

Bashmatic Functions Index

Table of Contents

1. Bashmatic Modules	1
2. List of Bashmatic Functions	4
2.1. Module <code>.audio</code>	4
2.2. Module <code>7z</code>	4
2.2.1. <code>7z.a</code>	4
2.2.2. <code>7z.install</code>	4
2.2.3. <code>7z.unzip</code>	4
2.2.4. <code>7z.x</code>	5
2.2.5. <code>7z.zip</code>	5
2.3. Module <code>array</code>	6
2.3.1. <code>array.eval.in-groups-of</code>	6
2.3.2. <code>array.force-range</code>	6
2.3.3. <code>array.from.command</code>	7
2.3.4. <code>array.from.command.bash</code>	7
2.3.5. <code>array.from.command.zsh</code>	7
2.3.6. <code>array.has-element</code>	8
2.3.7. <code>array.includes</code>	8
2.3.8. <code>array.includes-or-complain</code>	8
2.3.9. <code>array.includes-or-exit</code>	9
2.3.10. <code>array.join</code>	9
2.3.11. <code>array.max</code>	10
2.3.12. <code>array.min</code>	10
2.3.13. <code>array.sort</code>	10
2.3.14. <code>array.sort-numeric</code>	11
2.3.15. <code>array.to.bullet-list</code>	11
2.3.16. <code>array.to.csv</code>	11
2.3.17. <code>array.to.piped-list</code>	11
2.3.18. <code>array.uniq</code>	11
2.4. Module <code>asciidoc</code>	12
2.4.1. <code>asciidoc.rouge-themes</code>	12
2.5. Module <code>audio</code>	12
2.5.1. <code>æ.mp3</code>	12
2.5.2. <code>audio.dir.mp3-to-wav</code>	12
2.5.3. <code>audio.dir.rename-karaoke-wavs</code>	12
2.5.4. <code>audio.dir.rename-wavs</code>	13
2.5.5. <code>audio.file.frequency</code>	13
2.5.6. <code>audio.file.mp3-to-wav</code>	14
2.5.7. <code>audio.m4a.to.mp3</code>	14

2.5.8. <code>audio.m4as.to.mp3s</code>	15
2.5.9. <code>audio.make.mp3</code>	15
2.5.10. <code>audio.make.mp3.usage</code>	16
2.5.11. <code>audio.make.mp3s</code>	16
2.6. Module <code>aws</code>	17
2.6.1. <code>aws.ec2</code>	17
2.6.2. <code>aws.rds.hostname</code>	17
2.6.3. <code>aws.s3.upload</code>	18
2.7. Module <code>bashit</code>	19
2.7.1. <code>bashit-activate</code>	19
2.7.2. <code>bashit-colorscheme</code>	19
2.7.3. <code>bashit-colorschemes</code>	20
2.7.4. <code>bashit-init</code>	20
2.7.5. <code>bashit-install</code>	20
2.7.6. <code>bashit-list-prompts</code>	21
2.7.7. <code>bashit-prompt-aws</code>	21
2.7.8. <code>bashit-prompt-developer</code>	21
2.7.9. <code>bashit-prompt-gcloud</code>	21
2.7.10. <code>bashit-prompt-k8s</code>	21
2.7.11. <code>bashit-prompt-minimal</code>	22
2.7.12. <code>bashit-prompt-terraform</code>	22
2.7.13. <code>bashit-refresh</code>	22
2.8. Module <code>bashmatic</code>	22
2.8.1. <code>bashmatic.auto-update</code>	22
2.8.2. <code>bashmatic.auto-update-error</code>	23
2.8.3. <code>bashmatic.bash.exit-unless-version-four-or-later</code>	23
2.8.4. <code>bashmatic.bash.version</code>	23
2.8.5. <code>bashmatic.bash.version-four-or-later</code>	24
2.8.6. <code>bashmatic.cd-into</code>	24
2.8.7. <code>bashmatic.current-os</code>	24
2.8.8. <code>bashmatic.debug-off</code>	24
2.8.9. <code>bashmatic.debug-on</code>	24
2.8.10. <code>bashmatic.functions</code>	24
2.8.11. <code>bashmatic.functions-from</code>	25
2.8.12. <code>bashmatic.functions.output</code>	25
2.8.13. <code>bashmatic.functions.runtime</code>	25
2.8.14. <code>bashmatic.is-developer</code>	25
2.8.15. <code>bashmatic.is-installed</code>	26
2.8.16. <code>bashmatic.load-at-login</code>	26
2.8.17. <code>bashmatic.reload</code>	26
2.8.18. <code>bashmatic.reload-debug</code>	27

2.8.19. <code>bashmatic.reset.cache</code>	27
2.8.20. <code>bashmatic.setup</code>	27
2.8.21. <code>bashmatic.shell-check</code>	27
2.8.22. <code>bashmatic.source</code>	28
2.8.23. <code>bashmatic.source-dir</code>	29
2.8.24. <code>bashmatic.version</code>	30
2.9. Module <code>brew</code>	30
2.9.1. <code>brew.binary</code>	30
2.9.2. <code>brew.cache-reset</code>	30
2.9.3. <code>brew.cache-reset.delayed</code>	31
2.9.4. <code>brew.cask.all-installed</code>	31
2.9.5. <code>brew.cask.is-installed</code>	31
2.9.6. <code>brew.cask.list</code>	31
2.9.7. <code>brew.cask.tap</code>	31
2.9.8. <code>brew.install</code>	32
2.9.9. <code>brew.install.cask</code>	32
2.9.10. <code>brew.install.package</code>	33
2.9.11. <code>brew.install.packages</code>	33
2.9.12. <code>brew.is-installed</code>	34
2.9.13. <code>brew.package.all-installed</code>	34
2.9.14. <code>brew.package.available-versions</code>	34
2.9.15. <code>brew.package.is-installed</code>	34
2.9.16. <code>brew.package.link</code>	35
2.9.17. <code>brew.package.list</code>	35
2.9.18. <code>brew.reinstall.package</code>	35
2.9.19. <code>brew.reinstall.packages</code>	35
2.9.20. <code>brew.relink</code>	36
2.9.21. <code>brew.service.down</code>	36
2.9.22. <code>brew.service.restart</code>	36
2.9.23. <code>brew.service.up</code>	36
2.9.24. <code>brew.setup</code>	37
2.9.25. <code>brew.uninstall</code>	37
2.9.26. <code>brew.uninstall.package</code>	37
2.9.27. <code>brew.uninstall.packages</code>	37
2.9.28. <code>brew.upgrade</code>	38
2.9.29. <code>brew.upgrade.packages</code>	38
2.9.30. <code>cache-or-command</code>	38
2.9.31. <code>package.install</code>	39
2.9.32. <code>package.is-installed</code>	39
2.9.33. <code>package.uninstall</code>	39
2.10. Module <code>caller</code>	39

2.10.1. <code>caller.stack</code>	39
2.10.2. <code>stack.frame</code>	39
2.11. Module <code>color</code>	40
2.11.1. <code>bold</code>	40
2.11.2. <code>color.current-background</code>	40
2.11.3. <code>color.disable</code>	40
2.11.4. <code>color.enable</code>	41
2.11.5. <code>error-text</code>	44
2.11.6. <code>italic</code>	44
2.11.7. <code>red</code>	44
2.11.8. <code>reset-color</code>	44
2.11.9. <code>strikethrough</code>	44
2.11.10. <code>txt-err</code>	44
2.11.11. <code>txt-info</code>	45
2.11.12. <code>txt-warn</code>	45
2.11.13. <code>underline</code>	45
2.12. Module <code>config</code>	45
2.12.1. <code>config.dig</code>	45
2.12.2. <code>config.dig.pretty</code>	46
2.12.3. <code>config.get-file</code>	46
2.12.4. <code>config.get-format</code>	47
2.12.5. <code>config.set-file</code>	47
2.12.6. <code>config.set-format</code>	47
2.13. Module <code>db</code>	47
2.13.1. <code>db.actions.connect</code>	47
2.13.2. <code>db.actions.connections</code>	48
2.13.3. <code>db.actions.csv</code>	48
2.13.4. <code>db.actions.data-dir</code>	48
2.13.5. <code>db.actions.db-settings-pretty</code>	48
2.13.6. <code>db.actions.db-settings-toml</code>	48
2.13.7. <code>db.actions.explain</code>	49
2.13.8. <code>db.actions.explain-analyze-json</code>	49
2.13.9. <code>db.actions.explain-json</code>	49
2.13.10. <code>db.actions.list-indexes</code>	49
2.13.11. <code>db.actions.list-tables</code>	49
2.13.12. <code>db.actions.list-users</code>	49
2.13.13. <code>db.actions.pga</code>	50
2.13.14. <code>db.actions.run</code>	50
2.13.15. <code>db.actions.run-multiple</code>	50
2.13.16. <code>db.actions.set-max-query-len</code>	50
2.13.17. <code>db.actions.table-locks</code>	51

2.13.18. <code>db.actions.table-locks-query</code>	51
2.13.19. <code>db.actions.table-settings-set</code>	51
2.13.20. <code>db.actions.table-settings-show</code>	51
2.13.21. <code>db.actions.top</code>	51
2.13.22. <code>db.config.connections</code>	51
2.13.23. <code>db.config.connections-list</code>	52
2.13.24. <code>db.config.get-file</code>	52
2.13.25. <code>db.config.init</code>	52
2.13.26. <code>db.config.init-max-query-len</code>	52
2.13.27. <code>db.config.parse</code>	53
2.13.28. <code>db.config.set-file</code>	53
2.13.29. <code>db.datetime</code>	53
2.13.30. <code>db.pg.local.num-procs</code>	53
2.13.31. <code>db.postgres.version</code>	54
2.13.32. <code>db.psql.args</code>	54
2.13.33. <code>db.psql.args-data-only</code>	54
2.13.34. <code>db.psql.args.config</code>	54
2.13.35. <code>db.psql.args.localhost</code>	55
2.13.36. <code>db.psql.args.maintenance</code>	55
2.13.37. <code>db.psql.connect</code>	55
2.13.38. <code>db.psql.connect.db-settings-pretty</code>	56
2.13.39. <code>db.psql.connect.db-settings-toml</code>	56
2.13.40. <code>db.psql.connect.just-data</code>	57
2.13.41. <code>db.psql.connect.table-settings-set</code>	57
2.13.42. <code>db.psql.connect.table-settings-show</code>	57
2.13.43. <code>db.psql.db-settings</code>	57
2.13.44. <code>db.psql.explain</code>	58
2.13.45. <code>db.psql.list-indexes</code>	58
2.13.46. <code>db.psql.list-tables</code>	58
2.13.47. <code>db.psql.list-users</code>	58
2.13.48. <code>db.psql.report-error</code>	59
2.13.49. <code>db.psql.run</code>	59
2.13.50. <code>db.psql.run-multiple</code>	59
2.13.51. <code>db.psql.table-locks</code>	60
2.13.52. <code>db.psql.table-locks-query</code>	60
2.13.53. <code>db.psql.version</code>	60
2.13.54. <code>db.wait-until-db-online</code>	60
2.13.55. <code>is-quiet</code>	61
2.13.56. <code>is-verbose</code>	61
2.13.57. <code>print-cli</code>	61
2.14. Module <code>db_cli</code>	61

2.14.1. <code>db.actions.commands</code>	61
2.14.2. <code>db.cli-setup</code>	62
2.14.3. <code>db.commands-list</code>	62
2.14.4. <code>db.connections-list</code>	62
2.14.5. <code>db.examples</code>	62
2.14.6. <code>db.main</code>	63
2.14.7. <code>db.refresh.actions</code>	65
2.14.8. <code>db.usage</code>	65
2.15. Module <code>db_rails</code>	65
2.15.1. <code>db.rails.schema.checksum</code>	66
2.15.2. <code>db.rails.schema.file</code>	66
2.16. Module <code>db_top</code>	66
2.16.1. <code>db.top</code>	66
2.16.2. <code>db.top.set-refresh</code>	68
2.17. Module <code>deploy</code>	69
2.17.1. <code>deploy.slack</code>	69
2.17.2. <code>deploy.slack-ding</code>	69
2.17.3. <code>deploy.validate-vpn</code>	69
2.18. Module <code>dir</code>	70
2.18.1. <code>dir.count-slashes</code>	70
2.18.2. <code>dir.expand-dir</code>	70
2.18.3. <code>dir.is-a-dir</code>	70
2.18.4. <code>dir.rsync-from-mac</code>	70
2.18.5. <code>dir.rsync-from-mac-verbose</code>	70
2.18.6. <code>dir.rsync-to</code>	71
2.18.7. <code>dir.short-home</code>	72
2.18.8. <code>dir.with-file</code>	72
2.19. Module <code>docker</code>	72
2.19.1. <code>docker.abort-if-down</code>	72
2.19.2. <code>docker.actions.build</code>	73
2.19.3. <code>docker.actions.clean</code>	73
2.19.4. <code>docker.actions.pull</code>	73
2.19.5. <code>docker.actions.push</code>	73
2.19.6. <code>docker.actions.setup</code>	74
2.19.7. <code>docker.actions.start</code>	74
2.19.8. <code>docker.actions.stop</code>	74
2.19.9. <code>docker.actions.tag</code>	74
2.19.10. <code>docker.actions.up</code>	74
2.19.11. <code>docker.actions.update</code>	75
2.19.12. <code>docker.build.container</code>	75
2.19.13. <code>docker.containers.clean</code>	75

2.19.14. <code>docker.image.inspect</code>	75
2.19.15. <code>docker.image.rm</code>	75
2.19.16. <code>docker.images-named</code>	76
2.19.17. <code>docker.images.clean</code>	76
2.19.18. <code>docker.images.inspect</code>	76
2.19.19. <code>docker.last-version</code>	76
2.19.20. <code>docker.next-version</code>	77
2.19.21. <code>docker.set-repo</code>	77
2.20. Module <code>dropbox</code>	77
2.20.1. <code>dropbox.unignore</code>	77
2.21. Module <code>file</code>	77
2.21.1. <code>dir.find</code>	77
2.21.2. <code>dir.temp</code>	78
2.21.3. <code>file.ask.if-exists</code>	78
2.21.4. <code>file.count.lines</code>	78
2.21.5. <code>file.count.words</code>	78
2.21.6. <code>file.exists-and-newer-than</code>	79
2.21.7. <code>file.extension</code>	79
2.21.8. <code>file.extension.remove</code>	79
2.21.9. <code>file.extension.replace</code>	79
2.21.10. <code>file.find</code>	80
2.21.11. <code>file.first-is-newer-than-second</code>	80
2.21.12. <code>file.gsub</code>	80
2.21.13. <code>file.install-with-backup</code>	81
2.21.14. <code>file.last-modified-date</code>	82
2.21.15. <code>file.last-modified-millis</code>	82
2.21.16. <code>file.last-modified-year</code>	82
2.21.17. <code>file.list.filter-existing</code>	82
2.21.18. <code>file.list.filter-non-empty</code>	82
2.21.19. <code>file.normalize-files</code>	83
2.21.20. <code>file.print-normalized-name</code>	83
2.21.21. <code>file.size</code>	83
2.21.22. <code>file.size.gb</code>	84
2.21.23. <code>file.size.mb</code>	84
2.21.24. <code>file.source-if-exists</code>	84
2.21.25. <code>file.stat</code>	84
2.21.26. <code>file.strip.extension</code>	85
2.21.27. <code>file.temp</code>	85
2.21.28. <code>files.find</code>	85
2.21.29. <code>files.map</code>	86
2.21.30. <code>files.map.shell-scripts</code>	86

2.21.31. <code>ls.gb</code>	86
2.21.32. <code>ls.mb</code>	87
2.22. Module <code>file-helpers</code>	87
2.23. Module <code>files-normalize</code>	87
2.23.1. <code>files.normalize-tree</code>	87
2.24. Module <code>flatten</code>	87
2.24.1. <code>flatten-file</code>	87
2.25. Module <code>ftrace</code>	88
2.25.1. <code>ftrace-in</code>	88
2.25.2. <code>ftrace-off</code>	88
2.25.3. <code>ftrace-on</code>	88
2.25.4. <code>ftrace-out</code>	88
2.26. Module <code>gem</code>	89
2.26.1. <code>g-i</code>	89
2.26.2. <code>g-u</code>	89
2.26.3. <code>gem.cache-installed</code>	89
2.26.4. <code>gem.cache-list</code>	89
2.26.5. <code>gem.cache-refresh</code>	90
2.26.6. <code>gem.cache-reset</code>	90
2.26.7. <code>gem.changelog-generate</code>	90
2.26.8. <code>gem.clear-cache</code>	90
2.26.9. <code>gem.configure-cache</code>	91
2.26.10. <code>gem.ensure-gem-version</code>	91
2.26.11. <code>gem.gemfile.bundler-version</code>	91
2.26.12. <code>gem.gemfile.version</code>	91
2.26.13. <code>gem.global.latest-version</code>	92
2.26.14. <code>gem.global.versions</code>	92
2.26.15. <code>gem.install</code>	92
2.26.16. <code>gem.is-installed</code>	93
2.26.17. <code>gem.remote.version</code>	93
2.26.18. <code>gem.uninstall</code>	94
2.26.19. <code>gem.version</code>	94
2.27. Module <code>git</code>	94
2.27.1. <code>git.branch.current</code>	94
2.27.2. <code>git.cfg.get</code>	95
2.27.3. <code>git.cfgu</code>	95
2.27.4. <code>git.commits.last.message</code>	96
2.27.5. <code>git.commits.last.sha</code>	96
2.27.6. <code>git.config.kigster</code>	96
2.27.7. <code>git.configure-auto-updates</code>	96
2.27.8. <code>git.current-branch</code>	96

2.27.9. <code>git.generate-changelog</code>	97
2.27.10. <code>git.is-it-time-to-update</code>	97
2.27.11. <code>git.is-valid-repo</code>	97
2.27.12. <code>git.last-update-at</code>	98
2.27.13. <code>git.local-vs-remote</code>	98
2.27.14. <code>git.open.current</code>	99
2.27.15. <code>git.open.repo</code>	99
2.27.16. <code>git.parse-remote</code>	99
2.27.17. <code>git.quiet</code>	100
2.27.18. <code>git.remote</code>	100
2.27.19. <code>git.remotes</code>	100
2.27.20. <code>git.repo</code>	100
2.27.21. <code>git.repo-is-clean</code>	100
2.27.22. <code>git.repo.current</code>	101
2.27.23. <code>git.repo.latest-local-tag</code>	101
2.27.24. <code>git.repo.latest-remote-tag</code>	101
2.27.25. <code>git.repo.next-local-tag</code>	101
2.27.26. <code>git.repo.remote-to-git@</code>	102
2.27.27. <code>git.save-last-update-at</code>	102
2.27.28. <code>git.seconds-since-last-pull</code>	102
2.27.29. <code>git.squash</code>	103
2.27.30. <code>git.sync</code>	103
2.27.31. <code>git.sync-dirs</code>	103
2.27.32. <code>git.sync-remote</code>	104
2.27.33. <code>git.update-repo-if-needed</code>	104
2.27.34. <code>git.upstream</code>	104
2.28. Module <code>github</code>	105
2.28.1. <code>github.clone</code>	105
2.28.2. <code>github.org</code>	105
2.28.3. <code>github.setup</code>	105
2.28.4. <code>github.validate</code>	106
2.29. Module <code>gpg</code>	106
2.29.1. <code>gpg.install</code>	106
2.29.2. <code>gpg.install-deps</code>	106
2.29.3. <code>gpg.key-for-github</code>	107
2.29.4. <code>gpg.my-keys</code>	107
2.29.5. <code>gpg.name-and-email</code>	108
2.30. Module <code>is</code>	108
2.30.1. <code>is-validations</code>	108
2.30.2. <code>is.a-command</code>	109
2.30.3. <code>is.a-directory</code>	109

2.30.4. <code>is.a-function</code>	109
2.30.5. <code>is.a-function.invoke</code>	109
2.30.6. <code>is.a-non-empty-array</code>	109
2.30.7. <code>is.a-non-empty-file</code>	110
2.30.8. <code>is.a-script</code>	110
2.30.9. <code>is.a-variable</code>	110
2.30.10. <code>is.alias</code>	110
2.30.11. <code>is.an-empty-file</code>	111
2.30.12. <code>is.an-existing-file</code>	111
2.30.13. <code>is.an-integer</code>	111
2.30.14. <code>is.blank</code>	111
2.30.15. <code>is.command</code>	111
2.30.16. <code>is.empty</code>	111
2.30.17. <code>is.integer</code>	112
2.30.18. <code>is.missing</code>	112
2.30.19. <code>is.non.zero</code>	112
2.30.20. <code>is.not-a-blank-var</code>	112
2.30.21. <code>is.not-blank</code>	112
2.30.22. <code>is.numeric</code>	112
2.30.23. <code>is.sourced-in</code>	113
2.30.24. <code>is.variable</code>	113
2.30.25. <code>is.zero</code>	113
2.30.26. <code>unless</code>	113
2.30.27. <code>validations.add-error</code>	113
2.30.28. <code>validations.begin</code>	114
2.30.29. <code>validations.end</code>	114
2.30.30. <code>validations.print-errors</code>	114
2.30.31. <code>whenever</code>	114
2.31. Module <code>jemalloc</code>	115
2.31.1. <code>jm.check</code>	115
2.31.2. <code>jm.jemalloc.detect-loud</code>	116
2.31.3. <code>jm.jemalloc.detect-quiet</code>	116
2.31.4. <code>jm.jemalloc.stats</code>	116
2.31.5. <code>jm.ruby.detect</code>	116
2.31.6. <code>jm.ruby.report</code>	117
2.31.7. <code>jm.usage</code>	117
2.32. Module <code>json</code>	118
2.32.1. <code>json.begin-array</code>	118
2.32.2. <code>json.begin-hash</code>	118
2.32.3. <code>json.begin-key</code>	118
2.32.4. <code>json.end-array</code>	118

2.32.5. <code>json.end-hash</code>	118
2.32.6. <code>json.file-to-array</code>	119
2.33. Module <code>maths</code>	119
2.33.1. <code>maths.eval</code>	119
2.34. Module <code>memory</code>	120
2.34.1. <code>memory.bytes-to-units</code>	120
2.34.2. <code>memory.size-to-bytes</code>	120
2.35. Module <code>net</code>	121
2.35.1. <code>net.fast-scan</code>	121
2.35.2. <code>net.is-host-port-protocol-open</code>	121
2.35.3. <code>net.local-net</code>	122
2.35.4. <code>net.local-subnet</code>	122
2.36. Module <code>nvm</code>	122
2.36.1. <code>node.install.pin.version</code>	122
2.36.2. <code>nvm.activate</code>	122
2.36.3. <code>nvm.detect</code>	123
2.36.4. <code>nvm.install</code>	123
2.36.5. <code>nvm.is-valid-dir</code>	123
2.36.6. <code>nvm.load</code>	123
2.36.7. <code>nvm.use</code>	124
2.37. Module <code>openssl</code>	124
2.37.1. <code>openssl.certs.generate-csr</code>	124
2.38. Module <code>osx</code>	125
2.38.1. <code>afp.servers</code>	125
2.38.2. <code>bashmatic-set-fqdn</code>	125
2.38.3. <code>bashmatic-term</code>	125
2.38.4. <code>bashmatic-term-program</code>	125
2.38.5. <code>change-underscan</code>	125
2.38.6. <code>cookie-dump</code>	127
2.38.7. <code>http.servers</code>	127
2.38.8. <code>https.servers</code>	127
2.38.9. <code>osx.app.is-installed</code>	128
2.38.10. <code>osx.cookie-dump</code>	128
2.38.11. <code>osx.detect-cpu</code>	128
2.38.12. <code>osx.dropbox.exclude</code>	129
2.38.13. <code>osx.dropbox.exclude-pwd</code>	129
2.38.14. <code>osx.env-print</code>	129
2.38.15. <code>osx.local-servers</code>	129
2.38.16. <code>osx.ramdisk.mount</code>	130
2.38.17. <code>osx.ramdisk.unmount</code>	130
2.38.18. <code>osx.scutil-print</code>	131

2.38.19. <code>osx.set-fqdn</code>	131
2.38.20. <code>ssh.servers</code>	132
2.39. Module <code>output</code>	132
2.39.1. <code>ascii-clean</code>	132
2.39.2. <code>ascii-pipe</code>	132
2.39.3. <code>bashmatic.set-widget-width-to</code>	132
2.39.4. <code>center</code>	132
2.39.5. <code>cursor.at.x</code>	132
2.39.6. <code>cursor.at.y</code>	133
2.39.7. <code>cursor.down</code>	133
2.39.8. <code>cursor.left</code>	133
2.39.9. <code>cursor.restore</code>	133
2.39.10. <code>cursor.rewind</code>	133
2.39.11. <code>cursor.right</code>	133
2.39.12. <code>cursor.save</code>	134
2.39.13. <code>cursor.shift.x</code>	134
2.39.14. <code>cursor.up</code>	134
2.39.15. <code>duration</code>	134
2.39.16. <code>hr</code>	135
2.39.17. <code>hr.colored</code>	135
2.39.18. <code>inline.not-ok</code>	135
2.39.19. <code>inline.ok</code>	135
2.39.20. <code>left</code>	135
2.39.21. <code>left-prefix</code>	136
2.39.22. <code>not-ok</code>	136
2.39.23. <code>not-ok:</code>	136
2.39.24. <code>ok</code>	136
2.39.25. <code>ok:</code>	136
2.39.26. <code>output.color.off</code>	136
2.39.27. <code>output.color.on</code>	137
2.39.28. <code>output.constrain-screen-width</code>	137
2.39.29. <code>output.has-stdin</code>	137
2.39.30. <code>output.is-pipe</code>	137
2.39.31. <code>output.is-redirect</code>	137
2.39.32. <code>output.is-ssh</code>	138
2.39.33. <code>output.is-terminal</code>	138
2.39.34. <code>output.is-tty</code>	138
2.39.35. <code>output.print-at-x-y</code>	138
2.39.36. <code>output.reset-min-max-width</code>	138
2.39.37. <code>output.screen-height.actual</code>	139
2.39.38. <code>output.screen-width.actual</code>	139

2.39.39. <code>output.set-max-width</code>	140
2.39.40. <code>output.set-min-width</code>	140
2.39.41. <code>output.unconstrain-screen-width</code>	140
2.39.42. <code>screen-height</code>	140
2.39.43. <code>screen-width</code>	140
2.39.44. <code>screen.height</code>	140
2.39.45. <code>screen.height.actual</code>	141
2.39.46. <code>screen.width</code>	141
2.39.47. <code>screen.width.actual</code>	141
2.39.48. <code>section</code>	141
2.39.49. <code>stderr</code>	141
2.39.50. <code>stdout</code>	141
2.39.51. <code>ui.closer.kind-of-ok</code>	142
2.39.52. <code>ui.closer.kind-of-ok:</code>	142
2.39.53. <code>ui.closer.not-ok</code>	142
2.39.54. <code>ui.closer.not-ok:</code>	142
2.39.55. <code>ui.closer.ok</code>	142
2.39.56. <code>ui.closer.ok:</code>	143
2.40. Module <code>output-admonitions</code>	143
2.40.1. <code>alert</code>	143
2.40.2. <code>box-error</code>	143
2.40.3. <code>box-info</code>	143
2.40.4. <code>box-purple</code>	143
2.40.5. <code>box-success</code>	144
2.40.6. <code>box-warning</code>	144
2.40.7. <code>divider</code>	144
2.40.8. <code>divider.yellow</code>	144
2.40.9. <code>divider__</code>	144
2.40.10. <code>panel-error</code>	145
2.40.11. <code>panel-error-yellow</code>	145
2.40.12. <code>panel-green</code>	145
2.40.13. <code>panel-info</code>	145
2.40.14. <code>panel-info-blue</code>	146
2.40.15. <code>panel-info-blue-black</code>	146
2.40.16. <code>panel-info-dark</code>	146
2.40.17. <code>panel-info-yellow</code>	146
2.40.18. <code>panel-purple-red</code>	147
2.40.19. <code>panel-red-white</code>	147
2.40.20. <code>title</code>	147
2.40.21. <code>title-blue</code>	147
2.40.22. <code>title-box</code>	147

2.40.23. <code>title-green</code>	148
2.40.24. <code>title-panel</code>	148
2.40.25. <code>title-red</code>	149
2.40.26. <code>title-yellow</code>	149
2.41. Module <code>output-boxes</code>	149
2.41.1. <code>arrow-right</code>	149
2.41.2. <code>arrow.blk-on-blu</code>	149
2.41.3. <code>arrow.blk-on-cyn</code>	150
2.41.4. <code>arrow.blk-on-grn</code>	150
2.41.5. <code>arrow.blk-on-pur</code>	150
2.41.6. <code>arrow.blk-on-red</code>	150
2.41.7. <code>arrow.blk-on-wht</code>	150
2.41.8. <code>arrow.blk-on-ylw</code>	150
2.41.9. <code>arrow.wht-on-blu</code>	151
2.41.10. <code>arrow.wht-on-blu</code>	151
2.41.11. <code>arrow.wht-on-cyn</code>	151
2.41.12. <code>arrow.wht-on-grn</code>	151
2.41.13. <code>arrow.wht-on-pur</code>	151
2.41.14. <code>arrow.wht-on-red</code>	151
2.41.15. <code>arrow.wht-on-ylw</code>	152
2.41.16. <code>attention</code>	152
2.41.17. <code>box.black-on-blue</code>	152
2.41.18. <code>box.black-on-green</code>	152
2.41.19. <code>box.black-on-purple</code>	152
2.41.20. <code>box.black-on-red</code>	152
2.41.21. <code>box.black-on-yellow</code>	153
2.41.22. <code>box.blue-in-green</code>	153
2.41.23. <code>box.blue-in-yellow</code>	153
2.41.24. <code>box.green-in-cyan</code>	153
2.41.25. <code>box.green-in-green</code>	153
2.41.26. <code>box.green-in-magenta</code>	153
2.41.27. <code>box.green-in-yellow</code>	154
2.41.28. <code>box.magenta-in-blue</code>	154
2.41.29. <code>box.magenta-in-green</code>	154
2.41.30. <code>box.red-in-magenta</code>	154
2.41.31. <code>box.red-in-red</code>	154
2.41.32. <code>box.red-in-yellow</code>	154
2.41.33. <code>box.white-on-blue</code>	155
2.41.34. <code>box.white-on-green</code>	155
2.41.35. <code>box.white-on-red</code>	155
2.41.36. <code>box.yellow-in-blue</code>	155

2.41.37. <code>box.yellow-in-red</code>	155
2.41.38. <code>box.yellow-in-yellow</code>	155
2.41.39. <code>box.yellow-on-green</code>	156
2.41.40. <code>box.yellow-on-purple</code>	156
2.41.41. <code>h.black</code>	156
2.41.42. <code>h.blue</code>	156
2.41.43. <code>h.e</code>	156
2.41.44. <code>h.green</code>	156
2.41.45. <code>h.orange</code>	157
2.41.46. <code>h.orange-center</code>	157
2.41.47. <code>h.red</code>	157
2.41.48. <code>h.salmon</code>	157
2.41.49. <code>h.salmon-center</code>	157
2.41.50. <code>h.salmon-center-black</code>	157
2.41.51. <code>h.yellow</code>	158
2.41.52. <code>h.yellow-center-black</code>	158
2.41.53. <code>h1</code>	158
2.41.54. <code>h1.blue</code>	158
2.41.55. <code>h1.green</code>	158
2.41.56. <code>h1.purple</code>	158
2.41.57. <code>h1.red</code>	159
2.41.58. <code>h1.yellow</code>	159
2.41.59. <code>h1bg</code>	159
2.41.60. <code>h2</code>	159
2.41.61. <code>h2.green</code>	159
2.41.62. <code>h2bg</code>	159
2.41.63. <code>h3</code>	160
2.41.64. <code>h3.yellow</code>	160
2.41.65. <code>h3bg</code>	160
2.41.66. <code>h4</code>	160
2.41.67. <code>h4.red</code>	160
2.41.68. <code>h4bg</code>	160
2.41.69. <code>h5</code>	161
2.41.70. <code>h5bg</code>	161
2.41.71. <code>h6</code>	161
2.41.72. <code>h7</code>	161
2.41.73. <code>h8</code>	161
2.41.74. <code>hdr</code>	161
2.41.75. <code>h1.blue</code>	162
2.41.76. <code>h1.desc</code>	162
2.41.77. <code>h1.green</code>	162

2.41.78. <code>hl.subtle</code>	162
2.41.79. <code>hl.yellow</code>	162
2.41.80. <code>hl.yellow-on-black</code>	162
2.41.81. <code>hl.yellow-on-gray</code>	163
2.41.82. <code>notice</code>	163
2.41.83. <code>section.blue</code>	163
2.41.84. <code>section.cyan</code>	163
2.41.85. <code>section.gray</code>	163
2.41.86. <code>section.gray-yellow</code>	163
2.41.87. <code>section.green</code>	164
2.41.88. <code>section.purple</code>	164
2.41.89. <code>section.red</code>	164
2.41.90. <code>section.salmon</code>	164
2.41.91. <code>section.white</code>	164
2.41.92. <code>section.yellow</code>	164
2.41.93. <code>status.failed</code>	165
2.41.94. <code>status.ok</code>	165
2.42. Module <code>output-repeat-char</code>	165
2.43. Module <code>output-utils</code>	165
2.43.1. <code>abort</code>	165
2.43.2. <code>ask</code>	165
2.43.3. <code>br</code>	166
2.43.4. <code>columnize</code>	166
2.43.5. <code>dbg</code>	166
2.43.6. <code>dbg-off</code>	166
2.43.7. <code>dbg-on</code>	166
2.43.8. <code>dbgf</code>	167
2.43.9. <code>err</code>	167
2.43.10. <code>error</code>	167
2.43.11. <code>error-title</code>	168
2.43.12. <code>error:</code>	168
2.43.13. <code>fatal</code>	168
2.43.14. <code>inf</code>	168
2.43.15. <code>info</code>	168
2.43.16. <code>info-debug</code>	169
2.43.17. <code>info:</code>	169
2.43.18. <code>is-dbg</code>	169
2.43.19. <code>note</code>	169
2.43.20. <code>okay</code>	169
2.43.21. <code>puts</code>	170
2.43.22. <code>reset-color</code>	170

2.43.23. <code>reset-color</code>	170
2.43.24. <code>shutdown</code>	170
2.43.25. <code>skipping</code>	170
2.43.26. <code>success</code>	171
2.43.27. <code>warn</code>	171
2.43.28. <code>warning</code>	171
2.43.29. <code>warning:</code>	171
2.44. Module <code>package</code>	171
2.44.1. <code>package.ensure.command-available</code>	171
2.44.2. <code>package.ensure.is-installed</code>	172
2.45. Module <code>path</code>	172
2.45.1. <code>PATH.uniqify</code>	172
2.45.2. <code>path.absolute</code>	172
2.45.3. <code>path.append</code>	173
2.45.4. <code>path.dirs</code>	173
2.45.5. <code>path.dirs.delete</code>	173
2.45.6. <code>path.dirs.join</code>	174
2.45.7. <code>path.dirs.size</code>	174
2.45.8. <code>path.dirs.sort</code>	174
2.45.9. <code>path.dirs.sort-rev</code>	174
2.45.10. <code>path.dirs.uniq</code>	174
2.45.11. <code>path.mutate.append</code>	175
2.45.12. <code>path.mutate.delete</code>	175
2.45.13. <code>path.mutate.prepend</code>	175
2.45.14. <code>path.mutate.uniq</code>	175
2.45.15. <code>path.prepend</code>	175
2.45.16. <code>path.strip-slash</code>	176
2.45.17. <code>path.uniq</code>	176
2.46. Module <code>pdf</code>	176
2.46.1. <code>pdf.combine</code>	176
2.47. Module <code>pg</code>	177
2.47.1. <code>pg.is-running</code>	177
2.47.2. <code>pg.running.data-dirs</code>	177
2.47.3. <code>pg.running.server-binaries</code>	177
2.47.4. <code>pg.server-in-path.version</code>	177
2.48. Module <code>pids</code>	178
2.48.1. <code>pall</code>	178
2.48.2. <code>pid.alive</code>	178
2.48.3. <code>pid.sig</code>	178
2.48.4. <code>pid.stop</code>	179
2.48.5. <code>pid.stop-and-kill</code>	179

2.48.6. <code>pid.stop-if-listening-on-port</code>	180
2.48.7. <code>pids-with-args</code>	180
2.48.8. <code>pids.all</code>	181
2.48.9. <code>pids.for-each</code>	181
2.48.10. <code>pids.matching</code>	182
2.48.11. <code>pids.matching.regexp</code>	182
2.48.12. <code>pids.normalize.search-string</code>	183
2.48.13. <code>pids.stop</code>	183
2.48.14. <code>pids.stop-by-listen-tcp-ports</code>	184
2.48.15. <code>pstop</code>	184
2.48.16. <code>sig.is-valid</code>	184
2.48.17. <code>sig.list</code>	184
2.48.18. <code>top-most-program</code>	184
2.49. Module <code>pipe</code>	185
2.49.1. <code>pipe.extract-variables</code>	185
2.49.2. <code>pipe.remove-blank-lines</code>	185
2.49.3. <code>pipe.remove-hash-comments</code>	185
2.50. Module <code>progress-bar</code>	185
2.50.1. <code>progress.bar.auto-run</code>	185
2.50.2. <code>progress.bar.config</code>	186
2.50.3. <code>progress.bar.configure.color-green</code>	186
2.50.4. <code>progress.bar.configure.color-red</code>	186
2.50.5. <code>progress.bar.configure.color-yellow</code>	186
2.50.6. <code>progress.bar.configure.symbol-arrow</code>	187
2.50.7. <code>progress.bar.configure.symbol-bar</code>	187
2.50.8. <code>progress.bar.configure.symbol-block</code>	187
2.50.9. <code>progress.bar.configure.symbol-square</code>	187
2.50.10. <code>progress.bar.launch-and-wait</code>	187
2.51. Module <code>repositories</code>	188
2.51.1. <code>repo.rebase</code>	188
2.51.2. <code>repo.stash-and-rebase</code>	188
2.51.3. <code>repo.update</code>	188
2.51.4. <code>repos.catch-interrupt</code>	188
2.51.5. <code>repos.init-interrupt</code>	189
2.51.6. <code>repos.recursive-update</code>	189
2.51.7. <code>repos.update</code>	189
2.51.8. <code>repos.was-interrupted</code>	190
2.52. Module <code>ruby</code>	190
2.52.1. <code>bundle.gems-with-c-extensions</code>	190
2.52.2. <code>interrupted</code>	190
2.52.3. <code>ruby.aliases</code>	190

