one that speaks all punctuations.

`User Option` *voice-punctuations-none-settings* [Variable]
Turns current voice into one that speaks no punctuations.

`User Option` *voice-punctuations-some-settings* [Variable]
Turns current voice into one that speaks some punctuations.

`User Option` *voice-smoothen-extra-settings* [Variable]
Extra smoothen current voice.

`User Option` *voice-smoothen-medium-settings* [Variable]
Add medium smoothen current voice.

`User Option` *voice-smoothen-settings* [Variable]
Smoothen current voice.

`User Option` *voice-womanize-1-settings* [Variable]
Apply first female voice.

12.231 xbacklight

Provide an emacs front-end to xbacklight. This is a tool that controls the brightness on laptops. To install xbacklight, sudo apt-get install xbacklight

This module is most easily used in conjunction with hydra: M-x package-install hydra

12.231.1 Xbacklight Commands

12.231.1.1 xbacklight-black

`xbacklight-black` [Command]
Turn screen black.

12.231.1.2 xbacklight-decrement

`xbacklight-decrement` [Command]
Decrease brightness by by one step.

12.231.1.3 xbacklight-get

`xbacklight-get` [Command]

Get current brightness level.

12.231.1.4 xbacklight-increment

`xbacklight-increment` [Command]

Increase brightness by by one step.

12.231.1.5 xbacklight-set

`xbacklight-set` (*brightness*) [Command]

Set brightness to specified level.
'brightness' is a percentage value.

(fn BRIGHTNESS)

12.231.1.6 xbacklight-white

`xbacklight-white` [Command]

Turn screen white.

12.232 URL Templates

This section documents a total of 75 URL Templates.

All of these URL templates can be invoked via command *`M-x emacspeak-url-template-fetch`* normally bound to *`C-e u <fn> u`*. This command prompts for the name of the template, and completion is available via Emacs' minibuffer completion. Each URL template carries out the following steps:

- Prompt for the relevant information.
- Fetch the resulting URL using an appropriate fetcher.
- Set up the resulting resource with appropriate customizations.

1. **AQI From Wunderground**

   Air quality from Wunderground
2. **Air Traffic Control**

   Find live streams for Air Traffic Control.
3. **Airport conditions**

   Display airport conditions from the FAA.
4. **Amazon Product Details By ASIN**

   Retrieve product details from Amazon by either ISBN or ASIN.
5. **Anonymize Google Search**

   Logout from Google to do an anonymous search.
6. **ArchWiki Search**

   Search Linux ArchWiki

7. **BBC Podcast Directory**
   BBC PodCast Directory
8. **BBC World News Summary**
   BBC World News Summary
9. **Baseball Box Scores**
   Display baseball Play By Play.
10. **Baseball Game Details**
    Display baseball Play By Play.
11. **Baseball Game Index**
    Display baseball Play By Play.
12. **Baseball Highlights**
    Display baseball Video Highlights.
13. **Baseball scores**
    Display baseball scores.
14. **Bing News**
    Bing News results as RSS feed.
15. **Bing Search**
    Bing results as RSS feed.
16. **Bloomberg Stock Lookup**
    Lookup Stock Quote information on Bloomberg. Ticker is of the form goog:us
17. **CIA Leaders Of The World**
    Open CIA World Leaders For Specified Country.
18. **CIA World Fact Book**
    Open CIA World Fact Book For Specified Country.
19. **CNN Content**
    Filter down to CNN content area.
20. **CNN Headlines**
    News Headlines From CNN
21. **CNN Market Data**
    Market data filtered from CNN Money
22. **CNN PodCasts**
    List CNN Podcast media links.
23. **Currency Converter**
    Currency Converter. Currencies can be a comma-separated list of codes.
24. **Dictionary Lookup**
    Dictionary Lookup
25. **Finance Summary From Google**
    Display top stocks from Google Finance.

26. **FreeSound**
    Search FreeSound.
27. **GitHub Search**
    Perform a GitHub Search.
28. **GoLang Browse**
    Browse GoLang package documentation.
29. **GoLang Lookup**
    Lookup GoLang package documentation.
30. **GoLang Search**
    Search GoLang package documentation.
31. **Google Basic**
    Light-weight Google search.
32. **Google News Search**
    Search Google news.
33. **Google Scholar**
    Google Scholar Search
34. **Google Trends**
    Google Trends
35. **Google Weather**
    Light-weight Google search.
36. **Guardian RSS Feeds Directory**
    Guardian Feeds Directory
37. **Hacker News Frontpage**
    Display Hacker News Front Page
38. **Hacker News Search**
    Display Hacker News Front Page
39. **MLB Scorecard**
    Show MLB Scorecard.
40. **MLB standings**
    Display MLB standings.
41. **Money Headlines From CNN**
    Money Headlines From CNN
42. **NBA standings**
    Display NBA standings.
43. **NLS Bard Popular**
    NLS Bard Catalog: Most Popular. Login once before using this template.
44. **NLS Bard Recent**
    NLS Bard Catalog: Recently Added. Login once before using this template.

45. **NLS Bard Search**
    Search NLS Bard Catalog. Login once before using this template.
46. **NY Times Mobile**
    NYTimes Mobile Site
47. **Old Time Radio**
    This months Old Time Radio Programming
48. **OpenLibrary**
    Open Library Search
49. **Patent Search From Google**
    Perform patent search via Google
50. **RadioTime Browser**
    RadioTime Entry point.
51. **RadioTime Categories**
    RadioTime Categories .
52. **RadioTime Search**
    RadioTime Search.
53. **Reddit At Point.**
    Open RSS Feed for Reddit URL under point.
54. **Reddit By Topic.**
    Open RSS Feed for Reddit Topic.
55. **Reddit Front Page.**
    Open Feed for Reddit Front Page.
56. **Reddit Search.**
    Reddit Search Results Feed.
57. **Seeking Alpha Stock Search**
    Seeking Alpha search.
58. **Sign in to Google**
    Login to Google.
59. **StreamWorld Radio**
    Play radio stream. Example: kcbsFM. Format is stationid+AM/FM.
60. **Tech News From CNet**
    Display tech news from CNET
61. **TuneIn Radio**
    Translate StreamId to playable stream.
62. **UPS Packages**
    Display package tracking information from UPS.
63. **Washington Post**
    Washington Post Contents

64. **Wiki Data Search**
    Search WikiData.
65. **WordNet Search**
    Look up term in WordNet.
66. **Yahoo RSSNews**
    News From Yahoo As RSS.
67. **Youtube News**
    News Headlines From Youtube
68. **fedex packages**
    Display package tracking information from Fedex.
69. **html Google News Search**
    Search Google news.
70. **sourceforge Download**
    Download specified file.
71. **sourceforge browse mirrors**
    Retrieve download page at Sourceforge for specified project.
72. **sourceforge project**
    Open specified project page at SourceForge.
73. **w3c IRC Logs**
    Use this to pull up the archived logs from the W3C IRC. You need to know the exact name of the channel.
74. **w3c Lists**
    Use this to pull up the archived mail from the W3C list. You need to know the exact name of the list.
75. **world CNN Content**
    Filter down to CNN content area.

13 Emacspeak Keyboard Commands.

This chapter gives an overview of all the keymaps used by Emacspeak. For a complete reference, see See Section 12.9 [emacspeak], page 58. For basic usage, see See Chapter 6 [Basic Usage], page 9.

Emacspeak uses the following keymaps, each of which are invoked by a specific prefix key.

`C-e`	The main Emacspeak keymap.
`C-e d`	The text-to-speech keymap.
`C-;`	The Emacspeak hyper keymap.
`C-'`	The emacspeak super keymap.
`C-,`	The Emacspeak alt keymap.
`C-.`	The Emacspeak super keymap.
`C-e x`	The emacspeak `x` keymap.
`C-e C-x`	The Emacspeak `C-x` keymap.

Primary Emacspeak commands start with `C-e`. Following `C-e` with `d` invokes commands that control the text-to-speech engine. Note that silencing speech is an exception to this rule — Speech silence commands are placed directly on the primary emacspeak-keymap (`C-e s` and `C-e .`).

In addition, Emacspeak introduces five additional keymaps for binding its extensive set of facilities to convenient keystrokes.

When running under a windowing system, Emacs automatically receives keys `C-;`, `C-'`, `C-,` and `C-.`. When running on the Linux console, these keys become available after loading the custom Linux keymap found in `emacspeak/tvr/console-keymaps` after checking out the emacspeak repository from `https://github.com/tvraman/emacspeak`.

Emacspeak defines personal keymaps accessible via `C-e x` and `C-e C-x`. For now, emacspeak does not bind any commands in keymap `C-e C-x` — this keymap is left for end-user personalization.

Note that the information presented in the following subsections can also be viewed via Emacs' built-in Help system; e.g., Press `C-; C-h` to get a `*Help*` buffer that displays all keys bound in `emacspeak-hyper-keymap`.

14 TTS Servers

Emacspeak produces spoken output by communicating with one of many speech servers. This section documents the communication protocol between the client application i.e. Emacspeak, and the TTS (Text to Speech) server. This section is primarily intended for developers wishing to:

- Create new speech servers that comply with this communication protocol
- Developers of other client applications who wish to use the various Emacspeak speech servers.

For additional notes on how to log and view TTS server commands when developing a speech server, see `http://emacspeak.blogspot.com/2015/04/howto-log-speech-server-output-to-aid.html`.

14.1 High-level Overview

The TTS server reads commands from standard input, and script *speech-server* can be used to cause a TTS server to communicate via a TCP socket. Speech server commands are used by the client application to make specific requests of the server; the server listens for these requests in a non-blocking read loop and executes requests as they become available. Requests can be classified as follows:

- Commands that send text to be spoken.
- Commands that set *state* of the TTS server.

All commands are of the form

```
commandWord {arguments}
```

The braces are optional if the command argument contains no white space. The speech server maintains a *current state* that determines various characteristics of spoken output such as speech rate, punctuations mode etc. (see set of commands that manipulate speech state for complete list). The client application *queues* The text and non-speech audio output to be produced before asking the server to *dispatch* the set of queued requests, i.e. start producing output.

Once the server has been asked to produce output, it removes items from the front of the queue, sends the requisite commands to the underlying TTS engine, and waits for the engine to acknowledge that the request has been completely processed. This is a non-blocking operation, i.e., if the client application generates additional requests, these are processed *immediately*.

The above design allows the Emacspeak TTS server to be *highly* responsive; Client applications can queue large amounts of text (typically queued a clause at a time to achieve the best prosody), ask the TTS server to start speaking, and interrupt the spoken output at any time.

14.1.1 Commands That Queue Output.

This section documents commands that either produce spoken output, or queue output to be produced on demand. Commands that place the request on the queue are clearly marked.

```
version
```

Speaks the *version* of the TTS engine. Produces output immediately.

`tts_say text`

Speaks the specified *text* immediately. The text is not pre-processed in any way, contrast this with the primary way of speaking text which is to queue text before asking the server to process the queue.

Note that this command needs to handle the special syntax for morpheme boundaries '`[*]`'. The '`[*]`' syntax is specific to the Dectalk family of synthesizers; servers for other TTS engines need to map this pattern to the engine-specific code for each engine. As an example, see `servers/outloud` A morpheme boundary results in synthesizing compound words such as *left bracket* with the right intonation; using a space would result in that phrase being synthesized as two separate words.

`l c`

Speak *c* a single character, as a letter. The character is spoken immediately. This command uses the TTS engine's capability to speak a single character with the ability to flush speech *immediately*. Client applications wishing to produce character-at-a-time output, e.g., when providing character echo during keyboard input should use this command.

`d`

This command is used to *dispatch* all queued requests. It was renamed to a single character command (like many of the commonly used TTS server commands) to work more effectively over slow (9600) dialup lines. The effect of calling this command is for the TTS server to start processing items that have been queued via earlier requests.

`tts_pause`

This pauses speech *immediately*. It does not affect queued requests; when command *tts_resume* is called, the output resumes at the point where it was paused. Not all TTS engines provide this capability.

`tts_resume`

Resume spoken output if it has been paused earlier.

`s`

Stop speech *immediately*. Spoken output is interrupted, and all pending requests are flushed from the queue.

`q text`

Queues text to be spoken. No spoken output is produced until a *dispatch* request is received via execution of command *d*.

`c codes`

Queues synthesis codes to be sent to the TTS engine. Codes are sent to the engine with no further transformation or processing. The codes are inserted into the output queue and will be dispatched to the TTS engine at the appropriate point in the output stream.

`a filename`

Cues the audio file identified by filename for playing.

`t freq length`

Queues a tone to be played at the specified frequency and having the specified length. Frequency is specified in hertz and length is specified in milliseconds.

`sh duration`

Queues the specified duration of silence. Silence is specified in milliseconds.

14.1.2 Commands That Set State

```
tts_reset
```

Reset TTS engine to default settings.

```
tts_set_punctuations mode
```

Sets TTS engine to the specified punctuation mode. Typically, TTS servers provide at least three modes:

- None: Do not speak punctuation characters.
- Some: Speak some punctuation characters. Used for English prose.
- All: Speak out *all* punctuation characters; useful in programming modes.

```
tts_set_speech_rate rate
```

Sets speech rate. The interpretation of this value is typically engine specific.

```
tts_set_character_scale factor
```

Scale factor applied to speech rate when speaking individual characters.Thus, setting speech rate to 500 and character scale to 1.2 will cause command *l* to use a speech rate of $500 * 1.2 = 600$.

```
tts_split_caps flag
```

Set state of *split caps* processing. Turn this on to speak mixed-case (AKA Camel Case) identifiers.

```
tts_capitalize flag
```

Indicate capitalization via a beep tone or voice pitch.

```
tts_allcaps_beep flag
```

Setting this flag produces a high-pitched beep when speaking words that are in all-caps, e.g. abbreviations.

15 Emacspeak At Twenty.

This article was originally published to mark the 20th anniversary of Emacspeak. It has been incorporated as the final chapter of the Emacspeak manual for easy reference. The original article is available on the Web and in its original source form as an *org* file in the emacspeak distribution.

15.1 Turning Twenty

One afternoon in the third week of September 1994, I started writing myself a small Emacs extension using Lisp Advice to make Emacs speak to me so I could use a Linux laptop. As Emacspeak turns twenty, this article is both a quick look back over the twenty years of lessons learned, as well as a glimpse into what might be possible as we evolve to a world of connected, ubiquitous computing. This article draws on Learning To Program In 10 Years (`http://norvig.com/21-days.html`) by Peter Norvig for some of its inspiration.

15.2 Using UNIX With Speech Output — 1994

As a graduate student at Cornell (`http://www.cs.cornell.edu/info/people/raman/raman.html`), I accessed my Unix workstation (SunOS) from an Intel 486 PC running IBM Screen-Reader. There was no means of directly using a UNIX box at the time; after graduating, I continued doing the same for about six months at Digital Research in Cambridge — the only difference being that my desktop workstation was now a DEC-Alpha. Throughout this time, Emacs was my environment of choice for everything from software development and Internet access to writing documents.

In fall of 1994, I wanted to start using a laptop running Linux; a colleague (Dave Wecker) was retiring his 386mhz laptop that already had Linux on it and I decided to inherit it. But there was only one problem — until then I had always accessed a UNIX machine from a secondary PC running a screen-reader — something that would clearly make no sense with a laptop!

Another colleague, Win Treese, had pointed out the interesting possibilities presented by package **advice** in Emacs 19.23 — a few weeks earlier, he had sent around a small snippet of code that magically modified Emacs' version-control primitive to first create an *RCS* directory if none existed before adding a file to version control. When I speculated about using the Linux laptop, Dave remarked — you live in Emacs anyway — why dont you just make it talk!