2.52.4. <code>ruby.bundle-install</code>	191
2.52.5. <code>ruby.bundler-version</code>	191
2.52.6. <code>ruby.compiled-with</code>	191
2.52.7. <code>ruby.default-gems</code>	191
2.52.8. <code>ruby.ensure-rbenv</code>	192
2.52.9. <code>ruby.ensure-rbenv-or-complain</code>	192
2.52.10. <code>ruby.full-version</code>	192
2.52.11. <code>ruby.gemfile-lock-version</code>	192
2.52.12. <code>ruby.gems</code>	193
2.52.13. <code>ruby.gems.install</code>	193
2.52.14. <code>ruby.gems.uninstall</code>	194
2.52.15. <code>ruby.handle-missing</code>	194
2.52.16. <code>ruby.init</code>	195
2.52.17. <code>ruby.install</code>	195
2.52.18. <code>ruby.install-ruby</code>	195
2.52.19. <code>ruby.install-ruby-with-deps</code>	196
2.52.20. <code>ruby.install-ruby-with-readline-and-openssl</code>	196
2.52.21. <code>ruby.install-upgrade-bundler</code>	196
2.52.22. <code>ruby.installed-gems</code>	197
2.52.23. <code>ruby.kigs-gems</code>	197
2.52.24. <code>ruby.linked-libs</code>	197
2.52.25. <code>ruby.numeric-version</code>	197
2.52.26. <code>ruby.rbenv</code>	198
2.52.27. <code>ruby.rubygems-update</code>	198
2.52.28. <code>ruby.stop</code>	198
2.52.29. <code>ruby.top-versions</code>	199
2.52.30. <code>ruby.top-versions-as-yaml</code>	199
2.52.31. <code>ruby.validate-version</code>	199
2.53. Module <code>run</code>	200
2.53.1. <code>run</code>	200
2.53.2. <code>run.ui.ask</code>	200
2.53.3. <code>run.ui.ask-user-value</code>	201
2.53.4. <code>run.ui.get-user-value</code>	201
2.53.5. <code>run.ui.press-any-key</code>	201
2.53.6. <code>run.ui.retry-command</code>	201
2.54. Module <code>runtime</code>	202
2.54.1. <code>command-spacer</code>	202
2.54.2. <code>run</code>	202
2.54.3. <code>run.add-obfuscated-var</code>	202
2.54.4. <code>run.config.detail-is-enabled</code>	203
2.54.5. <code>run.config.is-dry-run</code>	203

2.54.6. <code>run.config.verbose-is-enabled</code>	203
2.54.7. <code>run.dry-run-prefix</code>	203
2.54.8. <code>run.filter-out-sensitive-vars</code>	203
2.54.9. <code>run.inspect</code>	204
2.54.10. <code>run.inspect-variable</code>	204
2.54.11. <code>run.inspect-variables</code>	206
2.54.12. <code>run.inspect-variables-that-are</code>	206
2.54.13. <code>run.inspect-vars</code>	207
2.54.14. <code>run.inspect.set-skip-false-or-blank</code>	207
2.54.15. <code>run.obfuscate-string-value</code>	207
2.54.16. <code>run.on-error.ask-is-enabled</code>	207
2.54.17. <code>run.post-command-with-output</code>	208
2.54.18. <code>run.print-command</code>	208
2.54.19. <code>run.print-command-full-screen</code>	208
2.54.20. <code>run.print-long-command</code>	209
2.54.21. <code>run.print-obfuscated-vars</code>	209
2.54.22. <code>run.print-variable</code>	209
2.54.23. <code>run.print-variables</code>	209
2.54.24. <code>run.register-obfuscated-vars</code>	210
2.54.25. <code>run.ui.press-any-key</code>	210
2.54.26. <code>run.variables-ending-with</code>	210
2.54.27. <code>run.variables-starting-with</code>	210
2.54.28. <code>run.was-successful</code>	211
2.54.29. <code>run.with.minimum-duration</code>	211
2.54.30. <code>run.with.ruby-bundle</code>	211
2.54.31. <code>run.with.ruby-bundle-and-output</code>	211
2.54.32. <code>var.is-falsy</code>	212
2.54.33. <code>var.is-truthy</code>	212
2.55. Module <code>runtime-config</code>	212
2.55.1. <code>is.dry-run.off</code>	212
2.55.2. <code>is.dry-run.on</code>	212
2.55.3. <code>run.inspect</code>	212
2.55.4. <code>run.set-all</code>	213
2.55.5. <code>run.set-all.list</code>	213
2.55.6. <code>run.set-next</code>	213
2.55.7. <code>run.set-next.list</code>	213
2.55.8. <code>set.dry-run.off</code>	214
2.55.9. <code>set.dry-run.on</code>	214
2.56. Module <code>sedx</code>	214
2.56.1. <code>sedx</code>	214
2.56.2. <code>sedx.cache-command</code>	214

2.56.3. <code>util.ensure-gnu-sed</code>	214
2.57. Module <code>serial</code>	215
2.57.1. <code>serial.baud</code>	215
2.57.2. <code>serial.device</code>	215
2.58. Module <code>set</code>	216
2.58.1. <code>set-e-restore</code>	216
2.58.2. <code>set-e-save</code>	216
2.58.3. <code>set-e-status</code>	216
2.59. Module <code>shasum</code>	217
2.59.1. <code>sha</code>	217
2.59.2. <code>shasum.all-files</code>	217
2.59.3. <code>shasum.all-files-in-dir</code>	217
2.59.4. <code>shasum.set-algo</code>	217
2.59.5. <code>shasum.set-command</code>	217
2.59.6. <code>shasum.sha</code>	218
2.59.7. <code>shasum.sha-only</code>	218
2.59.8. <code>shasum.sha-only-stdin</code>	218
2.59.9. <code>shasum.to-hash</code>	218
2.60. Module <code>shdoc</code>	219
2.60.1. <code>gawk.install</code>	219
2.61. Module <code>shell-set</code>	219
2.61.1. <code>save-restore-x</code>	219
2.61.2. <code>save-set-x</code>	219
2.61.3. <code>shell-set.init-stack</code>	219
2.61.4. <code>shell-set.is-set</code>	220
2.61.5. <code>shell-set.pop-stack</code>	220
2.61.6. <code>shell-set.push-stack</code>	220
2.61.7. <code>shell-set.show-stack</code>	221
2.62. Module <code>ssh</code>	221
2.62.1. <code>ssh.key.copy</code>	221
2.62.2. <code>ssh.key.filenames</code>	221
2.62.3. <code>ssh.keys.generate</code>	221
2.62.4. <code>ssh.load-keys</code>	222
2.63. Module <code>subshell</code>	222
2.63.1. <code>bashmatic.detect-subshell</code>	222
2.63.2. <code>bashmatic.run-if-subshell</code>	223
2.63.3. <code>bashmatic.subshell-init</code>	223
2.63.4. <code>bashmatic.validate-sourced-in</code>	224
2.63.5. <code>bashmatic.validate-subshell</code>	224
2.64. Module <code>sym</code>	224
2.64.1. <code>decrypt.secrets</code>	224

2.64.2. <code>dev.decrypt.chef</code>	225
2.64.3. <code>dev.decrypt.file</code>	225
2.64.4. <code>dev.decrypt.str</code>	225
2.64.5. <code>dev.edit.file</code>	225
2.64.6. <code>dev.encrypt.file</code>	225
2.64.7. <code>dev.encrypt.str</code>	226
2.64.8. <code>dev.sym</code>	226
2.64.9. <code>sym.dev.configure</code>	226
2.64.10. <code>sym.dev.files</code>	226
2.64.11. <code>sym.dev.have-key</code>	226
2.64.12. <code>sym.dev.import</code>	227
2.64.13. <code>sym.dev.install-shell-helpers</code>	228
2.64.14. <code>sym.install.symit</code>	229
2.65. Module <code>text</code>	229
2.65.1. <code>text.chr</code>	229
2.65.2. <code>text.markdown-to-asciidoc</code>	229
2.65.3. <code>text.ord</code>	230
2.66. Module <code>time</code>	230
2.66.1. <code>date.now.with-time</code>	230
2.66.2. <code>date.now.with-time.and.zone</code>	231
2.66.3. <code>epoch</code>	231
2.66.4. <code>millis</code>	231
2.66.5. <code>time.a-command</code>	231
2.66.6. <code>time.date-from-epoch</code>	231
2.66.7. <code>time.duration.humanize</code>	232
2.66.8. <code>time.duration.millis-to-secs</code>	232
2.66.9. <code>time.epoch-to-iso</code>	232
2.66.10. <code>time.epoch-to-local</code>	233
2.66.11. <code>time.epoch.minutes-ago</code>	233
2.66.12. <code>time.now.db</code>	233
2.66.13. <code>time.now.file-extension</code>	233
2.66.14. <code>time.now.with-ms</code>	233
2.66.15. <code>time.with-duration</code>	234
2.66.16. <code>time.with-duration.clear</code>	235
2.66.17. <code>time.with-duration.end</code>	235
2.66.18. <code>time.with-duration.start</code>	235
2.66.19. <code>today</code>	236
2.67. Module <code>trap</code>	236
2.67.1. <code>trap-setup</code>	236
2.67.2. <code>trap-was-fired</code>	236
2.67.3. <code>trapped</code>	236

2.68. Module <code>url</code>	237
2.68.1. <code>url.cert.domain</code>	237
2.68.2. <code>url.cert.info</code>	237
2.68.3. <code>url.cert.is-valid</code>	237
2.68.4. <code>url.downloader</code>	237
2.68.5. <code>url.host.is-valid</code>	238
2.68.6. <code>url.http-code</code>	238
2.68.7. <code>url.is-valid</code>	239
2.68.8. <code>url.shorten</code>	239
2.68.9. <code>url.valid-status</code>	240
2.69. Module <code>usage</code>	240
2.69.1. <code>help-command</code>	240
2.69.2. <code>help-comment</code>	240
2.69.3. <code>help-details</code>	240
2.69.4. <code>help-example</code>	240
2.69.5. <code>help-name</code>	241
2.69.6. <code>help-section</code>	241
2.69.7. <code>usage-box</code>	241
2.69.8. <code>usage-box.section</code>	241
2.69.9. <code>usage-box.sub-section</code>	241
2.69.10. <code>usage-widget</code>	242
2.69.11. <code>usage.set-min-flag-len</code>	242
2.70. Module <code>user</code>	243
2.70.1. <code>user</code>	243
2.70.2. <code>user.current-shell</code>	243
2.70.3. <code>user.current-shell-init-file</code>	243
2.70.4. <code>user.finger.name</code>	243
2.70.5. <code>user.first</code>	243
2.70.6. <code>user.gitconfig.email</code>	244
2.70.7. <code>user.gitconfig.name</code>	244
2.70.8. <code>user.host</code>	244
2.70.9. <code>user.login-shell</code>	244
2.70.10. <code>user.login-shell-init-file</code>	244
2.70.11. <code>user.login-shell-path</code>	245
2.70.12. <code>user.my.external-ip</code>	245
2.70.13. <code>user.my.ip</code>	245
2.70.14. <code>user.my.local-ip</code>	245
2.70.15. <code>user.my.reverse-ips</code>	245
2.70.16. <code>user.pairs.email</code>	246
2.70.17. <code>user.pairs.firstname</code>	246
2.70.18. <code>user.pairs.lastname</code>	246

2.70.19. <code>user.pairs.set-file</code>	247
2.70.20. <code>user.pairs.username</code>	247
2.70.21. <code>user.username</code>	247
2.71. Module <code>util</code>	247
2.71.1. <code>is-func</code>	247
2.71.2. <code>pause</code>	247
2.71.3. <code>pause.long</code>	248
2.71.4. <code>pause.medium</code>	248
2.71.5. <code>pause.short</code>	248
2.71.6. <code>system.save-os-name</code>	248
2.71.7. <code>system.uname</code>	248
2.71.8. <code>util.append-to-init-files</code>	249
2.71.9. <code>util.arch</code>	249
2.71.10. <code>util.call-if-function</code>	250
2.71.11. <code>util.dev-setup.update</code>	250
2.71.12. <code>util.eval-function-body</code>	250
2.71.13. <code>util.functions-matching</code>	250
2.71.14. <code>util.functions-matching.diff</code>	251
2.71.15. <code>util.functions-starting-with</code>	251
2.71.16. <code>util.functions-starting-with-csv</code>	251
2.71.17. <code>util.functions-starting-with-lines</code>	251
2.71.18. <code>util.generate-password</code>	252
2.71.19. <code>util.i-to-ver</code>	252
2.71.20. <code>util.install-direnv</code>	252
2.71.21. <code>util.invoke-if-function</code>	252
2.71.22. <code>util.is-a-function</code>	253
2.71.23. <code>util.is-numeric</code>	253
2.71.24. <code>util.is-variable-defined</code>	253
2.71.25. <code>util.lines-in-folder</code>	253
2.71.26. <code>util.os</code>	253
2.71.27. <code>util.random-number</code>	253
2.71.28. <code>util.random-string.of-length</code>	254
2.71.29. <code>util.remove-from-init-files</code>	254
2.71.30. <code>util.rot13</code>	255
2.71.31. <code>util.rot13-stdin</code>	255
2.71.32. <code>util.shell-init-files</code>	255
2.71.33. <code>util.ver-to-i</code>	255
2.71.34. <code>var.expand</code>	256
2.71.35. <code>watch.command</code>	256
2.71.36. <code>watch.ls-al</code>	256
2.71.37. <code>watch.set-refresh</code>	257

2.72. Module <code>video</code>	257
2.72.1. <code>video.encode</code>	257
2.72.2. <code>video.filename.encoded</code>	258
2.72.3. <code>video.install.dependencies</code>	258
2.72.4. <code>video.make.mp4</code>	259
2.72.5. <code>video.shrink</code>	259
2.72.6. <code>video.squeeze</code>	260
2.73. Module <code>vim</code>	260
2.73.1. <code>gvim.off</code>	260
2.73.2. <code>gvim.on</code>	260
2.73.3. <code>vim.gvim-off</code>	260
2.73.4. <code>vim.gvim-on</code>	261
2.73.5. <code>vim.setup</code>	261
2.74. Module <code>yaml</code>	262
2.74.1. <code>yaml-diff</code>	262
2.74.2. <code>yaml-dump</code>	262
2.74.3. <code>yaml.diff</code>	262
2.74.4. <code>yaml.dump</code>	262
2.74.5. <code>yaml.edit</code>	263
2.74.6. <code>yaml.expand-aliases</code>	263
2.75. Module <code>yarn</code>	263
2.75.1. <code>yarn_install</code>	263
2.75.2. <code>yarn_sha</code>	264
2.76. Module <code>zsh</code>	264
2.76.1. <code>zsh.install</code>	264
2.76.2. <code>zsh.install.oh-my-zsh</code>	264
2.76.3. <code>zsh.install.plugins</code>	265
3. Copyright	266

Chapter 1. Bashmatic Modules

- [.audio](#)
- [7z](#)
- [array](#)
- [asciidoc](#)
- [audio](#)
- [aws](#)
- [bashit](#)
- [bashmatic](#)
- [brew](#)
- [caller](#)
- [color](#)
- [config](#)
- [db](#)
- [db_cli](#)
- [db_rails](#)
- [db_top](#)
- [deploy](#)
- [dir](#)
- [docker](#)
- [dropbox](#)
- [file](#)
- [file-helpers](#)
- [files-normalize](#)
- [flatten](#)
- [ftrace](#)
- [gem](#)
- [git](#)
- [github](#)
- [gpg](#)
- [is](#)
- [jemalloc](#)
- [json](#)
- [maths](#)

- [memory](#)
- [net](#)
- [nvm](#)
- [openssl](#)
- [osx](#)
- [output](#)
- [output-admonitions](#)
- [output-boxes](#)
- [output-repeat-char](#)
- [output-utils](#)
- [package](#)
- [path](#)
- [pdf](#)
- [pg](#)
- [pids](#)
- [pipe](#)
- [progress-bar](#)
- [repositories](#)
- [ruby](#)
- [run](#)
- [runtime](#)
- [runtime-config](#)
- [sedx](#)
- [serial](#)
- [set](#)
- [shasum](#)
- [shdoc](#)
- [shell-set](#)
- [ssh](#)
- [subshell](#)
- [sym](#)
- [text](#)
- [time](#)
- [trap](#)
- [url](#)

- [usage](#)
- [user](#)
- [util](#)
- [video](#)
- [vim](#)
- [yaml](#)
- [yarn](#)
- [zsh](#)

Chapter 2. List of Bashmatic Functions

2.1. Module .audio

2.2. Module 7z

2.2.1. 7z.a

```
7z.a ()  
{  
    7z.zip "$@"  
}
```

2.2.2. 7z.install

```
7z.install ()  
{  
    [[ -n $(which 7z) ]] || run "brew install p7zip"  
    [[ -n $(which 7z) ]] || {  
        error "7z is not found after installation"  
        return 1  
    }  
    return 0  
}
```

2.2.3. 7z.unzip

```
7z.unzip ()  
{  
    7z.install  
    local archive="$1"  
    [[ -f ${archive} ]] || archive="${archive}.tar.7z"  
    [[ -f ${archive} ]] || {  
        error "Neither $1 nor ${archive} were found."  
        return 1  
    }  
    info "Unpacking archive ${txtylw}${archive}${txt-info}, total of ${file.size}"  
    ${archive}" bytes."  
    run.set-next show-output-on  
    run "7za x -so ${archive} | tar xfv -"  
}
```

2.2.4. *7z.x*

```
7z.x ()  
{  
    7z.unzip "$@"  
}
```

2.2.5. *7z.zip*

```
7z.zip ()  
{  
    local folder="$1"  
    shift  
    7z.install  
    local archive="${folder}"  
    [[ -f "${folder}" || -d "${folder}" ]] && archive=$(basename "${folder}" | sed -E  
's/./-/g').tar.7z  
    [[ -f ${archive} ]] && {  
        run.set-next on-decline-return  
        run.ui.ask "File ${archive} already exists. Press Y to remove it and  
continue." || return 1  
        run "rm -f ${archive}"  
    }  
    local -a flags=  
    local -a args=  
    for arg in $@  
    do  
        if [[ ${arg:0:1} == "-" ]]; then  
            flags=(${flags[@]} "${arg}")  
        else  
            args=(${args[@]} "${arg}")  
        fi  
    done  
    printf "${bldgrn}"  
    printf "${args[*]}\n"  
    printf "${bldylw}"  
    set +e  
    local command="tar cf - ${folder} ${args[*]} | 7za a ${flags[*]} -si -bd ${  
archive}"  
    run.print-command "${command}"  
    eval "${command}"  
    local code=$?  
    printf "${clr}"  
    if [[ ${code} -eq 0 ]]; then  
        success "${archive} created."  
    else  
        error "Tar/7z Exited with code ${code}"  
        return 1  
    fi
```

```
}
```

2.3. Module `array`

2.3.1. `array.eval.in-groups-of`

```
array.eval.in-groups-of ()  
{  
    local chunk="$1"  
    shift  
    local function="$1"  
    shift  
    local -a group  
    for item in "$@"  
    do  
        index=$(( index + 1 ))  
        if [[ ${#group[@]} -eq ${chunk} ]]; then  
            ${function} "${group[@]}"  
            group=("${item}")  
        else  
            group=("${group[@]}" "${item}")  
        fi  
    done  
    if [[ ${#group[@]} -gt 0 ]]; then  
        ${function} "${group[@]}"  
    fi  
    return 0  
}
```

2.3.2. `array.force-range`

```
array.force-range ()  
{  
    local n="$1"  
    is.numeric "${n}" || {  
        error "First argument to this function must be numeric, got ${n}" 1>&2  
        return 1  
    }  
    shift  
    [[ "${#@}" -gt 0 ]] || {  
        error "Please pass additional arguments to define min/max" 1>&2  
        return 1  
    }  
    local min=$(array.min "$@")  
    local max=$(array.max "$@")  
    if [[ $n -lt $min ]]; then  
        n=${min}  
    fi  
}
```

```

else
    if [[ $n -gt ${max} ]]; then
        n=${max}
    else
        n=$n
    fi
fi
printf -- "%d" "${n}"
}

```

2.3.3. array.from.command

```

array.from.command ()
{
    local func="array.from.command.${user.current-shell}"
    is.a-function "${func}" || return 1
    ${func} "$@"
}

```

2.3.4. array.from.command.bash

```

array.from.command.bash ()
{
    local array_name="$1"
    shift
    local command="$*"
    local IFS="$IFS"
    eval "IFS=\$'\\n'; read -r -d '' -a ${array_name} < <<( bash -c \"${command}\\" || true && printf '\0' ); export ${array_name} || true"
    export IFS="$IFS"
}

```

2.3.5. array.from.command.zsh

```

array.from.command.zsh ()
{
    local array_name="$1"
    shift
    local command="$*"
    eval "declare -a ${array_name}"
    eval "${array_name}=(\"${${(@f)$(($command))}}\"); export ${array_name}; true"
    return
}

```

2.3.6. array.has-element

```
array.has-element ()  
{  
    local search="$1"  
    shift  
    local r="false"  
    local e  
    [[ "$*" =~ ${search} ]] || {  
        echo -n $r  
        return 1  
    }  
    for e in "${@}"  
    do  
        [[ "$e" == "${search}" ]] && r="true"  
    done  
    echo -n $r  
    [[ $r == "false" ]] && return 1  
    return 0  
}
```

2.3.7. array.includes

```
array.includes ()  
{  
    local search="$1"  
    [[ -z $search ]] && return 1  
    shift  
    [[ "$*" =~ ${search} ]] || return 1  
    for e in "${@}"  
    do  
        [[ "$e" == "${search}" ]] && {  
            return 0  
        }  
    done  
    return 1  
}
```

2.3.8. array.includes-or-complain

```
array.includes-or-complain ()  
{  
    array.includes "${@}" || {  
        element="$1"  
        shift  
        local -a output=()  
        while true; do
```

```

[[ -z "$1" ]] && break
if [[ "$1" =~ " " ]]; then
    output=("${output[@]} $1")
else
    output=("$1")
fi
shift
done
if [[ ${#output[@]} -gt 10 ]]; then
    error "Value ${element} must be one of the supplied values."
else
    error "Value ${element} must be one of the supplied values:" "${output[@]:0:10}"
fi
echo
return 0
}
return 1
}

```

2.3.9. array.includes-or-exit

```

array.includes-or-exit ()
{
    array.includes-or-complain "$@" || exit 1
}

```

2.3.10. array.join

```

array.join ()
{
    local sep="$1"
    shift
    local lines="$1"
    if [[ ${lines} == true || ${lines} == false ]]; then
        shift
    else
        lines=false
    fi
    local elem
    local len=$#
    local last_index=$(( len - 1 ))
    local index=0
    for elem in "$@"
    do
        if ${lines}; then
            printf "${sep}%s\n" "${elem}"
        else

```

```

        printf "%s" "${elem}"
        [[ ${index} -lt ${last_index} ]] && printf '%s' "${sep}"
    fi
    index=$(( index + 1 ))
done
}

```

2.3.11. `array.max`

```

array.max ()
{
    local max="$1"
    shift
    for v in "$@"
    do
        is.numeric "$v" || continue
        [[ ${v} -gt ${max} ]] && max="$v"
    done
    printf -- "%d" "${max}"
}

```

2.3.12. `array.min`

```

array.min ()
{
    local min="$1"
    shift
    for v in "$@"
    do
        is.numeric "$v" || continue
        [[ ${v} -lt ${min} ]] && min="$v"
    done
    printf -- "%d" "${min}"
}

```

2.3.13. `array.sort`

```

array.sort ()
{
    local IFS_previous="${IFS}"
    export IFS=' '
    printf "%s\n" "$@" | sort | tr '\n' ' ' | sed 's/ //g'
    IFS="${IFS_previous}"
}

```

2.3.14. `array.sort-numeric`

```
array.sort-numeric ()  
{  
    local IFS_previous="${IFS}"  
    export IFS=''  
    printf "%s\n" "$@" | sort -n | tr '\n' ' ' | sed 's/ $//g'  
    IFS="${IFS_previous}"  
}
```

2.3.15. `array.to.bullet-list`

```
array.to.bullet-list ()  
{  
    array.join ' ' true "$@"  
}
```

2.3.16. `array.to.csv`

```
array.to.csv ()  
{  
    array.join ', ' false "$@"  
}
```

2.3.17. `array.to.piped-list`

```
array.to.piped-list ()  
{  
    array.join ' | ' false "$@"  
}
```

2.3.18. `array.uniq`

```
array.uniq ()  
{  
    local IFS_previous="${IFS}"  
    IFS=''  
    printf "%s\n" "$@" | sort -u | tr '\n' ' ' | sed 's/ $//g'  
    IFS="${IFS_previous}"  
}
```

2.4. Module asciidoc

2.4.1. asciidoc.rouge-themes

```
asciidoc.rouge-themes ()  
{  
    gem.install rouge  
    info "Available themes:"  
    local -a themes=$(ruby -e 'require :rouge.to_s; puts  
Rouge::Theme.registry.keys.sort.join ?\n')  
    array.to.bullet-list "${themes[@]}"  
    echo  
    info "To specify a theme in your *.adoc file, put this at the top:"  
    info "${bldlyw}:source-highlighter: ${bldgrn}rouge"  
    info "${bldlyw}:rouge-style: ${bldgrn}monokai"  
}
```

2.5. Module audio

2.5.1. æ.mp3

```
æ.mp3 ()  
{  
    audio.make.mp3 "$@"  
}
```

2.5.2. audio.dir.mp3-to-wav

```
audio.dir.mp3-to-wav ()  
{  
    local from="$1"  
    local to="$2"  
    run "cd \\"${from}\\""  
    trap "return 1" INT  
    while read -d '' filename; do  
        audio.file.mp3-to-wav "${filename}" "${to}" < /dev/null  
    done <<(find . -type f -name "*.mp3" -print0)  
    run "cd -"  
}
```

2.5.3. audio.dir.rename-karaoke-wavs

```
audio.dir.rename-karaoke-wavs ()  
{
```

```

        audio.dir.rename-wavs karaoke "$@"
}

```

2.5.4. audio.dir.rename-wavs

```

audio.dir.rename-wavs () {
    local format="$1"
    shift
    local func=".audio.${format}.format"
    is.a-function "${func}" || {
        error "Format not recognized: ${format}" "usage: audio.dir.rename-wavs
<renaming-scheme> [ optional-dir ]"
        return 1
    }
    local dir="$1"
    local pwd="$(pwd -P)"
    if [[ -n "${dir}" ]]; then
        [[ ! -d "${dir}" ]] && {
            error "First argument is either blank (current directory)" "or the folder
where *.wav files to be renamed are."
            return 1
        }
        cd "${dir}" || exit
    fi
    local nfile
    for file in $(ls -1 '*.*')
    do
        n="$(.audio.karaoke.format "${file}" | sed 's/-/_/g')"
        h1 "${file}" "${n}"
        run "mv -vn \"${file}\\" \"${n}\""
    done
    run "cd \\"${pwd}\\"
}

```

2.5.5. audio.file.frequency

```

audio.file.frequency () {
    local file="$1"
    [[ -z $(command -V mdls) ]] && return 1
    local frequency=$(mdls "${file}" | grep kMDItemAudioSampleRate | sed 's/.*= //g')
    [[ -z ${frequency} ]] && frequency=48000
    local kHz=$((maths.eval "${frequency} / 1000.0" 0))
    printf "${kHz}"
}

```

2.5.6. audio.file.mp3-to-wav

```
audio.file.mp3-to-wav ()  
{  
    local from="${1/.\\/\\/}"  
    local destination="$2"  
    if [[ -z ${destination} ]]; then  
        destination=$(dirname "${from}")"  
    else  
        destination="${destination}/$(dirname "${from}")"  
    fi  
    local to="${destination}/$(basename "${from/.mp3/.wav}")"  
    if [[ ${from} =~ ".mp3" ]]; then  
        h.blue "Source:      ${from}"  
        cursor.up 1  
        h.green "Destination: ${to}"  
        [[ -f "${to}" ]] && {  
            info: "File already converted."  
            return 0  
        }  
        run "mkdir -p \\"${destination}\\""  
        run.set-next show-output-on  
        run "lame --decode \\"${from}\\" \\"${to}\\""  
    else  
        error "File ${from} is not an MP3 file."  
        return 1  
    fi  
}
```

2.5.7. audio.m4a.to.mp3

```
audio.m4a.to.mp3 ()  
{  
    local file="$1"  
    [[ -z "${file}" ]] && return 0  
    [[ -f "${file}" ]] || {  
        error "File does not exist: ${file}"  
        return 1  
    }  
    local output="${file/.m4a/.mp3}"  
    [[ -s ${output} ]] && {  
        info "${file} has already been converted... Skipping."  
        return 0  
    }  
    h1 "From: ${bldylw}${file} " "To:   ${bldblu}${output}"  
    command -v ffmpeg > /dev/null || video.install.dependencies  
    local cmd="ffmpeg -i \"${file}\" -codec:a libmp3lame -qscale:a 1 \\"${output}\\""  
    h2 "${cmd}"  
    run "${cmd}"
```

```
}
```

2.5.8. audio.m4as.to.mp3s

```
audio.m4as.to.mp3s ()  
{  
    folder="${1:-" ."}"  
    info "Converting the following files:"  
    find "${folder}" -name '*.m4a'  
    hr  
    echo  
    info "Ctrl-C to abort."  
    echo  
    find "${folder}" -name '*.m4a' -exec bash -c 'source ~/.bashmatic/init.sh;  
audio.m4a.to.mp3 "{}"' \  
}
```

2.5.9. audio.make.mp3

```
audio.make.mp3 ()  
{  
    local file="$1"  
    shift  
    local nfile="$2"  
    shift  
    set +e  
    [[ -n "$(command -V lame)" ]] || brew.package.install lame  
    local default_options=" -m s -b 320 "  
    [[ -n "${file}" ]] || {  
        audio.make.mp3.usage && return 1  
    }  
    [[ -s "${file}" ]] || {  
        error "File '${file}' does not exist."  
        audio.make.mp3.usage && return 2  
    }  
    [[ -z ${nfile} ]] && nfile="$(echo "${file}" | sedx 's/\.(wav|aiff?)$/\.mp3/g')"  
    local khz=$(audio.file.frequency "${file}")  
    h2 "'$(basename "${file}")' -> ${txtylw}${nfile}${txtgrn}, sample rate: ${khz:-  
'Unknown'}kHz"  
    info "lame ${default_options} $* '${file}' '${nfile}'"  
    run.set-next show-output-on abort-on-error  
    run "lame ${default_options} $* '${file}' '${nfile}'"  
    hr  
    success "MP3 file ${nfile} is $(file.size.mb "${nfile}")Mb"  
}
```

2.5.10. audio.make.mp3.usage

```
audio.make.mp3.usage ()  
{  
    usage-box "audio.wav-to-mp3 [ file.wav | file.aif | file.aiff ] [ file.mp3 ] @  
Convert a RAW PCM Audio to highest quality MP3" "You can pass additional flags to  
${txtylw}lame" "" "Just run ${txtylw}lame --longhelp for more info." "" "Default  
Flags: ${default_options}" ""  
}
```

2.5.11. audio.make.mp3s

```
audio.make.mp3s ()  
{  
    local dir="${1:-"."}"  
    local kHz="${2:-"48"}"  
    local first=$(find "${dir}" -type f -a \(-name "*.aif*" -o -name "*.wav"\) -print | head -1)  
    h3 "Converting WAV and AIF files to MP3 in ${txtylw}${dir}."  
    if [[ -z ${first} ]]; then  
        error "No AIFF or WAV files in the folder ${bldgrn}${dir}"  
        return 1  
    fi  
    inf "Determining audio sampling rate (will apply the same rate to all files)... "  
    kHz=$((audio.file.frequency "${first}")  
    printf "${bldgrn} - ${kHz}kHz"  
    ok:  
    SAVEIFS=$IFS  
    run.set-all show-command-on show-output-off abort-on-error  
    find "${dir}" -type f -a \(-name "*.aif*" -o -name "*.wav"\) -print0 | while  
    read -d '' file; do  
        local fn=$(ascii-clean "${file}")  
        mp3=$(echo "${file}" | sedx 's/\.(wav|aiff?)$/.\mp3/g')  
        inf "checking ${txtylw}${file} ${txt-info} ... "  
        if [[ -f "${mp3}" && -z "${FORCE}" ]]; then  
            printf "${bldgrn} OK, already converted. Use FORCE=1 to overwrite. ${clr}"  
            ok:  
            continue  
        fi  
        printf "${txtcyn} Transcoding...${clr}"  
        ui.closer.kind-of-ok:  
        inf " ${txtylw}lame --silent -m s -b 320 \"${file}\\""  
        trap _term SIGINT  
        lame --silent -m s -b 320 "${fn}" & child_pid=$!  
        wait "$child_pid"  
        code=$?  
        if [[ ${code} -ne 0 ]]; then  
            ui.closer.not-ok:  
            info "${bakred}${bldwht} ERROR: lame exited with an error code ${code}."  
        fi  
    done  
}
```

```

Aborting! "
    [[ -f "${mp3}" ]] && {
        info "NOTE: removing unfinished MP3 file ${mp3}."
        rm -f "${mp3}" 1>&2 > /dev/null
    }
    break
}
else
    ok:
fi
done
success 'All done.'
}

```

2.6. Module aws

2.6.1. aws.ec2

```

aws.ec2 ()
{
    local cmd="$1"
    local command="$cmd"
    case $command in
        list | show | ls)
            _utf_table "$(aws ec2 describe-instances --query
'Reservations[*].Instances[*].{name: Name, instance_id: InstanceId, ip_address:
PrivateIpAddress, state: State.Name}' --output table 2> /dev/null)"
            return $?
        ;;
        *)
            error "Invalid Command: ${cmd}"
            return 1
        ;;
    esac
}

```

2.6.2. aws.rds.hostname

```

aws.rds.hostname ()
{
    local name=${1}
    [[ -z $(which jq) ]] && out=$(brew.install.package jq 2> /dev/null > /dev/null)
    [[ -z $(which aws) ]] && out=$(brew.install.package awscli 2> /dev/null >
/dev/null)
    [[ -n ${name} ]] && aws rds describe-db-instances | jq '.[][].Endpoint.Address' |
sedx 's//g' | ${GrepCommand} "^\${name}\\""
    [[ -z ${name} ]] && aws rds describe-db-instances | jq '.[][].Endpoint.Address' |
sedx 's//g'
}

```

```
}
```

2.6.3. aws.s3.upload

```
aws.s3.upload ()  
{  
    local pathname="$1"  
    shift  
    local skip_file_modification="$1"  
    [[ -n ${skip_file_modification} ]] && skip_file_modification=true  
    [[ -z ${skip_file_modification} ]] && skip_file_modification=false  
    if [[ -z "${LibAws__DefaultUploadBucket}" || -z "${LibAws__DefaultUploadFolder}"  
]]; then  
    error "Required AWS S3 configuration is not defined." "Please set variables:  
${bldylw}LibAws__DefaultUploadFolder" "and ${bldylw}LibAws__DefaultUploadBucket"  
"before using this function."  
    return 1  
fi  
if [[ ! -f "${pathname}" ]]; then  
    error "Local file was not found: ${bldylw}${pathname}"  
    return 1  
fi  
local file=$(basename "${pathname}")  
local remote_file="${file}"  
local year=$((file.last-modified-year "${pathname}")  
local date=$((file.last-modified-date "${pathname}")  
[[ -z ${year} ]] && year=$((date +'%Y'))  
[[ -z ${date} ]] && date=$(today)  
${skip_file_modification} || {  
    [[ "${remote_file}" =~ "${date}" ]] && remote_file=$(echo "${remote_file}" |  
sedx "s/[_.-]?${date}[_.-]//g")  
    [[ "${remote_file}" =~ "${date}" ]] || remote_file="${date}.${remote_file}"  
}  
remote_file=$(echo "${remote_file}" | sed -E 's/ /-/g;s/--+/-/g' | tr '[A-Z]' '[a-z]')  
local remote="s3://${LibAws__DefaultUploadBucket}/${LibAws__DefaultUploadFolder}  
/${year}/${remote_file}"  
run "aws s3 cp \"${pathname}\" \"${remote}\""  
if [[ ${LibRun__LastExitCode} -eq 0 ]]; then  
    local remoteUrl="https://s3-${LibAws__DefaultRegion}.amazonaws.com/  
${LibAws__DefaultUploadBucket}/${LibAws__DefaultUploadFolder}/${year}/${remote_file}"  
    [[ -n "${LibAws__ObjectUrlFile}" ]] && echo "${remoteUrl}" >  
${LibAws__ObjectUrlFile}  
    echo  
    info "NOTE: You should now be able to access your resource at the following  
URL:"  
    hr  
    info "${bldylw}${remoteUrl}"  
    hr
```

```

    else
        error "AWS S3 upload failed with code ${LibRun__LastExitCode}"
    fi
    return "${LibRun__LastExitCode}"
}

```

2.7. Module **bashit**

2.7.1. **bashit-activate**

```

bashit-activate ()
{
    local color
    local func
    export __bashmatic_bash_it_loaded=1
    bashit-install
    bashit-init
    bashit-colorscheme dark
    bashit-prompt-minimal
    while true; do
        local arg="$1"
        shift
        [[ -z "${arg}" ]] && break
        color=$(bashit-colorschemes | grep "${arg}")
        if [[ -n "${color}" ]]; then
            bashit-colorscheme "${color}"
        else
            local func=bashit-prompt-${arg}
            if is.a.function "${func}"; then
                ${func}
            else
                error "Unrecognized argument: [${arg}] is neither a color nor prompt
type."
                return 1
            fi
        fi
    done
}