Connecting the dots, I decided to write myself a tool that augmented Emacs' default behavior to *speak* — within about 4 hours, version 0.01 of Emacspeak was up and running.

15.3 Key Enabler — Emacs And Lisp Advice

It took me a couple of weeks to fully recognize the potential of what I had built with Emacs Lisp Advice. Until then, I had used screen-readers to listen to the contents of the visual display — but Lisp Advice let me do a lot more — it enabled Emacspeak to generate highly context-specific spoken feedback, augmented by a set of auditory icons. I later formalized this design under the name speech-enabled applications (`http://en.wikipedia.org/wiki/Self-voicing`). For a detailed overview of the architecture of

Emacspeak, see the chapter on Emacspeak (`http://emacspeak.sourceforge.net/raman/publications/bc-emacspeak/publish-emacspeak-bc.html`) in the book Beautiful Code (`http://emacspeak.blogspot.com/2007/07/emacspeak-and-beautiful-code.html`) from O'Reilly.

15.4 Key Component — Text To Speech (TTS)

Emacspeak is a speech-subsystem for Emacs; it depends on an external Text-To-Speech (TTS) engine to produce speech. In 1994, Digital Equipment released what would turn out to be the last in the line of hardware DECTalk synthesizers, the DECTalk Express. This was essentially an Intel 386with 1mb of flash memory that ran a version of the DECTalk TTS software — to date, it still remains my favorite Text-To-Speech engine. At the time, I also had a software version of the same engine running on my DEC-Alpha workstation; the desire to use either a software or hardware solution to produce speech output defined the Emacspeak speech-server architecture.

I went to IBM Research in 1999; this coincided with IBM releasing a version of the Eloquennce TTS engine on Linux under the name *ViaVoice Outloud*. My colleague Jeffrey Sorenson implemented an early version of the Emacspeak speech-server for this engine using the OSS API; I later updated it to use the ALSA library while on a flight back to SFO from Boston in 2001. That is still the TTS engine that is speaking as I type this article on my laptop.

20 years on, TTS continues to be the weakest link on Linux; the best available solution in terms of quality continues to be the Linux port of Eloquence TTS available from Voxin in Europe for a small price. Looking back across 20 years, the state of TTS on Linux in particular and across all platforms in general continues to be a disappointment; most of today's newer TTS engines are geared toward mainstream use-cases where *naturalness* of the voice tends to supersede intelligibility at higher speech-rates. Ironically, modern TTS engines also give applications far less control over the generated output — as a case in point, I implemented Audio System For Technical Readings (AsTeR) (`http://www.cs.cornell.edu/home/raman/aster/demo.html`) in 1994 using the DECTalk; 20 years later, we implemented MathML support (`http://allthingsd.com/20130604/t-v-ramans-audio-deja-vu-from-google-a-math-reading-system-for-the-web/`) in ChromeVox (`http://www.chromevox.com/`) using Google TTS. In 2013, it turned out to be difficult or impossible to implement the type of audio renderings that were possible with the admittedly less-natural sounding DECTalk!

15.5 Emacspeak And Software Development

Version 0.01 of Emacspeak was written using IBM Screen-Reader on a PC with a terminal emulator accessing a UNIX workstation. But in about 2 weeks, Emacspeak was already a better environment for developing Emacspeak in particular and software development in general. Here are a few highlights in 1994 that made Emacspeak a good software development environment, present-day users of Emacspeak will see that that was just scratching the surface.

- Audio formatting using voice-lock to provide aural syntax highlighting.
- Succinct auditory icons to provide efficient feedback.
- Emacs' ability to navigate code structurally —

as opposed to moving around by plain-text units such as characters, lines and words. S-Expressions are a major win!

- Emacs' ability to specialize behavior based on major and minor modes.
- Ability to browse program code using tags, and getting fluent spoken feedback.
- Completion **everywhere**.
- Everything is searchable — this is a huge win when you cannot see the screen.
- Interactive spell-checking using ISpell with continuous spoken feedback augmented by aural highlights.
- Running code compilation and being able to jump to errors with spoken feedback.
- Ability to move through diff chunks when working with source code and source control systems; refined diffs as provided by the **ediff** package when speech-enabled is a major productivity win.
- Ability to easily move between email, document authoring and programming — though this may appear trivial, it continues to be one of Emacs' biggest wins.

Long-term Emacs users will recognize all of the above as being among the reasons why they do most things inside Emacs — there is little that is Emacspeak specific in the above list — except that Emacspeak was able to provide fluent, well-integrated contextual feedback for all of these tasks. And that was a game-changer given what I had had before Emacspeak. As a case in point, I did not dare program in Python before I speech-enabled Emacs' Python-Mode; the fact that white space is significant in Python made it difficult to program using a plain screen-reader that was unaware of the semantics of the underlying content being accessed.

15.5.1 Programming Defensively

As an aside, note that all of Emacspeak has been developed over the last 20 years with Emacspeak being the only adaptive technology on my system. This has led to some interesting design consequences, primary among them being a strong education in **programming defensively**. Here are some other key features of the Emacspeak code-base:

1. The code-base is extremely **bushy** rather than deeply hierarchical — this means that when a module breaks, it does not affect the rest of the system.
2. Separation of concerns with respect to the various layers, a tightly knit core speech library interfaces with any one of many speech servers running as an external process.
3. Audio formatting is abstracted by using the formalism defined in Aural CSS.
4. Emacspeak integrates with Emacs' user interface conventions by taking over a single prefix key **C-e** with *all* Emacspeak commands accessed through that single keymap. This helps embedding Emacspeak functionality into a large variety of third party modules without any loss of functionality.

15.6 Emacspeak And Authoring Documents

In 1994, my preferred environment for authoring **all** documents was **LaTeX** using the Auctex package. Later I started writing either LaTeX or HTML using the appropriate support modes; today I use **org-mode** to do most of my content authoring. Personally, I have never been a fan of What You See Is What You Get (WYSIWYG) authoring tools — in my experience that places an undue burden on the author by drawing attention away from the content to focus on the final appearance. An added benefit of creating content in Emacs in the form of light-weight markup is that the content is long-lived — I can still usefully process and re-use things I have written 25 years ago.

Emacs, with Emacspeak providing audio formatting and context-specific feedback remains my environment of choice for writing all forms of content ranging from simple email messages to polished documents for print publishing. And it is worth repeating that I **never** need to focus on what the content is going to look like — that job is best left to the computer.

As an example of producing high-fidelity visual content, see this write-up on Polyhedral Geometry (`http://emacspeak.sourceforge.net/raman/publications/polyhedra/`) that I published in 2002; all of the content, including the drawings were created by me using Emacs.

15.7 Emacspeak And The Early Days Of The Web

Right around the time that I was writing version 0.01 of emacspeak, a far more significant software movement was under way — the World Wide Web was moving from the realms of academia to the mainstream world with the launch of NCSA Mosaic — and in late 1994 by the first commercial Web browser in Netscape Navigator. Emacs had always enabled integrated access to FTP archives via package *ange-ftp*; in late 1993, William Perry released Emacs-W3, a Web browser for Emacs written entirely in Emacs Lisp. W3 was one of the first large packages to be speech-enabled by Emacspeak — later it was the browser on which I implemented the first draft of the Aural CSS specification (`http://www.w3.org/TR/CSS2/aural.html`). Emacs-W3 enabled many early innovations in the context of providing non-visual access to Web content, including audio formatting and structured content navigation; in summer of 1995, Dave Raggett and I outlined a few extensions to HTML Forms, including the **label** element as a means of associating metadata with interactive form controls in HTML, and many of these ideas were prototyped in Emacs-W3 at the time. Over the years, Emacs-W3 fell behind the times — especially as the Web moved away from cleanly structured HTML to a massive soup of unmatched tags. This made parsing and error-correcting badly-formed HTML markup expensive to do in Emacs-Lisp — and performance suffered. To add to this, mainstream users moved away because Emacs' rendering engine at the time was not rich enough to provide the type of visual renderings that users had come to expect. The advent of DHTML, and JavaScript based Web Applications finally killed off Emacs-W3 as far as most Emacs users were concerned.

But Emacs-W3 went through a revival on the emacspeak audio desktop in late 1999 with the arrival of XSLT, and Daniel Veillard's excellent implementation via the **libxml2** and **libxslt** packages. With these in hand, Emacspeak was able to hand-off the bulk of HTML error correction to the **xsltproc** tool. The lack of visual fidelity didn't matter much for an eyes-free environment; so Emacs-W3 continued to be a useful tool for consuming large amounts of Web content that did not require JavaScript support.

During the last 24 months, **libxml2** has been built into Emacs; this means that you can now parse arbitrary HTML as found in the wild without incurring a performance hit. This functionality was leveraged first by package **shr** (Simple HTML Renderer) within the **gnus** package for rendering HTML email. Later, the author of **gnus** and **shr** created a new light-weight HTML viewer called **eww** that is now part of Emacs 24. With improved support for variable pitch fonts and image embedding, Emacs is once again able to provide visual renderings for a large proportion of text-heavy Web content where it becomes useful for mainstream Emacs users to view at least some Web content within Emacs; during the last year, I have added support within emacspeak to extend package **eww** (`http://emacspeak.blogspot.com/2014/05/emacspeak-eww-updates-for-complete.html`) with support for DOM filtering and quick content navigation.

15.8 Audio Formatting — Generalizing Aural CSS

A key idea in Audio System For Technical Readings (AsTeR) (`http://www.cs.cornell.edu/home/raman/aster/aster-toplevel.html`) was the use of various voice properties in combination with non-speech auditory icons to create rich aural renderings. When I implemented Emacspeak, I brought over the notion of audio formatting to all buffers in Emacs by creating a **voice-lock** module that paralleled Emacs' **font-lock** module. The visual medium is far richer in terms of available fonts and colors as compared to voice parameters available on TTS engines — consequently, it did not make sense to directly map Emacs' **face** properties to voice parameters. To aid in projecting visual formatting onto auditory space, I created property **personality** analogous to Emacs' **face** property that could be applied to content displayed in Emacs; module **voice-lock** applied that property appropriately, and the Emacspeak core handled the details of mapping personality values to the underlying TTS engine.

The values used in property **personality** were abstract, i.e., they were independent of any given speech engine. Later in the fall of 1995, I re-expressed these set of abstract voice properties in terms of Aural CSS; the work was published as a first draft toward the end of 1995, and implemented in Emacs-W3 in early 1996. Aural CSS was an appendix in the CSS-1.0 specification; later, it graduated to being its own module within CSS-2.0.

Later in 1996, all of Emacs' **voice-lock** functionality was re-implemented in terms of Aural CSS; the implementation has stood the test of time in that as I added support for more TTS engines, I was able to implement engine-specific mappings of Aural-CSS values. This meant that the rest of Emacspeak could define various types of voices for use in specific contexts without having to worry about individual TTS engines. Conceptually, property **personality** can be thought of as holding an **aural display list** — various parts of the system can annotate pieces of text with relevant properties that finally get rendered in the aggregate. This model also works well with the notion of Emacs overlays where a moving overlay is used to temporarily highlight text that has other context-specific properties applied to it.

Audio formatting as implemented in Emacspeak is extremely effective when working with all types of content ranging from richly structured mark-up documents ($\LaTeX$, org-mode) and formatted Web pages to program source code. Perceptually, switching to audio formatted output feels like switching from a black-and-white monitor to a rich color display. Today, Emacspeak's audio formatted output is the only way I can correctly write **else if** vs **elsif** in various programming languages!

15.9 Conversational Gestures For The Audio Desktop

By 1996, Emacspeak was the only piece of adaptive technology I used; in fall of 1995, I had moved to Adobe Systems from DEC Research to focus on enhancing the Portable

Document Format (PDF) to make PDF content repurposable. Between 1996 and 1998, I was primarily focused on electronic document formats — I took this opportunity to step back and evaluate what I had built as an auditory interface within Emacspeak. This retrospect proved extremely useful in gaining a sense of perspective and led to formalizing the high-level concept of *Conversational Gestures* and structured browsing/searching as a means of thinking about user interfaces.

By now, Emacspeak was a complete environment — I formalized what it provided under the moniker *Complete Audio Desktop*. The fully integrated user experience allowed me to move forward with respect to defining interaction models that were highly optimized to eyes-free interaction — as an example, see how Emacspeak interfaces with modes like **dired** (Directory Editor) for browsing and manipulating the filesystem, or **proced** (Process Editor) for browsing and manipulating running processes. Emacs' integration with **ispell** for spell checking, as well as its various completion facilities ranging from minibuffer completion to other forms of dynamic completion while typing text provided more opportunities for creating innovative forms of eyes-free interaction. With respect to what had gone before (and is still par for the course as far as traditional screen-readers are concerned), these types of highly dynamic interfaces present a challenge. For example, consider handling a completion interface using a screen-reader that is speaking the visual display. There is a significant challenge in deciding *what to speak* e.g., when presented with a list of completions, the currently typed text, and the default completion, which of these should you speak, and in what order? The problem gets harder when you consider that the underlying semantics of these items is generally not available from examining the visual presentation in a consistent manner. By having direct access to the underlying information being presented, Emacspeak had a leg up with respect to addressing the higher-level question — when you do have access to this information, how do you present it effectively in an eyes-free environment? For this and many other cases of dynamic interaction, a combination of audio formatting, auditory icons, and the ability to synthesize succinct messages from a combination of information items — rather than having to forcibly speak each item as it is rendered visually provided for highly efficient eyes-free interaction.

This was also when I stepped back to build out Emacspeak's table browsing facilities — see the online Emacspeak documentation for details on Emacspeak's table browsing functionality which continues to remain one of the richest collection of end-user affordances for working with two-dimensional data.

15.9.1 Speech-Enabling Interactive Games

So in 1997, I went the next step in asking — given access to the underlying infromation, is it possible to build effective eyes-free interaction to highly interactive tasks? I picked **Tetris** as a means of exploring this space, the result was an Emacspeak extension to speech-enable module **tetris.el**. The details of what was learned were published as a paper in Assets 98, and expanded as a chapter on Conversational Gestures in my book on Auditory Interfaces; that book was in a sense a culmination of stepping back and gaining a sense of perspective of what I had build during this period. The work on Conversational Gestures also helped in formalizing the abstract user interface layer that formed part of the XForms (`http://www.w3.org/MarkUp/Forms/`) work at the W3C.

Speech-enabling games for effective eyes-free interaction has proven highly educational. Interactive games are typically built to challenge the user, and if the eyes-free interface is inefficient, you just wont play the game — contrast this with a task that you **must** perform, where you're likely to make do with a sub-optimal interface. Over the years, Emacspeak has come to include eyes-free interfaces to several games including Tetris (`http://en.wikipedia.org/wiki/Tetris`), Sudoku (`http://en.wikipedia.org/wiki/2048_(video_game)`), and of late the popular 2048 game (`http://en.wikipedia.org/wiki/2048_(video_game)`). Each of these have in turn contributed to enhancing the interaction model in Emacspeak, and those innovations typically make their way to the rest of the environment.

15.10 Accessing Media Streams

Streaming real-time audio on the Internet became a reality with the advent of RealAudio in 1995; soon there were a large number of media streams available on the Internet ranging from music streams to live radio stations. But there was an interesting twist — for the most part, all of these media streams expected one to look at the screen, even though the primary content was purely audio (streaming video hadn't arrived yet!). Starting in 1996, Emacspeak started including a variety of eyes-free front-ends for accessing media streams. Initially, this was achieved by building a wrapper around **trplayer** — a headless version of RealPlayer; later I built Emacspeak module **emacspeak-m-player** for interfacing with package **mplayer**. A key aspect of streaming media integration in emacspeak is that one can launch and control streams without ever switching away from one's primary task; thus, you can continue to type email or edit code while seamlessly launching and controlling media streams. Over the years, Emacspeak has come to integrate with Emacs packages like **emms** as well as providing wrappers for **mplayer** and **alsaplayer** — collectively, these let you efficiently launch all types of media streams, including streaming video, without having to explicitly switch context.