```

2.7.2. **bashit-colorscheme**

```

bashit-colorscheme ()
{
    local scheme="$1"
    if [[ -z "${scheme}" ]]; then
        if [[ "${!ITERM_PROFILE}" =~ "Light" || "${!ITERM_PROFILE}" =~ "light" ]]; then
            export scheme=light
        else
            export scheme=dark
        fi
    else
        export scheme=$scheme
    fi
}

```

```

        else
            export scheme=dark
        fi
    else
        local theme="${BASH_IT}/colorschemes/${scheme}.colorscheme.bash"
        if [[ -f ${theme} ]]; then
            source "${theme}"
        else
            error "Color theme ${scheme} does not exist." 1>&2
        fi
    fi
}

```

2.7.3. bashit-colorschemes

```

bashit-colorschemes ()
{
    find "${BASH_IT}/colorschemes" -type f -name '*.colorscheme.bash' 2> /dev/null |
    tr '\n' '\0' | xargs -0 -n1 basename | sed 's/\.\colorscheme\.\bash//g'
}

```

2.7.4. bashit-init

```

bashit-init ()
{
    export BASH_IT="${HOME}/.bash_it"
    export BASH_IT_THEME="powerline"
    export BASH_IT_THEME="powerline-multiline"
    export BASH_IT_REMOTE="${__bashmatic_bash_it_remote}"
    source "${BASH_IT}/bash_it.sh"
    export GIT_HOSTING='git@git.domain.com'
    export SHORT_HOSTNAME="$(hostname -s)"
    export SHORT_TERM_LINE=true
    export BASH_IT_P4_DISABLED=true
    export SCM=git
    export SCM_CHECK=true
}

```

2.7.5. bashit-install

```

bashit-install ()
{
    if [[ ! -d "${HOME}/.bash_it" && -n $(command -v git 2> /dev/null) ]]; then
        git clone -q "${__bashmatic_bash_it_source}" ~/.bash_it > /dev/null
    fi
    [[ -d ${HOME}/.bash_it ]] || return 1
}

```

```
}
```

2.7.6. **bashit-list-prompts**

```
bashit-list-prompts ()  
{  
    regex="^bashit-prompt-([a-zA-Z0-9]*)"  
    for f in $(set | grep -E "$regex")  
    do  
        [[ $f =~ $regex ]] && echo "${BASH_REMATCH[1]}"  
    done  
}
```

2.7.7. **bashit-prompt-aws**

```
bashit-prompt-aws ()  
{  
    powerline.prompt.git.max  
    powerline.prompt.left aws_profile scm cwd shlvl last_status  
    powerline.prompt.right clock battery user_info hostname  
}
```

2.7.8. **bashit-prompt-developer**

```
bashit-prompt-developer ()  
{  
    powerline.prompt.git.max  
    powerline.prompt.left go node ruby scm cwd shlvl last_status  
    powerline.prompt.right clock battery user_info hostname  
}
```

2.7.9. **bashit-prompt-gcloud**

```
bashit-prompt-gcloud ()  
{  
    powerline.prompt.git.max  
    powerline.prompt.left gcloud scm cwd shlvl last_status  
    powerline.prompt.right clock battery user_info hostname  
}
```

2.7.10. **bashit-prompt-k8s**

```
bashit-prompt-k8s ()
```

```
{  
    powerline.prompt.git.max  
    powerline.prompt.left k8s_context scm cwd shlvl last_status  
    powerline.prompt.right clock battery user_info hostname  
}
```

2.7.11. **bashit-prompt-minimal**

```
bashit-prompt-minimal ()  
{  
    powerline.prompt.git.min  
    powerline.prompt.left scm cwd last_status  
    powerline.prompt.right go node ruby clock battery  
}
```

2.7.12. **bashit-prompt-terraform**

```
bashit-prompt-terraform ()  
{  
    powerline.prompt.git.max  
    powerline.prompt.left terraform scm cwd shlvl last_status  
    powerline.prompt.right clock battery user_info hostname  
}
```

2.7.13. **bashit-refresh**

```
bashit-refresh ()  
{  
    bashit-activate "$@"  
}
```

2.8. Module **bashmatic**

2.8.1. **bashmatic.auto-update**

```
bashmatic.auto-update ()  
{  
    ( unset -f _direnv_hook > /dev/null 2>&1  
    [[ ${Bashmatic__Test} -eq 1 ]] && return 0  
    local pwd="$(pwd -P)"  
    cd "${BASHMATIC_HOME:=${HOME}/.bashmatic}" || exit  
    git.configure-auto-updates  
    git.repo-is-clean || {  
        output.is-ssh || {
```

```

        output.is-terminal && bashmatic.auto-update-error
        cd "${pwd}" > /dev/null || exit
        return 1
    }
}
git.update-repo-if-needed
cd "${pwd}" > /dev/null || exit )
}

```

2.8.2. `bashmatic.auto-update-error`

```

bashmatic.auto-update-error ()
{
    bashmatic.is-developer || return
    file.exists-and-newer-than "${__bashmatic_warning_notification}" 10 || return
    touch "${__bashmatic_warning_notification}"
    if [[ -f ${__bashmatic_auto_update_help_file} ]]; then
        cat "${__bashmatic_auto_update_help_file}" 1>&2
    else
        output.constrain-screen-width 60
        box.black-on-yellow "${bldwht}Warning! BASHMATIC_HOME contains local
modifications." "Automatic update is disabled until git state is clean again." | tee
-a "${__bashmatic_auto_update_help_file}" 1>&2
    fi
}

```

2.8.3. `bashmatic.bash.exit-unless-version-four-or-later`

```

bashmatic.bash.exit-unless-version-four-or-later ()
{
    bashmatic.bash.version-four-or-later || {
        error "Sorry, this functionality requires BASH version 4 or later."
        exit 1 > /dev/null
    }
}

```

2.8.4. `bashmatic.bash.version`

```

bashmatic.bash.version ()
{
    echo "${BASH_VERSION:0:1}"
}

```

2.8.5. **bashmatic.bash.version-four-or-later**

```
bashmatic.bash.version-four-or-later ()  
{  
    [[ $(bashmatic.bash.version) -gt 3 ]]  
}
```

2.8.6. **bashmatic.cd-into**

```
bashmatic.cd-into ()  
{  
    [[ -d ${BASHMATIC_HOME} ]] || return 1  
    cd "${BASHMATIC_HOME}" || exit 1  
}
```

2.8.7. **bashmatic.current-os**

```
bashmatic.current-os ()  
{  
    printf "%s" "${BASHMATIC_OS}"  
}
```

2.8.8. **bashmatic.debug-off**

```
bashmatic.debug-off ()  
{  
    unset DEBUG  
    unset BASHMATIC_DEBUG  
    unset BASHMATIC_PATH_DEBUG  
}
```

2.8.9. **bashmatic.debug-on**

```
bashmatic.debug-on ()  
{  
    export DEBUG=1  
    export BASHMATIC_DEBUG=1  
    export BASHMATIC_PATH_DEBUG=1  
}
```

2.8.10. **bashmatic.functions**

```
bashmatic.functions ()  
{  
    bashmatic.functions-from '*.sh' "$@"  
}
```

2.8.11. `bashmatic.functions-from`

```
bashmatic.functions-from ()  
{  
    local pattern="${1}"  
    [[ -n ${pattern} ]] && shift  
    [[ -z ${pattern} ]] && pattern="[a-z]*.sh"  
    cd "${BASHMATIC_HOME}/lib" > /dev/null || return 1  
    local screen_width=$(screen.width.actual)  
    if [[ -n $(echo "${pattern}" | eval "${GrepCommand} '\*$' ") || ! ${pattern} =~  
        \.sh$ ]]; then  
        pattern="${pattern}.sh"  
    fi  
    eval "${GrepCommand} '^[_a-zA-Z0-9]+.*\(\)' ${pattern}" | sedx  
    's/^lib\//.*\.sh://g' | sedx 's/^function //g' | sedx 's/\(\) *$//g'  
    /usr/bin/tr -d '()' | sedx '/^ *$/d' | eval "${GrepCommand} '^(_|.)' -v" | sort |  
    uniq | columnize "$@"  
    cd - > /dev/null || return 1  
}
```

2.8.12. `bashmatic.functions.output`

```
bashmatic.functions.output ()  
{  
    bashmatic.functions-from 'output.sh' "$@"  
}
```

2.8.13. `bashmatic.functions.runtime`

```
bashmatic.functions.runtime ()  
{  
    bashmatic.functions-from 'run*.sh' "$@"  
}
```

2.8.14. `bashmatic.is-developer`

```
bashmatic.is-developer ()  
{  
    [[ ${BASHMATIC_DEVELOPER} -eq 1 || -f ${BASHMATIC_HOME}/.envrc.local ]]
```

```
}
```

2.8.15. `bashmatic.is-installed`

```
bashmatic.is-installed ()  
{  
    export bashmatic_home="${1:-${BASHMATIC_HOME}}"  
    export bashmatic_min_functions=917  
    set +e  
    /usr/bin/env bash -c " [[ -x ${bashmatic_home}/init.sh ]] || exit 1  
    source ${bashmatic_home}/init.sh >/dev/null 2>/dev/null; declare -i total;  
    type bashmatic.functions 2>/dev/null | grep -q function || exit 2; total=  
    $(bashmatic.functions 1 | wc -l | sed -E 's/[ \t]*//g'); if [[ $total -ge  
    ${bashmatic_min_functions} ]]; then exit 0; else exit 3; fi "  
    local code=$?  
    return $code  
}
```

2.8.16. `bashmatic.load-at-login`

```
bashmatic.load-at-login ()  
{  
    local file="${1}"  
    [[ -z ${file} ]] && file="${user.login-shell-init-file}"  
    grep -q -E 'BASHMATIC_HOME' "${file}" || {  
        {  
            echo "export BASHMATIC_HOME=\${BASHMATIC_HOME:-~/bashmatic}"  
            echo '[[ -f ${BASHMATIC_HOME}/init.sh ]] && source  
${BASHMATIC_HOME}/init.sh'  
            echo 'export PATH="\${PATH}:\${BASHMATIC_HOME}/bin"'  
        } >> "${file}"  
        source "${file}"  
    }  
}
```

2.8.17. `bashmatic.reload`

```
bashmatic.reload ()  
{  
    __bashmatic.set-is-not-loaded  
    [[ -f "${BASHMATIC_HOME}/.envrc.no-debug" ]] && source "${BASHMATIC_HOME}  
/.envrc.no-debug"  
    source "${BASHMATIC_INIT}" --reload  
}
```

2.8.18. `bashmatic.reload-debug`

```
bashmatic.reload-debug ()  
{  
    __bashmatic.set-is-not-loaded  
    source "${BASHMATIC_HOME}/.envrc.debug"  
    source "${BASHMATIC_INIT}" --reload  
}
```

2.8.19. `bashmatic.reset.cache`

```
bashmatic.reset.cache ()  
{  
    unset load_cache  
    bashmatic.bash.version-four-or-later && {  
        ${GLOBAL} -A load_cache  
        load_cache=()  
    }  
    rm -f "${__bashmatic_library_last_sourced}"  
}
```

2.8.20. `bashmatic.setup`

```
bashmatic.setup ()  
{  
    [[ -z ${BashMatic__Downloader} && -n $(command -v curl) ]] && export  
    BashMatic__Downloader="curl -fsSL --connect-timeout 5"  
    [[ -z ${BashMatic__Downloader} && -n $(command -v wget) ]] && export  
    BashMatic__Downloader="wget -q -O --connect-timeout=5 -"  
    if [[ ! -d "${BASHMATIC_LIB}" ]]; then  
        .err "Unable to file BashMatic's library source folder - ${BASHMATIC_LIB}"  
        return 1  
    fi  
    declare -a preload_modules=(time.sh output.sh output-utils.sh output-repeat-  
    char.sh output-boxes.sh is.sh user.sh util.sh git.sh file.sh color.sh brew.sh)  
    bashmatic.source "${preload_modules[@]}"  
    bashmatic.shell-check || return 1  
    bashmatic.source-dir "${BASHMATIC_LIB}"  
    output.unconstrain-screen-width  
    [[ -d ${BASHMATIC_HOME}/.git ]] && bashmatic.auto-update 1>&2 2> /dev/null  
    return 0  
}
```

2.8.21. `bashmatic.shell-check`

```
bashmatic.shell-check ()
```

```

{
    local shell="$(user.current-shell)"
    if [[ "${shell}" =~ bash$ || "${shell}" =~ zsh$ ]]; then
        return 0
    else
        cat "${BASHMATIC_HOME}/.init.sh" 1>&2
        return 120
    fi
}

```

2.8.22. `bashmatic.source`

```

bashmatic.source () {
    local __path="${BASHMATIC_LIB}"
    local file
    local total=0
    local files=0
    local last_loaded_at=0
    [[ -f "${__bashmatic_library_last_sourced}" ]] && last_loaded_at=$(cat "${__bashmatic_library_last_sourced}")
    for file in "${@}"
    do
        local t1=$(millis)
        [[ "${file}" =~ "/" ]] || file="${__path}/${file}"
        bashmatic.bash.version-four-or-later && {
            local cached_at=${load_cache["${file}"]}
            cached_at=${cached_at:-0}
            local modified_at="$(file.last-modified-millis "${file}")"
            [[ ${modified_at} -le ${cached_at} && ${modified_at} -le ${last_loaded_at} ]]
        } && {
            is-debug && printf -- "${bldred} (cached) ${txtgrn} \u2022 %s${clr}\n" "${file/*//}"
            continue
        }
    }
    [[ -s "${file}" ]] || {
        .err "Can't source file ${file} – file is invalid."
        return 1
    }
    if [[ -n ${SOURCE_DEBUG} || ${DEBUG} -eq 1 ]]; then
        is-debug && printf -- " ${txtylw} \u2022 %s${clr}" "${file/*//}"
        source "${file}" > /dev/null
        is-debug && {
            cursor.rewind -120
            local code=$?
            local t2=$(millis)
            local duration=$(( t2 - t1 ))
            total=$(( total + duration ))
        }
    }
}

```

```

        files=$(( files + 1 ))
    }
bashmatic.bash.version-four-or-later && {
    ((code)) || load_cache[$file]=${t1}
}
is-debug && {
    local color=${txtblu}
    [[ ${duration} -gt 20 ]] && color="${bldred}"
    printf "${color}${duration}ms [%3d] ${code}"
    printf "\n"
    unset t1
    unset t2
}
else
    source "${file}"
fi
done
bashmatic.bash.version-four-or-later && {
    [[ ${#load_cache[@]} -gt 0 ]] && millis > "${__bashmatic_library_last_sourced}"
}
is-debug && printf "${files} sourced in, taking ${total}ms total.\n"
return 0
}

```

2.8.23. bashmatic.source-dir

```

bashmatic.source-dir ()
{
    local folder="${1}"
    local loaded=false
    local file
    unset files
    declare -a files
    eval "$(files.map.shell-scripts "${folder}" files)"
    if [[ ${#files[@]} -eq 0 ]]; then
        .err "No files were returned from files.map in \"\n    ${bldylw}${folder}\n"
        return 1
    fi
    local -a sources=()
    for file in "${files[@]}"
    do
        local n="$(basename "${file}")"
        [[ ${n:0:1} == . ]] && continue
        sources+=("${file}")
    done
    bashmatic.source "${sources[@]}"
    loaded=true
    unset files
}

```

```

${loaded} || {
    .err "Unable to find BashMatic library folder with files:" "${BASHMATIC_LIB}"
    return 1
}
if [[ ${LoadedShown} -eq 0 ]]; then
    hr
    success "BashMatic was loaded! Happy Bashing :)"
    hr
    export LoadedShown=1
fi
}

```

2.8.24. `bashmatic.version`

```

bashmatic.version ()
{
    cat "$(dirname "${BASHMATIC_INIT}")/.version"
}

```

2.9. Module `brew`

2.9.1. `brew.binary`

```

brew.binary ()
{
    brew.is-installed && command -v brew
}

```

2.9.2. `brew.cache-reset`

```

brew.cache-reset ()
{
    if [[ "$1" == "cask" ]]; then
        rm -f "${LibBrew__CaskCacheList}"
    else
        if [[ "$2" == "package" ]]; then
            rm -f "${LibBrew__PackageCacheList}"
        else
            rm -f "${LibBrew__PackageCacheList}" "${LibBrew__CaskCacheList}"
        fi
    fi
}

```

2.9.3. brew.cache-reset.delayed

```
brew.cache-reset.delayed ()  
{  
    ((BASH_IN_SUBSHELL)) || brew.cache-reset both  
}
```

2.9.4. brew.cask.all-installed

```
brew.cask.all-installed ()  
{  
    local -a installed_casks=$(brew.cask.list)  
    for item in "$@"  
    do  
        array.includes "${item}" "${installed_casks[@]}" || return 1  
    done  
    return 0  
}
```

2.9.5. brew.cask.is-installed

```
brew.cask.is-installed ()  
{  
    if brew.cask.all-installed "$@"; then  
        echo "true"  
    else  
        echo "false"  
    fi  
}
```

2.9.6. brew.cask.list

```
brew.cask.list ()  
{  
    cache-or-command "${LibBrew__CaskCacheList}" 10 --cask -1  
}
```

2.9.7. brew.cask.tap

```
brew.cask.tap ()  
{  
    run "brew tap homebrew/cask-cask"  
}
```

2.9.8. brew.install

```
brew.install ()  
{  
    inf "Checking if a local brew command exists already ..." 1>&2  
    if brew.is-installed; then  
        ok:  
        local brew=$(brew.binary)  
        info "Excellent: an existing Homebrew Version: ${bldylw}$(${brew} --version 2>  
/dev/null | head -1) exists"  
        run "${brew} update"  
    else  
        not-ok:  
        info "Brew wasn't found – installing Homebrew, ${bldgrn}please wait..."  
        hl.yellow "Please enter your SUDO password, if prompted:"  
        sudo echo  
        run "/bin/bash -c \"\$curl -fsSL  
https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh\""  
    fi  
}
```

2.9.9. brew.install.cask

```
brew.install.cask ()  
{  
    local cask=$1  
    local force=  
    local verbose=  
    [[ -n "${opts_force}" ]] && force="--force"  
    [[ -n "${opts_verbose}" ]] && verbose="--verbose"  
    local installed_app=$(osx.app.is-installed "${cask}")  
    inf "checking if cask is installed: ${bldylw}${cask}"  
    brew.cask.all-installed "${cask}" && {  
        ok:  
        return 0  
    }  
    if [[ -n "${installed_app}" && -z "${opts_force}" ]]; then  
        ui.closer.ok:  
        return 0  
    else  
        ui.closer.kind-of-ok:  
        run "brew install --cask ${cask} ${force} ${verbose}"  
        brew.cache-reset cask  
    fi  
}
```

2.9.10. brew.install.package

```
brew.install.package ()  
{  
    local package="$1"  
    local force=  
    local verbose=  
    local code  
    [[ -n "${opts_force}" ]] && force="--force"  
    [[ -n "${opts_verbose}" ]] && verbose="--verbose"  
    [[ -z "${opt_terse}" ]] && inf "brewing  ${bldylw}${package}..."  
    if brew.package.all-installed "${package}"; then  
        [[ -z "${opt_terse}" ]] && ok:  
        [[ -z "${opt_terse}" ]] || printf "${bldgrn} "  
        export LibRun__LastExitCode=0  
    else  
        if [[ -z "${opt_terse}" ]]; then  
            ui.closer.kind-of-ok:  
            run "brew install ${force} ${verbose} ${package}"  
            code="${LibRun__LastExitCode}"  
        else  
            brew install ${force} ${verbose} "${package}" > /dev/null 2>&1  
            code=$?  
        fi  
        brew.cache-reset package  
        brew.package.all-installed "${package}" && code=0  
        [[ -n ${force} ]] && {  
            run.set-next continue-on-error  
            run "brew link --force --overwrite ${verbose} ${package}"  
        }  
        hash -r > /dev/null  
        ((code)) && {  
            warning "Reinstalling ${package} as I couldn't find it after instal..."  
            brew.reinstall.package "${package}"  
        }  
        export LibRun__LastExitCode=0  
        if [[ "$(brew.package.is-installed "${package}")" == "true" ]]; then  
            [[ -n "${opt_terse}" ]] && printf "\n "  
        else  
            [[ -n "${opt_terse}" ]] && printf "\n "  
            export LibRun__LastExitCode=1  
        fi  
    fi  
    return ${LibRun__LastExitCode}  
}
```

2.9.11. brew.install.packages

```
brew.install.packages ()
```

```
{
    local force=
    [[ -n "${opts_force}" ]] && force="--force"
    for package in "$@"
    do
        brew.install.package "${package}"
    done
}
```

2.9.12. `brew.is-installed`

```
brew.is-installed ()
{
    command -v brew 2>&1 > /dev/null && return 0
    return 1
}
```

2.9.13. `brew.package.all-installed`

```
brew.package.all-installed ()
{
    local -a installed_packages=$(brew.package.list)
    for item in "$@"
    do
        array.includes "${item}" "${installed_packages[@]}" || return 1
    done
    return 0
}
```

2.9.14. `brew.package.available-versions`

```
brew.package.available-versions ()
{
    local package="$1"
    [[ -z "$1" ]] && return 1
    brew search "${package}@" | tr -d 'a-zA-Z =>--' | sed '/^$/d' | sort -nr | tr
    '\n' ' '
}
```

2.9.15. `brew.package.is-installed`

```
brew.package.is-installed ()
{
    if brew.package.all-installed "$@"; then
        echo "true"
    fi
}
```

```
    else
        echo "false"
    fi
}
```

2.9.16. `brew.package.link`

```
brew.package.link ()
{
    local package="${1}"
    shift
    [[ -n "${opts_verbose}" ]] && verbose="--verbose"
    run "brew link --force --overwrite ${verbose} ${package} $*"
}
```

2.9.17. `brew.package.list`

```
brew.package.list ()
{
    cache-or-command "${LibBrew__PackageCacheList}" 10 --formula -1
}
```

2.9.18. `brew.reinstall.package`

```
brew.reinstall.package ()
{
    local package="${1}"
    local force=
    local verbose=
    [[ -n "${opts_force}" ]] && force="--force"
    [[ -n "${opts_verbose}" ]] && verbose="--verbose"
    run "brew unlink --quiet ${package}"
    run "brew uninstall ${force} ${verbose} ${package}"
    brew.install.package "${package}"
}
```

2.9.19. `brew.reinstall.packages`

```
brew.reinstall.packages ()
{
    local force=
    local result=0
    [[ -n "${opts_force}" ]] && force="--force"
    for package in "$@"
    do
```

```
    brew.uninstall.package "${package}"
    brew.install.package "${package}"
    local result=$?
done
return ${result}
}
```

2.9.20. `brew.relink`

```
brew.relink () {
  local package="${1}"
  local verbose=
  [[ -n "${opts_verbose}" ]] && verbose="--verbose"
  run "brew unlink --quiet ${package}"
  run "brew link --force --overwrite ${verbose} ${package}"
}
```

2.9.21. `brew.service.down`

```
brew.service.down () {
  local svc="$1"
  run "brew services stop ${svc}"
}
```

2.9.22. `brew.service.restart`

```
brew.service.restart () {
  local svc="$1"
  run "brew services restart ${svc}"
}
```

2.9.23. `brew.service.up`

```
brew.service.up () {
  local svc="$1"
  run "brew services start ${svc}"
}
```

2.9.24. brew.setup

```
brew.setup ()  
{  
    brew.upgrade  
}
```

2.9.25. brew.uninstall

```
brew.uninstall ()  
{  
    run.set-next show-output-on  
    run "$(brew.binary) install --cask uninstallpkg"  
}
```

2.9.26. brew.uninstall.package

```
brew.uninstall.package ()  
{  
    local package=$1  
    local force=  
    local verbose=  
    [[ -n "${opts_force}" ]] && force="--force"  
    [[ -n "${opts_verbose}" ]] && verbose="--verbose"  
    run.set-next continue-on-error  
    run "brew unlink ${package} ${force} ${verbose}"  
    run.set-next continue-on-error  
    run "brew uninstall ${package} ${force} ${verbose}"  
    brew.cache-reset.delayed  
}
```

2.9.27. brew.uninstall.packages

```
brew.uninstall.packages ()  
{  
    local force=  
    [[ -n "${opts_force}" ]] && force="--force"  
    for package in "$@"  
    do  
        brew.uninstall.package "${package}"  
    done  
}
```

2.9.28. brew.upgrade

```
brew.upgrade ()  
{  
    brew.is-installed || brew.install  
    local brew_cmd=$(command -v brew 2> /dev/null)  
    if [[ -z ${brew_cmd} ]]; then  
        warn "brew is not installed...., brew_command is blank..."  
        return 1  
    fi  
    run "${brew_cmd} update --force"  
    run "${brew_cmd} upgrade"  
    run "${brew_cmd} cleanup -s"  
}
```

2.9.29. brew.upgrade.packages

```
brew.upgrade.packages ()  
{  
    brew.is-installed || brew.install  
    [[ -z $1 ]] && {  
        error "usage: brew.upgrade.packages package1 package2 ..."  
        return 1  
    }  
    run "brew upgrade $*"  
}
```

2.9.30. cache-or-command

```
cache-or-command ()  
{  
    local file="$1"  
    shift  
    local stale_minutes="$1"  
    shift  
    if file.exists-and-newer-than "${file}" "${stale_minutes}"; then  
        if [[ -s "${file}" ]]; then  
            cat "${file}"  
            return 0  
        fi  
    fi  
    is-dbg && info "REFRESHING CACHE with command: ${bldylw}brew list ${*} >${file}"  
    brew list "$@" > "${file}"  
    cat "${file}"  
}
```

2.9.31. package.install

```
package.install ()  
{  
    brew.install.packages "$@"  
    hash -r 2> /dev/null  
}
```

2.9.32. package.is-installed

```
package.is-installed ()  
{  
    [[ "$(brew.package.is-installed "$@")" == "true" ]]  
}
```

2.9.33. package.uninstall

```
package.uninstall ()  
{  
    brew.uninstall.packages "$@"  
}
```

2.10. Module caller

2.10.1. caller.stack

```
caller.stack ()  
{  
    local index=${1:-"-1"}  
    while true; do  
        index=$((index + 1))  
        caller ${index} 2>&1 > /dev/null || break  
        local -a frame=($(caller ${index} | tr ' ' '\n'))  
        printf "%3d [ %-40.40s ]: %s\n" ${index} "${frame[2]}:${frame[0]}" "${frame  
[1]}"  
        done  
}
```

2.10.2. stack.frame

```
stack.frame ()  
{  
    caller.stack 0
```

```
}
```

2.11. Module color

2.11.1. bold

```
bold ()  
{  
    .ansi 1 "$@"  
}
```

2.11.2. color.current-background

```
color.current-background ()  
{  
    printf "\e]11;?\a"  
}
```

2.11.3. color.disable

```
color.disable ()  
{  
    export clr='\e[0m'  
    unset txtblk  
    unset txtred  
    unset txtgrn  
    unset txtylw  
    unset txtblu  
    unset txtpur  
    unset txtcyn  
    unset txtwht  
    unset bldblk  
    unset bldred  
    unset bldgrn  
    unset bldylw  
    unset bldblu  
    unset bldpur  
    unset bldcyn  
    unset bldwht  
    unset unkblk  
    unset undred  
    unset undgrn  
    unset undylw  
    unset undblu  
    unset undpur  
    unset undcyn
```

```
unset undwht
unset bakblk
unset bakred
unset bakgrn
unset bakylw
unset bakblu
unset bakpur
unset bakcyn
unset bakwht
unset txtrst
unset italic
unset bold
unset strikethrough
unset underlined
unset white_on_orange
unset white_on_yellow
unset white_on_red
unset white_on_pink
unset white_on_salmon
unset yellow_on_gray
unset bg_blood
unset bg_blue_on_gray
unset bg_bright_green
unset bg_dark_green
unset bg_bright_red
unset bg_deep_blue
unset bg_deep_green
unset bg_green_on_gray
unset bg_grey
unset bg_mustard
unset bg_pink
unset bg_sky_blue
unset bg_yellow_on_gray
unset fg_dark_red
unset fg_bright_green
unset fg_sky_blue
unset fg_deep_green
unset fg_doll
unset fg_grey
unset fg_light_green
unset fg_mustard
unset fg_mustard
unset fg_pink
unset fg_purr
export BashMatic__ColorLoaded=0
}
```

2.11.4. `color.enable`

```

color.enable ()
{
    export BashMatic__ColorLoaded=1
    export txtblk='\e[0;30m'
    export txtred='\e[0;31m'
    export txtgrn='\e[0;32m'
    export txtylw='\e[0;33m'
    export txtblu='\e[0;34m'
    export txtpur='\e[0;35m'
    export txtcyn='\e[0;36m'
    export txtwht='\e[0;37m'
    export bldblk='\e[1;30m'
    export bldred='\e[1;31m'
    export bldgrn='\e[1;32m'
    export bldylw='\e[1;33m'
    export bldblu='\e[1;34m'
    export bldpur='\e[1;35m'
    export bldcyn='\e[1;36m'
    export bldwht='\e[1;37m'
    export unkblk='\e[4;30m'
    export undred='\e[4;31m'
    export undgrn='\e[4;32m'
    export undylw='\e[4;33m'
    export undblu='\e[4;34m'
    export undpur='\e[4;35m'
    export undcyn='\e[4;36m'
    export undwht='\e[4;37m'
    export bakblk='\e[40m'
    export bakred='\e[41m'
    export bakgrn='\e[42m'
    export bakylw='\e[43m'
    export bakblu='\e[44m'
    export bakpur='\e[45m'
    export bakcyn='\e[46m'
    export bakwht='\e[47m'
    export txtrst='\e[0m'
    export rst='\e[0m'
    export clr='\e[0m'
    export bold='\e[1m'
    export italic='\e[3m'
    export underlined='\e[4m'
    export strikethrough='\e[9m'
    export inverse_on='\e[7m'
    export inverse_off='\e[27m'
    export default_bg='\e[49m'
    export default_fg='\e[39m'
    export black_on_orange="\e[48;5;208m\e[48;30;208m"
    export black_on_yellow="\e[48;5;11m\e[48;30;209m"
    export white_on_orange="\e[48;5;208m"
    export white_on_yellow="\e[48;5;214m"
}

```

```
export white_on_red="\e[48;5;9m"
export white_on_pink="\e[48;5;199m"
export white_on_salmon="\e[48;5;196m"
export yellow_on_gray="\e[38;5;220m\e[48;5;242m"
export bg_blood="\e[41m"
export bg_blue_on_gray="\e[90;7;42m"
export bg_bright_green="\e[48;5;82m"
export bg_dark_green="\e[48;5;82m"
export bg_bright_red="\e[1;31m\e[48;5;196m"
export bg_deep_blue="\e[48;5;37m"
export bg_deep_green="\e[48;5;28m"
export bg_green_on_gray="\e[90;7;102m"
export bg_grey="\e[48;5;239m"
export bg_mustard="\e[48;5;178m"
export bg_pink="\e[48;5;89m"
export bg_sky_blue="\e[48;5;39m"
export bg_yellow_on_gray="\e[90;7;43m"
export fg_dark_red="\e[38;5;88m"
export fg_bright_green="\e[38;5;82m"
export fg_sky_blue="\e[38;5;39m"
export fg_deep_green="\e[38;5;28m"
export fg_doll="\e[38;5;183m"
export fg_grey="\e[38;5;239m"
export fg_light_green="\e[38;5;108m"
export fg_mustard="\e[38;5;178m"
export fg_mustard="\e[38;5;178m"
export fg_pink="\e[38;5;89m"
export fg_purr="\e[38;5;219m"
export BashMatic__ColorLoaded=1
export bg_blood="\e[41m"
export bg_blue_on_gray="\e[90;7;42m"
export bg_bright_green="\e[48;5;82m"
export bg_dark_green="\e[48;5;82m"
export bg_bright_red="\e[1;31m\e[48;5;196m"
export bg_deep_blue="\e[48;5;37m"
export bg_deep_green="\e[48;5;28m"
export bg_green_on_gray="\e[90;7;102m"
export bg_grey="\e[48;5;239m"
export bg_mustard="\e[48;5;178m"
export bg_pink="\e[48;5;89m"
export bg_sky_blue="\e[48;5;39m"
export bg_yellow_on_gray="\e[90;7;43m"
export fg_dark_red="\e[38;5;88m"
export fg_bright_green="\e[38;5;82m"
export fg_sky_blue="\e[38;5;39m"
export fg_deep_green="\e[38;5;28m"
export fg_doll="\e[38;5;183m"
export fg_grey="\e[38;5;239m"
export fg_light_green="\e[38;5;108m"
export fg_mustard="\e[38;5;178m"
export fg_mustard="\e[38;5;178m"
```

```
    export fg_pink="\e[38;5;89m"
    export fg_purr="\e[38;5;219m"
}
```

2.11.5. error-text

```
error-text ()
{
    printf "${txtred}"
}
```

2.11.6. italic

```
italic ()
{
    .ansi 3 "$@"
}
```

2.11.7. red

```
red ()
{
    .ansi 31 "$@"
}
```

2.11.8. reset-color

```
reset-color ()
{
    echo -en "${clr}"
}
```

2.11.9. strikethrough

```
strikethrough ()
{
    .ansi 9 "$@"
}
```

2.11.10. txt-err

```
txt-err ()
```

```
{  
    printf "${clr}${bldylw}${bakred}"  
}
```

2.11.11. **txt-info**

```
txt-info ()  
{  
    printf "${clr}${txtblu}"  
}
```

2.11.12. **txt-warn**

```
txt-warn ()  
{  
    printf "${clr}${bldylw}"  
}
```

2.11.13. **underline**

```
underline ()  
{  
    .ansi 4 "$@"  
}
```

2.12. Module config

2.12.1. **config.dig**

```
config.dig ()  
{  
    local key="$1"  
    local subkey="$2"  
    local format="$(config.get-format)"  
    local format_lower="$(config.get-format | tr '[[:upper:]]' '[[:lower:]]')"  
    local rf="require '${format_lower}'; "  
    local load_config="${rf}; def config; ${format}.load(File.read('"  
${bashmatic__config_file}')); end"  
    local interpreter="$(command -v ruby)"  
    local script  
    if [[ -z ${key} ]]; then  
        script="${load_config}; ${rf} pp config"  
    else  
        if [[ -n ${subkey} ]]; then  
            script="${load_config}; ${rf} pp ${key}."  
        else  
            script="${load_config}; ${rf} pp config."  
        fi  
    fi  
    eval $script  
}
```

```

        script="${load_config}; ${rf} puts config['${key}'][ '${subkey}' ]"
    else
        script="${load_config}; ${rf} pp config['${key}']"
    fi
fi
${interpreter} -e "${script}" || {
    error "ERROR while evaluating the following script with ruby $(ruby --version ):"
    "${script}"
    return 1
}
return 0
}

```

2.12.2. config.dig.pretty

```

config.dig.pretty ()
{
    local -a args
    local -a keys
    for a in "$@"
    do
        if [[ $a =~ ^- ]]; then
            args+=("$a")
        else
            keys+=("$a")
        fi
    done
    is-dbg && {
        dbg "args: ${args[*]}"
        dbg "keys: ${keys[*]}"
    }
    local format=$(config.get-format)
    config.set-format JSON
    command -v jq > /dev/null || package.install.packages jq
    config.dig "${keys[@]}" | jq "${args[@]}" | tr -d ''
    config.set-format "${format}"
    return 0
}

```

2.12.3. config.get-file

```

config.get-file ()
{
    printf "%s" "${bashmatic__config_file}"
}

```

2.12.4. config.get-format

```
config.get-format ()  
{  
    echo -n "${bashmatic__config_format}"  
}
```

2.12.5. config.set-file

```
config.set-file ()  
{  
    export bashmatic__config_file="$1"  
    if [[ ${bashmatic__config_file} =~ \.yml$ || ${bashmatic__config_file} =~ \.yaml$  
    || ${bashmatic__config_file} =~ \.YML$ || ${bashmatic__config_file} =~ \. YAML$ ]];  
    then  
        config.set-format yaml  
    else  
        if [[ ${bashmatic__config_file} =~ \.json$ || ${bashmatic__config_file} =~  
        \.JSON$ ]]; then  
            config.set-format json  
        else  
            warning "File extension is not recognized." "Use config.set-format  
[json|yaml]" 1>&2  
        fi  
    fi  
}
```

2.12.6. config.set-format

```
config.set-format ()  
{  
    local format="${1^^}"  
    if [[ "$format" == "YAML" || "$format" == "JSON" ]]; then  
        export bashmatic__config_format="$format"  
    else  
        error "Invalid format $1: only YAML or JSON is supported."  
        return 1  
    fi  
}
```

2.13. Module db

2.13.1. db.actions.connect

```
db.actions.connect ()
```

```
{  
    db.psql.connect "$@"  
}
```

2.13.2. db.actions.connections

```
db.actions.connections ()  
{  
    db.config.connections  
    echo  
}
```

2.13.3. db.actions.csv

```
db.actions.csv ()  
{  
    local dbname=${1}  
    shift  
    [[ -z ${dbname} ]] && return 1  
    export flag_quiet=1  
    db.psql.connect "${dbname}" -P border=0 -P fieldsep="," --csv -A -X -P pager=off  
    -P footer=off -c "\$*"  
}
```

2.13.4. db.actions.data-dir

```
db.actions.data-dir ()  
{  
    db.psql.connect "$@" $(db.psql.args-data-only) -c "'show data_directory'"  
}
```

2.13.5. db.actions.db-settings-pretty

```
db.actions.db-settings-pretty ()  
{  
    db.psql.connect.db-settings-pretty "$@"  
}
```

2.13.6. db.actions.db-settings-toml

```
db.actions.db-settings-toml ()  
{  
    db.psql.connect.db-settings-toml "$@"  
}
```

```
}
```

2.13.7. db.actions.explain

```
db.actions.explain ()  
{  
    db.psql.explain "$@"  
}
```