In the mid-90's, Emacspeak started including a directory of media links to some of the more popular radio stations — primarily as a means of helping users getting started — Emacs' ability to rapidly complete directory and file-names turned out to be the most effective means of quickly launching everything from streaming radio stations to audio books. And even better — as the Emacs community develops better and smarter ways of navigating the filesystem using completions, e.g., package **ido**, these types of actions become even more efficient!

15.11 EBooks — Ubiquitous Access To Books

AsTeR — was motivated by the increasing availability of technical material as online electronic documents. While AsTeR processed the $\TeX$ family of markup languages, more general ebooks came in a wide range of formats, ranging from plain text generated from various underlying file formats to structured EBooks, with Project Gutenberg (`http://www.gutenberg.org/`) leading the way. During the mid-90's, I had access to a wide range of electronic materials from sources such as O'Reilly Publishing and various electronic journals — The Perl Journal (TPJ) is one that I still remember fondly.

Emacspeak provided fairly light-weight but efficient access to all of the electronic books I had on my local disk — Emacs' strengths with respect to browsing textual documents meant that I needed to build little that was specific to Emacspeak. The late 90's saw the arrival of Daisy as an XML-based format for accessible electronic books. The last decade has seen the rapid convergence to **epub** as a distribution format of choice for electronic books. Emacspeak provides interaction modes that make organizing, searching and reading these materials on the Emacspeak Audio Desktop a pleasant experience. Emacspeak also provides an OCR-Mode — this enables one to call out to an external OCR program and read the content efficiently.

The somewhat informal process used by publishers like O'Reilly to make technical material available to users with print impairments was later formalized by BookShare (`https://www.bookshare.org/`) — today, qualified users can obtain a large number of books and periodicals initially as Daisy-3 and increasingly as **EPub**. BookShare provides a RESTful API for searching and downloading books; Emacspeak module **emacspeak-bookshare** implements this API to create a client for browsing the BookShare library, downloading and organizing books locally, and an integrated ebook reading mode to round off the experience.

A useful complement to this suite of tools is the Calibre package for organizing ones ebook collection; Emacspeak now implements an **EPub Interaction** mode that leverages Calibre (actually sqlite3) to search and browse books, along with an integrated **EPub mode** for reading books.

15.12 Leveraging Computational Tools — From SQL And R To IPython Notebooks

The ability to invoke external processes and interface with them via a simple read-eval-loop (REPL) is perhaps one of Emacs' strongest extension points. This means that a wide variety of computational tools become immediately available for embedding within the Emacs environment — a facility that has been widely exploited by the Emacs community. Over the years, Emacspeak has leveraged many of these facilities to provide a well-integrated auditory interface.

Starting from a tight code, eval, test form of iterative programming as encouraged by Lisp. Applied to languages like Python and Ruby to exploratory computational tools such as R for data analysis and SQL for database interaction, the Emacspeak Audio Desktop has come to encompass a collection of rich computational tools that provide an efficient eyes-free experience.

In this context, module **ein** — Emacs IPython Notebooks — provides another excellent example of an Emacs tool that helps interface seamlessly with others in the technical domain. IPython Notebooks provide an easy means of reaching a large audience when publishing technical material with interactive computational content; module **ein** brings the power and convenience of Emacs ' editting facilities when developing the content. Speech-enabling package **ein** is a major win since editting program source code in an eyes-free environment is far smoother in Emacs than in a browser-based editor.

15.13 Social Web — EMail, Instant Messaging, Blogging And Tweeting Using Open Protocols

The ability to process large amounts of email and electronic news has always been a feature of Emacs. I started using package **vm** for email in 1990, along with **gnus** for Usenet access many years before developing Emacspeak. So these were the first major packages that Emacspeak speech-enabled. Being able to access the underlying data structures used to visually render email messages and Usenet articles enabled Emacspeak to produce rich, succinct auditory output — this vastly increased my ability to consume and organize large amounts of information. Toward the turn of the century, instant messaging arrived in the mainstream — package **tnt** provided an Emacs implementation of a chat client that could communicate with users on the then popular AOL Instant Messenger platform. At the time, I worked at IBM Research, and inspired by package **tnt**, I created an Emacs client called **ChatterBox** using the Lotus Sametime API — this enabled me to communicate with colleagues at work from the comfort of Emacs. Packages like **vm**, **gnus**, **tnt** and **ChatterBox** provide an interesting example of how availability of a clean underlying API to a specific service or content stream can encourage the creation of efficient (and different) user interfaces. The touchstone of such successful implementations is a simple test — can the user of a specific interface tell if the person whom he is communicating with is also using the same interface? In each of the examples enumerated above, a user at one end of the communication chain cannot tell, and in fact shouldn't be able to tell what client the user at the other end is using. Contrast this with closed services that have an inherent *lock-in* model e.g., proprietary word processors that use undocumented serialization formats — for a fun read, see this write-up on Universe Of Fancy Colored Paper (`http://emacspeak.sourceforge.net/publications/colored-paper.html`).

Today, my personal choice for instant messaging is the open Jabber platform. I connect to Jabber via Emacs package **emacs-jabber** and with Emacspeak providing a light-weight wrapper for generating the eyes-free interface, I can communicate seamlessly with colleagues and friends around the world.

As the Web evolved to encompass ever-increasing swathes of communication functionality that had already been available on the Internet, we saw the world move from Usenet groups to **Blogs** — I remember initially dismissing the blogging phenomenon as just a re-invention of Usenet in the early days. However, mainstream users flocked to Blogging, and I later realized that blogging as a publishing platform brought along interesting features that made communicating and publishing information **much** easier. In 2005, I joined Google; during the winter holidays that year, I implemented a light-weight client for Blogger that became the start of Emacs package **g-client** — this package provides Emacs wrappers for Google services that provide a RESTful API.

15.14 The RESTful Web — Web Wizards And URL Templates For Faster Access

Today, the Web, based on URLs and HTTP-style protocols is widely recognized as a platform in its own right. This platform emerged over time — to me, Web APIs arrived in the late 90's when I observed the following with respect to my own behavior on many popular sites:

1. I opened a Web page that took a while to load (remember, I was still using Emacs-W3),
2. I then searched through the page to find a form-field that I filled out, e.g., start and end destinations on Yahoo Maps,
3. I hit **submit**, and once again waited for a heavy-weight HTML page to load,
4. And finally, I hunted through the rendered content to find what I was looking for.

This pattern repeated across a wide-range of interactive Web sites ranging from AltaVista for search (this was pre-Google), Yahoo Maps for directions, and Amazon for product searches to name but a few. So I decided to automate away the pain by creating Emacspeak module **emacspeak-websearch** that did the following:

1. Prompt via the minibuffer for the requisite fields,
2. Consed up an HTTP GET URL,
3. Retrieved this URL,
4. And filtered out the specific portion of the HTML DOM that held the generated response.

Notice that the above implementation hard-wires the CGI parameter names used by a given Web application into the code implemented in module **emacspeak-websearch**. REST as a design pattern had not yet been recognized, leave alone formalized, and module **emacspeak-websearch** was initially criticized as being fragile.

However, over time, the CGI parameter names remained fixed — the only things that have required updating in the Emacspeak code-base are the content filtering rules that extract the response — for popular services, this has averaged about one to two times a year.

I later codified these filtering rules in terms of XPath, and also integrated XSLT-based pre-processing of incoming HTML content before it got handed off to Emacs-W3 — and yes, Emacs/Advice once again came in handy with respect to injecting XSLT pre-processing into Emacs-W3!

Later, in early 2000, I created companion module **emacspeak-url-templates** — partially inspired by Emacs' **webjump** module. URL templates in Emacspeak leveraged the recognized REST interaction pattern to provide a large collection of Web widgets that could be quickly invoked to provide rapid access to the right pieces of information on the Web.

The final icing on the cake was the arrival of RSS and Atom feeds and the consequent deep-linking into content-rich sites — this meant that Emacspeak could provide audio renderings of useful content without having to deal with complex visual navigation! While Google Reader existed, Emacspeak provided a light-weight **greader** client for managing ones feed subscriptions; with the demise of Google Reader, I implemented module **emacspeak-feeds** for organizing feeds on the Emacspeak desktop. A companion package **emacspeak-webspace** implements additional goodies including a continuously updating ticker of headlines taken from the user's collection of subscribed feeds.

15.15 Mashing It Up — Leveraging Evolving Web APIs

The next step in this evolution came with the arrival of richer Web APIs — especially ones that defined a clean client/server separation. In this respect, the world of Web APIs is a somewhat mixed bag in that many Web sites equate a Web API with a JS-based API that can be exclusively invoked from within a Web-Browser run-time. The issue with that type of API binding is that the only runtime that is supported is a full-blown Web browser; but the arrival of native mobile apps has actually proven a net positive in encouraging sites to create a cleaner separation. Emacspeak has leveraged these APIs to create Emacspeak front-ends to many useful services, here are a few:

1. Minibuffer completion for Google Search using Google Suggest to provide completions.
2. Librivox for browsing and playing free audio books.

3. NPR for browsing and playing NPR archived programs.
4. BBC for playing a wide variety of streaming content available from the BBC.
5. A Google Maps front-end that provides instantaneous access to directions and Places search.
6. Access to Twitter via package **twittering-mode**.

And a lot more than will fit this margin! This is an example of generalizing the concept of a mashup as seen on the Web with respect to creating hybrid applications by bringing together a collection of different Web APIs. Another way to think of such separation is to view an application as a **head** and a **body** — where the **head** is a specific user interface, with the **body** implementing the application logic. A cleanly defined separation between the **head** and **body** allows one to attach *different* user interfaces i.e., **heads** to the given **body** without any loss of functionality, or the need to re-implement the entire application. Modern platforms like Android enable such separation via an Intent (`http://developer.android.com/reference/android/content/Intent.html`) mechanism. The Web platform as originally defined around URLs is actually well-suited to this type of separation — though the full potential of this design pattern remains to be fully realized given today's tight association of the Web to the Web Browser.

15.16 Conclusion — Turning Twenty

In 1996, I wrote an article entitled User Interface — A Means To An End (`http://www.drdobbs.com/user-interface-a-means-to-an-end/184410453`) pointing out that the size and shape of computers were determined by the keyboard and display. This is even more true in today's world of tablets, phablets and large-sized phones — with the only difference that the keyboard has been replaced by a touch screen. The next generation in the evolution of **personal** devices is that they will become truly personal by being wearables — this once again forces a separation of the user interface peripherals from the underlying compute engine. Imagine a variety of wearables that collectively connect to ones cell phone, which itself connects to the cloud for all its computational and information needs. Such an environment is rich in possibilities for creating a wide variety of user experiences to a single underlying body of information; Eyes-Free interfaces as pioneered by systems like Emacspeak will come to play an increasingly vital role alongside visual interaction when this comes to pass.

–T.V. Raman, San Jose, CA, September 12, 2014

15.17 References

- Auditory User Interfaces (`http://emacspeak.sourceforge.net/raman/aui/aui.html`) Klewer Publishing, 1997.
- Advice An Emacs Lisp package by Hans Chalupsky (`http://www.isi.edu/~hans/`) that became part of Emacs 19.23.
- Beautiful Code (`http://emacspeak.blogspot.com/2007/07/emacspeak-and-beautiful-code.html`) An overview of the Emacspeak architecture.
- Speech-Enabled Applications (`http://emacspeak.sourceforge.net/raman/publications/chi96-emacspeak/`) Emacspeak at CHI 1996.

- EWW Emacspeak extends EWW (`http://emacspeak.blogspot.com/2014/05/emacspeak-eww-updates-for-complete.html`).
- In The Beginning Was The Command Line (`http://artlung.com/smorgasborg/C_R_Y_P_T_O_N_O_M_I_C_O_N.shtml`) By Neal Stephenson

16 Acknowledgments

Thanks.

17 Concept Index

A

AUCTeX 27
Audio Desktop 19
Audio Formatting 20

B

Browsing Structure 31

C

character echo 13
Context-Sensitive Interaction 20

D

Desktop 38
Desktop Applications 39
Desktop Navigation 20
Desktop Objects 19
Document Authoring 26
Document Creation 26

E

Emacs Keyboard Commands 365
Emacs Packages 26

F

Finding 20

I

Introduction 6

L

line echo 13

M

Messaging 38

O

Object-Oriented Desktop 19
Online Help 24

P

Personal Information Management 39
Programming 38

R

Replace 27

S

Search 27
Searching 20
Software Development 38
speech settings 13
speech system 13
Spell Check 28
Structured Navigation 20