2.13.8. db.actions.explain-analyze-json

```
db.actions.explain-analyze-json ()  
{  
    export _bashmatic_db_explain="${_bashmatic_db_explain_sql}"  
    .db.actions.explain "$@"  
}
```

2.13.9. db.actions.explain-json

```
db.actions.explain-json ()  
{  
    export _bashmatic_db_explain="${_bashmatic_db_analyze_sql}"  
    .db.actions.explain "$@"  
}
```

2.13.10. db.actions.list-indexes

```
db.actions.list-indexes ()  
{  
    db.psql.list-indexes "$@"  
}
```

2.13.11. db.actions.list-tables

```
db.actions.list-tables ()  
{  
    db.psql.list-tables "$@"  
}
```

2.13.12. db.actions.list-users

```
db.actions.list-users ()
```

```
{  
    db.psql.list-users "$@"  
}
```

2.13.13. `db.actions.pga`

```
db.actions.pga ()  
{  
    local name="$1"  
    command -v python3 > /dev/null || brew.install.packages python3  
    command -v pg_activity > /dev/null || run "python3 -m pip install pg_activity  
psycopg2-binary"  
    command -v pg_activity > /dev/null || {  
        local binary=$(find /usr/local/Cellar -type f -name 'pg_activity')  
        run "ln -nfs ${binary} /usr/local/bin/pg_activity"  
    }  
    command -v pg_activity > /dev/null || {  
        error "Can't find pg_activity even after install + symlink".  
        return 1  
    }  
    local args=$(db.psql.args.config "${name}")  
    db.psql.args.config "${name}" > /dev/null  
    pg_activity "${args}" --verbose-mode=1 --rds --no-app --no-database --no-user  
}
```

2.13.14. `db.actions.run`

```
db.actions.run ()  
{  
    db.psql.run "$@"  
}
```

2.13.15. `db.actions.run-multiple`

```
db.actions.run-multiple ()  
{  
    db.psql.run-multiple "$@"  
}
```

2.13.16. `db.actions.set-max-query-len`

```
db.actions.set-max-query-len ()  
{  
    db.config.set-max-query-len "$@"  
}
```

```
}
```

2.13.17. db.actions.table-locks

```
db.actions.table-locks ()  
{  
    db.psql.table-locks "$@"  
}
```

2.13.18. db.actions.table-locks-query

```
db.actions.table-locks-query ()  
{  
    db.psql.table-locks-query "$@"  
}
```

2.13.19. db.actions.table-settings-set

```
db.actions.table-settings-set ()  
{  
    db.psql.connect.table-settings-set "$@"  
}
```

2.13.20. db.actions.table-settings-show

```
db.actions.table-settings-show ()  
{  
    db.psql.connect.table-settings-show "$@"  
}
```

2.13.21. db.actions.top

```
db.actions.top ()  
{  
    db.top "$@"  
}
```

2.13.22. db.config.connections

```
db.config.connections ()  
{  
    ascii-clean "$(db.config.connections-list | awk '{print $1}')"
```

```
}
```

2.13.23. db.config.connections-list

```
db.config.connections-list ()
{
    [[ -f ${bashmatic_db_config} ]] || return 2
    ruby.handle-missing
    gem.install colorize > /dev/null
    __yaml_source="${bashmatic_db_config}" ruby <<RUBY
    require 'yaml'
    require 'colorize'
    h = YAML.load(File.read(ENV['__yaml_source']), aliases: true)
    h.each_pair do |name, params|
        printf "%50s → %s@%s/%s\n",
            name.yellow,
            params['username'].blue,
            params['host'].green,
            params['database'].cyan
    end
}
RUBY

}
```

2.13.24. db.config.get-file

```
db.config.get-file ()
{
    echo "${bashmatic_db_config}"
}
```

2.13.25. db.config.init

```
db.config.init ()
{
    export bashmatic_db_connection=(host database username password)
}
```

2.13.26. db.config.init-max-query-len

```
db.config.init-max-query-len ()
{
    local w=$(screen.width.actual)
    local max_query_len
    max_query_len=$(( w - 20 ))
```

```

max_query_len=$(array.force-range "${max_query_len}" 80 1000)
((flag_verbose)) && {
    attention "Queries will be truncated to ${max_query_len} characters" 1>&2
}
printf "%d" "${max_query_len}"
}

```

2.13.27. db.config.parse

```

db.config.parse ()
{
    local db="$1"
    [[ -z ${db} ]] && return 1
    [[ -f ${bashmatic_db_config} ]] || return 2
    db.config.init
    local -a script=(`require 'yaml'; h = YAML.load(STDIN); `)
    for field in "${bashmatic_db_connection[@]}"
    do
        script+=("h.key?('$db') && h['$db'].key?('$field') ? print(h['$db']
        ['$field']) : print('null'); print ' ' ")
    done
    is.a-function ruby.handle-missing || source "${BASHMATIC_LIB}/ruby.sh"
    ruby.handle-missing
    ruby -e "${script[*]}" < "${bashmatic_db_config}"
}

```

2.13.28. db.config.set-file

```

db.config.set-file ()
{
    [[ -s "$1" ]] || return 1
    export bashmatic_db_config="$1"
}

```

2.13.29. db.datetime

```

db.datetime ()
{
    date '+%Y%m%d-%H%M%S'
}

```

2.13.30. db.pg.local.num-procs

```

db.pg.local.num-procs ()
{

```

```
/bin/ps -ef | /bin/grep "[p]ostgres" | wc -l | awk '{print $1}'  
}
```

2.13.31. db.postgres.version

```
db.postgres.version ()  
{  
    command -v postgres > /dev/null || return 1  
    postgres --version | sed -E 's/[^\0-9.]/g'  
}
```

2.13.32. db.psql.args

```
db.psql.args ()  
{  
    if [[ -z "${bashmatic_db_database}" || -z "${bashmatic_db_host}" ]]; then  
        if [[ -n "$1" ]]; then  
            db.psql.args.config "$1"  
        else  
            error "Unable to determine DB connection parameters"  
            return 1  
        fi  
    else  
        export PGPASSWORD="${bashmatic_db_password}"  
        printf -- "-U ${bashmatic_db_username} -h ${bashmatic_db_host}  
${bashmatic_db_database}"  
        fi  
}
```

2.13.33. db.psql.args-data-only

```
db.psql.args-data-only ()  
{  
    printf -- "%s" "-t --no-align --pset footer -q -X --tuples-only"  
}
```

2.13.34. db.psql.args.config

```
db.psql.args.config ()  
{  
    local output=$(db.config.parse "$1")  
    local -a params  
    [[ -z ${output} || "${output}" =~ "null" ]] && {  
        section.red 65 "Unknown database connection - ${bldylw}${1}." 1>&2  
        info "The following are connections defined in ${bldylw}${bashmatic_db_config}
```

```

/${{HOME}}/\`~\`:\`n" 1>&2
    for c in $(db.config.connections)
    do
        info " \ ${c}" 1>&2
    done
    echo 1>&2
    exit 1
}
params=($(db.config.parse "$1"))
local dbhost
local dbname
local dbuser
local dbpass
dbhost=${params[0]}
dbname=${params[1]}
dbuser=${params[2]}
dbpass=${params[3]}
export PGPASSWORD="${dbpass}"
printf -- "-U ${dbuser} -h ${dbhost} -d ${dbname}"
}

```

2.13.35. db.psql.args.localhost

```

db.psql.args.localhost ()
{
    printf -- "-U postgres -h localhost $*"
}

```

2.13.36. db.psql.args.maintenance

```

db.psql.args.maintenance ()
{
    db.psql.args.localhost "--maintenance-db=postgres $*"
}

```

2.13.37. db.psql.connect

```

db.psql.connect ()
{
    local dbname="$1"
    shift
    if [[ -z ${dbname} ]]; then
        h1 "USAGE: db.connect connection-name" "WHERE: connection-name is defined by
your ${bldylw}${bashmatic_db_config}${clr} file." 1>&2
        return 0
    fi
}

```

```

export __pgsql_stderr="$(_file.temp)"
cp /dev/null "$__pgsql_stderr"
local tempfile=$(mktemp)
db.psql.args.config "${dbname}" > "${tempfile}"
local -a args=($(cat "${tempfile}"))
[[ -n "${pgsql_extra_args[*]}"] ] && args+=("${pgsql_extra_args[@]}")
rm -f "${tempfile}" > /dev/null
[[ ${flag_quiet} -eq 0 ]] && {
    printf "${txtpur}export PGPASSWORD=[redacted]${clr}\n" 1>&2
    printf "${txtylw}$(which psql) ${args[*]}${clr}\n" 1>&2
    ( hr
        echo ) 1>&2
}
set +e
is-verbose && echo
if [[ ${action} == "run" ]]; then
    print-cli "psql ${args[*]} --echo-errors $*"
    psql "${args[@]}" --echo-errors "$@" 2> "$__pgsql_stderr"
    local code=$?
    [[ ${code} -ne 0 || -s "$__pgsql_stderr" ]] && db.psql.report-error "${args[@]}" "$@"
else
    print-cli "psql ${args[*]} --echo-errors $*"
    eval "psql ${args[*]} --echo-errors $*"
    local code=$?
fi
return ${code}
}

```

2.13.38. `db.psql.connect.db-settings-pretty`

```

db.psql.connect.db-settings-pretty ()
{
    db.psql.connect "$@" -A -X -q -c "\show all\" | grep -v 'rows)' | sort | awk
"BEGIN{FS=\"|\"}{ printf(\"%-40.40s %-30.30s ## %s\\n\", \$1, \$2, \$3) }" | sedx
'##\$*/d' | GREP_COLOR="1;32" grep -E -C 1000 -i --color=always -e '^([^\n]*'
GREP_COLOR="3;0;34" grep -E -C 1000 -i --color=always -e '##.*$|$'
}

```

2.13.39. `db.psql.connect.db-settings-toml`

```

db.psql.connect.db-settings-toml ()
{
    db.psql.connect.just-data "$1" -c "\show all\" | awk 'BEGIN{FS=""}{printf
"%s=%s\\n", \$1, \$2}' | sort
}

```

2.13.40. db.psql.connect.just-data

```
db.psql.connect.just-data ()  
{  
    local dbname="$1"  
    shift  
    db.psql.connect "${dbname}" ${db.psql.args-data-only} "$@"  
}
```

2.13.41. db.psql.connect.table-settings-set

```
db.psql.connect.table-settings-set ()  
{  
    local dbname="$1"  
    shift  
    local table="$1"  
    shift  
    local setting="$1"  
    shift  
    local value="$1"  
    shift  
    [[ -z ${setting} || -z ${value} ]] && {  
        error "Either setting or value are not defined." 1>&2  
        return 1  
    }  
    info "Setting ${setting} = ${value} on table ${table}...."  
    db.psql.connect "${dbname}" ${db.psql.args-data-only} -c "ALTER TABLE \"${table}\"  
SET (${setting} = ${value});"  
}
```

2.13.42. db.psql.connect.table-settings-show

```
db.psql.connect.table-settings-show ()  
{  
    local dbname="$1"  
    shift  
    local table="$1"  
    shift  
    db.psql.connect "${dbname}" ${db.psql.args-data-only} -c "SELECT relname,  
reloptions FROM pg_class WHERE relname='${table}';"  
}
```

2.13.43. db.psql.db-settings

```
db.psql.db-settings ()  
{
```

```
psql "$*" -X -q -c "\show all" | sort | awk '{ printf("%s=%s\n", $1, $3) }' |  
sed -E 's/[()]-//g;/name=setting/d;/^[-+=]*$/d;/^[0-9]*=$/d'  
}
```

2.13.44. db.psql.explain

```
db.psql.explain ()  
{  
    local dbname="$1"  
    shift  
    db.psql.connect "${dbname}" -t -A -X --pset border=0 -c "'explain $*'"  
}
```

2.13.45. db.psql.list-indexes

```
db.psql.list-indexes ()  
{  
    local dbname="$1"  
    shift  
    db.psql.connect "${dbname}" ${db.psql.args-data-only} -c '\\\di' | awk  
'BEGIN{FS="|"}{print $2}'  
}
```

2.13.46. db.psql.list-tables

```
db.psql.list-tables ()  
{  
    local dbname="$1"  
    shift  
    db.psql.connect "${dbname}" ${db.psql.args-data-only} -c '\\\dt' | awk  
'BEGIN{FS="|"}{print $2}'  
}
```

2.13.47. db.psql.list-users

```
db.psql.list-users ()  
{  
    local dbname="$1"  
    shift  
    db.psql.connect "${dbname}" ${db.psql.args-data-only} -c '\\\du' | awk  
'BEGIN{FS="|"}{print $2}'  
}
```

2.13.48. db.psql.report-error

```
db.psql.report-error ()  
{  
    local -a argv=("$@")  
    [[ -z "${__pgsql_stderr}" ]] && return 0  
    [[ -s "${__pgsql_stderr}" ]] || return 0  
    error "Error running command: \"${bldylw}psql ${argv[*]}\""  
    printf -- "${txtred}$(cat "${__pgsql_stderr}" | sed -E 's/^/\t/g')${clr}\n"  
    hr  
    rm -f "${__pgsql_stderr}"  
    unset __pgsql_stderr  
}
```

2.13.49. db.psql.run

```
db.psql.run ()  
{  
    local dbname="$1"  
    shift  
    local query="$1"  
    shift  
    db.psql.connect.just-data "${dbname}" -c "${query}" "@"  
}
```

2.13.50. db.psql.run-multiple

```
db.psql.run-multiple ()  
{  
    local dbname="$1"  
    shift  
    local commands  
    for arg in "$@"  
    do  
        if [[ ${arg} =~ \\" ]]; then  
            commands="${commands} -c '$(printf "%s" "${arg}")'"  
        else  
            commands="${commands} -c \"$(printf "%s" "${arg})\"\n"  
        fi  
    done  
    echo "${commands}" > /tmp/a  
    db.psql.connect "${dbname}" -t -A -X --pset border=0 "${commands}"  
}
```

2.13.51. db.psql.table-locks

```
db.psql.table-locks ()  
{  
    local db="$1"  
    shift  
    local table="$1"  
    shift  
    local file=$(./db.locks.generate-sql "${table}")"  
    db.psql.connect "${db}" "$*" -X -x --pset=pager -q -f "${file}" | GREP_COLOR=41  
grep --color=always -E "(${table}:-table_name.*$)|$)"  
    [[ -f "${file}" ]] && rm -f "${file}"  
}
```

2.13.52. db.psql.table-locks-query

```
db.psql.table-locks-query ()  
{  
    local db="$1"  
    shift  
    local table="$1"  
    shift  
    local file=$(./db.locks.generate-sql "${table}")"  
    printf "\n${txtgrn}"  
    cat "${file}"  
    printf "\n${clr}"  
    [[ -f "${file}" ]] && rm -f "${file}"  
}
```

2.13.53. db.psql.version

```
db.psql.version ()  
{  
    command -v psql > /dev/null || return 1  
    psql --version | sed -E 's/[^\d.]///g'  
}
```

2.13.54. db.wait-until-db-online

```
db.wait-until-db-online ()  
{  
    local db="${1}"  
    inf 'waiting for the database to come up...'  
    while true; do  
        out=$(psql -c "select count(*) from pg_stat_user_tables" "${db}" 2>&1)  
    done  
}
```

```
        code=$?
        [[ ${code} == 0 ]] && break
        [[ ${code} == 1 ]] && break
        sleep 1
        [[ ${out} =~ 'does not exist' ]] && break
    done
    ui.closer.ok:
    return 0
}
```

2.13.55. `is-quiet`

```
is-quiet ()
{
    ((flag_quiet))
}
```

2.13.56. `is-verbose`

```
is-verbose ()
{
    ((flag_verbose))
}
```

2.13.57. `print-cli`

```
print-cli ()
{
    is-verbose || return
    notice "Running command line:"
    cursor.up 2
    notice "${itablk}$*"
}
```

2.14. Module `db_cli`

2.14.1. `db.actions.commands`

```
db.actions.commands ()
{
    db.commands-list
}
```

2.14.2. db.cli-setup

```
db.cli-setup ()  
{  
    db.refresh.actions  
    color.enable > /dev/null  
    output.constrain-screen-width 110  
    if [[ $(screen.width) -lt 110 ]]; then  
        error "Please resize your terminal to have at least 110 columns."  
        return 1  
    fi  
}
```

2.14.3. db.commands-list

```
db.commands-list ()  
{  
    h5 "Available Commands"  
    source "${BASHMATIC_LIB}/util.sh"  
    db.refresh.actions  
    printf "${bldgrn}"  
    array.to.bullet-list "${db_actions[@]}" | sed 's/^/      /g'  
    echo  
    hr  
    echo  
    exit 0  
}
```

2.14.4. db.connections-list

```
db.connections-list ()  
{  
    h4 "Available Database Connections"  
    local -a connections  
    connections=$(db.actions.connections)  
    printf "${bldblu}"  
    array.to.bullet-list "${connections[@]}" | sed 's/^/      /g'  
    echo  
    hr  
    echo  
    exit 0  
}
```

2.14.5. db.examples

```
db.examples ()
```

```

{
    h2 EXAMPLES "${txtblu}#${italic}# List available connection names" "${bldylw}db
--connections" " " "${txtblu}#${italic}# List available sub-commands" "${bldylw}db
--commands" " " "${txtblu}#${italic}# Connect to the database named 'staging.core'
using psql" "${bldylw}db connect staging.core" " " "${txtblu}#${italic}# Show 'db top'
for up to 3 databases at once:" "${bldylw}db top prod.core prod.replica1
prod.replica2" " " "${txtblu}#${italic}# Use 'pg_activity' to show db top for one
connection:" "${bldylw}db pga prod.core" " " "${txtblu}#${italic}# Show all settings
currently active on production DB in TOML/ini format:" "${txtblu}#${italic}# and
suppress the header with -q:" "${bldylw}db db-settings-toml prod.core -q" " "
"${txtblu}#${italic}# Run a query with the default output" "${bldylw}db run -q prod.core
'select relname,n_live_tup from pg_stat_user_tables order by n_live_tup desc'" " "
"${txtblu}#${italic}# Run the same query, but this time output in a CSV format" "
"${txtblu}#${italic}# NOTE: majority of the flags are passed to the ${bldgrn}psql${clr}#${txtblu}#${italic} to format the output," "${txtblu}#${italic}#     except -q is
consumed by the script and turns off the script header." "${txtblu}#${italic}#
While -P flag is equivalent to \pset in psql session." "${bldylw}export query='select
relname,n_live_tup from pg_stat_user_tables order by n_live_tup desc'" "${bldylw}db
run staging.core \"\$query limit 10\" -q -AX -P pager=0 -P fieldsep=, -P footer=off"
" " "${txtblu}#${italic}#NOTE: read more about psql formatting options via \pset and
--pset flags:" "${txtblu}#${italic}      ${undgrn}https://bit.ly/psql-pset""
}

```

2.14.6. db.main

```

db.main () {
    declare -a psql_extra_args
    export psql_extra_args=()
    [[ -z "$*" ]] && {
        db.usage
        return
    }
    while :; do
        case $1 in
            -h | --help)
                shift
                db.usage
                return
            ;;
            -e | --examples)
                shift
                db.examples
                return
            ;;
            -c | --connections)
                shift
                db.connections-list
                return
            ;;
        esac
    done
}

```

```

;
-q | --quiet)
    shift
    export flag_quiet=1
;
-v | --verbose)
    shift
    export flag_verbose=1
;
-C | --commands)
    shift
    h3 "Valid actions are:" "${db_actions[@]}"
    exit 0
;
-f | --filter)
    shift
    [[ "$1" =~ "%" ]] && {
        error "The filter expression is automatically wrapped in
'<expression>%',' so - no need to add '%' yourself."
        exit 1
    }
    export flag_filter_active="$1"
    shift
;
-w | --width)
    shift
    is.numeric "$1" || {
        error "Argument to -w | --width must be a number, dear."
        exit 1
    }
    export flag_width="$1"
    shift
;
[a-z]*)
    [[ -n ${action} ]] && break
    export action="$1"
    shift
    array.includes "${action}" "${db_actions[@]}" || {
        error "Invalid Command" "${action}"
        db.commands-list
        return 1
    }
    export func="db.actions.${action}"
;
--)
    shift
    export psql_extra_args="$@"
    while (( $#)); do
        shift
    done
;

```

```

        *)
            [[ -z "$1" ]] && break
            error "Unknown flag $1 — if it's intended for psql, please add -- before it."
        return 2
    ;
    esac
done
is.a-function "${func}" || {
    error "Invalid action ${action}!"
    db.usage
    return 3
}
${func} "$@"
}

```

2.14.7. db.refresh.actions

```

db.refresh.actions ()
{
    export db_actions=$(util.functions-matching.diff db.actions.)
}

```

2.14.8. db.usage

```

db.usage ()
{
    local config="$(basename $(dirname "${bashmatic_db_config}"))/$(basename "${bashmatic_db_config}")"
    usage-box "db [global flags] command [command flags] connection [-- psql flags] ©
    Performs one of many supported actions against PostgreSQL" "-q / --quiet" "Suppress
    the colorful header messages" "-v / --verbose" "Show additional output" "-n / --dry
    -run" "Only print commands, but do not run them" " |-GLOBAL FLAGS:" " " " -w / --width
    WIDTH" "Use specified screen width instead of automatic." "-f / --filter REGEX"
    "Defaults to 'idle', but can be set" " " " to remove some active queries from the
    view." "-C / --commands" "List all sub-commands to the db script" "-c / --connections"
    "List all available database connections" "-e / --examples" "Show script usage
    examples" "-h / --help" "Show this help screen" " " " " |-SUMMARY:" " " " "
    "This
    tool uses a list of database connections defined in the" " " " YAML file that must be
    installed at: ${bldylw}${config}" " " " "
}

```

2.15. Module db_rails

2.15.1. db.rails.schema.checksum

```
db.rails.schema.checksum ()  
{  
    if [[ -d db/migrate ]]; then  
        find db/migrate -type f -ls | awk '{printf("%10d-%s\n",$7,$11)}' | sort |  
shasum | awk '{print $1}'  
    else  
        local schema=$(db.rails.schema.file)  
        [[ -s ${schema} ]] || error "can not find Rails schema in either  
${RAILS_SCHEMA_RB} or ${RAILS_SCHEMA_SQL}"  
        [[ -s ${schema} ]] && shasum.sha-only "${schema}"  
    fi  
}
```

2.15.2. db.rails.schema.file

```
db.rails.schema.file ()  
{  
    if [[ -f "${RAILS_SCHEMA_RB}" && -f "${RAILS_SCHEMA_SQL}" ]]; then  
        if [[ "${RAILS_SCHEMA_RB}" -nt "${RAILS_SCHEMA_SQL}" ]]; then  
            printf "${RAILS_SCHEMA_RB}"  
        else  
            printf "${RAILS_SCHEMA_SQL}"  
        fi  
    else  
        if [[ -f "${RAILS_SCHEMA_RB}" ]]; then  
            printf "${RAILS_SCHEMA_RB}"  
        else  
            if [[ -f "${RAILS_SCHEMA_SQL}" ]]; then  
                printf "${RAILS_SCHEMA_SQL}"  
            fi  
        fi  
    fi  
}
```

2.16. Module db_top

2.16.1. db.top

```
db.top ()  
{  
    local dbname  
    local width_min=90  
    local height_min=30  
    local width=$(screen.width)
```

```

local height=$(screen.height)
if [[ "$1" == "-l" || "$1" == "--list" ]]; then
    db.config.connections
    echo
    return
fi
output.unconstrain-screen-width
if [[ ${width} -lt ${width_min} || ${height} -lt ${height_min} ]]; then
    error "Your screen is too small for db.top."
    info "Minimum required screen dimensions are ${width_min} columns,
${height_min} rows."
    info "Your screen is ${bldred}${width}x${height}.""
    return
fi
local -a connections_arguments
local -a connections_names
local -a connections_passwords
local code=0
local i=0
local arguments
local tof="$(mktemp -d "${TMPDIR:-/tmp/}.XXXXXXXXXXXXXX").db.top.$$"
cp /dev/null "${tof}" > /dev/null
for connection in "$@"
do
    db.psql.args.config "${connection}" > /dev/null || return 1
    db.psql.args "${connection}" > "${tof}"
    arguments=$(cat "${tof}" | tr -d '\n')
    connections_arguments+=("${arguments}")
    connections_names+=("${connection}")
    connections_passwords+=("${PGPASSWORD}")
    i=$((i + 1))
done
if [[ ${#connections_names[@]} -eq 0 ]]; then
    h1 "${bldgrn}USAGE: db.top db1 db2 ... " "
    h1 "EG: db.top prod-master prod-
replica1 prod-replica2"
    return 1
fi
((BASH_IN_SUBSHELL)) && {
    trap "clear" TERM
    trap "clear" EXIT
}
local interval=${bashmatic_db_top_refresh:-1}
local num dbs=${#connections_names[@]}
h1 "Refreshing activity for ${num} databases..."
while true; do
    local index=0
    rm -f "${tof}.errors"
    cp /dev/null "${tof}"
    local screen_height=$(screen.height)
    for dbname in "${connections_names[@]}"
    do
        if [[ ${index} -lt ${num} ]]; then
            db.psql.args.config "${dbname}" > /dev/null
            db.psql.args "${dbname}" > "${tof}"
            arguments=$(cat "${tof}" | tr -d '\n')
            connections_arguments+=("${arguments}")
            connections_names+=("${dbname}")
            connections_passwords+=("${PGPASSWORD}")
            i=$((i + 1))
        fi
        index=$((index + 1))
    done
    sleep ${interval}
done

```

```

local percent_total_height
local percent_total_height_next
percent_total_height=$(.db.top.vertical-offset "${num_dbs}" ${index})
percent_total_height_next=$(.db.top.vertical-offset "${num_dbs}" $((index
+ 1)))
local vertical_shift=$((percent_total_height * screen_height / 100))
local vertical_shift_next=$((percent_total_height_next * screen_height /
100))
local height=$((vertical_shift_next - vertical_shift - 7))
cursor.at.y ${vertical_shift} >> "${tov}"
[[ -n ${BASHMATIC_DEBUG} ]] && {
    .output.set-indent 0
    info "CURRENT ${bldylw} % = ${percent_total_height}, %++ =
${percent_total_height_next}" >> "${tov}"
    h1 -- "Database: ${dbname}" "PSQL arguments:" "${
connections_arguments[$index]}" >> "${tov}"
}
.db.top.connection "${tov}" "${dbname}" "${connections_passwords[$index
]}" "${height}" "${connections_arguments[$index]}"
index=$((index + 1))
done
clear
if [[ -s "${tov}.errors" ]]; then
    error "ERROR running psql with args: ${bldylw}${connections_arguments[
${index}]}"
    printf "${bldred}"
    cat "${tov}.errors"
    printf "${clr}\n"
    h3 "Output:"
    cat "${tov}"
    code=111
    break
else
    .output.set-indent 0
    hl.green "DbTop0 v1.1.0 © 2016-2022 Konstantin Gredeskoul © All Rights
Reserved © MIT License — "
    cat "${tov}"
    cursor.at.y $(((${output.screen-height} + 1)))
    printf "${bldwht}Press Ctrl-C to quit.${clr}"
    code=0
fi
sleep "${interval}"
done
return ${code}
}

```

2.16.2. db.top.set-refresh

```
db.top.set-refresh ()
```

```
{  
    export bashmatic_db_top_refresh="$1"  
}
```

2.17. Module deploy

2.17.1. `deploy.slack`

```
deploy.slack ()  
{  
    local original_text="$*"  
    [[ -z ${LibDeploy__SlackHookUrl} ]] && return 1  
    local text=$(echo "${original_text}" | sed -E 's/"/\\"/g' | sed -E "s/'/\\'/g")  
    local json="{\"text\": \"$text\"}"  
    local slack_url="${LibDeploy__SlackHookUrl}"  
    [[ ${LibRun__DryRun} -eq ${False} ]] && {  
        if ${LibDeploy__NoSlack}; then  
            hl.green "${original_text}"  
        else  
            curl -s -d "payload=$json" "${slack_url}" > /dev/null  
            if [[ $? -eq 0 ]]; then  
                info: "sent to Slack: [${text}]"  
            else  
                warning: "error sending to Slack, is your SLACK_URL set?"  
            fi  
        fi  
    }  
    [[ ${LibRun__DryRun} -eq ${True} ]] && run "send to slack [${text}]"  
}
```

2.17.2. `deploy.slack-ding`

```
deploy.slack-ding ()  
{  
    deploy.slack "<!here> $@"  
}
```

2.17.3. `deploy.validate-vpn`

```
deploy.validate-vpn ()  
{  
    .deploy.check-vpn "$@" || .deploy.vpn-error "$@"  
}
```

2.18. Module `dir`

2.18.1. `dir.count-slashes`

```
dir.count-slashes ()  
{  
    local dir="${1}"  
    echo "${dir}" | sed 's/[^\//g' | tr -d '\n' | wc -c | tr -d ' '  
}
```

2.18.2. `dir.expand-dir`

```
dir.expand-dir ()  
{  
    local dir="${1}"  
    if [[ "${dir:0:1}" != "/" && "${dir:0:1}" != "~" ]]; then  
        dir="$(pwd)/${dir}"  
    else  
        if [[ "${dir:0:1}" == "~" ]]; then  
            dir="${HOME}/${dir:1:1000}"  
        fi  
    fi  
    printf -- "%s" "${dir}"  
}
```

2.18.3. `dir.is-a-dir`

```
dir.is-a-dir ()  
{  
    local dir="${1}"  
    [[ -d "${dir}" ]]  
}
```

2.18.4. `dir.rsync-from-mac`

```
dir.rsync-from-mac ()  
{  
    dir.rsync-to "$1" "$2" "--iconv=utf-8-mac,utf-8 --xattrs" "${@:3}"  
}
```

2.18.5. `dir.rsync-from-mac-verbose`

```
dir.rsync-from-mac-verbose ()
```

```
{
    dir.rsync-to "$1" "$2" "--iconv=utf-8-mac,utf-8 --xattrs" "-v" "${@:3}"
}
```

2.18.6. `dir.rsync-to`

```
dir.rsync-to () {
    local from="$1"
    shift
    local to="$1"
    shift
    local -a extra_rsync_flags
    local sudo=false
    for flag in "$@"
    do
        if [[ ${flag} == "--sudo" ]]; then
            sudo=true
        else
            extra_rsync_flags+=("${flag}")
        fi
    done
    local extra_flags="${extra_rsync_flags[*]}"
    [[ -d "${from}" ]] || {
        error "usage: dir.rsync-to [ from-dir ] [ enclosing-to-dir ] [ custom-flags ]"
        "Directory ${from} does not exist."
        return 1
    }
    [[ -d "${to}" ]] || {
        error "usage: dir.rsync-to [ from-dir ] [ enclosing-to-dir ] [ custom-flags ]"
        "Directory ${to} does not exist."
        return 1
    }
    command -v rsync > /dev/null || package.install rsync
    command -v rsync > /dev/null || {
        error "Unable to find rsync even after installing. Aborting"
        return 2
    }
    local from_pwd=$(cd "${from}" || exit 1; pwd -P)
    h1.yellow "Starting RSync" "From      →  ${txtgrn}[${from_pwd}]" "Dest      →
${txtcyn}[${to}]" "Flags      →  ${txtred}[${extra_flags}]"
    local flags="-ah"
    local command="rsync ${flags} ${extra_flags} \"${from_pwd}\" \"${to}\"
${sudo} && command=\"sudo ${command}\""
    h1 "Rsync Command:" "${bldylw}${command}"
    run.ui.press-any-key "Press any key to run this command, or Ctrl-C to abort."
    h2 "Starting rsync process..."
    run.set-next show-output-on
    run "${command}"
```

```
}
```

2.18.7. `dir.short-home`

```
dir.short-home ()  
{  
    local dir="$1"  
    printf -- "%s" "${dir}" | sed -E "s#${HOME}#\~#g"  
}
```

2.18.8. `dir.with-file`

```
dir.with-file ()  
{  
    local file="$1"  
    local dir="${2:-$(pwd -P)}"  
    if [[ ${dir:0:1} != "/" ]]; then  
        dir="$(pwd -P)/${dir}"  
    fi  
    local _d="${dir}"  
    while true; do  
        local try="${_d}/${file}"  
        [[ -f "${try}" ]] && {  
            echo "${_d}"  
            return 0  
        }  
        _d="$(dirname "${_d}")"  
        if [[ "${_d}" == "/" || "${_d}" == "" ]]; then  
            [[ -f "${_d}/${file}" ]] || {  
                echo "No file ${file} was found in the path." 1>&2  
                return 1  
            }  
            echo "${_d}"  
            exit 0  
        fi  
    done  
    return 2  
}
```

2.19. Module `docker`

2.19.1. `docker.abort-if-down`

```
docker.abort-if-down ()  
{  
    local should_exit="${1:-true}"
```

```

inf 'Checking if Docker is running...'
docker ps 2> /dev/null > /dev/null
code=$?
if [[ ${code} == 0 ]]; then
    ui.closer.ok:
else
    ui.closer.not-ok:
    error "docker ps returned ${code}, is Docker running?"
[[ "${should_exit}" == "true" ]] && exit 127
    return 127
fi
}

```

2.19.2. docker.actions.build

```

docker.actions.build ()
{
    docker.build.container "$@"
}

```

2.19.3. docker.actions.clean

```

docker.actions.clean ()
{
    .docker.exec "docker-compose rm"
}

```

2.19.4. docker.actions.pull

```

docker.actions.pull ()
{
    local tag=${1:-'latest'}
    .docker.check-repo "${2}" || return 1
    .docker.exec "docker pull ${AppDockerRepo}:${tag}"
}

```

2.19.5. docker.actions.push

```

docker.actions.push ()
{
    local tag=${1:-$(.docker.next-version)}
    .docker.check-repo "${2}" || return 1
    docker.actions.tag latest
    [[ -n ${tag} ]] && docker.actions.tag "${tag}"
    .docker.check-repo || return 1
}

```

```
.docker.exec docker push "${AppDockerRepo}:${tag}"
[[ ${tag} != 'latest' ]] && .docker.exec docker push "${AppDockerRepo}:latest"
}
```

2.19.6. `docker.actions.setup`

```
docker.actions.setup ()
{
    setup.docker
    docker.pull
    docker.build
}
```

2.19.7. `docker.actions.start`

```
docker.actions.start ()
{
    .docker.exec "docker-compose start"
}
```

2.19.8. `docker.actions.stop`

```
docker.actions.stop ()
{
    .docker.exec "docker-compose stop"
}
```

2.19.9. `docker.actions.tag`

```
docker.actions.tag ()
{
    local tag=${1}
    [[ -z ${tag} ]] && return 1
    .docker.check-repo "${2}" || return 1
    .docker.exec docker tag "${AppDockerRepo}" "${AppDockerRepo}:${tag}"
}
```

2.19.10. `docker.actions.up`

```
docker.actions.up ()
{
    .docker.exec "docker-compose up"
```

```
}
```

2.19.11. docker.actions.update

```
docker.actions.update ()  
{  
    docker.build  
    docker.push  
}
```

2.19.12. docker.build.container

```
docker.build.container ()  
{  
    .docker.check-repo "${1}" || return 1  
    local tag=${AppDockerRepo}  
    .docker.exec "docker build -m 3G -c 4 --pull -t ${tag} . $*"  
}
```

2.19.13. docker.containers.clean

```
docker.containers.clean ()  
{  
    local -a args=("$@")  
    run "docker rm $(docker ps -q -a) ${args[*]}"  
}
```

2.19.14. docker.image.inspect

```
docker.image.inspect ()  
{  
    run.set-next show-output-on  
    local jq=" | jq"  
    [[ -z $(command -v jq) ]] && jq=  
    run "docker image inspect ${*} ${jq}"  
}
```

2.19.15. docker.image.rm

```
docker.image.rm ()  
{  
    run "docker image rm ${*}"
```

```
}
```

2.19.16. docker.images-named

```
docker.images-named ()  
{  
    local name="${1}"  
    local func="${2}"  
    docker.abort-if-down false || return 127  
    hl.subtle "Processing Docker images matching ${name} with function ${func}..."  
    local images=$(docker images | grep "^${name}" | sed 's/ */ /g' | cut -d ' ' -f  
3 | tr '\n' ' ')  
    ${func} "${images}"  
}
```

2.19.17. docker.images.clean

```
docker.images.clean ()  
{  
    local name=${1:-<none>}  
    docker.images-named "${name}" "docker.image.rm"  
}
```

2.19.18. docker.images.inspect

```
docker.images.inspect ()  
{  
    local name=${1:-<none>}  
    docker.images-named "${name}" "docker.image.inspect"  
}
```

2.19.19. docker.last-version

```
docker.last-version ()  
{  
    .docker.check-repo "${1}" || return 1  
    [[ -z ${AppDockerRepo} ]] && {  
        error "usage: docker.last-version organization/reponame:version"  
        return 1  
    }  
    .docker.last-version "$@"  
}
```

2.19.20. docker.next-version

```
docker.next-version ()  
{  
    .docker.check-repo "${1}" || return 1  
    [[ -z ${AppDockerRepo} ]] && {  
        error "usage: docker.next-version [ organization/repo-name:version ]"  
        return 1  
    }  
    .docker.next-version "$@"  
}
```

2.19.21. docker.set-repo

```
docker.set-repo ()  
{  
    [[ -n "$1" ]] && export AppDockerRepo="$1"  
}
```

2.20. Module dropbox

2.20.1. dropbox.unignore

```
dropbox.unignore ()  
{  
    local file=$(path.absolute "$1")  
    util.os  
    case "${BASHMATIC_OS}" in  
        darwin)  
            run "xattr -d com.dropbox.ignored \"${file}\""  
        ;  
        linux)  
            run "attr -r com.dropbox.ignored \"${file}\""  
        ;  
    esac  
}
```

2.21. Module file

2.21.1. dir.find

```
dir.find ()  
{  
    find . -name "*$1*" -type d -print
```

```
}
```