T

TTS 13

W

Web Browsing 31
word echo 13

18 Key Index

<

<C-left> ... 308

<C-right> ... 308

<C-up> ... 241

<f1> " ... 322

<f1> , ... 314

<f1> = ... 328

<f1> \ ... 314

<f1> C-e ... 306

<f1> C-l ... 307

<f1> C-s ... 315

<f1> C-v ... 357

<f1> M ... 310

<f1> T ... 261

<f1> TAB ... 164

<fn> ! ... 255

<fn> " ... 256

<fn> # a ... 159

<fn> # l ... 159

<fn> # s ... 160

<fn> $ ... 243

<fn> % ... 247

<fn> & ... 326

<fn> ' ... 228

<fn> (... 239

<fn>) ... 239

<fn> , ... 257

<fn> 246

<fn> / ... 311

<fn> : ... 185

<fn> ; ... 187

<fn> < ... 254

<fn> <(deletechar> ... 312

<fn> <C-left> ... 309

<fn> <C-right> ... 308

<fn> <delete> ... 312

<fn> <down> ... 243

<fn> <f1> ... 307

<fn> <f11> ... 327

<fn> <insert> ... 307

<fn> <left> ... 311

<fn> <right> ... 311

<fn> <up> ... 243

<fn> = ... 246

<fn> > ... 251

<fn> ? ... 300

<fn> [... 252

<fn>] ... 252

<fn> ^ ... 141

<fn> @ ... 250

<fn> \ ... 261

<fn> | ... 249

<fn> 0 ... 254

<fn> 1 ... 254

<fn> 2 ... 254

<fn> 3 ... 254

<fn> 4 ... 254

<fn> 5 ... 254

<fn> 6 ... 254

<fn> 7 ... 254

<fn> 8 ... 253

<fn> 9 ... 253

<fn> a ... 250

<fn> A ... 80

<fn> b ... 244

<fn> B ... 245

<fn> c ... 245

<fn> C ... 306

<fn> C-@ ... 247

<fn> C-a ... 240

<fn> C-b ... 71

<fn> C-c ... 305

<fn> C-d ... 260

<fn> C-f ... 245

<fn> C-j ... 161

<fn> C-l ... 249

<fn> C-M-@ ... 257

<fn> C-M-b ... 312

<fn> C-M-l ... 252

<fn> C-M-SPC ... 257

<fn> C-n ... 251

<fn> C-o ... 213

<fn> C-p ... 254

<fn> C-q ... 86

<fn> C-s ... 56

<fn> C-SPC ... 247

<fn> C-t # ... 276

<fn> C-t , ... 269

<fn> C-t 276

<fn> C-t < ... 270

<fn> C-t <down> ... 274

<fn> C-t <left> ... 274

<fn> C-t <right> ... 273

<fn> C-t <S-tab> ... 274

<fn> C-t <up> ... 274

<fn> C-t = ... 277

<fn> C-t > ... 270

<fn> C-t a ... 275

<fn> C-t A ... 270

<fn> C-t b ... 276

<fn> C-t B ... 270

<fn> C-t c ... 276

<fn> C-t C ... 275

<fn> C-t C-b ... 274

<fn> C-t C-f ... 273

<fn> C-t C-n ... 274

<fn> C-t C-p ... 274

<fn> C-t E ... 270

<fn> C-t f ... 277
<fn> C-t g ... 276
<fn> C-t h ... 275
<fn> C-t j ... 270
<fn> C-t k ... 269
<fn> C-t M-< ... 270
<fn> C-t M-> ... 270
<fn> C-t M-l ... 277
<fn> C-t M-s ... 277
<fn> C-t n ... 274
<fn> C-t p ... 274
<fn> C-t Q ... 307
<fn> C-t r ... 277
<fn> C-t R ... 275
<fn> C-t s ... 275
<fn> C-t SPC ... 276
<fn> C-t T ... 270
<fn> C-t TAB ... 273
<fn> C-t v ... 278
<fn> C-t w ... 268
<fn> C-t x ... 268
<fn> C-u ... 140
<fn> C-w ... 258
<fn> C-y ... 305
<fn> d , ... 54
<fn> d 0 ... 52
<fn> d 1 ... 52
<fn> d 2 ... 52
<fn> d 3 ... 52
<fn> d 4 ... 52
<fn> d 5 ... 52
<fn> d 6 ... 52
<fn> d 7 ... 52
<fn> d 8 ... 52
<fn> d 9 ... 52
<fn> d a ... 49
<fn> d c ... 54
<fn> d C ... 54
<fn> d C-c ... 49
<fn> d C-d ... 48
<fn> d C-e ... 334
<fn> d C-n ... 50
<fn> d C-o ... 347
<fn> d C-s ... 50
<fn> d d ... 50
<fn> d f ... 51
<fn> d i ... 259
<fn> d k ... 260
<fn> d l ... 260
<fn> d L ... 49
<fn> d n ... 55
<fn> d N ... 51
<fn> d o ... 56
<fn> d p ... 53
<fn> d P ... 53
<fn> d r ... 53
<fn> d R ... 50
<fn> d RET ... 51
<fn> d s ... 55
<fn> d S ... 51
<fn> d SPC ... 55
<fn> d v ... 356
<fn> d V ... 56
<fn> d w ... 261
<fn> d z ... 262
<fn> DEL ... 47
<fn> g ... 109
<fn> h ... 248
<fn> i ... 278
<fn> I ... 310
<fn> j ... 161
<fn> k ... 246
<fn> l ... 248
<fn> L ... 249
<fn> m ... 251
<fn> M ... 250
<fn> M-% ... 241
<fn> M-; ... 133
<fn> M-b ... 252
<fn> M-c ... 306
<fn> M-d ... 231
<fn> M-f ... 307
<fn> M-h ... 310
<fn> M-i ... 269
<fn> M-l ... 308
<fn> M-m ... 260
<fn> M-p ... 309
<fn> M-q ... 357
<fn> M-s ... 312
<fn> M-t ... 279
<fn> M-u ... 139
<fn> M-v ... 309
<fn> M-w ... 258
<fn> M-x ... 317
<fn> n ... 255
<fn> N ... 312
<fn> o ... 85
<fn> p ... 253
<fn> P ... 253
<fn> q ... 261
<fn> r ... 255
<fn> R ... 255
<fn> RET ... 246
<fn> s ... 54
<fn> SPC ... 248
<fn> t ... 257
<fn> T ... 312
<fn> TAB ... 242
<fn> u ... 287
<fn> V ... 258
<fn> w ... 258
<fn> W ... 279
<fn> x , ... 327
<fn> x 326
<fn> x = ... 318
<fn> x | ... 328

<fn> x 0 326
<fn> x 1 326
<fn> x 2 326
<fn> x 3 326
<fn> x 4 326
<fn> x 5 326
<fn> x 6 326
<fn> x 7 326
<fn> x 8 326
<fn> x 9 325
<fn> x a 313
<fn> x c 314
<fn> x C 315
<fn> x e a 298
<fn> x e b 296
<fn> x e c 292
<fn> x e C 293
<fn> x e C-c 296
<fn> x e C-f 292
<fn> x e C-p 298
<fn> x e C-t 292
<fn> x e C-x 292
<fn> x e d 296
<fn> x e D 296
<fn> x e e 297
<fn> x e f 298
<fn> x e i 293
<fn> x e I 293
<fn> x e j 299
<fn> x e k 297
<fn> x e m 295
<fn> x e M 295
<fn> x e o 298
<fn> x e p 298
<fn> x e P 296
<fn> x e r 293
<fn> x e s 299
<fn> x e S 297
<fn> x e t 295
<fn> x e T 295
<fn> x e u 294
<fn> x e v 292
<fn> x e x 294
<fn> x e X 294
<fn> x e y 291
<fn> x e z 294
<fn> x h 320
<fn> x j 167
<fn> x o 324
<fn> x q 324
<fn> x s 307
<fn> x SPC 167
<fn> x t 310
<fn> x u 329
<fn> x v 330
<fn> x w 323
<help> " 322
<help> , 314
<help> = 328
<help> \ 314
<help> C-e 306
<help> C-l 307
<help> C-s 315
<help> C-v 357
<help> M 310
<help> T 261
<help> TAB 164
<print> 204
<S-down> 310
<S-up> 309
<silence> 244

C

C-' . 326
C-' a 317
C-' C-n 320
C-' d 94
C-' g 158
C-' h 179
C-' l 180
C-' m 330
C-' n 320
C-' o 338
C-' q 321
C-' r 349
C-' R 302
C-' s 348
C-' S 349
C-' SPC 325
C-' t 350
C-' u 350
C-' v 340
C-, , 65
C-, / 341
C-, a 139
C-, b 354
C-, c 330
C-, f 317
C-, h 204
C-, n 315
C-, o 140
C-, p 315
C-, q 321
C-, r 140
C-, s 329
C-, SPC 136
C-, t 329
C-, u 186
C-, y 187
C-. . 326
C-. a 317
C-. C-n 320
C-. d 94
C-. g 158
C-. h 179

C-. l ... 180
C-. m ... 330
C-. n ... 320
C-. o ... 338
C-. q ... 321
C-. r ... 349
C-. R ... 302
C-. s ... 348
C-. S ... 349
C-. SPC ... 325
C-. t ... 350
C-. u ... 350
C-. v ... 340
C-/ ... 207
C-; ' ... 186
C-; : ... 330
C-; ; ... 186
C-; a ... 328
C-; C-u ... 140
C-; e ... 342
C-; h ... 219
C-; j ... 327
C-; k ... 154
C-; l ... 172
C-; p ... 324
C-; s ... 325
C-; v ... 208
C-; w ... 323
C-c (... 43
C-c) ... 43
C-c , ... 339
C-c 0 ... 43
C-c 1 ... 43
C-c C-\ ... 44
C-c C-a ... 44
C-c C-c ... 43, 44
C-c C-d ... 43, 44
C-c C-f ... 42
C-c C-j ... 42
C-c C-k ... 44
C-c C-u ... 44
C-c C-w ... 44
C-c C-x C-c ... 43
C-c C-z ... 44
C-c e ... 43
C-c k ... 43
C-c o ... 42, 207
C-e ! ... 255
C-e " ... 256
C-e # a ... 159
C-e # l ... 159
C-e # s ... 160
C-e $ ... 243
C-e % ... 18, 247
C-e & ... 326
C-e ' ... 228
C-e (... 239
C-e) ... 239
C-e , ... 256
C-e 246
C-e / ... 11, 311
C-e : ... 185
C-e ; ... 187
C-e < ... 254
C-e <(deletechar> ... 312
C-e <C-left> ... 309
C-e <C-right> ... 308
C-e <delete> ... 312
C-e <down> ... 243
C-e <f1> ... 307
C-e <f11> ... 327
C-e <insert> ... 307
C-e <left> ... 311
C-e <right> ... 311
C-e <up> ... 243
C-e = ... 17, 246
C-e > ... 251
C-e ? ... 300
C-e [... 12, 252
C-e] ... 252
C-e ^ ... 141
C-e @ ... 250
C-e \ ... 261
C-e | ... 249
C-e 0 ... 253
C-e 1 ... 253
C-e 2 ... 253
C-e 3 ... 253
C-e 4 ... 253
C-e 5 ... 253
C-e 6 ... 253
C-e 7 ... 253
C-e 8 ... 253
C-e 9 ... 253
C-e a ... 17, 249
C-e A ... 80
C-e b ... 11, 244
C-e B ... 245
C-e c ... 10, 245
C-e C ... 306
C-e C-@ ... 17, 246
C-e C-a ... 240
C-e C-b ... 71
C-e C-c ... 305
C-e C-d ... 260
C-e C-f ... 245
C-e C-j ... 161
C-e C-l ... 17, 249
C-e C-M-@ ... 257
C-e C-M-b ... 312
C-e C-M-l ... 252
C-e C-M-SPC ... 257
C-e C-n ... 12, 251
C-e C-o ... 213
C-e C-p ... 12, 254
C-e C-q ... 86