2.21.2. `dir.temp`

```
dir.temp ()  
{  
    local dir="$(file.temp)/$$/${RANDOM/284/_-}"  
    mkdir -p "${dir}" 2> /dev/null  
    [[ -n ${BASHMATIC_DEBUG} ]] && {  
        info "temporary folder is: ${dir}"  
        inf "it exists? "  
        [[ -d ${dir} ]] && ok:  
        [[ -d ${dir} ]] || not-ok:  
    }  
    printf "%s" "${dir}"  
    trap "rm -rf ${dir}" EXIT  
}
```

2.21.3. `file.ask.if-exists`

```
file.ask.if-exists ()  
{  
    local file="$1"  
    shift  
    local message="$*"  
    [[ -z "${message}" ]] && message="File ${file} exists. Overwrite?"  
    if [[ -f ${file} ]]; then  
        run.set-next on-decline-return  
        run.ui.ask "${message}" || return 1  
    fi  
    return 0  
}
```

2.21.4. `file.count.lines`

```
file.count.lines ()  
{  
    [[ -f "$1" ]] || return 1  
    wc -l "$1" | awk '{print $1}' | tr -d '\n'  
}
```

2.21.5. `file.count.words`

```
file.count.words ()  
{
```

```
[[ -f "$1" ]] || return 1
wc -w "$1" | awk '{print $1}' | tr -d '\n'
}
```

2.21.6. `file.exists-and-newer-than`

```
file.exists-and-newer-than () {
    local file="${1}"
    shift
    local minutes="${1}"
    shift
    if [[ -n "$(find "${file}" -mmin -"${minutes}" -print 2> /dev/null)" ]]; then
        return 0
    else
        return 1
    fi
}
```

2.21.7. `file.extension`

```
file.extension () {
    local filename="$1"
    printf "${filename##*.}"
}
```

2.21.8. `file.extension.remove`

```
file.extension.remove () {
    local filename="$1"
    printf "${filename%.*}"
}
```

2.21.9. `file.extension.replace`

```
file.extension.replace () {
    local ext="$1"
    shift
    [[ -z "$1" ]] && {
        info "USAGE: file.extension.replace <new-extension> file1 file2 ... "
        return 1
    }
```

```

ext=".${echo "${ext}" | tr -d '.'}"
local first=true
for file in "$@"
do
    ${first} || printf " "
    printf "%s%s" "$(file.strip.extension "${file}")" "${ext}"
    first=false
done
}

```

2.21.10. `file.find`

```

file.find ()
{
    find . -name "*$1*" -type f -print
}

```

2.21.11. `file.first-is-newer-than-second`

```

file.first-is-newer-than-second ()
{
    [[ "$1" -nt "$2" ]]
}

```

2.21.12. `file.gsub`

```

file.gsub ()
{
    local file="$1"
    shift
    local find="$1"
    shift
    local replace="$1"
    shift
    local runtime_options="$*"
    [[ ! -s "${file}" || -z "${find}" || -z "${replace}" ]] && {
        error "Invalid usage of file.gsub - " "USAGE: file.gsub <file> <find-regex>
<replace-regex>" "EG: file.gsub ~/.bashrc '^export EDITOR=vi' 'export EDITOR=gvim'"
        return 1
    }
    ${GrepCommand} -q "${find}" "${file}" || return 0
    [[ -z "${runtime_options}" ]] || run.set-next "${runtime_options}"
    run "sed -i'' -E 's/${find}/${replace}/g' \"${file}\""
}

```

2.21.13. file.install-with-backup

```
file.install-with-backup ()  
{  
    local source="$1"  
    shift  
    if [[ ! -f "${source}" ]]; then  
        error "file ${source} can not be found"  
        return 1  
    fi  
    local dest="$1"  
    shift  
    [[ -z ${dest} ]] && {  
        error "usage: file.install-with-backup <source> <dest> [ bak | folder ]"  
        return 1  
    }  
    local backup_strategy="${1:-bak}"  
    shift  
    local backup_fn=".file.backup.strategy.${backup_strategy}"  
    is.a-function "${backup_fn}" || {  
        error "Invalid backup method '${backup_strategy}': supported are: 'bak' and  
'folder'"  
        return 1  
    }  
    if [[ -f "${dest}" ]]; then  
        if [[ -z $(diff "${dest}" "${source}" 2>&1) ]]; then  
            info: "The two files ${source} and ${dest} are identical."  
            return 0  
        else  
            if [[ ${LibFile__ForceOverwrite} -eq 1 ]]; then  
                info "file ${dest} already exists, but overwrite flag was set, so  
overwriting."  
                run.set-next show-output-on  
                run "cp -v \"${source}\" \"${dest}\""  
                return 0  
            else  
                info "creating a backup of ${dest}..."  
                ${backup_fn} "${dest}" && {  
                    success "File ${dest} has been backed up."  
                }  
            fi  
        fi  
    fi  
    run "mkdir -p $(dirname "${dest}")"  
    run.set-next show-output-on  
    run "cp -v ${source} ${dest}"  
}
```

2.21.14. `file.last-modified-date`

```
file.last-modified-date ()  
{  
    stat -f "%Sm" -t "%Y-%m-%d" "$1"  
}
```

2.21.15. `file.last-modified-millis`

```
file.last-modified-millis ()  
{  
    echo -n "$( /usr/bin/stat -f %m "$1" )000"  
}
```

2.21.16. `file.last-modified-year`

```
file.last-modified-year ()  
{  
    stat -f "%Sm" -t "%Y" "$1"  
}
```

2.21.17. `file.list.filter-existing`

```
file.list.filter-existing ()  
{  
    for file in "$@"  
    do  
        [[ -f "${file}" ]] && echo "${file}"  
    done  
}
```

2.21.18. `file.list.filter-non-empty`

```
file.list.filter-non-empty ()  
{  
    for file in "$@"  
    do  
        [[ -s "${file}" ]] && echo "${file}"  
    done  
}
```

2.21.19. file.normalize-files

```
file.normalize-files ()  
{  
    trap 'return 1' INT  
    run.set-all abort-on-error  
    local file  
    local counter=0  
    for file in "$@"  
    do  
        [[ -f "${file}" ]] || {  
            error "File '${file}' does not exist, continue."  
            continue  
        }  
        local new_name=$(file.print-normalized-name "${file}")  
        [[ "${file}" == "${new_name}" ]] && continue  
        local dir_name=$(dirname "${new_name}")  
        [[ -n ${dir_name} ]] && {  
            [[ -d ${dir_name} ]] || run "mkdir -p \${dir_name}\\"  
        }  
        if run.config.is-dry-run; then  
            info "mv -v \"${file}\" \"${new_name}\""  
        else  
            run.set-next show-output-on  
            run "mv -fv \"${file}\" \"${new_name}\""  
            counter=$(( counter + 1 ))  
        fi  
    done  
    hr  
    inf "Total of ${counter} files have been renamed."  
    return  
}
```

2.21.20. file.print-normalized-name

```
file.print-normalized-name ()  
{  
    local file="$1"  
    echo -e "${file}" | ascii-pipe | tr '[:upper:]' '[:lower:]' | sed -E 's/ /-/g;  
s/[^A-Za-z0-9.-]/-/g; s/---+---/g' | tr -d '\r\n'
```

2.21.21. file.size

```
file.size ()  
{  
    util.os
```

```

if [[ ${BASHMATIC_OS} =~ linux ]]; then
    stat -c %s "$1"
else
    file.stat "$1" st_size
fi
}

```

2.21.22. `file.size.gb`

```

file.size.gb ()
{
    local file="$1"
    shift
    local s=$(file.size "${file}")
    local gb=$(echo $((s / 10 / 1024 / 1024 )) | sedx 's/([0-9][0-9])$/.\1/g')
    printf "%1f Gb" "${gb}"
}

```

2.21.23. `file.size.mb`

```

file.size.mb ()
{
    local file="$1"
    shift
    local s=$(file.size "${file}")
    local mb=$(echo $((s / 10/ 1024)) | sedx 's/([0-9][0-9])$/.\1/g')
    printf "%1f MB" "${mb}"
}

```

2.21.24. `file.source-if-exists`

```

file.source-if-exists ()
{
    local file
    for file in "$@"
    do
        [[ -f "${file}" ]] && source "${file}"
    done
}

```

2.21.25. `file.stat`

```

file.stat ()
{
    local file="$1"

```

```

local field="$2"
[[ -f ${file} ]] || {
    error "file ${file} is not found. Usage: file.stat <filename> <stat-field-name>"
    info "eg: ${bldylw}file.stat README.md st_size"
    return 1
}
[[ -n ${field} ]] || {
    error "Second argument field is required."
    info "eg: ${bldylw}file.stat README.md st_size"
    return 2
}
eval "$(stat -s "${file}" | tr ' ' '\n' | sed 's/^/local /g')"
echo "${!field}"
}

```

2.21.26. `file.strip.extension`

```

file.strip.extension ()
{
    file.extension.remove "$@"
}

```

2.21.27. `file.temp`

```

file.temp ()
{
    local host="${HOST:-${HOSTNAME:-$(hostname)}}"
    local user="${USER:-$(whoami)}"
    local temp_file_pattern=".bashmatic.${host}.${user}.${*: /}"
    local n=$((epoch))
    local t=$((n % 99991))
    local file="/tmp/${temp_file_pattern}${n}$$$${t}${RANDOM}${RANDOM}"
    find "$(dirname ${file})" -maxdepth 1 -type f -name "${temp_file_pattern}*" \
-mtime +1 -delete > /dev/null 2>&1
    echo "${file}"
}

```

2.21.28. `files.find`

```

files.find ()
{
    local folder="$1"
    local pattern="${2}"
    [[ -z ${folder} || -z ${pattern} ]] && {
        echo "usage: files.find <folder> <pattern>" 1>&2
    }
}

```

```

        return 1
    }
    find "$1" -name "${pattern}"
}

```

2.21.29. **files.map**

```

files.map ()
{
    local folder="${1}"
    local pattern="${2}"
    local array="${3}"
    local -a files
    if bashmatic.bash.version-four-or-later; then
        mapfile -t files <<(files.find "${folder}" "${pattern}")
    else
        files=()
        while IFS=' ' read -r line; do
            files+=("$line")
        done <<(files.find "${folder}" "${pattern}")
    fi
    if [[ -n ${array} ]]; then
        printf "%s" "unset ${array}; declare -a ${array}; ${array}=(${files[*]})"
    export ${array}
    else
        printf "%s" "${files[*]}"
    fi
}

```

2.21.30. **files.map.shell-scripts**

```

files.map.shell-scripts ()
{
    files.map "$1" '*.sh' "$2"
}

```

2.21.31. **ls.gb**

```

ls.gb ()
{
    du -g -d 1 "$@" | sort -rn
}

```

2.21.32. ls.mb

```
ls.mb ()  
{  
    du -m -d 1 "$@" | sort -rn  
}
```

2.22. Module file-helpers

2.23. Module files-normalize

2.23.1. files.normalize-tree

```
files.normalize-tree ()  
{  
    local command="__files.normalize-tree"  
    ("${command}" "$@")  
}
```

2.24. Module flatten

2.24.1. flatten-file

```
flatten-file ()  
{  
    local path="$1"  
    local dest="${2}"  
    [[ ${dest[-1]} == "/" ]] && dest="${dest:0:-1}"  
    errors=()  
    [[ -f "${path}" ]] || errors+=("File [${path}] was not found.")  
    [[ -d "${dest}" ]] || errors+=("Directory [${dest}] does not exist.")  
    [[ ${#errors[@]} -eq 0 ]] || {  
        error "${errors[@]}" 1>&2  
        return 1  
    }  
    local target  
    local space=" "  
    local enclosing="$(dirname "$(dirname "${path}")")"  
    target="$(echo "$(dirname "${path}")"-$(basename "${path}") | sed -E "s/[${space}]/-/g;")"  
    target="${dest} ${bldred}${target}/${enclosing}/\}\}"  
    if ((flag_verbose)); then  
        inf "COPY: ${bldgrn}\\""${path}"\\" [${bldylw}\\"${target}${bldgrns}\"]" 1>&2  
    else  
        printf "${txtblu}" 1>&2
```

```

fi
local command="cp -v \"${path}\\" ${bldylw}\"${target}\""
printf -- "${command}"
return 0
}

```

2.25. Module ftrace

2.25.1. ftrace-in

```

ftrace-in ()
{
    local func=$1
    shift
    local args="$*"
    [[ -z ${TraceON} ]] && return
    export __LibTrace__StackLevel=$(( ${__LibTrace__StackLevel} + 1 ))
    printf "    %*s ${bldylw}%s${bldblu}(%s)${clr}\n" ${__LibTrace__StackLevel} \
"$func" "$args" 1>&2
}

```

2.25.2. ftrace-off

```

ftrace-off ()
{
    unset TraceON
}

```

2.25.3. ftrace-on

```

ftrace-on ()
{
    export TraceON=true
}

```

2.25.4. ftrace-out

```

ftrace-out ()
{
    local func=$1
    shift
    local code=$1
    shift
    local msg="$*"

```

```

[[ -z ${TraceON} ]] && return
local color="${bldgrn}"
[[ ${code} -ne 0 ]] && color="${bldred}"
printf "%*s ${bldlw}%s() ${color} \u201d %d %s\n\n" ${__LibTrace__StackLevel} ' '
"${func}" "${code}" "${msg}" 1>&2
export __LibTrace__StackLevel=$(( ${__LibTrace__StackLevel} - 1 ))
}

```

2.26. Module gem

2.26.1. g-i

```

g-i ()
{
    gem.install "$@"
}

```

2.26.2. g-u

```

g-u ()
{
    gem.uninstall "$@"
}

```

2.26.3. gem.cache-installed

```

gem.cache-installed ()
{
    gem.configure-cache
    if [[ ! -f "${LibGem__GemListCache}" || ! -s "${LibGem__GemListCache}" || -z "${LibGem__GemListCache}" ]]; then
        mkdir -p $(dirname ${LibGem__GemListCache})
        gem list > "${LibGem__GemListCache}" 2> /dev/null
    fi
    [[ -s "${LibGem__GemListCache}" ]]
}

```

2.26.4. gem.cache-list

```

gem.cache-list ()
{
    cat "${LibGem__GemListCache}"
}

```

2.26.5. `gem.cache-refresh`

```
gem.cache-refresh ()  
{  
    ( gem.configure-cache  
    gem.clear-cache  
    gem.cache-installed ) > /dev/null  
}
```

2.26.6. `gem.cache-reset`

```
gem.cache-reset ()  
{  
    gem.cache-refresh  
}
```

2.26.7. `gem.changelog-generate`

```
gem.changelog-generate ()  
{  
    local project="$1"  
    [[ -z ${project} ]] && {  
        error "usage: gem.changelog-generate username/repo"  
        return 1  
    }  
    local user  
    local repo  
    user="${project%/*/}"  
    repo="${project%/*//}"  
    gem.install github_changelog_generator  
    [[ -z ${GITHUB_TOKEN} ]] && {  
        error "Please set GITHUB_TOKEN to avoid hitting 50 reqs/minute API limit."  
        exit 1  
    }  
    run "github_changelog_generator --project ${repo} --user ${user} -t ${  
GITHUB_TOKEN} --no-verbose"  
    ls -al CHANGELOG.md  
}
```

2.26.8. `gem.clear-cache`

```
gem.clear-cache ()  
{  
    cp /dev/null "${LibGem__GemListCache}"  
}
```

2.26.9. `gem.configure-cache`

```
gem.configure-cache ()  
{  
    export LibGem__GemListCacheBase="${BASHMATIC_TEMP}/.gem/gem.list"  
    export LibGem__GemListCache=  
    export LibGem__GemInstallFlags=" -N --force --quiet "  
    local ruby_version=$(ruby.numeric-version)  
    export LibGem__GemListCache="${LibGem__GemListCacheBase}.${ruby_version}"  
    local dir=$(dirname "${LibGem__GemListCache}")  
    [[ -d ${dir} ]] || mkdir -p "${dir}" > /dev/null  
}
```

2.26.10. `gem.ensure-gem-version`

```
gem.ensure-gem-version ()  
{  
    local gem=$1  
    local gem_version=$2  
    [[ -z ${gem} || -z ${gem_version} ]] && return  
    gem.cache-installed  
    if [[ -z $(grep "${gem} (${gem_version}) < ${LibGem__GemListCache}" ) ]]; then  
        gem.uninstall "${gem}"  
        gem.install "${gem}" "${gem_version}"  
    else  
        info "gem ${gem} version ${gem_version} is already installed."  
    fi  
}
```

2.26.11. `gem.gemfile.bundler-version`

```
gem.gemfile.bundler-version ()  
{  
    [[ -f Gemfile.lock ]] && grep -A2 BUNDLED Gemfile.lock | tail -1 | tr -d '\n'  
}
```

2.26.12. `gem.gemfile.version`

```
gem.gemfile.version ()  
{  
    local gem=$1  
    local gemfile="${2:-Gemfile.lock}"  
    [[ -z ${gem} ]] && return  
    [[ -f ${gemfile} ]] || {  
        error "Can't find Gemfile ${gemfile} in the current directory" 1>&2  
        return 1  
    }  
}
```

```

    }
    grep -E -e " ${gem} \([0-9]\) \"${gemfile}\" | awk '{print $2}' | sed 's/[()]///g'
}

```

2.26.13. `gem.global.latest-version`

```

gem.global.latest-version ()
{
    local gem="$1"
    [[ -z ${gem} ]] && return
    declare -a versions=(${(gem.global.versions "${gem}")})
    local max=0
    local max_version=${versions[0]}
    for v in "${versions[@]}"
    do
        vi=$(util.ver-to-i "${v}")
        if [[ ${vi} -gt ${max} ]]; then
            max=${vi}
            max_version="${v}"
        fi
    done
    printf "%s" "${max_version}"
}

```

2.26.14. `gem.global.versions`

```

gem.global.versions ()
{
    local gem=$1
    [[ -z ${gem} ]] && return
    gem.cache-installed
    grep -E -e "^\${gem}" < "${LibGem_GemListCache}" | sedx "s/^${gem}/;s/[(),]/;/g"
}

```

2.26.15. `gem.install`

```

gem.install ()
{
    .gem.verify-name "$@" || return 1
    local gem_name="$1"
    local gem_version="$2"
    local gem_version_flags=
    local gem_version_name=
    gem_version=${gem_version:-$(gem.version "${gem_name}")}

    if [[ -z ${gem_version} ]]; then

```

```

    gem_version_name=latest
    gem_version_flags=
else
    gem_version_name="${gem_version}"
    gem_version_flags="--version ${gem_version}"
fi
if gem.is-installed "${gem_name}" "${gem_version}"; then
    info "gem ${bldylw}${gem_name} (${bldgrn}${gem_version_name}${bldylw})"
${txtblu} is already installed"
else
    info "installing ${bldylw}${gem_name} ${bldgrn}(${gem_version_name})${txtblu}
}..."
run "gem install ${gem_name} ${gem_version_flags} ${LibGem__GemInstallFlags}"
if [[ ${LibRun__LastExitCode} -eq 0 ]]; then
    rbenv rehash > /dev/null 2> /dev/null
    gem.cache-refresh
else
    error "Unable to install gem ${bldylw}${gem_name}"
fi
return "${LibRun__LastExitCode}"
fi
}

```

2.26.16. `gem.is-installed`

```

gem.is-installed ()
{
    local gem=$1
    local version=$2
    gem.cache-installed > /dev/null
    if [[ -z ${version} ]]; then
        grep -q -E -e "^\${gem} \\" "\${LibGem__GemListCache}"
    else
        grep -E -e "^\${gem} \\" "\${LibGem__GemListCache}" | grep -E -q -e "\${version}"
    fi
}

```

2.26.17. `gem.remote.version`

```

gem.remote.version ()
{
    [[ -z "$1" ]] && return
    gem search "$1" --remote -e | sedx "s/^${1} //g; s/[(),]///g"
}

```

2.26.18. `gem.uninstall`

```
gem.uninstall ()  
{  
    .gem.verify-name "$@" || return 1  
    local gem_name=$1  
    local gem_version=$2  
    gem.is-installed "${gem_name}" "${gem_version}" || {  
        info "gem ${bldylw}${gem_name}${txtblu} is not installed"  
        return  
    }  
    local gem_flags="-x -I --force"  
    if [[ -z ${gem_version} ]]; then  
        gem_flags="${gem_flags} -a"  
    else  
        gem_flags="${gem_flags} --version ${gem_version}"  
    fi  
    run "gem uninstall ${gem_name} ${gem_flags}"  
    gem.clear-cache  
    return "${LibRun__LastExitCode}"  
}
```

2.26.19. `gem.version`

```
gem.version ()  
{  
    local gem="$1"  
    local default="$2"  
    [[ -z ${gem} ]] && return  
    local version  
    [[ -f Gemfile.lock ]] && version=$(gem.gemfile.version "${gem}")  
    if [[ -z ${version} ]]; then  
        if gem.is-installed "${gem}"; then  
            version=$(gem.global.latest-version "${gem}")  
        else  
            version=$(gem.remote.version "${gem}")  
        fi  
    fi  
    [[ -z ${version} && -n ${default} ]] && version=${default}  
    printf "%s" "${version}"  
}
```

2.27. Module `git`

2.27.1. `git.branch.current`

```

git.branch.current ()
{
    git rev-parse --abbrev-ref HEAD
}

```

2.27.2. `git.cfg.get`

```

git.cfg.get ()
{
    local section="global"
    if [[ "$1" == "local" || "$1" == "global" ]]; then
        section="$1"
        shift
    fi
    local command="get"
    local cmd
    if [[ -z "$*" ]]; then
        cmd="git config --${section} --list"
    else
        if [[ "$*" =~ ^[a-zA-Z]*$ ]]; then
            for token in "$@"
            do
                cmd="git config --${section} --${command} ${token}"
            done
        else
            cmd="git config --${section} --get-all '$*'"
        fi
    fi
    h1 "${cmd}"
    eval "${cmd}"
}

```

2.27.3. `git.cfgu`

```

git.cfgu ()
{
    [[ -z $1 ]] && {
        git config --global -l
        return
    }
    if [[ -n $2 ]]; then
        rm -f ~/.gitconfig.lock
        git config --global --replace-all user."$1" "$2"
    else
        if [[ $1 =~ - ]]; then
            git config --global "$1"
        else

```

```
        git config --global user."$1"
    fi
fi
}
```

2.27.4. `git.commits.last.message`

```
git.commits.last.message ()
{
    git log --pretty=format:"%s" -1
}
```

2.27.5. `git.commits.last.sha`

```
git.commits.last.sha ()
{
    git log --pretty=format:"%H" -1
}
```

2.27.6. `git.config.kigster`

```
git.config.kigster ()
{
    [[ ${git.cfgu name} == "Konstantin Gredeskoul" && ${git.cfgu email} ==
"kigster@gmail.com" ]] && return 0
    git.cfgu name "Konstantin Gredeskoul"
    git.cfgu email "kigster@gmail.com"
}
```

2.27.7. `git.configure-auto-updates`

```
git.configure-auto-updates ()
{
    export LibGit__StaleAfterThisManyHours="${LibGit__StaleAfterThisManyHours:-1}"
    export LibGit__LastUpdateTimestampFile="${BASHMATIC_TEMP}/.config/${(echo "${USER}" |
shasum.sha-only-stdin)}"
    mkdir -p "$(dirname "${LibGit__LastUpdateTimestampFile}")"
}
```

2.27.8. `git.current-branch`

```
git.current-branch ()
{
```

```
git branch --no-color | grep -F "*" | cut -f 2 -d " "
```

2.27.9. `git.generate-changelog`

```
git.generate-changelog ()  
{  
    local token=$(git config user.token)  
    git.is-valid-repo || return 2  
    gem.install github_changelog_generator  
    local -a remote_parts  
    remote_parts=($(git.parse-remote "$(git.remote)"))  
    local user=${remote_parts[3]}  
    local repo=${remote_parts[4]}  
    local host=${remote_parts[2]}  
    [[ ${host} =~ github.com ]] || {  
        error "Can only generate changelog for Github Repos at the moment, sorry."  
        return 1  
    }  
    run "rm -f CHANGELOG.md"  
    run "bundle exec github_changelog_generator --project ${repo}/.git/ --user ${user} -t ${token} --no-verbose"  
    [[ -s "CHANGELOG.md" ]] || {  
        error "CHANGELOG.md has not been generated."  
        return 1  
    }  
    success "CHANGELOG.md is ready."  
    return 0  
}
```

2.27.10. `git.is-it-time-to-update`

```
git.is-it-time-to-update ()  
{  
    local last_update_at=$(git.last-update-at)  
    local second_since_update=$(git.seconds-since-last-pull "${last_update_at}")  
    local update_period_seconds=$((LibGit__StaleAfterThisManyHours * 60 * 60))  
    [[ ${second_since_update} -gt ${update_period_seconds} ]]  
}
```

2.27.11. `git.is-valid-repo`

```
git.is-valid-repo ()  
{  
    if [[ ! -d .git ]]; then  
        error "Please run this script at the root of your project / git repo." 1>&2  
    fi  
}
```

```

        return 1
    fi
}
```

2.27.12. `git.last-update-at`

```

git.last-update-at ()
{
    git.configure-auto-updates
    local file="${1:+"${LibGit__LastUpdateTimestampFile}""
    local last_update=0
    if [[ ${LibGit__ForceUpdate} -eq 0 && -f ${file} ]]; then
        last_update=$(cat "$file" | tr -d '\n')
    else
        last_update=0
    fi
    printf "%d" ${last_update}
}
```

2.27.13. `git.local-vs-remote`

```

git.local-vs-remote ()
{
    local upstream=${1:-'@{u}'}
    local local_repo=$(git rev-parse @)
    local remote_repo=$(git rev-parse "$upstream")
    local base=$(git merge-base @ "$upstream")
    if [[ -n ${BASHMATIC_DEBUG} ]]; then
        printf "
    pwd      = $(pwd)
    remote   = $(git.remotes)
    base     = ${base}
    upstream = ${upstream}
    local_repo = ${local_repo}
    remote_repo = ${remote_repo}
"
    fi
    local result=
    if [[ "${local_repo}" == "${remote_repo}" ]]; then
        result="ok"
    else
        if [[ "${local_repo}" == "${base}" ]]; then
            result="behind"
        else
            if [[ "${remote_repo}" == "${base}" ]]; then
                result="ahead"
            else
                result="diverged"
            fi
        fi
    fi
}
```

```

        fi
    fi
fi
printf '%s' ${result}
[[ ${result} == "ok" ]] && return 0
return 1
}

```

2.27.14. `git.open.current`

```

git.open.current ()
{
    local url=$(git.repo.current)
    info "git.open() -> ${bldylw}${url}"
    open -a 'Google Chrome' "${url}"
}

```

2.27.15. `git.open.repo`

```

git.open.repo ()
{
    local url=$(git.repo)
    info "git.open.repo() -> ${bldylw}${url}"
    open -a 'Google Chrome' "${url}"
}

```

2.27.16. `git.parse-remote`

```

git.parse-remote ()
{
    local url="${1:-$(git.remote)}"
    local re="^(\https|git)(:\//|@)([^/:]+)[/:]([^\/:]+)\//(.+)(\..git)?$"
    [[ $url =~ $re ]] || {
        error "git remote [${url}] does not match regular expression." 1>&2
        return 1
    }
    local protocol="${BASH_REMATCH[1]}"
    local separator="${BASH_REMATCH[2]}"
    local hostname="${BASH_REMATCH[3]}"
    local user="${BASH_REMATCH[4]}"
    local repo="${BASH_REMATCH[5]}"
    printf "%s %s %s %s %s\n" "${protocol}" "${separator}" "${hostname}" "${user}" "${repo}"
}

```

2.27.17. `git.quiet`

```
git.quiet ()  
{  
    [[ -n ${LibGit__QuietUpdate} ]]  
}
```

2.27.18. `git.remote`

```
git.remote ()  
{  
    if git.remotes | grep -q "git@"; then  
        git.remotes | egrep "git@" | sort | head -1  
    else  
        git.remotes | sort | head -1  
    fi  
}
```

2.27.19. `git.remotes`

```
git.remotes ()  
{  
    git remote -v | awk '{print $2}' | uniq
```

2.27.20. `git.repo`

```
git.repo ()  
{  
    local url=$(git remote get-url origin | sed -E  
's/git@https:\//\//g;s/com:/com\//g')  
    printf -- "%s" "${url}"  
}
```

2.27.21. `git.repo-is-clean`

```
git.repo-is-clean ()  
{  
    local repo="${1:-$BASHMATIC_HOME:="$HOME/.bashmatic"}"  
    cd "${repo}" > /dev/null  
    if [[ -z $(git status -s) ]]; then  
        cd - > /dev/null  
        return 0  
    else
```

```
    cd - > /dev/null
    return 1
}
}
```

2.27.22. `git.repo.current`

```
git.repo.current ()
{
    local url="$(git.repo)"
    local current_branch="$(git.branch.current)"
    local readme="README.md"
    for file in README.md README README.txt README.adoc CHANGELOG.md
    do
        if [[ -s "${file}" ]]; then
            export readme="${file}"
            break
        fi
    done
    local full_url="${url}/blob/${current_branch}/${readme}"
    printf -- "%s" "${full_url}"
}
```

2.27.23. `git.repo.latest-local-tag`

```
git.repo.latest-local-tag ()
{
    git tag -l | sort | tail -1
}
```

2.27.24. `git.repo.latest-remote-tag`

```
git.repo.latest-remote-tag ()
{
    local repo_url="$1"
    git ls-remote --tags --sort="v:refname" "${repo_url}" | grep -E '\-v
'(latest|stable)' | grep -E -v '^{' | tail -1 | awk 'BEGIN{FS="/" }{print $3}'
}
```

2.27.25. `git.repo.next-local-tag`

```
git.repo.next-local-tag ()
{
    local tag=$(git.repo.latest-local-tag)
    [[ -z ${tag} ]] && tag="0.0.0"
```

```

ruby -e "prefix='${tag}'.gsub(/^(^\d+).*/,'\\1'); version='${tag}'.gsub(/^\d.*/, '').split(/\./).map(&to_i); version[2]+=1; puts \"#${prefix}#{version.join('.')}\""
}

```

2.27.26. `git.repo.remote-to-git@`

```

git.repo.remote-to-git@ ()
{
    local f=".git/config"
    if [[ -f "$f" ]]; then
        grep -q "url = git@" "$f" && {
            info "The repo is already using git@ syntax for the remote."
            return 0
        }
        cat "${f}" | sed -E 's#url = https://github\.com/([^\/]*)/#url = git@github\.com:\1#g' > "${f}.ssh"
        mv "${f}" "${f}.https"
        cd .git
        ln -nfs config.ssh config
        cd - > /dev/null
        hr
        info "Created an ssh version of .git/config file, and symlinked it:"
        ls -l .git/config*
        info "Your new remote:"
        info $(grep "git@" "${f}")
        hr
    fi
}

```

2.27.27. `git.save-last-update-at`

```

git.save-last-update-at ()
{
    echo $(epoch) > "${LibGit__LastUpdateTimestampFile}"
}

```

2.27.28. `git.seconds-since-last-pull`

```

git.seconds-since-last-pull ()
{
    local last_update="$1"
    local now=$(epoch)
    printf $((now - last_update))
}

```

2.27.29. git.squash

```
git.squash ()  
{  
    local number="${1}"  
    is.numeric "${number}" || {  
        info "USAGE: git.squash <number> # of commits to go back"  
        return  
    }  
    run "git reset --soft HEAD~${number}"  
    info "We've squashed down ${number} commits locally."  
    info "Now, you must commit this squash, and likely force push."  
}
```

2.27.30. git.sync

```
git.sync ()  
{  
    local dir="$(pwd -P)"  
    cd "${BASHMATIC_HOME}" > /dev/null  
    git.repo-is-clean || {  
        output.is-ssh || warning "${BASHMATIC_HOME} has locally modified files."  
        "Please commit or stash them to allow auto-upgrade to function as designed." 1>&2  
        cd "${dir}" > /dev/null  
        return 1  
    }  
    if is-debug; then  
        git.update-repo-if-needed  
    else  
        git.update-repo-if-needed 1>&2 > /dev/null  
    fi  
    cd "${dir}" > /dev/null  
    return 0  
}
```

2.27.31. git.sync-dirs

```
git.sync-dirs ()  
{  
    local pattern="${1:-'*'}"  
    set -e  
    run.set-all abort-on-error  
    for dir in $(find . -type d -maxdepth 1 -name "${pattern}*")  
    do  
        hl.yellow-on-gray "syncing [$dir]..."  
        cd "$dir" > /dev/null  
        run "git pull --rebase"
```

```
    cd - > /dev/null
    done
}
```

2.27.32. `git.sync-remote`

```
git.sync-remote () {
    git.is-it-time-to-update || return 0
    if git.quiet; then
        ( git remote update && git fetch ) 2>&1 > /dev/null
    else
        run "git remote update && git fetch"
    fi
    local git_status=$(git.local-vs-remote)
    if [[ ${git_status} == "behind" ]]; then
        git.quiet || run "git pull --rebase"
        git.quiet && git pull --rebase 2>&1 > /dev/null
    else
        if [[ ${git_status} != "ahead" ]]; then
            git.save-last-update-at
        else
            if [[ ${git_status} != "ok" ]]; then
                error "Report $(pwd) is ${status} compared to the remote." "Please fix
manually to continue."
                return 1
            fi
        fi
    fi
    git.save-last-update-at
    return 0
}
```

2.27.33. `git.update-repo-if-needed`

```
git.update-repo-if-needed () {
    git.is-it-time-to-update && git.sync-remote
}
```

2.27.34. `git.upstream`

```
git.upstream () {
    local this_branch=$(git.current-branch)
    this_branch=${this_branch:-main}
```

```
    run.set-next show-output-on
    run "git branch --set-upstream-to=origin/${this_branch} ${this_branch}"
}
```

2.28. Module `github`

2.28.1. `github.clone`

```
github.clone ()
{
    test -n "$1" && github.validate && run "git clone git@github.com:${(github.org)}/$1"
}
```

2.28.2. `github.org`

```
github.org ()
{
    local namespace="$1"
    if [[ -z ${namespace} ]]; then
        git config --global --get user.github
    else
        git config --global --unset user.github
        git config --global --add user.github "${namespace}"
    fi
}
```

2.28.3. `github.setup`

```
github.setup ()
{
    local namespace="$(github.org)"
    if [[ -z "${namespace}" ]]; then
        unset GITHUB_ORG
        run.ui.ask-user-value GITHUB_ORG "Please enter the name of your Github Organization:" || return 1
        github.org "${GITHUB_ORG}"
        echo
        h2 "Your github organization was saved in your ~/.gitconfig file." "To change it in the future, run: ${bldylw}github.org ${blgrn}new-organization"
        echo
    fi
    github.org > /dev/null
}
```

2.28.4. `github.validate`

```
github.validate ()  
{  
    inf "Validating Github Configuration..."  
    if github.org > /dev/null; then  
        ok:  
        return 0  
    else  
        not-ok:  
        github.setup  
        return $?  
    fi  
}
```

2.29. Module `gpg`

2.29.1. `gpg.install`

```
gpg.install ()  
{  
    [[ -z ${BASHMATIC_OS} ]] && util.os  
    gpg.install-deps  
    case "${BASHMATIC_OS}" in  
        darwin)  
            brew.install.packages "gnupg"  
            ;  
        linux)  
            run "sudo apt-get install gnupg -yyq"  
            ;  
    esac  
}
```

2.29.2. `gpg.install-deps`

```
gpg.install-deps ()  
{  
    [[ -z ${BASHMATIC_OS} ]] && util.os  
    case "${BASHMATIC_OS}" in  
        darwin)  
            brew.install.packages "coreutils gawk gnu-sed git curl gzip"  
            ;  
        linux)  
            run "sudo apt-get install -yqq bash gnupg2 git tar xz-utils coreutils gawk  
grep sed"  
            run "sudo apt-get install -yqq gzip bzip lzip file jq curl"
```

```
;  
esac  
}
```

2.29.3. gpg.key-for-github

```
gpg.key-for-github ()  
{  
    [[ -z ${BASHMATIC_OS} ]] && util.os  
    if ! command -v gpg > /dev/null; then  
        gpg.install  
    fi  
    local -a info=(${gpg.name-and-email})  
    local name="${info[0]}"  
    local email="${info[1]}"  
    local -a keys=(${gpg.my-keys})  
    if [[ ${#keys[@]} -gt 0 ]]; then  
        gpg.my-keys  
        return 0  
    fi  
    local key_spec=$(mktemp)  
    echo "%echo Generating a basic OpenPGP key  
Key-Type: 1  
Key-Length: 4096  
Name-Real: ${name}  
Name-Email: ${email}  
Expire-Date: 0  
%no-protection  
%commit  
%echo done  
" > "${key_spec}"  
    cat "${key_spec}"  
    gpg --batch --gen-key "${key_spec}" > /dev/null  
}
```

2.29.4. gpg.my-keys

```
gpg.my-keys ()  
{  
    local -a info=(${gpg.name-and-email})  
    local name="${info[0]}"  
    local email="${info[1]}"  
    declare -a keys=(${gpg --list-secret-keys --keyid-format=long | grep -B 3 -E "^\uid  
*\[ultimate\] ${name}.*$" | grep -E '^sec' | cut -d '/' -f 2 | sed 's/ .*$/\g')})  
    if [[ ${#keys[@]} -gt 0 ]]; then  
        printf "\n${bldylw}Your GPG keys are:${clr}\n" 1>&2  
        echo "${keys[*]}" | tr ' ' "\n"  
        local len=${#keys[@]}
```

```

local index
while true; do
    if [[ -n ${index} && ${index} -ge 0 && ${index} -lt ${#keys[@]} ]]; then
        local key_id="${keys[$index]}"
        printf -- "Key ID is ${bldylw}${key_id}\n\n"
        run "git config --global user.signingkey ${key_id}"
        gpg --armor --export "${key_id}" | pbcopy
        gpg --armor --export "${key_id}"
        printf -- "${clr}NOTE: ${bldylw}the key is now in your clipboard${clr}\n\n"
    printf -- "${clr}NOTE: gpg key for your ~/.gitconfig is ${bldgrn}\n${key_id}\n\n"
    hr
    return $?
else
    [[ -n ${index} ]] && printf "${bldred}Invalid answer, expecting a\nnumber between 1 and ${len}.${clr}\n"
    run.ui.ask-user-value index "Which key would you like to print [1-\n${len}]? ${clr}"
    index=$(( index - 1 ))
fi
done
else
    echo "No gpg keys found matching name ${name}." 1>&2
    return 1
fi
return 0
}

```

2.29.5. `gpg.name-and-email`

```

gpg.name-and-email ()
{
    local name=$(git config --global --get user.name)
    local email=$(git config --global --get user.email)
    [[ -z ${name} ]] && run.ui.ask-user-value name "Your full name:"
    [[ -z ${email} ]] && run.ui.ask-user-value email "Your full Email:"
    echo "${name} ${email}"
}

```

2.30. Module `is`

2.30.1. `is-validations`

```

is-validations ()
{
    util.functions-matching.diff is\\.| sedx 's/^/is./g'
}

```

```
}
```

2.30.2. `is.a-command`

```
is.a-command ()  
{  
    is.command "$@"  
}
```

2.30.3. `is.a-directory`

```
is.a-directory ()  
{  
    [[ -d "${1}" ]]  
}
```

2.30.4. `is.a-function`

```
is.a-function ()  
{  
    if [[ -n $1 ]] && typeset -f "$1" > /dev/null 2>&1; then  
        return 0  
    else  
        return 1  
    fi  
}
```

2.30.5. `is.a-function.invoke`

```
is.a-function.invoke ()  
{  
    local func="$1"  
    shift  
    is.a-function "${func}" && eval "${func} \\$@\\\""  
}
```

2.30.6. `is.a-non-empty-array`

```
is.a-non-empty-array ()  
{  
    local var_name="$1"  
    local -a array  
    echo "array=( \\$\\${var_name}[@]\\ )"
```

```

eval "array=( \"\$${var_name}[@]\")"
[[ -n ${array[*]} && ${var_name} =~ ^[0-9a-zA-Z_]+\$ && ${#array[@]} -gt 0 ]]
}

```

2.30.7. **is.a-non-empty-file**

```

is.a-non-empty-file ()
{
    [[ -s "${1}" ]]
}

```

2.30.8. **is.a-script**

```

is.a-script ()
{
    bashmatic.detect-subshell
    [[ ${BASH_IN_SUBSHELL} -eq 1 ]]
}

```

2.30.9. **is.a-variable**

```

is.a-variable ()
{
    local var_name="$1"
    local shell="$(user.current-shell)"
    case $shell in
        bash)
            [[ -n ${var_name} && ${var_name} =~ ^[0-9a-zA-Z_]+\$ && -n ${!var_name+x} ]]
        ]] && return 0
        ;
        zsh)
            eval "[[ -v ${var_name} ]]" && return 0
            ;
            *)
                return 1
            ;
    esac
}

```

2.30.10. **is.alias**

```

is.alias ()
{
    alias "$1" 2> /dev/null
}

```

```
}
```

2.30.11. `is.an-empty-file`

```
is.an-empty-file ()  
{  
    [[ ! -s "${1}" ]]  
}
```

2.30.12. `is.an-existing-file`

```
is.an-existing-file ()  
{  
    [[ -f "${1}" ]]  
}
```

2.30.13. `is.an-integer`

```
is.an-integer ()  
{  
    is.integer "$@"  
}
```

2.30.14. `is.blank`

```
is.blank ()  
{  
    [[ -z "${1}" ]]  
}
```

2.30.15. `is.command`

```
is.command ()  
{  
    command -v "$1" > /dev/null  
}
```

2.30.16. `is.empty`

```
is.empty ()  
{  
    is.blank "$@"  
}
```

```
}
```

2.30.17. **is.integer**

```
is.integer ()  
{  
    [[ $1 =~ ^[+-]?[0-9]+$ ]]  
}
```

2.30.18. **is.missing**

```
is.missing ()  
{  
    ! is.command "$@"  
}
```

2.30.19. **is.non.zero**

```
is.non.zero ()  
{  
    [[ $1 -ne 0 ]]  
}
```

2.30.20. **is.not-a-blank-var**

```
is.not-a-blank-var ()  
{  
    local var="$1"  
    [[ -n "${!var}" ]]  
}
```

2.30.21. **is.not-blank**

```
is.not-blank ()  
{  
    [[ -n "${1}" ]]  
}
```

2.30.22. **is.numeric**

```
is.numeric ()  
{
```

```
[[ $1 =~ ^[+-]?([0-9]+([.][0-9]*?)?|\.?[0-9]+)$ ]]
}
```

2.30.23. `is.sourced-in`

```
is.sourced-in () {
    bashmatic.detect-subshell
    [[ ${BASH_IN_SUBSHELL} -eq 0 ]]
}
```

2.30.24. `is.variable`

```
is.variable () {
    is.a-variable "$@"
}
```

2.30.25. `is.zero`

```
is.zero () {
    [[ $1 -eq 0 ]]
}
```

2.30.26. `unless`

```
unless () {
    ! __is.validation.error "${2}" "${1}" __is.validation.ignore-error "${@:3}"
}
```

2.30.27. `validations.add-error`

```
validations.add-error () {
    local error="$1"
    local value="$2"
    is.not-blank "${error}" && is.not-blank "${value}" && {
        export Bashmatic__IsErrors+=("${error}")
        export Bashmatic__IsValues+=("${value}")
    }
}
```

```
}
```

2.30.28. validations.begin

```
validations.begin ()  
{  
    export Bashmatic__IsErrors=()  
    export Bashmatic__IsValues=()  
    export Bashmatic__IsSessionName="$1"  
}
```

2.30.29. validations.end

```
validations.end ()  
{  
    validations.print-errors "$@"  
    validations.begin ""  
}
```

2.30.30. validations.print-errors

```
validations.print-errors ()  
{  
    local error_count="${#Bashmatic__IsErrors}"  
    ((error_count)) || return 0  
    is.not-blank "${Bashmatic__IsSessionName}" && hl.salmon "Reporing errors for  
${Bashmatic__IsSessionName}"  
    for i in $(seq 1 "${error_count}")  
    do  
        local error=${Bashmatic__IsErrors[$((i - 1))]}  
        local value=${Bashmatic__IsValues[$((i - 1))]}  
        error "${error}" "Invalid value: ${value}"  
    done  
    return "${error_count}"  
}
```

2.30.31. whenever

```
whenever ()  
{  
    __is.validation.error "${2}" "${1}" __is.validation.report-error "${@:3}"  
}
```

2.31. Module `jemalloc`

2.31.1. `jm.check`

```
jm.check ()  
{  
    local JM_Quiet=false  
    local JM_Ruby=false  
    local JM_Stats=false  
    while :; do  
        case $1 in  
            -q | --quiet)  
                shift  
                export JM_Quiet=true  
            ;  
            -r | --ruby)  
                shift  
                export JM_Ruby=true  
            ;  
            -s | --stats)  
                shift  
                export JM_Stats=true  
                exit $?  
            ;  
            -h | -\? | --help)  
                shift  
                jm.usage  
                exit 0  
            ;  
            --)  
                shift  
                break  
            ;  
            *)  
                break  
            ;  
        esac  
    done  
    ${JM_Ruby} && {  
        jm.ruby.report  
        exit 0  
    }  
    ${JM_Quiet} && {  
        jm.jemalloc.detect-quiet  
        code=$?  
        exit ${code}  
    }  
    ${JM_Stats} && {  
        jm.jemalloc.stats  
        exit 0  
    }
```

```
        }
    jm.jemalloc.detect-loud
}
```

2.31.2. jm.jemalloc.detect-loud

```
jm.jemalloc.detect-loud ()
{
    jm.jemalloc.detect-quiet
    local code=$?
    local local_ruby=$(jm.ruby.detect)
    printf "${ColorBlue}Checking if ruby ${ColorYellow}${local_ruby}${ColorBlue} is
linked with jemalloc... \n\n"
    if [[ ${code} -eq 0 ]]; then
        printf " ${ColorGreen} - jemalloc was detected.\n"
    else
        printf " ${ColorRed} - jemalloc was not detected.\n"
    fi
    printf "${ColorReset}\n"
    return ${code}
}
```

2.31.3. jm.jemalloc.detect-quiet

```
jm.jemalloc.detect-quiet ()
{
    MALLOC_CONF=stats_print:true ruby -e "exit" 2>&1 | grep -q "jemalloc statistics"
    return $?
}
```

2.31.4. jm.jemalloc.stats

```
jm.jemalloc.stats ()
{
    jm.jemalloc.detect-quiet || {
        printf "No Jemalloc was found for the current ruby $(jm.ruby.detect)\n"
        return 1
    }
    MALLOC_CONF=stats_print:true ruby -e "exit" 2>&1 | less -S
}
```

2.31.5. jm.ruby.detect

```
jm.ruby.detect ()
{
```

```

local ruby_loc
if [[ -n $(which rbenv) ]]; then
    ruby_loc=$(rbenv versions | grep '*' | awk '{print $2}')
    [[ -n ${ruby_loc} ]] && ruby_loc="(rbenv) ${ruby_loc}"
else
    ruby_loc="$(which ruby) $(ruby -e 'puts "#{RUBY_VERSION} (#{$RUBY_PLATFORM})"'"
)
fi
printf "%s" "${ruby_loc}"
}

```

2.31.6. jm.ruby.report

```

jm.ruby.report ()
{
    printf "Ruby version being tested:\n → ${ColorBlue}${(which ruby)} ${ColorYellow}
}${jm.ruby.detect}${ColorReset}\n"
}

```

2.31.7. jm.usage

```

jm.usage ()
{
    printf "
${ColorBlue}USAGE:${ColorReset}
$(basename "$0") [ -q/--quiet ]
                [ -r/--ruby ]
                [ -s/--stats ]
                [ -h/--help ]

${ColorBlue}DESCRIPTION:${ColorReset}
Determines whether the currently defined in the PATH ruby
interpreter is linked with libjemalloc memory allocator.

${ColorBlue}OPTIONS${ColorReset}
-q--quiet      Do not print output, exit with 1 if no jemalloc
-r--ruby       Print which ruby is currently in the PATH
-s--stats      Print the jemalloc stats
-h--help       This page.

%S
" """
    exit 0
}

```

2.32. Module json

2.32.1. json.begin-array

```
json.begin-array ()  
{  
    [[ -n "$1" ]] && json.begin-key "$1"  
    echo "["  
}
```

2.32.2. json.begin-hash

```
json.begin-hash ()  
{  
    [[ -n "$1" ]] && json.begin-key "$1"  
    echo "{"  
}
```

2.32.3. json.begin-key

```
json.begin-key ()  
{  
    if [[ -n "$1" ]]; then  
        printf "\\"$1\\": "  
    fi  
}
```

2.32.4. json.end-array

```
json.end-array ()  
{  
    printf "]"  
    [[ "$1" == "true" ]] && printf ","  
    echo  
}
```

2.32.5. json.end-hash

```
json.end-hash ()  
{  
    printf "}"  
    [[ "$1" == "true" ]] && printf ","  
    echo
```

```
}
```

2.32.6. json.file-to-array

```
json.file-to-array ()  
{  
    json.begin-array "$1"  
    cat "$2" | tr -d '\r' | tr -d '\015' | sed 's/^"/g;s/$"/,/g' | tail -r | awk -F,  
'{if (FNR!=1) print; else print $1}' | tail -r  
    json.end-array "$3"  
}
```

2.33. Module maths

2.33.1. maths.eval

```
maths.eval ()  
{  
    local __math_chars  
=(!123456789101112131415161718192021222324252627282930313233343536373839404142434445464748495051525354555657585960616263646566676869707172737475767778798081828384858687888990919293949596979899100101102103104105106107108109110111112113114115116117118119120121122123124125126127128129130131132133134135136137138139140141142143144145146147148149150151152153154155156157158159160161162163164165166167168169170171172173174175176177178179180181182183184185186187188189190191192193194195196197198199200201202203204205206207208209210211212213214215216217218219220221222223224225226227228229230231232233234235236237238239240241242243244245246247248249250251252253254255256257258259260261262263264265266267268269270271272273274275276277278279280281282283284285286287288289290291292293294295296297298299300301302303304305306307308309310311312313314315316317318319320321322323324325326327328329330331332333334335336337338339340341342343344345346347348349350351352353354355356357358359360361362363364365366367368369370371372373374375376377378379380381382383384385386387388389390391392393394395396397398399400401402403404405406407408409410411412413414415416417418419420421422423424425426427428429430431432433434435436437438439440441442443444445446447448449450451452453454455456457458459460461462463464465466467468469470471472473474475476477478479480481482483484485486487488489490491492493494495496497498499500501502503504505506507508509510511512513514515516517518519520521522523524525526527528529530531532533534535536537538539540541542543544545546547548549550551552553554555556557558559560561562563564565566567568569570571572573574575576577578579580581582583584585586587588589590591592593594595596597598599599600)  
    local -a __math_chars_array=(${echo "${__math_chars}" | sedx 's/(.)/\1 /g'})  
    local __math_chars_array  
    [[ -z "$1" ]] && {  
        output.set-max-width 100  
        output.set-min-width 40  
        usage-box "maths.eval 'expression' [ floating precitions [ total width ] ] @  
Computes a mathematical expression with UTF support" "Example 1." "maths.eval  
'!(57)*x(sin(\u03c0\u207d(1.3)))' => 1.66882" "Example 2." "maths.eval '5!x()' => 326.19382"  
"Special Characters:" "" " 0 through 23" "${__math_chars_array[*]:0:24}" "24 through  
48" "${__math_chars_array[*]:24:24}" "48 through 72" "${__math_chars_array[*]:48:24}"  
"72 through 96" "${__math_chars_array[*]:72:24}"  
        info "NOTE: ensure to use () brackets to group items you want to compute."  
        info "NOTE: if in doubt, add more brackets :)"  
        output.reset-min-max-width  
        return 0  
    }  
    gem.install unicode_math > /dev/null  
    local expression="$1"  
    shift  
    local output_precision="${1:-"5"}"  
    shift  
    local output_width="${1}"  
    local ruby_script="require 'unicode_math'; printf('%${output_width}."  
"${output_precision}f', (Math.module_eval { ${expression} }))"  
    ruby_script=$(echo "${ruby_script}" | sedx 's/ ?(x|!)/\.\1/g')"  
    local temp_file  
    temp_file="$(mktemp)"  
}
```

```

ruby -r 'unicode_math' -e "${ruby_script}" 2> "${temp_file}"
local code="$?"
[[ ${code} -ne 0 ]] && {
    error "Unable to perform an arithmetic expression: ${bldylw}${ruby_script}"
1>&2
    info "Error: \n${bldylw}$(cat "${temp_file}")"
    return 1
}
rm -f "${temp_file}"
return 0
}

```

2.34. Module memory

2.34.1. memory.bytes-to-units

```

memory.bytes-to-units ()
{
    local bytes="$1"
    local format="${2:-%.1f}"
    local suffix="${3:-b}"
    command -v numfmt > /dev/null || {
        command -v brew && brew install coreutils -q
        [[ -x $(brew --prefix)/bin/numfmt ]] || {
            error "Can't find numfmt installed, even after installing coreutils"
            return 1
        }
    }
    is.numeric "${bytes}" && numfmt --to=iec "${bytes}" --format="${format}" --suffix
    ="${suffix}/ */"
}

```

2.34.2. memory.size-to-bytes

```

memory.size-to-bytes ()
{
    local value="$1"
    local upper="${value^^}"
    local numeric
    local bytes
    numeric=$(echo "${value}" | tr -d 'a-zA-Z ')
    if [[ ${upper} =~ M\$ || ${upper} =~ MB\$ ]]; then
        bytes=$((numeric * 1024 * 1024))
    else
        if [[ ${upper} =~ TB\$ || ${upper} =~ T\$ ]]; then
            bytes=$((numeric * 1024 * 1024 * 1024 * 1024))
        else

```

```

if [[ ${upper} =~ GB\$ || ${upper} =~ G\$ ]]; then
    bytes=$((numeric * 1024 * 1024 * 1024))
else
    if [[ ${upper} =~ K\$ || ${upper} =~ KB\$ ]]; then
        bytes=$((numeric * 1024))
    else
        bytes=${numeric}
    fi
fi
fi
printf "%d" "${bytes}"
}

```

2.35. Module net

2.35.1. net.fast-scan

```

net.fast-scan ()
{
    local subnet="${1:-$(net.local-subnet)}"
    local out=$(mktemp)
    run.set-next show-output-on
    local colored=/tmp/colored.$$
    run "sudo nmap --min-parallelism 15 -O --host-timeout 5 -F ${subnet} > ${out}"
    run "echo 'printf \\' > ${colored}"
    cat "${out}" | sed -E "s/Nmap scan report for (.*)$/\\n${bldylw}Nmap scan report
for \1\\${clr}\\n/g" >> ${colored}
    run "echo '\\' >> ${colored}"
    bash ${colored}
}

```

2.35.2. net.is-host-port-protocol-open

```

net.is-host-port-protocol-open ()
{
    local host="$1"
    local port="$2"
    local protocol="${3:-tcp}"
    local command="nmap"
    [[ ${protocol} =~ udp ]] && command="sudo nmap -sU"
    command -v nmap > /dev/null || brew.install.package nmap 1>&2
    ${command} -Pn -p "${port}" "${host}" 2>&1 | ascii-pipe | grep -q -E "${port}/
${protocol} open "
}

```

2.35.3. `net.local-net`

```
net.local-net ()  
{  
    ifconfig -a | grep inet | grep broadcast | awk '{print $2}' | awk  
    'BEGIN{FS="."){printf "%d.%d.%d.%s", $1, $2, $3, "0/24"}'  
}
```

2.35.4. `net.local-subnet`

```
net.local-subnet ()  
{  
    local subnet=$(ifconfig -a | grep inet | grep broadcast | grep -v 'inet 169' |  
    grep -v 'inet 127' | awk '{print $2}' | cut -d '.' -f 1,2,3 | sort | uniq | head -1  
).0/24"  
    printf '%s' "${subnet}"  
}
```

2.36. Module `nvm`

2.36.1. `node.install.pin.version`

```
node.install.pin.version ()  
{  
    local node_version="${1:-${node_version}}"  
    if is.command volta; then  
        volta install "node@${node_version}"  
        volta pin "node@${node_version}"  
    else  
        nvm.activate  
    fi  
}
```

2.36.2. `nvm.activate`

```
nvm.activate ()  
{  
    is.an-existing-file .nvmrc && {  
        export node_version=$(cat .nvmrc | tr -d 'v')  
        info "Detected node version ${node_version}"  
    }  
    nvm.load  
}
```

2.36.3. `nvm.detect`

```
nvm.detect ()  
{  
    for dir in "${__bashmatic_nvm_dirs[@]}"  
    do  
        nvm.is-valid-dir "${dir}" && {  
            export NVM_DIR="${dir}"  
            source "${NVM_DIR}/nvm.sh"  
            return 0  
        }  
    done  
    return 1  
}
```

2.36.4. `nvm.install`

```
nvm.install ()  
{  
    h.blue "Installing nvm..."  
    local temp="/tmp/nvm_$$"  
    ( curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.37.2/install.sh 2>  
    /dev/null | bash 2> /dev/null ) | grep NVM_DIR > "${temp}"  
    source "${temp}"  
    rm -f "${temp}"  
    return 0  
}
```

2.36.5. `nvm.is-valid-dir`

```
nvm.is-valid-dir ()  
{  
    [[ -z $1 ]] && return 1  
    local d="$1"  
    [[ -d "${d}" && -s "${d}/nvm.sh" ]]  
}
```

2.36.6. `nvm.load`

```
nvm.load ()  
{  
    nvm.detect || nvm.install  
    is.a-function nvm || {  
        [[ -s "$NVM_DIR/nvm.sh" ]] && source "$NVM_DIR/nvm.sh"  
        [[ -s "$NVM_DIR/bash_completion" ]] && source "$NVM_DIR/bash_completion"  
    }  
}
```

```

[[ -f .nvmrc ]] && {
    nvm use > /dev/null 2>&1 || {
        h3 "NVM must install Node version $(cat .nvmrc), please wait..."
        run "nvm install && nvm use"
    }
}
}

```

2.36.7. `nvm.use`

```

nvm.use () {
    [[ -f .nvmrc ]] || return 1
    is.a.function nvm || nvm.load
    local node_version=$(cat .nvmrc | head -1)
    info "Activating NodeJS ${bldred}${node_version}..."
    nvm use "${node_version}" && return 0
    warning "No version ${node_version} was detected, installing"
    run.set-next show-output-on
    run "nvm install ${node_version}"
    nvm use "${node_version}" && return 0
    error "Unable to install node version ${node_version}"
    return 1
}

```

2.37. Module `openssl`

2.37.1. `openssl.certs.generate-csr`

```

openssl.certs.generate-csr () {
    local domain="${1:-"domain}"
    local subject="${2:-"${default_certificate_info}"}${domain}"
    local server="${domain}/\*/star"
    hl.desc "Generating CSR + Private Key for domain [${domain}]"
    set -e
    openssl req -new -newkey rsa:4096 -nodes -keyout "${server}.key" -out "${server}.csr" -subj "${subject}"
    pbcopy < "${server}.csr"
    success "CSR is now in the file ${server}.csr and your clipboard."
    set +e
    .openssl.certs.print-generated "${server}.csr" 2> /dev/null
    return 0
}

```

2.38. Module osx

2.38.1. `afp.servers`

```
afp.servers ()  
{  
    osx.local-servers afp  
}
```

2.38.2. `bashmatic-set-fqdn`

```
bashmatic-set-fqdn ()  
{  
    osx.set-fqdn "$@"  
}
```

2.38.3. `bashmatic-term`

```
bashmatic-term ()  
{  
    open "$(bashmatic-term-program)"  
}
```

2.38.4. `bashmatic-term-program`

```
bashmatic-term-program ()  
{  
    if [[ -d /Applications/iTerm.app ]]; then  
        printf '%s' /Applications/iTerm.app  
    else  
        if [[ -d /Applications/Utilities/Terminal.app ]]; then  
            printf '%s' /Applications/Utilities/Terminal.app  
        else  
            printf '%s' "echo 'No TERMINAL application found'"  
        fi  
    fi  
}
```

2.38.5. `change-underscan`

```
change-underscan ()  
{  
    set +
```

```

local amount_percentage="$1"
if [[ -z "${amount_percentage}" ]]; then
    printf "%s\n\n" "USAGE: change-underscan percent"
    printf "%s\n" "    eg: change-underscan 5 # underscan by 5%"
    printf "%s\n" "    eg: change-underscan -10 # overscan by 10%"
    return -1
fi
local file="/var/db/.com.apple.iokit.graphics"
local backup="/var/db/.com.apple.iokit.graphics.bak.$(date '+%F.%X')"
local new_value=$(ruby -e "puts (10000.0 + 10000.0 * ${amount_percentage}.to_f / 100.0).to_i")
h1 'This utility allows you to change underscan/overscan' 'on monitors that do not offer that option via GUI.'
run.ui.ask "Continue?"
info "Great! First we need to identify your monitor."
hl.yellow "Please make sure that the external monitor is plugged in."
run.ui.ask "Is it plugged in?"
info "Making a backup of your current graphics settings..."
inf "Please enter your password, if asked:"
set -
bash -c 'set - ; sudo ls -l > /dev/null; set + '
ok
run "sudo rm -f \"${backup}\""
export LibRun__AbortOnError=${True}
run "sudo cp -v \"${file}\" \"${backup}\""
h2 "Now: please change the resolution ${bldylw}on the problem monitor." "NOTE: it's ${italic}not important what resolution you choose," "as long as it's different than what you had previously..." "Finally: exit Display Preferences once you changed resolution."
run "open /System/Library/PreferencePanes/Displays.prefPane"
run.ui.ask "Have you changed the resolution and exited Display Prefs? "
local line=$(sudo diff "${file}" "${backup}" 2> /dev/null | head -1 | /usr/bin/env ruby -ne 'puts $_.to_i')
[[ -n ${BASHMATIC_DEBUG} ]] && info "diff line is at ${line}"
value=
if [[ "${line}" -gt 0 ]]; then
    line_pscn_key=$((line - 4))
    line_pscn_value=$((line - 3))
    ( awk "NR==${line_pscn_key}{print;exit}" "${file}" | grep -q pscn ) && {
        value=$(awk "NR==${line_pscn_value}{print;exit}" "${file}" | awk
'BEGIN{FS="[<>]"}{print $3}')
        [[ -n ${BASHMATIC_DEBUG} ]] && info "current value is ${value}"
    }
else
    error "It does not appear that anything changed, sorry."
    return -1
fi
h2 "Now, please unplug the problem monitor temporarily..."
run.ui.ask "...and press Enter to continue"
if [[ -n ${value} && ${value} -ne ${new_value} ]]; then
    export LibRun__AbortOnError=${True}

```

```

        run "sudo sed -i.backup \"${line_pscn_value}s/${value}/${new_value}/g\" \""
${file}\"
echo
h2 "Congratulations!" "Your display underscan value has been changed."
info "Previous Value - ${bldpur}${value}"
info "New value:      - ${bldgrn}${new_value}"
hr
info "${bldylw}IMPORTANT!"
info "You must restart your computer for the settings to take affect."
echo
run.ui.ask "Should I reboot your computer now? "
info "Very well, rebooting!"
run "sudo reboot"
else
warning "Unable to find the display scan value to change. "
info "Could it be that you haven't restarted since your last run?"
echo
info "Feel free to edit file directly, using:"
info "eg: ${bldylw}vim ${file} +${line_pscn_value}"
fi
}

```

2.38.6. cookie-dump

```

cookie-dump ()
{
    osx.cookie-dump "$@"
}

```

2.38.7. http.servers

```

http.servers ()
{
    osx.local-servers http
}

```

2.38.8. https.servers

```

https.servers ()
{
    osx.local-servers https
}

```

2.38.9. osx.app.is-installed

```
osx.app.is-installed ()  
{  
    local app="$1"  
    shift  
    /bin/ls -1 ~/Applications | grep -E '\.app$' | sort -u | grep "$@"  
-E -i "${app}|${app/*-/}"  
}
```

2.38.10. osx.cookie-dump

```
osx.cookie-dump ()  
{  
    local file="$1"  
    local tmp  
    if [[ ! -s ${file} ]]; then  
        tmp=$(mktemp)  
        file=${tmp}  
        pbpaste > "${file}"  
        local size=$(file.size "${file}")  
        if [[ ${size} -lt 4 ]]; then  
            error "Pasted data is too small to be a valid cookie?"  
            info "Here is what we got in your clipboard:\n\n$(cat "${file}")\n"  
            return 1  
        fi  
    fi  
    if [[ -s ${file} ]]; then  
        cat "${file}" | tr ' ' '\n' | sed '/^$/d' | awk 'BEGIN{FS="="}{printf( "%10d  
= %s\n", length($2), $1 )}' | sort -n  
    else  
        info "File ${file} does not exist or is empty."  
        info "Copy the value of the ${bldylw}Set-Cookie:${txtblu} header into the  
clipboard,"  
        info "and rerun this function."  
    fi  
    [[ -z ${tmp} ]] || rm -f "${tmp}"  
}
```

2.38.11. osx.detect-cpu

```
osx.detect-cpu ()  
{  
    local arch_name=$(uname -m)  
    if [[ "${arch_name}" = "x86_64" ]]; then  
        if [[ "$(sysctl -in sysctl.proc_translated)" = "1" ]]; then  
            echo m1 rosetta
```

```

        else
            echo intel native
        fi
    else
        if [ "${arch_name}" = "arm64" ]; then
            echo m1 native
        else
            echo unknown unknown
        fi
    fi
}

```

2.38.12. osx.dropbox.exclude

```

osx.dropbox.exclude ()
{
    local dir="$1"
    if [[ -d "${dir}" ]]; then
        xattr -w com.dropbox.ignored 1 "$1"
    else
        error "Folder '${dir}' does not exist or is blank."
        return 1
    fi
}

```

2.38.13. osx.dropbox.exclude-pwd

```

osx.dropbox.exclude-pwd ()
{
    xattr -w com.dropbox.ignored 1 "${PWD}"
}

```

2.38.14. osx.env-print

```

osx.env-print ()
{
    local var="$1"
    printf "${bldylw}%20s: ${bldgrn}%s\n" "${var}" "${!var}"
}

```

2.38.15. osx.local-servers

```

osx.local-servers ()
{
    local protocol="${1:-"ssh"}"

```

```

run.set-next show-output-on
run "timeout 20 dns-sd -B _${protocol}._tcp ."
}

```

2.38.16. osx.ramdisk.mount

```

osx.ramdisk.mount ()
{
    local size="${1:-"8"}"
    local diskname="${2:-"ramdisk"}"
    local total=$((size * 2 * 1024))
    util.os
    [[ ${BASHMATIC_OS} != "darwin" ]] && {
        error "This function only works on OSX"
        return 1
    }
    if [[ "${diskname}" =~ ' ' ]]; then
        error "Disk name can not contain spaces."
        return 1
    fi
    local path="/Volumes/${diskname}"
    if ( mount | ${GrepCommand} -q "/[V]olumes/${diskname}" ); then
        info "Looks like RAM disk already exists at ${path}..."
        return 1
    else
        run.ui.ask "Creating RAM disk sized ${size}Mb at ${path}"
        run.set-next show-output-on
        run "diskutil erasevolume HFS+ '${diskname}' $(hdiutil attach -nomount ram://$total)"
    fi
}

```

2.38.17. osx.ramdisk.unmount

```

osx.ramdisk.unmount ()
{
    local diskname="${2:-"ramdisk"}"
    [[ ${os.util} != "darwin" ]] && {
        error "This function only works on OSX"
        return 1
    }
    local path="/Volumes/${diskname}"
    if ( mount | ${GrepCommand} -q "/[V]olumes/${diskname}" ); then
        run.ui.ask "Unmount RAM disk at ${path}? "
        run "umount ${path}"
    else
        info "Couldn't find volume ${bldylw}${path}. Does the RAM disk exist?"
        return 1
    fi
}

```

```
    fi  
}
```

2.38.18. osx.scutil-print

```
osx.scutil-print ()  
{  
    local var="$1"  
    printf "${bldylw}%20s: ${bldgrn}%s\n" "${var}" $(sudo scutil --get "${var}" | tr -d '\n')  
}
```

2.38.19. osx.set-fqdn

```
osx.set-fqdn ()  
{  
    local fqdn="$1"  
    local domain=$(echo "${fqdn}" | sed -E 's/^.*\..*/.*/g')  
    local host=$(echo "${fqdn}" | sed -E 's/\..*/-/g')  
    h1 "Current HostName: ${bldylw}${HOSTNAME}"  
    echo  
    info "■ You provided the following FQDN : ${bldylw}${fqdn}"  
    echo  
    info "■ Hostname will be set to: ${bldgrn}${host}"  
    info "■ Domain will also change: ${bldgrn}${domain}"  
    echo  
    run.ui.ask "Does that look correct to you?"  
    echo  
    inf "Now, please provide your SUDO password, if asked:"  
    sudo printf '' || {  
        ui.closer.not-ok:  
        exit 1  
    }  
    ui.closer.ok:  
    run "sudo scutil --set HostName ${fqdn}"  
    run "sudo scutil --set LocalHostName ${host}.local 2>/dev/null|| true"  
    run "sudo scutil --set ComputerName ${host}"  
    run "dscacheutil -flushcache"  
    echo  
    h2 "Result of the changes:"  
    osx.scutil-print HostName  
    osx.scutil-print LocalHostName  
    osx.scutil-print ComputerName  
    osx.env-print HOSTNAME  
    echo  
    hr  
}
```

2.38.20. ssh.servers

```
ssh.servers ()  
{  
    osx.local-servers ssh  
}
```

2.39. Module output

2.39.1. ascii-clean

```
ascii-clean ()  
{  
    .output.clean "$@"  
}
```

2.39.2. ascii-pipe

```
ascii-pipe ()  
{  
    cat | .output.clean.pipe  
}
```

2.39.3. bashmatic.set-widget-width-to

```
bashmatic.set-widget-width-to ()  
{  
    output.constrain-screen-width "$@"  
}
```

2.39.4. center

```
center ()  
{  
    .output.center "$@"  
}
```

2.39.5. cursor.at.x

```
cursor.at.x ()  
{  
    .output.cursor-move-to-x "$@"  
}
```

```
}
```

2.39.6. `cursor.at.y`

```
cursor.at.y ()  
{  
    .output.cursor-move-to-y "$@"  
}
```

2.39.7. `cursor.down`

```
cursor.down ()  
{  
    .output.cursor-down-by "$@"  
}
```

2.39.8. `cursor.left`

```
cursor.left ()  
{  
    .output.cursor-left-by "$@"  
}
```

2.39.9. `cursor.restore`

```
cursor.restore ()  
{  
    printf "\e["  
}
```

2.39.10. `cursor.rewind`

```
cursor.rewind ()  
{  
    local x=${1:-0}  
    .output.cursor-move-to-x "${x}"  
}
```

2.39.11. `cursor.right`

```
cursor.right ()  
{
```

```
.output.cursor-right-by "$@"
}
```

2.39.12. `cursor.save`

```
cursor.save ()
{
    printf "\e[s"
}
```

2.39.13. `cursor.shift.x`

```
cursor.shift.x ()
{
    local shift="$1"
    if [[ "${shift:0:1}" == "-" ]]; then
        .output.cursor-left-by "${shift:1}"
    else
        .output.cursor-right-by "${shift}"
    fi
}
```

2.39.14. `cursor.up`

```
cursor.up ()
{
    .output.cursor-up-by "$@"
}
```

2.39.15. `duration`

```
duration ()
{
    local millis="$1"
    local exit_code="$2"
    [[ -n $(which bc) ]] || return
    if [[ -n ${millis} && ${millis} -ge 0 ]]; then
        local pattern
        pattern=" %6.6s ms "
        pattern="${txtblu}${pattern}${clr}"
        printf "${txtblu}${pattern}" "${millis}"
    fi
    if [[ -n ${exit_code} ]]; then
        [[ ${exit_code} -eq 0 ]] && printf " ${txtblk}${bakgrn} %3d ${clr}" " ${exit_code}"
    fi
}
```

```
    [[ ${exit_code} -gt 0 ]] && printf "${bldwht}${bakred} %3d ${clr}" "
${exit_code}"
    fi
}
```

2.39.16. `hr`

```
hr () {
    [[ -z "$*" ]] || printf "$*"
    .output.hr
}
```

2.39.17. `hr.colored`

```
hr.colored () {
    local color="$*"
    [[ -z ${color} ]] && color="${bldred}"
    .output.hr "$(screen-width)" "-" "${*}"
}
```

2.39.18. `inline.not-ok`

```
inline.not-ok () {
    printf "${txtwht}${bakred} ✘ ${clr} "
}
```

2.39.19. `inline.ok`

```
inline.ok () {
    printf "${txtblk}${bakgrn} ✓ ${clr} "
}
```

2.39.20. `left`

```
left () {
    .output.left-justify "$@"
}
```

2.39.21. **left-prefix**

```
left-prefix ()  
{  
    [[ -z ${LibOutput__LeftPrefix} ]] && {  
        export LibOutput__LeftPrefix=$(output.replicate-to " " "  
${LibOutput__LeftPrefixLen}")  
    }  
    printf "${LibOutput__LeftPrefix}"  
}
```

2.39.22. **not-ok**

```
not-ok ()  
{  
    ui.closer.not-ok "$@"  
}
```

2.39.23. **not-ok:**

```
not-ok: ()  
{  
    ui.closer.not-ok: "$@"  
}
```

2.39.24. **ok**

```
ok ()  
{  
    ui.closer.ok "$@"  
}
```

2.39.25. **ok:**

```
ok: ()  
{  
    ui.closer.ok: "$@"  
}
```

2.39.26. **output.color.off**

```
output.color.off ()
```

```
{  
    reset-color: 1>&2  
    reset-color: 1>&1  
}
```

2.39.27. `output.color.on`

```
output.color.on ()  
{  
    printf "${bldred}" 1>&2  
    printf "${bldblu}" 1>&1  
}
```

2.39.28. `output.constrain-screen-width`

```
output.constrain-screen-width ()  
{  
    export LibOutput__WidthDetectionStrategy="constrained"  
    [[ $1 -gt 0 ]] && output.set-max-width "$1"  
    [[ $2 -gt 0 ]] && output.set-min-width "$2"  
    return 0  
}
```

2.39.29. `output.has-stdin`

```
output.has-stdin ()  
{  
    [[ ! -t 0 ]]  
}
```

2.39.30. `output.is-pipe`

```
output.is-pipe ()  
{  
    [[ -p /dev/stdout ]]  
}
```

2.39.31. `output.is-redirect`

```
output.is-redirect ()  
{  
    [[ ! -t 1 && ! -p /dev/stdout ]]
```

```
}
```

2.39.32. `output.is-ssh`

```
output.is-ssh ()  
{  
    [[ -n "${SSH_CLIENT}" || -n "${SSH_CONNECTION}" ]]  
}
```

2.39.33. `output.is-terminal`

```
output.is-terminal ()  
{  
    output.is-tty || output.is-redirect || output.is-pipe || output.is-ssh  
}
```

2.39.34. `output.is-tty`

```
output.is-tty ()  
{  
    [[ -t 1 ]]  
}
```

2.39.35. `output.print-at-x-y`

```
output.print-at-x-y ()  
{  
    local x=$1  
    shift  
    local y=$1  
    shift  
    .output.cursor-move-to-x "${x}"  
    cursor.up "${y}"  
    printf "%s" "$*"  
    cursor.down "${y}"  
    .output.cursor-move-to-x 0  
}
```

2.39.36. `output.reset-min-max-width`

```
output.reset-min-max-width ()  
{  
    export LibOutput__MinWidth=${LibOutput__MinWidth:-${LibOutput__MinWidth__Default}}  
}
```

```
export LibOutput__MaxWidth=${LibOutput__MaxWidth:-${LibOutput__MaxWidth__Default}}
export LibOutput__MinHeight=${LibOutput__MaxHeight:-${LibOutput__MinHeight__Default}}
}
```