C-e C-s 15, 56
C-e C-SPC 246
C-e C-t # 276
C-e C-t , ... 269
C-e C-t 276
C-e C-t < 270
C-e C-t <down> 274
C-e C-t <left> 274
C-e C-t <right> 273
C-e C-t <S-tab> 274
C-e C-t <up> 274
C-e C-t = ... 277
C-e C-t > 270
C-e C-t a 275
C-e C-t A 270
C-e C-t b 276
C-e C-t B 270
C-e C-t c 276
C-e C-t C 275
C-e C-t C-b 274
C-e C-t C-f 273
C-e C-t C-n 274
C-e C-t C-p 274
C-e C-t E 270
C-e C-t f 277
C-e C-t g 276
C-e C-t h 275
C-e C-t j 270
C-e C-t k 269
C-e C-t M-< 270
C-e C-t M-> 270
C-e C-t M-l 277
C-e C-t M-s 277
C-e C-t n 274
C-e C-t p 274
C-e C-t Q 307
C-e C-t r 277
C-e C-t R 275
C-e C-t s 275
C-e C-t SPC 276
C-e C-t T 270
C-e C-t TAB 273
C-e C-t v 277
C-e C-t w 268
C-e C-t x 268
C-e C-u 140
C-e C-w 17, 258
C-e C-y 305
C-e cap M 17
C-e cap R 11
C-e cap V 17
C-e d , ... 54
C-e d 0 52
C-e d 1 52
C-e d 2 52
C-e d 3 52
C-e d 4 52
C-e d 5 52
C-e d 6 52
C-e d 7 52
C-e d 8 52
C-e d 9 51
C-e d a 15, 49
C-e d c 14, 54
C-e d C 54
C-e d C-c 49
C-e d C-d 48
C-e d C-e 334
C-e d C-n 50
C-e d C-o 347
C-e d C-s 50
C-e d cap C 14
C-e d cap V 16
C-e d d 15, 50
C-e d DIGIT 14
C-e d f 14, 51
C-e d i 14, 259
C-e d k 13, 260
C-e d l 13, 260
C-e d L 49
C-e d m 14
C-e d n 55
C-e d N 51
C-e d o 56
C-e d p 14, 53
C-e d P 53
C-e d q 15
C-e d r 13, 53
C-e d R 50
C-e d RET 51
C-e d RETURN 16
C-e d s 14, 55
C-e d S 51
C-e d SPACE 15
C-e d SPC 55
C-e d t 16
C-e d v 356
C-e d V 56
C-e d w 13, 261
C-e d z 16, 262
C-e DEL 47
C-e DIGIT 12
C-e DOWN 11
C-e f 17
C-e g 108
C-e h 17, 248
C-e i 278
C-e I 310
C-e j 161
C-e k 17, 246
C-e l 11, 248
C-e L 249
C-e LEFT 12
C-e m 17, 251
C-e M 250
C-e M-% 241

C-e M-; 133
C-e M-b 251
C-e M-c 306
C-e M-d 231
C-e M-f 307
C-e M-h 310
C-e M-i 269
C-e M-l 307
C-e M-m 260
C-e M-p 309
C-e M-q 357
C-e M-s 312
C-e M-t 279
C-e M-u 139
C-e M-v 309
C-e M-w 258
C-e M-x 317
C-e meta C-@ 13
C-e n 11, 255
C-e N 312
C-e o 85
C-e p 11, 253
C-e P 253
C-e q 261
C-e r 11, 255
C-e R 255
C-e RET 246
C-e RIGHT 12
C-e s 15, 54
C-e SPC 17, 248
C-e t 17, 257
C-e T 312
C-e TAB 242
C-e u 286
C-e UP 11
C-e v 17
C-e V 258
C-e w 10, 258
C-e W 279
C-e x , 327
C-e x . 326
C-e x = 318
C-e x | 328
C-e x 0 325
C-e x 1 325
C-e x 2 325
C-e x 3 325
C-e x 4 325
C-e x 5 325
C-e x 6 325
C-e x 7 325
C-e x 8 325
C-e x 9 325
C-e x a 313
C-e x c 314
C-e x C 314
C-e x e a 298
C-e x e b 296
C-e x e c 292
C-e x e C 293
C-e x e C-c 296
C-e x e C-f 292
C-e x e C-p 298
C-e x e C-t 292
C-e x e C-x 292
C-e x e d 296
C-e x e D 296
C-e x e e 297
C-e x e f 298
C-e x e i 293
C-e x e I 293
C-e x e j 299
C-e x e k 297
C-e x e m 295
C-e x e M 295
C-e x e o 298
C-e x e p 297
C-e x e P 296
C-e x e r 293
C-e x e s 299
C-e x e S 297
C-e x e t 295
C-e x e T 295
C-e x e u 294
C-e x e v 292
C-e x e x 294
C-e x e X 294
C-e x e y 291
C-e x e z 294
C-e x h 320
C-e x j 167
C-e x o 324
C-e x q 324
C-e x s 307
C-e x SPC 167
C-e x t 310
C-e x u 329
C-e x v 330
C-e x w 323
C-h " 322
C-h , 314
C-h = 328
C-h \ 314
C-h C-e 306
C-h C-l 307
C-h C-s 315
C-h C-v 357
C-h M 310
C-h T 261
C-h TAB 164
C-M-y 205
C-x @ a , 65
C-x @ a / 341
C-x @ a a 139
C-x @ a b 354
C-x @ a c 330

C-x @ a f 317
C-x @ a h 204
C-x @ a n 315
C-x @ a o 140
C-x @ a p 315
C-x @ a q 321
C-x @ a r 140
C-x @ a s 329
C-x @ a SPC 136
C-x @ a t 329
C-x @ a u 186
C-x @ a y 187
C-x @ h ' 186
C-x @ h : 330
C-x @ h ; 186
C-x @ h a 328
C-x @ h C-u 140
C-x @ h e 342
C-x @ h h 219
C-x @ h j 327
C-x @ h k 154
C-x @ h l 173
C-x @ h p 324
C-x @ h s 325
C-x @ h v 208
C-x @ h w 323
C-x @ s 326
C-x @ s a 317
C-x @ s C-n 320
C-x @ s d 94
C-x @ s g 158
C-x @ s h 179
C-x @ s l 180
C-x @ s m 331
C-x @ s n 320
C-x @ s o 338
C-x @ s q 321
C-x @ s r 349
C-x @ s R 302
C-x @ s s 348
C-x @ s S 349
C-x @ s SPC 325
C-x @ s t 350
C-x @ s u 350
C-x @ s v 340
C-x r e 133
C-z b 330
C-z e 317
C-z f 320
C-z n 315
C-z p 315

E

ESC <down> 242
ESC <next> 242
ESC <prior> 242
ESC <select> 243
ESC <up> 242
ESCAPE DOWN 12
ESCAPE next 12
ESCAPE prior 12
ESCAPE UP 12

M

M-ESC : 327

S

s-' 339
s-; 337
s-m 335
s-n 206
s-SPC 205