2.39.37. `output.screen-height.actual`

```
output.screen-height.actual ()
{
    local h
    case ${BASHMATIC_OS_NAME} in
        darwin)
            h=$(./output.stty.field rows)"
            ;
        linux)
            h=$(stty -a 2> /dev/null | grep rows | awk '{print $5}' | sedx 's;//g')"
            ;
        *)
            error "Unsupported OS: ${BASHMATIC_OS_NAME}"
            return 1
            ;
    esac
    printf -- "%d" "${h}"
}
```

2.39.38. `output.screen-width.actual`

```
output.screen-width.actual ()
{
    local w
    case "${BASHMATIC_OS_NAME}" in
        darwin)
            w=$(./output.stty.field columns)"
            ;
        linux)
            w=$(stty -a 2> /dev/null | grep columns | awk '{print $7}' | sedx 's;//g')"
            ;
        *)
            error "Unsupported OS: ${BASHMATIC_OS_NAME}"
            return 1
            ;
    esac
    printf -- "%d" "${w}"
}
```

2.39.39. `output.set-max-width`

```
output.set-max-width ()  
{  
    [[ $1 -gt 0 ]] && export LibOutput__MaxWidth="$1"  
}
```

2.39.40. `output.set-min-width`

```
output.set-min-width ()  
{  
    [[ $1 -gt 0 ]] && export LibOutput__MinWidth="$1"  
}
```

2.39.41. `output.unconstrain-screen-width`

```
output.unconstrain-screen-width ()  
{  
    export LibOutput__WidthDetectionStrategy="unconstrained"  
    return 0  
}
```

2.39.42. `screen-height`

```
screen-height ()  
{  
    .output.screen-height  
}
```

2.39.43. `screen-width`

```
screen-width ()  
{  
    .output.screen-width  
}
```

2.39.44. `screen.height`

```
screen.height ()  
{  
    .output.screen-height  
}
```

2.39.45. `screen.height.actual`

```
screen.height.actual ()  
{  
    .output.screen-height  
}
```

2.39.46. `screen.width`

```
screen.width ()  
{  
    .output.screen-width  
}
```

2.39.47. `screen.width.actual`

```
screen.width.actual ()  
{  
    .output.current-screen-width.unconstrained  
}
```

2.39.48. `section`

```
section ()  
{  
    .output.left-powerline pur "$@"  
}
```

2.39.49. `stderr`

```
stderr ()  
{  
    local file=$1  
    hl.subtle STDERR  
    printf "${txtred}"  
    [[ -s ${file} ]] && cat "${file}"  
    reset-color  
}
```

2.39.50. `stdout`

```
stdout ()
```

```
{  
    local file=$1  
    hl.subtle STDOUT  
    printf "${clr}"  
    [[ -s ${file} ]] && cat "${file}"  
    reset-color  
}
```

2.39.51. ui.closer.kind-of-ok

```
ui.closer.kind-of-ok ()  
{  
    .output.cursor-left-by 1000  
    printf "${bakylw}${bldwht} ${clr} "  
}
```

2.39.52. ui.closer.kind-of-ok:

```
ui.closer.kind-of-ok: ()  
{  
    ui.closer.kind-of-ok "$@"  
    echo  
}
```

2.39.53. ui.closer.not-ok

```
ui.closer.not-ok ()  
{  
    .output.cursor-left-by 1000  
    printf "${bakred}${bldwht} ${clr} "  
}
```

2.39.54. ui.closer.not-ok:

```
ui.closer.not-ok: ()  
{  
    ui.closer.not-ok "$@"  
    echo  
}
```

2.39.55. ui.closer.ok

```
ui.closer.ok ()
```

```
{  
  .output.cursor-left-by 1000  
  printf "${txtblk}${bakgrn} ✓ ${clr}"  
}
```

2.39.56. ui.closer.ok:

```
ui.closer.ok: ()  
{  
  ui.closer.ok "$@"  
  echo  
}
```

2.40. Module output-admonitions

2.40.1. alert

```
alert ()  
{  
  export title_text_color=${bold_yellow}  
  export title_border_color=${color_red}  
  export title_shadow_color=${bold_red}  
  title-box "$@"  
}
```

2.40.2. box-error

```
box-error ()  
{  
  panel-error "$@"  
}
```

2.40.3. box-info

```
box-info ()  
{  
  panel-info "$@"  
}
```

2.40.4. box-purple

```
box-purple ()
```

```
{  
    panel-purple-red "$@"  
}
```

2.40.5. **box-success**

```
box-success ()  
{  
    panel-green "$@"  
}
```

2.40.6. **box-warning**

```
box-warning ()  
{  
    panel-red-white "$@"  
}
```

2.40.7. **divider**

```
divider ()  
{  
    divider__ "$@"  
    printf "${clr}\n"  
}
```

2.40.8. **divider.yellow**

```
divider.yellow ()  
{  
    printf "${clr}\n${txtylw}[]"  
    divider__ "${txtylw}${bakylw}"  
    printf "${clr}${txtylw}[]${clr}\n"  
}
```

2.40.9. **divider__**

```
divider__ ()  
{  
    local color="${1:-${fg_bright_green}}"  
    printf "${color}  
"}  
  
=====
```

```
—${clr}"  
}
```

2.40.10. panel-error

```
panel-error ()  
{  
    export title_title_color="${bg_bright_red}${bldwht}"  
    export title_text_color="${bg_bright_red}${bldwht}"  
    export title_border_color="${bg_bright_red}${fg_dark_red}"  
    export title_shadow_color="${fg_dark_red}"  
    title-panel "$@"  
}
```

2.40.11. panel-error-yellow

```
panel-error-yellow ()  
{  
    export title_title_color="${bldwht}${bakred}"  
    export title_text_color="${bldylw}${bakred}"  
    export title_border_color="${bldylw}${bakred}"  
    export title_shadow_color="${fg_dark_red}"  
    title-panel "$@"  
}
```

2.40.12. panel-green

```
panel-green ()  
{  
    export title_title_color="${txtblk}${bg_bright_green}"  
    export title_text_color="${txtblk}${bg_bright_green}"  
    export title_border_color="${fg_deep_green}${bg_bright_green}"  
    export title_shadow_color="${fg_deep_green}"  
    title-panel "$@"  
}
```

2.40.13. panel-info

```
panel-info ()  
{  
    export title_title_color="${bldwht}${bg_sky_blue}"  
    export title_text_color="${bldylw}${bg_sky_blue}"  
    export title_border_color="${bg_sky_blue}"  
    export title_shadow_color="${fg_deep_blue}"  
    title-panel "$@"  
}
```

```
}
```

2.40.14. panel-info-blue

```
panel-info-blue ()  
{  
    export title_title_color="${bldwht}${bg_sky_blue}"  
    export title_text_color="${bldwht}${bg_sky_blue}"  
    export title_border_color="${bldwht}${fg_deep_blue}"  
    export title_shadow_color="${fg_deep_blue}"  
    title-panel "$@"  
}
```

2.40.15. panel-info-blue-black

```
panel-info-blue-black ()  
{  
    export title_title_color="${txtblk}${bg_sky_blue}"  
    export title_text_color="${txtblk}${bg_sky_blue}"  
    export title_border_color="${txtblk}${bg_sky_blue}"  
    export title_shadow_color="${fg_deep_blue}"  
    title-panel "$@"  
}
```

2.40.16. panel-info-dark

```
panel-info-dark ()  
{  
    export title_text_color="${txtblk}${bakcyn}"  
    export title_border_color="${txtblk}${bakcyn}"  
    export title_shadow_color="${txtblu}"  
    title-panel "$@"  
}
```

2.40.17. panel-info-yellow

```
panel-info-yellow ()  
{  
    export title_title_color="${bldwht}${bakgrn}"  
    export title_text_color="${bldylw}${bakgrn}"  
    export title_border_color="${bldylw}${bakgrn}"  
    export title_shadow_color="${fg_deep_blue}"  
    title-panel "$@"  
}
```

2.40.18. panel-purple-red

```
panel-purple-red ()  
{  
    export title_title_color="${fg_bright_red}"  
    export title_text_color="${bg_pink}${bldwht}"  
    export title_border_color="${bg_pink}${bldred}"  
    export title_shadow_color="${fg_dark_red}"  
    title-panel "$@"  
}
```

2.40.19. panel-red-white

```
panel-red-white ()  
{  
    export title_title_color="${fg_bright_red}"  
    export title_text_color="${bg_bright_red}${bg_mustard}"  
    export title_border_color="${bg_mustard}${bldwht}"  
    export title_shadow_color="${fg_dark_red}"  
    title-panel "$@"  
}
```

2.40.20. title

```
title ()  
{  
    title-blue "$@"  
}
```

2.40.21. title-blue

```
title-blue ()  
{  
    export title_text_color="${color_black}${bg_sky_blue}"  
    export title_border_color="${bg_sky_blue}"  
    export title_shadow_color=${txtblu}  
    title-box "$@"  
}
```

2.40.22. title-box

```
title-box ()  
{  
    local len
```

2.40.23. title-green

```
title-green ()  
{  
    export title_text_color=${color_black}${bakgrn}  
    export title_border_color=${fg_bright_green}${bakgrn}  
    export title_shadow_color=${fg_deep_green}  
    title-box "$@"  
}
```

2.40.24. title-panel

```
title-panel ()  
{  
    local line="$1"  
    shift  
    local shadow="${title_shadow_color} ${clr}"  
    printf "${clr}\n"  
    ${title_border_color} ${box_top} ${clr}\n"  
    .title.line "${shadow}" "${title_title_color:-${title_text_color}}" "${line}"  
    if [[ ${#@} -gt 0 ]]; then
```

```

printf " ${title_border_color} ${box_divider} ${shadow}\n"
for line in "$@"
do
    .title.line "${shadow}" "${title_text_color}" "${line}"
done
fi
printf " ${title_border_color} ${box_bottom} ${shadow}\n"
printf " ${title_shadow_color}  ${box_bottom_shadow}${clr}\n\n"
}

```

2.40.25. title-red

```

title-red ()
{
    export title_text_color="${txtwht}${bakred}"
    export title_border_color="${bldwht}${bakred}"
    export title_shadow_color="${fg_dark_red}"
    title-box "$@"
}

```

2.40.26. title-yellow

```

title-yellow ()
{
    export title_text_color="${color_black}${bg_yellow_on_gray}"
    export title_border_color="${bg_yellow_on_gray}"
    export title_shadow_color="${fg_mustard}"
    title-box "$@"
}

```

2.41. Module output-boxes

2.41.1. arrow-right

```

arrow-right ()
{
    .output.left-as-is "$@"
}

```

2.41.2. arrow.blk-on-blu

```

arrow.blk-on-blu ()
{
    arrow-right "${bakblu}" "${txtblk}" "$@"
}

```

```
}
```

2.41.3. **arrow.blk-on-cyn**

```
arrow.blk-on-cyn ()  
{  
    arrow-right "${bakcyn}" "${txtblk}" "$@"  
}
```

2.41.4. **arrow.blk-on-grn**

```
arrow.blk-on-grn ()  
{  
    arrow-right "${bakgrn}" "${txtblk}" "$@"  
}
```

2.41.5. **arrow.blk-on-pur**

```
arrow.blk-on-pur ()  
{  
    arrow-right "${bakpur}" "${txtblk}" "$@"  
}
```

2.41.6. **arrow.blk-on-red**

```
arrow.blk-on-red ()  
{  
    arrow-right "${bakred}" "${txtblk}" "$@"  
}
```

2.41.7. **arrow.blk-on-wht**

```
arrow.blk-on-wht ()  
{  
    arrow-right "${bakwht}" "${txtblk}" "$@"  
}
```

2.41.8. **arrow.blk-on-ylw**

```
arrow.blk-on-ylw ()  
{  
    arrow-right "${bakylw}" "${txtblk}" "$@"  
}
```

```
}
```

2.41.9. **arrow.wht-on-blk**

```
arrow.wht-on-blk ()  
{  
    arrow-right "${bakblk}" "${txtwht}" "$@"  
}
```

2.41.10. **arrow.wht-on-blu**

```
arrow.wht-on-blu ()  
{  
    arrow-right "${bakblu}" "${txtwht}" "$@"  
}
```

2.41.11. **arrow.wht-on-cyn**

```
arrow.wht-on-cyn ()  
{  
    arrow-right "${bakcyn}" "${txtwht}" "$@"  
}
```

2.41.12. **arrow.wht-on-grn**

```
arrow.wht-on-grn ()  
{  
    arrow-right "${bakgrn}" "${txtwht}" "$@"  
}
```

2.41.13. **arrow.wht-on-pur**

```
arrow.wht-on-pur ()  
{  
    arrow-right "${bakpur}" "${txtwht}" "$@"  
}
```

2.41.14. **arrow.wht-on-red**

```
arrow.wht-on-red ()  
{  
    arrow-right "${bakred}" "${txtwht}" "$@"  
}
```

```
}
```

2.41.15. **arrow.wht-on-ylw**

```
arrow.wht-on-ylw ()  
{  
    arrow-right "${bakylw}" "${txtwht}" "$@"  
}
```

2.41.16. **attention**

```
attention ()  
{  
    .output.left-powerline blu 65 "${txtwht}${bakblu}" $* "  
}
```

2.41.17. **box.black-on-blue**

```
box.black-on-blue ()  
{  
    .output.box "${txtblk}${bakblu}" "${bakblu}" "$@"  
}
```

2.41.18. **box.black-on-green**

```
box.black-on-green ()  
{  
    .output.box "${txtblk}${bakgrn}" "${bakgrn}" "$@"  
}
```

2.41.19. **box.black-on-purple**

```
box.black-on-purple ()  
{  
    .output.box "${txtblk}${bakpur}" "${bakpur}" "$@"  
}
```

2.41.20. **box.black-on-red**

```
box.black-on-red ()  
{  
    .output.box "${txtblk}${bakred}" "${bakred}" "$@"  
}
```

```
}
```

2.41.21. **box.black-on-yellow**

```
box.black-on-yellow ()  
{  
    .output.box "${txtblk}${bakylw}" "${txtblk}${bakylw}" "$@"  
}
```

2.41.22. **box.blue-in-green**

```
box.blue-in-green ()  
{  
    .output.box "${bldblu}" "${bldgrn}" "$@"  
}
```

2.41.23. **box.blue-in-yellow**

```
box.blue-in-yellow ()  
{  
    .output.box "${bldylw}" "${bldblu}" "$@"  
}
```

2.41.24. **box.green-in-cyan**

```
box.green-in-cyan ()  
{  
    .output.box "${bldgrn}" "${bldcyn}" "$@"  
}
```

2.41.25. **box.green-in-green**

```
box.green-in-green ()  
{  
    .output.box "${bldgrn}" "${bldgrn}" "$@"  
}
```

2.41.26. **box.green-in-magenta**

```
box.green-in-magenta ()  
{  
    .output.box "${bldgrn}" "${bldpur}" "$@"  
}
```

```
}
```

2.41.27. **box.green-in-yellow**

```
box.green-in-yellow ()  
{  
    .output.box "${bldgrn}" "${bldylw}" "$@"  
}
```

2.41.28. **box.magenta-in-blue**

```
box.magenta-in-blue ()  
{  
    .output.box "${bldblu}" "${bldpur}" "$@"  
}
```

2.41.29. **box.magenta-in-green**

```
box.magenta-in-green ()  
{  
    .output.box "${bldpur}" "${bldgrn}" "$@"  
}
```

2.41.30. **box.red-in-magenta**

```
box.red-in-magenta ()  
{  
    .output.box "${bldred}" "${bldpur}" "$@"  
}
```

2.41.31. **box.red-in-red**

```
box.red-in-red ()  
{  
    .output.box "${txtred}" "${txtred}" "$@"  
}
```

2.41.32. **box.red-in-yellow**

```
box.red-in-yellow ()  
{  
    .output.box "${bldred}" "${bldylw}" "$@"  
}
```

```
}
```

2.41.33. **box.white-on-blue**

```
box.white-on-blue ()
{
    .output.box "${bakblu}${bldwht}" "${bakblu}${bldwht}" "$@"
}
```

2.41.34. **box.white-on-green**

```
box.white-on-green ()
{
    .output.box "${bakgrn}${bldwht}" "${bakgrn}${bldwht}" "$@"
}
```

2.41.35. **box.white-on-red**

```
box.white-on-red ()
{
    .output.box "${bakred}${bldwht}" "${bakred}${bldwht}" "$@"
}
```

2.41.36. **box.yellow-in-blue**

```
box.yellow-in-blue ()
{
    .output.box "${bldylw}" "${bldblu}" "$@"
}
```

2.41.37. **box.yellow-in-red**

```
box.yellow-in-red ()
{
    .output.box "${bldred}" "${bldylw}" "$@"
}
```

2.41.38. **box.yellow-in-yellow**

```
box.yellow-in-yellow ()
{
    .output.box "${bldylw}" "${txtylw}" "$@"
}
```

```
}
```

2.41.39. **box.yellow-on-green**

```
box.yellow-on-green ()  
{  
    .output.box "${bakgrn}${bldwht}" "${bakgrn}${bldylw}" "$@"  
}
```

2.41.40. **box.yellow-on-purple**

```
box.yellow-on-purple ()  
{  
    .output.box "${bakpur}" "${bldylw}" "$@"  
}
```

2.41.41. **h.black**

```
h.black ()  
{  
    center "${bldylw}${bakblk}" "$@"  
}
```

2.41.42. **h.blue**

```
h.blue ()  
{  
    center "${txtblk}${bakblu}" "$@"  
}
```

2.41.43. **h.e**

```
h.e ()  
{  
    .output.box "${bakred}${txtblk}" "${bakred}" "$@"  
}
```

2.41.44. **h.green**

```
h.green ()  
{  
    center "${txtblk}${bakgrn}" "$@"  
}
```

```
}
```

2.41.45. **h.orange**

```
h.orange ()  
{  
    left "${white_on_orange}" "$@"  
}
```

2.41.46. **h.orange-center**

```
h.orange-center ()  
{  
    center "${white_on_orange}" "$@"  
}
```

2.41.47. **h.red**

```
h.red ()  
{  
    center "${txtblk}${bakred}" "$@"  
}
```

2.41.48. **h.salmon**

```
h.salmon ()  
{  
    left "${white_on_salmon}" "$@"  
}
```

2.41.49. **h.salmon-center**

```
h.salmon-center ()  
{  
    center "${white_on_salmon}" "$@"  
}
```

2.41.50. **h.salmon-center-black**

```
h.salmon-center-black ()  
{  
    center "${black_on_salmon}" "$@"  
}
```

```
}
```

2.41.51. **h.yellow**

```
h.yellow ()  
{  
    center "${txtblk}${bakylw}" "$@"  
}
```

2.41.52. **h.yellow-center-black**

```
h.yellow-center-black ()  
{  
    center "${black_on_yellow}" "$@"  
}
```

2.41.53. **h1**

```
h1 ()  
{  
    box.blue-in-yellow "$@"  
}
```

2.41.54. **h1.blue**

```
h1.blue ()  
{  
    h1bg "${@}"  
}
```

2.41.55. **h1.green**

```
h1.green ()  
{  
    box.green-in-magenta "$@"  
}
```

2.41.56. **h1.purple**

```
h1.purple ()  
{  
    box.magenta-in-green "$@"  
}
```

```
}
```

2.41.57. h1.red

```
h1.red ()  
{  
    box.red-in-red "$@"  
}
```

2.41.58. h1.yellow

```
h1.yellow ()  
{  
    box.yellow-in-red "$@"  
}
```

2.41.59. h1bg

```
h1bg ()  
{  
    box.white-on-blue "$@"  
}
```

2.41.60. h2

```
h2 ()  
{  
    box.blue-in-green "$@"  
}
```

2.41.61. h2.green

```
h2.green ()  
{  
    h2bg "${{@}}"  
}
```

2.41.62. h2bg

```
h2bg ()  
{  
    box.black-on-green "$@"  
}
```

```
}
```

2.41.63. h3

```
h3 ()  
{  
    box.magenta-in-green "$@"  
}
```

2.41.64. h3.yellow

```
h3.yellow ()  
{  
    h3bg "${{@}}"  
}
```

2.41.65. h3bg

```
h3bg ()  
{  
    box.black-on-yellow "$@"  
}
```

2.41.66. h4

```
h4 ()  
{  
    section.blue "$@"  
}
```

2.41.67. h4.red

```
h4.red ()  
{  
    h4bg "${{@}}"  
}
```

2.41.68. h4bg

```
h4bg ()  
{  
    box.white-on-blue "$@"  
}
```

```
}
```

2.41.69. h5

```
h5 ()  
{  
    section.green "$@"  
}
```

2.41.70. h5bg

```
h5bg ()  
{  
    box.yellow-on-green "$@"  
}
```

2.41.71. h6

```
h6 ()  
{  
    section.yellow "$@"  
}
```

2.41.72. h7

```
h7 ()  
{  
    section.salmon "$@"  
}
```

2.41.73. h8

```
h8 ()  
{  
    section.cyan "$@"  
}
```

2.41.74. hdr

```
hdr ()  
{  
    h1 "$@"  
}
```

```
}
```

2.41.75. **hl.blue**

```
hl.blue ()  
{  
    left "${bldwht}${bakpur}" "$@"  
}
```

2.41.76. **hl.desc**

```
hl.desc ()  
{  
    left "${bakylw}${txtblk}${bakylw}" "$@"  
}
```

2.41.77. **hl.green**

```
hl.green ()  
{  
    left "${txtblk}${bakgrn}" "$@"  
}
```

2.41.78. **hl.subtle**

```
hl.subtle ()  
{  
    left "${bldwht}${bakblk}${underlined}" "$@"  
}
```

2.41.79. **hl.yellow**

```
hl.yellow ()  
{  
    left "${txtblk}${bakylw}" "$@"  
}
```

2.41.80. **hl.yellow-on-black**

```
hl.yellow-on-black ()  
{  
    left "${yellow_on_black}" "$@"  
}
```

```
}
```

2.41.81. `hl.yellow-on-gray`

```
hl.yellow-on-gray ()  
{  
    left "${yellow_on_gray}" "$@"  
}
```

2.41.82. `notice`

```
notice ()  
{  
    .output.left-powerline ylw 65 "${itablk}${bakylw} $* "
```

2.41.83. `section.blue`

```
section.blue ()  
{  
    .output.left-powerline blu "$@"  
}
```

2.41.84. `section.cyan`

```
section.cyan ()  
{  
    .output.left-powerline cyn "$@"  
}
```

2.41.85. `section.gray`

```
section.gray ()  
{  
    .output.left-powerline blk "$@"  
}
```

2.41.86. `section.gray-yellow`

```
section.gray-yellow ()  
{  
    section.gray "${bldylw}${bakblk}$*"
```

```
}
```

2.41.87. `section.green`

```
section.green ()  
{  
    .output.left-powerline grn "$@"  
}
```

2.41.88. `section.purple`

```
section.purple ()  
{  
    .output.left-powerline pur "$@"  
}
```

2.41.89. `section.red`

```
section.red ()  
{  
    .output.left-powerline red "$@"  
}
```

2.41.90. `section.salmon`

```
section.salmon ()  
{  
    .output.left-powerline wht 65 "${txtwht}${white_on_salmon} $* "
```

2.41.91. `section.white`

```
section.white ()  
{  
    .output.left-powerline wht "$@"  
}
```

2.41.92. `section.yellow`

```
section.yellow ()  
{  
    .output.left-powerline ylw "$@"  
}
```

```
}
```

2.41.93. `status.failed`

```
status.failed ()
{
    cursor.right 5
    section.red "$*"
    cursor.up 2
    not-ok:
    cursor.down 2
}
```

2.41.94. `status.ok`

```
status.ok ()
{
    cursor.right 5
    section.cyan "$*"
    cursor.up 2
    ok:
    cursor.down 2
}
```

2.42. Module `output-repeat-char`

2.43. Module `output-utils`

2.43.1. `abort`

```
abort ()
{
    echo
    printf -- "${LibOutput__LeftPrefix}${txtblk}${bakred} « ABORT » ${clr} ${bldwht}
✓ ${bldgrn}*${clr}" 1>&2
    echo
}
```

2.43.2. `ask`

```
ask ()
{
    printf -- "%s${txtylw}**${clr}\n" "${LibOutput__LeftPrefix}"
```

```
    printf -- "%s${bldgrn} ${bldylw}" "${LibOutput__LeftPrefix}"
}
```

2.43.3. **br**

```
br () {
    echo
}
```

2.43.4. **columnize**

```
columnize () {
    local columns="${1:-2}"
    [[ -n $1 ]] && shift
    local sw="${2:-$(screen.width.actual)}"
    [[ ${sw} -lt 90 ]] && sw=100
    pr -l 10000 -"${columns}" -e4 -w ${sw} | expand -8 | sed -E '/^ *$/d' | grep -v
'Page '
```

2.43.5. **dbg**

```
dbg () {
    is-dbg && printf "${txtgrn}[DEBUG | ${txtylw}${(time.now.with-ms)}${txtgrn}]"
    ${txtblu}$(txt-info)*\n" 1>&2
    return 0
}
```

2.43.6. **dbg-off**

```
dbg-off () {
    unset BASHMATIC_DEBUG
    unset BASHMATIC_PATH_DEBUG
    [[ -f ${BASHMATIC_HOME}/.envrc.no-debug ]] && source "${BASHMATIC_HOME}"
    "/.envrc.no-debug
}
```

2.43.7. **dbg-on**

```

dbg-on ()
{
    export BASHMATIC_DEBUG=1
    [[ -f ${BASHMATIC_HOME}/.envrc.debug ]] && source "${BASHMATIC_HOME}"/.envrc.debug
}

```

2.43.8. `dbgf`

```

dbgf ()
{
    local func="$1"
    shift
    is.a-function "${func}" || {
        error "${func} is not a function"
        return 1
    }
    dbg "${func}(" "$@" ")"
    ${func} "$@"
    local code=$?
    is-dbg || return "${code}"
    cursor.up 1
    cursor.at.x 0
    if [[ ${code} -eq 0 ]]; then
        ok:
    else
        not-ok:
    fi
    return ${code}
}

```

2.43.9. `err`

```

err ()
{
    echo
    printf -- "${LibOutput__LeftPrefix}${bldylw}${bakred} « ERROR! » ${clr} ${
bldwht}*${clr}" 1>&2
    echo
}

```

2.43.10. `error`

```

error ()
{
    panel-error "ERROR" "$@"
}

```

```
}
```

2.43.11. **error-title**

```
error-title ()  
{  
    title-red "$(printf "%60.60s" "----- ERROR -----")" "$@"  
}
```

2.43.12. **error:**

```
error: ()  
{  
    err "$*"  
    ui.closer.not-ok:  
}
```

2.43.13. **fatal**

```
fatal ()  
{  
    header=$(printf -- "${clr}${bldwht} « ABORT »")  
    box.black-on-red "${header} ${clr}${txtblk}${bakred} - $1" "${@:2}" 1>&2  
    exit 1  
}
```

2.43.14. **inf**

```
inf ()  
{  
    printf -- "${LibOutput__LeftPrefix}${clr}${txtcyn}*$${clr}"  
}
```

2.43.15. **info**

```
info ()  
{  
    inf "$@"  
    echo  
}
```

2.43.16. `info-debug`

```
info-debug ()  
{  
    [[ -z ${BASHMATIC_DEBUG} ]] && return  
    printf -- "${LibOutput__LeftPrefix}${bakpur}[ debug ] $* ${clr}\n"  
}
```

2.43.17. `info:`

```
info: ()  
{  
    inf "$*"  
    ui.closer.ok:  
}
```

2.43.18. `is-dbg`

```
is-dbg ()  
{  
    [[ -n ${BASHMATIC_DEBUG} ]]  
}
```

2.43.19. `note`

```
note ()  
{  
    printf -- "\n${bldwht}${bakblu} « NOTE » ${clr} ${bldwht} ✓ ${bldgrn}$*${clr}"  
1>&2  
    printf -- "\n\n" 1>&2  
}
```

2.43.20. `okay`

```
okay ()  
{  
    printf -- " ${txtblk}${bakgrn} ✓ ${clr} □ ${clr}${italic}${bldcyn}$*${clr} □  
${bldgrn} ALL GOOD, YO □ ${clr}" 1>&2  
    echo  
}
```

2.43.21. puts

```
puts ()  
{  
    printf " \${txtwht}*$\{clr}"  
}
```

2.43.22. reset-color

```
reset-color ()  
{  
    echo -en "\${clr}"  
}
```

2.43.23. reset-color:

```
reset-color: ()  
{  
    printf "\${clr}"  
}
```

2.43.24. shutdown

```
shutdown ()  
{  
    local message=${1:-"Shutting down..."}  
    echo  
    box.red-in-red "\${message}"  
    echo  
    exit 1  
}
```

2.43.25. skipping

```
skipping ()  
{  
    printf -- "\n${bldwht}${bakcyn} « SKIPPING » ${clr} ${bldwht} ✓ ${bldgrn}*$  
${clr}" 1>&2  
    printf -- "\n\n" 1>&2  
}
```

2.43.26. `success`

```
success ()
{
    printf -- "\n${txtblk}${bakgrn} « SUCCESS » ${clr} ${bldwht} ✓ ${bldgrn}$*
${clr}" 1>&2
    printf -- "\n\n" 1>&2
}
```

2.43.27. `warn`

```
warn ()
{
    printf -- "${LibOutput_LeftPrefix}${bldwht}${bakylw} « WARNING! » ${clr} ${bldylw}$*${clr}" 1>&2
}
```

2.43.28. `warning`

```
warning ()
{
    local first="$1"
    shift
    box.black-on-yellow "${header} ${clr}${txtblk}${bakylw} - $first" "$@" 1>&2
}
```

2.43.29. `warning:`

```
warning: ()
{
    warn "$*"
    ui.closer.kind-of-ok:
}
```

2.44. Module package

2.44.1. `package.ensure.command-available`

```
package.ensure.command-available ()
{
    local package="$1"
    shift
    local binary="$1"
```

```

shift
is.a-command "${binary}" && return 0
package.ensure.is-installed "${package}"
hash -r > /dev/null 2>&1
is.a-command "${binary}" && return 0
error "After installing package ${package}, binary ${binary} still is not found."
}

```

2.44.2. package.ensure.is-installed

```

package.ensure.is-installed ()
{
    for pkg in "$@"
    do
        package.is-installed "${pkg}" || package.install
        package.is-installed "${pkg}" || {
            error "Package ${pkg} does not appear installed, broken package or
version?"
            return 1
        }
    done
}

```

2.45. Module path

2.45.1. PATH.uniqify

```

PATH.uniqify ()
{
    local _p=$(printf "${PATH}" | sed 's/:/\n/g' | uniq | tr '\n' ':')
    [[ "${_p}" == "${PATH}" ]] || export PATH="${_p}"
}

```

2.45.2. path.absolute

```

path.absolute ()
{
    if [[ -d "$1" ]]; then
        pushd "$1" > /dev/null || exit
        pwd
        popd > /dev/null || exit
    else
        if [[ -e "$1" ]]; then
            pushd "$(dirname "$1")" > /dev/null || exit
            echo "$(pwd)/${basename "$1"}"
            popd > /dev/null || exit
        fi
    fi
}

```

```

        else
            echo "$1" does not exist! 1>&2
            return 127
        fi
    fi
}

```

2.45.3. path.append

```

path.append ()
{
    local new_path="${PATH}"
    for __path in "$@"
    do
        is.a-directory "${__path}" || {
            error "Argument ${__path} is not a valid directory, abort." 1>&2
            return 1
        }
        path.dirs.uniq | grep -q -E "^\${__path}\$" && continue
        new_path="${new_path}:${__path}"
    done
    echo "${new_path}"
}

```

2.45.4. path.dirs

```

path.dirs ()
{
    local path
    if output.has-stdin; then
        while IFS= read -r line; do
            path="${path}:${line}"
        done
    fi
    [[ -z "${path}" ]] && path="${1:-$PATH}"
    echo "${path//:/`"
    ``} | /usr/bin/tr -d '"' | sedx '/^$/d; s:///g'
}

```

2.45.5. path.dirs.delete

```

path.dirs.delete ()
{
    local path="$1"
    shift
    local opath=""

```

```

for p in "$@"
do
    opath="${path}"
    path=$(eval "echo '$${path}' | sedx 's#:${p}:#:#g'")
    [[ ${path} == "${opath}" ]] && path=$(eval "echo '$${path}' | sedx 's#(^|:)${p}(:^|$)##g'")
    is-dbg && dbg "path=${path} after removing ${p}"
done
echo "${path}"
return
}

```

2.45.6. `path.dirs.join`

```

path.dirs.join ()
{
    sedx 's/$/:/g' | tr -d '\n'
}

```

2.45.7. `path.dirs.size`

```

path.dirs.size ()
{
    path.dirs "$@" | /usr/bin/wc -l | /usr/bin/tr -d ' '
}

```

2.45.8. `path.dirs.sort`

```

path.dirs.sort ()
{
    path.dirs.uniq "$@" | sort
}

```

2.45.9. `path.dirs.sort-rev`

```

path.dirs.sort-rev ()
{
    path.dirs.uniq "$@" | sort -r
}

```

2.45.10. `path.dirs.uniq`

```

path.dirs.uniq ()

```

```
{
    local -a paths=($(path.dirs "$@"))
    for path in $(array.uniq "${paths[@]}")
    do
        echo "${path}"
    done | sort
}
```

2.45.11. `path.mutate.append`

```
path.mutate.append ()
{
    export PATH=$(path.append "$@")"
}
```

2.45.12. `path.mutate.delete`

```
path.mutate.delete ()
{
    export PATH=$(path.dirs.delete "$@")"
}
```

2.45.13. `path.mutate.prepend`

```
path.mutate.prepend ()
{
    export PATH=$(path.prepend "$@")"
}
```

2.45.14. `path.mutate.uniq`

```
path.mutate.uniq ()
{
    export PATH=$(path.uniq "$@")"
}
```

2.45.15. `path.prepend`

```
path.prepend ()
{
    local new_path="${PATH}"
    for __path in "$@"
    do
```

```

is.a-directory "${__path}" || {
    error "Argument ${__path} is not a valid directory, abort." 1>&2
    return 1
}
local p=$(path.dirs.uniq | grep -v -E "^\${__path}\$" | tr '\n' ':')
new_path="${__path}:$p"
done
echo "${new_path}"
}

```

2.45.16. path.strip-slash

```

path.strip-slash ()
{
    local path=$(echo "$1" | sed -E 's#\//+##g')
    printf -- "%s" "${path}"
}

```

2.45.17. path.uniq

```

path.uniq ()
{
    array.join ':' $(path.dirs.uniq "$@")
}

```

2.46. Module pdf

2.46.1. pdf.combine

```

pdf.combine ()
{
    local merged="${1}"
    shift
    local files=""
    for f in "$@"
    do
        [[ -f "${f}" ]] && {
            info "Appending file ${bldylw}${f}"
            files="${files} ${f}"
        }
    done
    [[ -s "${merged}" ]] && {
        warning "Merged file ${merged} already exists, removing..."
        run "rm -f \\"${merged}""
    }
    unalias gs 2> /dev/null
}

```

```

[[ -n $(command -V gs) ]] || brew.install.package gs
run "mkdir -p $(dirname "${merged}")"
info "Please wait while GhostScript combines your PDFs into"
info "destination file: ${bldylw}${merged}"
run "gs -q -dNOPAUSE -dBATCH -sDEVICE=pdfwrite -sOutputFile=${merged} ${files}"
return "${LibRun__LastExitCode}"
}

```

2.47. Module pg

2.47.1. pg.is-running

```

pg.is-running ()
{
  [[ $(/bin/ps -ef | grep -c '[p]ostgres:') -gt 4 ]]
}

```

2.47.2. pg.running.data-dirs

```

pg.running.data-dirs ()
{
  ps -eo 'args' | $(which grep) '[p]ostgres.*-D' | awk 'BEGIN{FS="-D"}{print $2}' |
  awk '{print $1}' | sort
}

```

2.47.3. pg.running.server-binaries

```

pg.running.server-binaries ()
{
  ps -eo 'args' | $(which grep) '[p]ostgres.*-D' | awk '{print $1}' | sort
}

```

2.47.4. pg.server-in-path.version

```

pg.server-in-path.version ()
{
  is.command postgres || return 1
  $(which postgres) -V | sed -E 's/[^0-9.]/g;s/\..*$/g'
}

```

2.48. Module pids

2.48.1. pall

```
pall ()  
{  
    pids.all "$@"  
}
```

2.48.2. pid.alive

```
pid.alive ()  
{  
    local pid="$1"  
    [[ -z ${pid} ]] && {  
        error "usage: pid.alive PID"  
        return 1  
    }  
    is.numeric "${pid}" || {  
        error "The argument to pid.alive() must be a numeric Process ID"  
        return 1  
    }  
    [[ -n "${pid}" && -n $(ps -p "${pid}" | grep -v TTY) ]]  
}
```

2.48.3. pid.sig

```
pid.sig ()  
{  
    local pid="${1}"  
    shift  
    local signal="${1}"  
    shift  
    [[ -z "${pid}" || -z "${signal}" ]] && {  
        printf "  
USAGE:  
    pid.sig pid signal  
"  
        return 1  
    }  
    is.numeric "${pid}" || {  
        error "First argument to pid.sig must be numeric."  
        return 1  
    }  
    is.numeric "${signal}" || sig.is-valid "${signal}" || {  
        error "First argument to pid.sig must be numeric."  
    }  
}
```

```

        return 1
    }
    if pid.alive "${pid}"; then
        info "sending ${bldred}${signal}${txt-info} to ${bldylw}${pid}..."
        /bin/kill -s "${signal}" "${pid}" 2>&1 | cat > /dev/null
    else
        warning "pid ${pid} was dead by the time we tried sending ${sig} to it."
        return 1
    fi
}

```

2.48.4. pid.stop

```

pid.stop ()
{
    local pid=${1}
    shift
    local delay=${1:-"0.3"}
    shift
    if [[ -z ${pid} ]]; then
        printf "
DESCRIPTION:
If the given PID is active, first sends kill -TERM, waits a bit,
then sends kill -9.

USAGE:
${bldgrn}pid.stop pid${clr}

EXAMPLES:
# stop all sidekiqs, waiting half a sec in between
${bldgrn}pid.stop sidekiq 0.5${clr}
"
        return 1
    fi
    pid.alive "${pid}" && ( pid.sig "${pid}" "TERM" || true ) && sleep "${delay}"
    pid.alive "${pid}" && pid.sig "${pid}" "KILL"
}

```

2.48.5. pid.stop-and-kill

```

pid.stop-and-kill ()
{
    local pid="$1"
    delta=1
    sig=STOP
    while true; do
        pid.alive "$pid" || return 0
        kill -${sig} "${pid}" 2>&1 > /dev/null

```

```

    delta=$((delta * 2))
    [[ ${delta} -gt 16 ]] && sig="KILL"
    sleep "0.${delta}"
done
pid.alive "$pid" && {
    error "PID ${pid} is miraculously still alive..." 1>&2
    return 1
}
}

```

2.48.6. pid.stop-if-listening-on-port

```

pid.stop-if-listening-on-port ()
{
    local port="$1"
    local protocol="${2:-tcp}"
    local -a pids
    pids=($(lsof -i "${protocol}":${port} | grep -v PID | awk '{print $2}'))
    local pids_string="${pids[*]}"
    if [[ ${#pids[@]} -eq 0 ]]; then
        return 0
    else
        info "Found ${#pids[@]} processes attached to port ${port}/${protocol}."
        info "Process IDs attached to ${bldcyn}${port}/${protocol}: ${bldylw}
${pids_string// /, }"
    fi
    pids.stop "${pids[@]}"
}

```

2.48.7. pids-with-args

```

pids-with-args ()
{
    local -a permitted=(%cpu "%mem" acflag acflg args blocked caught comm command
cpu cputime etime f flags gid group ignored inblk inblock jobc ktrace ktracep lim
login logname lstart majflt minflt msgrcv msgsnd ni nice nivcsn nsigals nsigs nswap
nvcsw nwchan oublk oublock p_ru paddr pagein pcpu pending pgid pid pmem ppid pri
pstime putime re rgid rgroup rss ruid ruser sess sig sigmask sl start stat stime
svgid svuid tdev time tpgid tsess tsiz tt tty ucomm uid upr user usrpri utime vsize
vsz wchan wq wqb wqr xstat)
    local -a additional=()
    local -a matching=()
    for arg in $@
    do
        array.includes "${arg}" "${permitted[@]}" && additional=(${additional[@]}
$arg) && continue
        matching=("${matching[@]} ${arg}")
    done
}

```

```

local columns="pid,ppid,user,%cpu,%mem,command"
if [[ ${#additional[@]} -gt 0 ]]; then
    columns="${columns},${array.join ', ' "${additional[@]}"}"
fi
pids.matching-regexp "${matching[*]}" | xargs /bin/ps -www -o"${columns}" -p
}

```

2.48.8. pids.all

```

pids.all ()
{
    if [[ -z "${1}" ]]; then
        printf "
DESCRIPTION:
    prints processes matching a given pattern

USAGE:
${bldgrn}pids.all pattern${clr}

EXAMPLES:
${bldgrn}pids.all puma${clr}
"
        return 0
    fi
    local pattern=$(pids.normalize.search-string "$1")
    shift
    ps -ef | ${GrepCommand} "${pattern}" | ${GrepCommand} -v grep
}

```

2.48.9. pids.for-each

```

pids.for-each ()
{
    if [[ -z "${1}" || -z "${2}" ]]; then
        printf "
DESCRIPTION:
    loops over matching PIDs and calls a named BASH function

USAGE:
${bldgrn}pids.for-each pattern function${clr}

EXAMPLES:
${bldgrn}pids.for-each puma echo
function hup() { kill -HUP \$1; }; pids.for-each sidekiq hup${clr}
"
        return 0
    fi
    local pattern=$(pids.normalize.search-string "$1")

```

```

shift
local func=${1:-"echo"}
if [[ -z $(which "${func}") && -z $(type "${func}" 2> /dev/null) ]]; then
    error "Function ${func} does not exist."
    return 1
fi
while true; do
    local -a pids=(${pids.matching "${pattern}"})
    [[ ${#pids[@]} == 0 ]] && break
    eval "${func} ${pids[0]}"
    sleep 0.1
done
}

```

2.48.10. `pids.matching`

```

pids.matching ()
{
    local pattern="${1}"
    if [[ -z "${pattern}" ]]; then
        printf "
DESCRIPTION:
    Finds process IDs matching a given string.

USAGE:
${bldgrn}pids.matching string${clr}

EXAMPLES:
${bldgrn}pids.matching sidekiq${clr}
"
        return 0
    fi
    pattern=$(pids.normalize.search-string "${pattern}")
    pids.matching.regexp "${pattern}"
}

```

2.48.11. `pids.matching.regexp`

```

pids.matching.regexp ()
{
    local pattern="${1}"
    if [[ -z "${pattern}" ]]; then
        printf "
DESCRIPTION:
    Finds process IDs matching a given regexp.

USAGE:
${bldgrn}pids.matching regular-expression${clr}

```

EXAMPLES:

```
 ${bldgrn}pids.matching '[s]idekiq\s+' ${clr}
"
    return 0
fi
ps -ef | ${GrepCommand} "${pattern}" | ${GrepCommand} -v grep | awk '{print $2}' |
sort -n
}
```

2.48.12. **pids.normalize.search-string**

```
pids.normalize.search-string ()
{
    local pattern="$*"
    [[ "${pattern:0:1}" == '[' ]] || pattern="${pattern:0:1]}${pattern:1}"
    printf "%s" "$pattern"
}
```

2.48.13. **pids.stop**

```
pids.stop ()
{
    if [[ -z "${1}" ]]; then
        printf "
DESCRIPTION:
    finds and stops IDs matching a given pattern
```

USAGE:

```
 ${bldgrn}pids.stop <pattern>${clr}
 ${bldgrn}pids.stop pid pid ... >${clr}
```

EXAMPLES:

```
 ${bldgrn}pids.stop puma${clr}
"
    return 0
fi
for pid in $@
do
    if is.numeric "${pid}"; then
        pid.stop "${pid}"
    else
        pids.for-each "${pid}" "pid.stop"
    fi
done
}
```

2.48.14. `pids.stop-by-listen-tcp-ports`

```
pids.stop-by-listen-tcp-ports ()
{
    for port in "$@"
    do
        pid.stop-if-listening-on-port "${port}"
    done
}
```

2.48.15. `pstop`

```
pstop ()
{
    pids.stop "$@"
}
```

2.48.16. `sig.is-valid`

```
sig.is-valid ()
{
    [[ -n $(kill -l "${1}" 2> /dev/null) ]]
}
```

2.48.17. `sig.list`

```
sig.list ()
{
    /bin/kill -l | sed -E 's/([ 0-9][0-9]\) SIG)//g; s/\s+/\n/g' | tr 'a-z' 'A-Z' |
sort
}
```

2.48.18. `top-most-program`

```
top-most-program ()
{
    pid=$$
    while true; do
        declare -a output=($(ps -o ppid,pid,args -p $pid | grep -v PPID))
        if [[ ${output[0]} -eq 1 ]]; then
            echo ${output[2]}
            break
        else
            if [[ ${output[0]} -gt 0 ]]; then
```

```

        pid=${output[0]}
    else
        if [[ -z ${output} ]]; then
            return 1
        fi
    fi
done
}

```

2.49. Module pipe

2.49.1. pipe.extract-variables

```

pipe.extract-variables ()
{
    sedx '/^(\s*\#.*$/D; /^\s*$/D; /^[^A-Z].*$/D' | sedx 's/export //ig' | cut -d
'=' -f 1
}

```

2.49.2. pipe.remove-blank-lines

```

pipe.remove-blank-lines ()
{
    sedx '/^$/D'
}

```

2.49.3. pipe.remove-hash-comments

```

pipe.remove-hash-comments ()
{
    sedx '/^(\s*\#.*$/D; s/#.*$/g;'
}

```

2.50. Module progress-bar

2.50.1. progress.bar.auto-run

```

progress.bar.auto-run ()
{
    .progress.reset
    .progress.bar "$@"
    code=$?
}

```

```
if [[ ${code} -ne 0 ]]; then
    .progress.reset
    return 1
fi
return 0
}
```

2.50.2. `progress.bar.config`

```
progress.bar.config ()
{
    while true; do
        local setting="$1"
        shift
        [[ -z ${setting} ]] && break
        local key=${setting/=*/}
        local value=${setting/*=/}
        eval "export LibProgress__${key}=\"${value}\""
    done
}
```

2.50.3. `progress.bar.configure.color-green`

```
progress.bar.configure.color-green ()
{
    progress.bar.config BarColor="${bldgrn}"
}
```

2.50.4. `progress.bar.configure.color-red`

```
progress.bar.configure.color-red ()
{
    progress.bar.config BarColor="${bldred}"
}
```

2.50.5. `progress.bar.configure.color-yellow`

```
progress.bar.configure.color-yellow ()
{
    progress.bar.config BarColor="${bldylw}"
}
```

2.50.6. progress.bar.configure.symbol-arrow

```
progress.bar.configure.symbol-arrow ()  
{  
    progress.bar.config BarChar="█"  
}
```

2.50.7. progress.bar.configure.symbol-bar

```
progress.bar.configure.symbol-bar ()  
{  
    progress.bar.config BarChar="█"  
}
```

2.50.8. progress.bar.configure.symbol-block

```
progress.bar.configure.symbol-block ()  
{  
    progress.bar.config BarChar="${LibProgress__BarChar__Default}"  
}
```

2.50.9. progress.bar.configure.symbol-square

```
progress.bar.configure.symbol-square ()  
{  
    progress.bar.config BarChar="██"  
}
```

2.50.10. progress.bar.launch-and-wait

```
progress.bar.launch-and-wait ()  
{  
    local command="$*"  
    run.print-command "${command}\n"  
    ${command} > /dev/null 2>&1 & local pid=$!  
    info "Waiting for background process to finish; PID=${bldylw}${pid}"  
    set -e  
    while .progress.bar.check-pid-alive $pid; do  
        progress.bar.auto-run 0.5 10  
    done  
    set +e  
    return 0  
}
```

2.51. Module repositories

2.51.1. repo.rebase

```
repo.rebase ()  
{  
    run "git pull origin main --rebase"  
}
```

2.51.2. repo.stash-and-rebase

```
repo.stash-and-rebase ()  
{  
    run "git stash >/dev/null"  
    run "git reset --hard"  
    repo.rebase  
}
```

2.51.3. repo.update

```
repo.update ()  
{  
    local folder="$1"  
    h2 "Entering repo ${bldgren}${folder}"  
    [[ -d "${folder}" ]] || return 1  
    [[ -d "${folder}/.git" ]] || return 1  
    [[ "$(pwd)" != "${folder}" ]] && {  
        cd "${folder}" || return 2  
    }  
    if [[ -z "$(git status -s)" ]]; then  
        repo.rebase  
    else  
        repo.stash-and-rebase  
    fi  
}
```

2.51.4. repos.catch-interrupt

```
repos.catch-interrupt ()  
{  
    export LibRepo__Interrupted=true  
}
```

2.51.5. repos.init-interrupt

```
repos.init-interrupt ()  
{  
    export LibRepo__Interrupted=false  
    trap 'repos.catch-interrupt' SIGINT  
}
```

2.51.6. repos.recursive-update

```
repos.recursive-update ()  
{  
    local repo="${1}"  
    run.set-all show-output-off  
    if [[ ${LibRepo__Interrupted} == true ]]; then  
        warn "Detected SINGINT, exiting..."  
        return 2  
    fi  
    if [[ -n "$repo" ]]; then  
        repo.update "$repo"  
    else  
        for dir in $(find . -type d -name '.git')  
        do  
            local subdir=$(dirname "$dir")  
            [[ -n "${BASHMATIC_DEBUG}" ]] && info "checking out sub-folder ${bldcyn}  
${subdir}..."  
            repos.recursive-update "${subdir}"  
            if [[ $? -eq 2 ]]; then  
                error "folder ${bldylw}${subdir}${bldred} return error!"  
                return 2  
            fi  
        done  
    fi  
    if [[ -n ${repo} ]]; then  
        info "returning to the root dir ${bldylw}${root_folder}..."  
        cd "${root_folder}" > /dev/null || return 2  
    fi  
}
```

2.51.7. repos.update

```
repos.update ()  
{  
    export root_folder="$(pwd)"  
    bash -c "  
[[ -d ${BASHMATIC_HOME} ]] || {  
    echo 'Can not find bashmatic installation sorry'  
}
```

```
    return
}
source ${BASHMATIC_HOME}/init.sh
repos.init-interrupt
repos.recursive-update '$*'
"
}
```

2.51.8. repos.was-interrupted

```
repos.was-interrupted ()
{
  [[ ${LibRepo__Interrupted} == true ]]
}
```

2.52. Module ruby

2.52.1. bundle.gems-with-c-extensions

```
bundle.gems-with-c-extensions ()
{
  run.set-next show-output-on
  run "bundle show --paths | ruby -e \"STDIN.each_line{|dep| puts
dep.split('/').last if File.directory?(File.join(dep.chomp, 'ext')) }\""
}
```

2.52.2. interrupted

```
interrupted ()
{
  export BashMatic__Interrupted=true
}
```

2.52.3. ruby.aliases

```
ruby.aliases ()
{
  alias b="bundle"
  alias be="bundle exec"
  alias ber="bundle exec rake"
  alias bert="bundle exec rake -T"
  alias berr="bundle exec rspec"
  alias rdb="set -ex; bundle exec rake db:drop:all; bundle exec rake db:create:all;
bundle exec rake db:migrate db:seed; bundle exec rake db:test:prepare; set +ex"
```

```
}
```

2.52.4. ruby.bundle-install

```
ruby.bundle-install ()  
{  
    if [[ -f Gemfile.lock ]]; then  
        run "bundle install"  
    fi  
}
```

2.52.5. ruby.bundler-version

```
ruby.bundler-version ()  
{  
    if [[ ! -f Gemfile.lock ]]; then  
        error "Can not find Gemfile.lock"  
        return 1  
    fi  
    tail -1 Gemfile.lock | sedx 's/ //g'  
}
```

2.52.6. ruby.compiled-with

```
ruby.compiled-with ()  
{  
    if [[ -z "$*" ]]; then  
        error "usage: ruby.compiled-with <library>"  
        return 1  
    fi  
    ruby -r rbconfig -e "puts RbConfig::CONFIG['LIBS']" | grep -q "$*"  
}
```

2.52.7. ruby.default-gems

```
ruby.default-gems ()  
{  
    declare -a DEFAULT_RUBY_GEMS=(bundler rubocop relaxed-rubocop rubocop-performance  
warp-dir colored2 sym pry pry-doc pry-byebug rspec rspec-its awesome_print  
activesupport pivotal_git_scripts git-smart travis awscli irbtools kramdown-asciidoc  
asciidoctor kramdown gemsmith rspec rspec-its)  
    export DEFAULT_RUBY_GEMS  
    printf "${DEFAULT_RUBY_GEMS[*]}"  
}
```

2.52.8. ruby.ensure-rbenv

```
ruby.ensure-rbenv ()  
{  
    [[ -n $(command -v rbenv) ]] && return 0  
    brew.install  
    brew.install.package rbenv ruby-build  
    grep -q "rbenv init" ~/.bash_profile && echo 'eval "$(rbenv init -)"' >>  
    ~/.bash_profile  
    [[ -n $(command -V rbenv) ]] && return 0  
    return 1  
}
```

2.52.9. ruby.ensure-rbenv-or-complain

```
ruby.ensure-rbenv-or-complain ()  
{  
    ruby.ensure-rbenv || {  
        error "Can't install rbenv via HomeBrew, please try manually."  
        return 1  
    }  
    return 0  
}
```

2.52.10. ruby.full-version

```
ruby.full-version ()  
{  
    /usr/bin/env ruby --version  
}
```

2.52.11. ruby.gemfile-lock-version

```
ruby.gemfile-lock-version ()  
{  
    local gem=${1}  
    if [[ ! -f Gemfile.lock ]]; then  
        error "Can not find Gemfile.lock"  
        return 1  
    fi  
    ${GrepCommand} " ${gem} \([0-9]\)" Gemfile.lock | sed -e 's/[\(\)]//g' | awk '{print  
$2}'  
}
```

2.52.12. ruby.gems

```
ruby.gems ()  
{  
    ruby.gems.install "$@"  
}
```

2.52.13. ruby.gems.install

```
ruby.gems.install ()  
{  
    local -a gems=("$@")  
    gem.clear-cache  
    [[ ${#gems[@]} -eq 0 ]] && gems=(${ruby.default-gems})  
    local -a existing=(${ruby.installed-gems})  
    [[ ${#gems[@]} -eq 0 ]] && {  
        error 'Unable to determine what gems to install. ' "Argument is empty, so is  
        ${DEFAULT_RUBY_GEMS[@]}" "USAGE: ${bldgrn}ruby.gems ${bldred} rails rubocop puma pry"  
        return 1  
    }  
    h2 "There are a total of ${#existing[@]} of globally installed Gems." "Total of  
    ${#gems[@]} need to be installed unless they already exist. " "${bldylw}Checking for  
    gems that still missing..."  
    local -a gems_to_be_installed()  
    for gem in "${gems[@]}"  
    do  
        local gem_info=  
        if [[ $(array.has-element "${gem}" "${existing[@]}") == "true" ]]; then  
            gem_info="${bldgrn} ✓ ${gem}${clr}\n"  
        else  
            gem_info="${bldred} ✗ ${gem}${clr}\n"  
            gems_to_be_installed+=("${gem}")  
        fi  
        printf "    ${gem_info}"  
    done  
    if [[ ${#gems_to_be_installed[@]} -eq 0 ]]; then  
        info "All gems are already installed. ☺"  
        return 0  
    fi  
    info "Looks like ${#gems_to_be_installed[@]} gems are left to install..."  
    local -a gem_installed  
    for gem in "${gems_to_be_installed[@]}"  
    do  
        run "gem install -q --force --no-document $gem"  
        if [[ ${LibRun__LastExitCode} -ne 0 ]]; then  
            error "Gem ${gem} refuses to install." "Perhaps try installing it  
manually?" "${bldgrn}Action: Skip and Continuing..."  
            break  
        else
```

```

        gem_installed+="$gem"
        continue
    fi
done
gem.clear-cache
info "Total of ${#gem_installed[@]} gems were installed."
echo
}

```

2.52.14. ruby.gems.uninstall

```

ruby.gems.uninstall ()
{
    local -a gems
    gems="$@"
    gem.clear-cache
    [[ ${#gems[@]} -eq 0 ]] && declare -a gems=(${!ruby.default-gems})
    local -a existing=(${!ruby.installed-gems})
    [[ ${#gems[@]} -eq 0 ]] && {
        error "Unable to determine what gems to remove. Argument is empty, so is
${DEFAULT_RUBY_GEMS[@]}"
        "USAGE: ${bldgrn}ruby.gems.uninstall ${bldred} rails rubocop
puma pry"
        return 1
    }
    h1.blue "There are a total of ${#existing[@]} of gems installed in a global
namespace." "Total of ${#gems[@]} need to be removed."
    local deleted=0
    for gem in "${gems[@]}"
    do
        local gem_info=
        if [[ ${array.has-element "$gem" "${existing[@]}"} == "true" ]]; then
            run "gem uninstall -a -x -I -D --force ${gem}"
            deleted=$((deleted + 1))
        else
            gem_info="${bldred} x [not found] ${bldylw}${gem}${clr}\n"
        fi
        printf "  ${gem_info}"
    done
    gem.clear-cache
    echo
    success "Total of ${deleted} gems were successfully obliterated."
    echo
}

```

2.52.15. ruby.handle-missing

```

ruby.handle-missing ()
{

```

```

command -v ruby > /dev/null || {
    info "Couldn't find Ruby, installing it..." 1>&2
    ruby.install-ruby "$(ruby.numeric-version)"
}
command -v ruby > /dev/null
}

```

2.52.16. `ruby.init`

```

ruby.init ()
{
    h1 "Installing Critical Gems for Your Glove, Thanos..."
    ruby.rubygems-update
    ruby.install-upgrade-bundler
    ruby.gems.install
    ruby.kigs-gems
}

```

2.52.17. `ruby.install`

```

ruby.install ()
{
    ruby.install-ruby "$@"
}

```

2.52.18. `ruby.install-ruby`

```

ruby.install-ruby ()
{
    local version="$1"
    shift
    local version_source="provided as an argument"
    if [[ -z ${version} && -f .ruby-version ]]; then
        version=$(cat .ruby-version | tr -d '\n')
        version_source="auto-detected from .ruby-version file"
    fi
    [[ -z ${version} ]] && {
        error "USAGE: ruby.install-ruby VERSION" "Or, you can create a local .ruby-
version file"
        return 1
    }
    local -a required_packages
    required_packages=(rbenv ruby-build)
    h3 "Installing Ruby Version ${bldpur}${version} ${bldblu}${version_source}."
    ruby.validate-version "${version}" || return 1
    brew.install.packages "${required_packages[@]}"
}

```

```

brew.upgrade.packages "${required_packages[@]}"
if [[ -n "$*" ]]; then
    info "Attempting to install additional packages via Brew:"
    for package in "$@"
    do
        run.set-next abort-on-error
        brew.install.package "${package}"
        local func=".ruby.configure-with.${package}"
        util.is-a-function "${func}" && ${func}
    done
fi
eval "$(rbenv init -)"
h2 "RUBY_CONFIGURE_OPTS: ${bldgrn}${RUBY_CONFIGURE_OPTS}"
run "RUBY_CONFIGURE_OPTS=\"${RUBY_CONFIGURE_OPTS}\\" rbenv install -s ${version}"
return "${LibRun__LastExitCode:-"0"}"
}

```

2.52.19. ruby.install-ruby-with-deps

```

ruby.install-ruby-with-deps ()
{
    local version="$1"
    declare -a packages=(cask bash bash-completion git go haproxy htop jemalloc
libxslt jq libiconv libzip netcat nginx openssl pcre pstree p7zip rbenv redis
ruby_build readline tree vim watch wget zlib)
    run.set-next show-output-on
    run "brew install --display-times ${packages[*]}"
}

```

2.52.20. ruby.install-ruby-with-readline-and-openssl

```

ruby.install-ruby-with-readline-and-openssl ()
{
    local version="$1"
    [[ -z ${version} ]] && {
        error "usage: ruby.install-ruby-with-readline-and-openssl ruby-version"
        return 1
    }
    shift
    ruby.install-ruby "${version}" openssl readline "$@"
}

```

2.52.21. ruby.install-upgrade-bundler

```

ruby.install-upgrade-bundler ()
{

```

```
gem.install bundler
run "bundle --update bundler || true"
}
```

2.52.22. ruby.installed-gems

```
ruby.installed-gems ()
{
    gem list | cut -d ' ' -f 1 | uniq
}
```

2.52.23. ruby.kigs-gems

```
ruby.kigs-gems ()
{
    if [[ -z $(type wd 2> /dev/null) && -n $(command -v warp-dir) ]]; then
        [[ -f ~/.bash_wd ]] || {
            warp-dir install --dotfile ~/.bashrc > /dev/null
            source ~/.bash_wd
        }
    fi
    [[ -n $(command -v sym) ]] && {
        [[ -f ~/.sym.completion.bash ]] || {
            sym -B ~/.bashrc
        }
    }
}
```

2.52.24. ruby.linked-libs

```
ruby.linked-libs ()
{
    ruby -r rbconfig -e "puts RbConfig::CONFIG['LIBS']"
}
```

2.52.25. ruby.numeric-version

```
ruby.numeric-version ()
{
    /usr/bin/env ruby --version | sed 's/^ruby //g; s/ (.*//g'
```

2.52.26. ruby.rbenv

```
ruby.rbenv ()
{
    ruby.ensure-rbenv-or-complain || return 1
    if [[ -n "$*" ]]; then
        rbenv "$*"
    else
        eval "$(rbenv init -)"
    fi
    run "rbenv rehash"
}
```

2.52.27. ruby.rubygems-update

```
ruby.rubygems-update ()
{
    info "Updating RubyGems..."
    run.set-next show-output-on
    run "gem update --system -N"
    gem.clear-cache
}
```

2.52.28. ruby.stop

```
ruby.stop ()
{
    local regex='/[r]uby| [p]uma| [i]rb| [r]ails | [b]undle| [u]nicorn| [r]ake'
    local procs=$(ps -ef | ${GrepCommand} "${regex}" | ${GrepCommand} -v grep | awk
'{print $2}' | sort | uniq | wc -l)
    [[ ${procs} -eq 0 ]] && {
        info: "No ruby processes were found."
        return 0
    }
    local -a pids=$(ps -ef | ${GrepCommand} "${regex}" | ${GrepCommand} -v grep | awk
'{print $2}' | sort | uniq | tr '\n' ' -p ')
    h2 "Detected ${#pids[@]} Ruby Processes..., here is the tree:"
    printf "${txtcyn}"
    pstree "${pids[*]}"
    printf "${clr}"
    hr
    printf "To abort, press Ctrl-C. To kill them all press any key.."
    run.ui.press-any-key
    ps -ef | ${GrepCommand} "${regex}" | ${GrepCommand} -v grep | awk '{print $2}' |
    sort | uniq | xargs kill -9
}
```

2.52.29. `ruby.top-versions`

```
ruby.top-versions ()  
{  
    local platform="${1}"  
    local arg="$(.ruby.ruby-build.list-argument)"  
    local filter="cat"  
    [[ -n ${platform} ]] && filter="grep -E '^${platform}'"  
    eval "rbenv install ${arg}" | eval "${filter}" | ruby -e '  
        last_v = nil  
        last_m = nil  
        ARGF.each do |line|  
            v = line.split(".")[0..1].join(".")  
            if last_v != v  
                puts last_m if last_m  
                last_v = v  
            end  
            last_m = line  
        end  
        puts last_m if last_m'  
}
```

2.52.30. `ruby.top-versions-as-yaml`

```
ruby.top-versions-as-yaml ()  
{  
    ruby.top-versions | sed 's/^/ - /g'  
}
```

2.52.31. `ruby.validate-version`

```
ruby.validate-version ()  
{  
    local version="$1"  
    local -a ruby_versions=()  
    run "brew upgrade ruby-build || true"  
    [[ -d ~/.rbenv/plugins/ruby-build ]] && {  
        run "cd ~/.rbenv/plugins/ruby-build && git reset --hard && git pull --rebase"  
    }  
    local arg="$(.ruby.ruby-build.list-argument)"  
    array.from.command ruby_versions "rbenv install ${arg}"  
    inf "Validating ruby version: ${version}"  
    array.includes "${version}" "${ruby_versions[@]}" || {  
        not-ok:  
        error "Ruby Version provided was NOT found by rbenv: ${bldylw}${version}"  
    "Found a total of ${bldgrn}${#ruby_versions[*]} ruby versions."  
    return 1  
}
```

```
    }
    ok:
    return 0
}
```

2.53. Module `run`

2.53.1. `run`

```
run () {
    .run "$@"
    return ${LibRun__LastExitCode}
}
```

2.53.2. `run.ui.ask`

```
run.ui.ask () {
    local question=$*
    local func="${LibRun__AskDeclineFunction}"
    export LibRun__AskDeclineFunction="${LibRun__AskDeclineFunction__Default}"
    echo
    inf "${bldcyn}${question}${clr} [Y/n] ${bldylw}"
    read a 2> /dev/null
    code=$?
    if [[ ${code} != 0 ]]; then
        error "Unable to read from STDIN."
        eval "${func} 12"
    fi
    echo
    if [[ ${a} == 'y' || ${a} == 'Y' || ${a} == '' ]]; then
        info "${bldblu}Roger that."
        info "Let's just hope it won't go nuclear on us :)"
        hr
        echo
    else
        info "${bldred}(Great idea!) Abort! Abandon ship!  ↵  " 1>&2
        hr 1>&2
        echo
        eval "${func} 1"
    fi
}
```

2.53.3. run.ui.ask-user-value

```
run.ui.ask-user-value ()  
{  
    local variable="$1"  
    shift  
    local text="$*"  
    local user_input  
    trap 'echo; echo Aborting at user request... ; echo; abort; return' int  
    ask "${text}"  
    read user_input  
    if [[ -z "${user_input}" ]]; then  
        error "Sorry, I didn't get that. Please try again or press Ctrl-C to abort."  
        return 1  
    else  
        eval "export ${variable}=\${user_input}"  
        return 0  
    fi  
}
```

2.53.4. run.ui.get-user-value

```
run.ui.get-user-value ()  
{  
    run.ui.retry-command run.ui.ask-user-value "${@}"  
}
```

2.53.5. run.ui.press-any-key

```
run.ui.press-any-key ()  
{  
    local prompt="$*"  
    [[ -z ${prompt} ]] && prompt="Press any key to continue..."  
    br  
    printf "    ${txtgrn}${italic}${prompt} ${clr}  "  
    read -r -s -n1 key  
    cursor.rewind  
    printf ""  
    cursor.up 2  
    cursor.rewind  
    echo  
}
```

2.53.6. run.ui.retry-command

```
run.ui.retry-command ()
```

```
{
    local command="$*"
    local retries=5
    n=0
    until [ $n -ge ${retries} ]; do
        [[ ${n} -gt 0 ]] && info "Retry number ${n}..."
        command && break
        n=$((n + 1))
        sleep 1
    done
}
```

2.54. Module `runtime`

2.54.1. `command-spacer`

```
command-spacer () {
    local color="${LibRun__CommandColor}"
    [[ ${LibRun__LastExitCode} -ne 0 ]] && color="${txtred}"
    [[ -z ${LibRun__AssignedWidth} || -z ${LibRun__CommandLength} ]] && return
    printf "%s${color}" ""
    local __width=$((LibRun__AssignedWidth - LibRun__CommandLength - 10))
    [[ ${__width} -gt 0 ]] && .output.replicate-to "${LibRun__CommandColor} " "
    ${__width}"
    printf "${clr}"
}
```

2.54.2. `run`

```
run () {
    .run "$@"
    return ${LibRun__LastExitCode}
}
```

2.54.3. `run.add-obfuscated-var`

```
run.add-obfuscated-var () {
    while true; do
        local variable="$1"
        shift
        [[ -n ${variable} ]] || break
        export OBFUSCATED_VARIABLES+=("${variable}")
    done
}
```

```
        export OBFUSCATED_VARIABLES=(${array.uniq "${OBFUSCATED_VARIABLES[@]}"})
    }
```

2.54.4. `run.config.detail-is-enabled`

```
run.config.detail-is-enabled ()
{
    [[ ${LibRun__Detail} -eq ${True} ]]
}
```

2.54.5. `run.config.is-dry-run`

```
run.config.is-dry-run ()
{
    [[ ${LibRun__DryRun} -eq ${True} ]]
}
```

2.54.6. `run.config.verbose-is-enabled`

```
run.config.verbose-is-enabled ()
{
    [[ ${LibRun__Verbose} -eq ${True} ]]
}
```

2.54.7. `run.dry-run-prefix`

```
run.dry-run-prefix ()
{
    if [[ ${LibRun__DryRun} == ${True} ]]; then
        printf "${txtcyn}${italic}« dry run »${clr} "
    fi
}
```

2.54.8. `run.filter-out-sensitive-vars`

```
run.filter-out-sensitive-vars ()
{
    local a="${1:-""}"
    if [[ "$a" == "-" ]]; then
        read -r a
    else
        if [[ -f "${a}" ]]; then
            grep -v -E "${SENSITIVE_VARS_REGEX}" "${a}"
        fi
    fi
}
```

```

        else
            echo | grep -v -E "${SENSITIVE_VARS_REGEX}"
        fi
    fi
}

```

2.54.9. `run.inspect`

```

run.inspect ()
{
    if [[ ${#@} -eq 0 || ${array.has-element "config" "$@") == "true" ]]; then
        run.inspect-variables-that-are starting-with LibRun
    fi
    if [[ ${#@} -eq 0 || ${array.has-element "totals" "$@") == "true" ]]; then
        hl.subtle "TOTALS"
        info "${bldgrn}${commands_completed} commands completed successfully"
        [[ ${commands_failed} -gt 0 ]] && info "${bldred}${commands_failed} commands
failed"
        [[ ${commands_ignored} -gt 0 ]] && info "${bldylw}${commands_ignored} commands
failed, but were ignored."
        echo
    fi
    if [[ ${#@} -eq 0 || ${array.has-element "current" "$@") == "true" ]]; then
        run.inspect-variables-that-are ending-with __LastExitCode
    fi
    reset-color
}

```

2.54.10. `run.inspect-variable`

```

run.inspect-variable ()
{
    local var_name=${1}
    local var_value=${!var_name}
    local value=""
    local print_value
    local obfuscated_value
    local max_len=100
    local avail_len=$(((${screen.width.actual} - 45)))
    local lcase_var_name=$(echo "${var_name}" | tr '[:upper:]' '[:lower:]')
    print_value=1
    local print_value
    array.includes "${var_name}" "${OBFUSCATED_VARIABLES[@]}" && obfuscated_value=
    $(run.obfuscate-string-value "${var_name}")
    array.includes "${lcase_var_name}" "${OBFUSCATED_VARIABLES[@]}" &&
    obfuscated_value=$(run.obfuscate-string-value "${lcase_var_name}")
    local color="${txtblk}${bakpur}"
    local value_off=" "

```

```

local value_check=" ✓"
local value_color=""
[[ -n ${obfuscated_value} ]]; && {
    obfuscated_value=<obfuscated-value-${obfuscated_value}>
    value_color="${italic}${txtred}"
}
if [[ -n "${var_value}" ]]; then
    if [[ ${var_name} =~ ${SENSITIVE_VARS_REGEX} || ${var_name} =~
${SENSITIVE_VARS_REGEX^^} ]]; then
        var_value=$(run.obfuscate-string-value ${var_name}) [obfuscated]
        color="${itawht}${italic}${bakcyn}"
    else
        if [[ ${lcase_var_name} =~ 'exit' ]]; then
            if [[ ${var_value} -eq 0 ]]; then
                var_value="${value_check} [zero]"
                color="${bakgrn}"
            else
                print_value=1
                var_value=${var_value}
                color="${bakred}"
                avail_len=$((avail_len + 5))
            fi
            avail_len=$((avail_len + 5))
        else
            if var.is-truthy "${var_value}"; then
                var_value="${value_check} [true]"
                color="${bakgrn}"
                avail_len=$((avail_len + 5))
            else
                if var.is-falsy "${var_value}"; then
                    var_value="${value_off} [false]"
                    color="${bakred}"
                    avail_len=$((avail_len + 2))
                fi
            fi
        fi
    fi
else
    var_value=" — [empty]"
    color="${bakpur}"
    avail_len=$((avail_len + 2))
fi
printf -- " ${txtylw} %-40s ${txtblk}${color} " "${var_name}"
[[ ${avail_len} -gt ${max_len} ]] && avail_len=${max_len}
local dot_count=$(echo "${var_value}" | sedx -E 's/[^.]/\./g' | tr -d '\n' | wc -c
)
if [[ "${print_value}" -eq 1 ]]; then
    if [[ -n ${obfuscated_value} ]]; then
        print_value="${obfuscated_value}"
        var_value="${obfuscated_value}"
        value="${obfuscated_value}"
    else
        if [[ ${lcase_var_name} =~ 'exit' ]]; then
            if [[ ${var_value} -eq 0 ]]; then
                print_value="${value_check} [zero]"
            else
                print_value=" ${txtylw} ${var_value} ${txtblk} "
            fi
        else
            if var.is-truthy "${var_value}"; then
                print_value=" ${txtylw} ${var_value} ${txtblk} "
            else
                if var.is-falsy "${var_value}"; then
                    print_value=" ${txtylw} ${var_value} ${txtblk} "
                fi
            fi
        fi
    fi
fi

```

```

    fi
    if [[ -n "${value}" ]]; && ! is.numeric "${var_value}"; then
        echo "XXX"
    else
        if is.numeric "${var_value}"; then
            avail_len=$((avail_len))
            if [[ ${dot_count} -gt 1 || ${dot_count} -gt 1 ]]; then
                echo "XXX"
                printf -- "${value_color}[%-*.s]" "${avail_len}" "${avail_len}"
                "${var_value}"
            else
                if [[ "${var_value}" =~ \. ]]; then
                    printf -- "${value_color}%-*.*f" "$((avail_len))" "${var_value}"
                else
                    printf -- "${value_color}%-*d" "$((avail_len))" "${var_value}"
                fi
            fi
        else
            printf -- "${value_color}%-*.*s" "${avail_len}" "${avail_len}" "${var_value}"
        fi
    else
        avail_len=$((avail_len))
        printf -- "${value_color}%-*.*s" "${avail_len}" "${avail_len}" "${var_value}"
    fi
    printf "${clr}\n"
}

```

2.54.11. `run.inspect-variables`

```

run.inspect-variables ()
{
    local title=${1}
    shift
    output.constrain-screen-width 100
    h3bg "${title}"
    for var in $@
    do
        run.inspect-variable "${var}"
    done
}

```

2.54.12. `run.inspect-variables-that-are`

```

run.inspect-variables-that-are ()
{

```

```

local pattern_type="${1}"
local pattern="${2}"
run.inspect-variables "VARIABLES $(echo "${pattern_type}" | tr '[:lower:]'
'[:upper:]') ${pattern}" "$(run.variables-"${pattern_type}" "${pattern}" | tr '\n' '
')
}

```

2.54.13. `run.inspect-vars`

```

run.inspect-vars ()
{
    for var in "$@"
    do
        run.inspect-variables-that-are-starting-with "$var"
    done
}

```

2.54.14. `run.inspect.set-skip-false-or-blank`

```

run.inspect.set-skip-false-or-blank ()
{
    local value="${1}"
    [[ -n "${value}" ]] && export LibRun__Inspect__SkipFalseOrBlank=${value}
    [[ -z "${value}" ]] && export LibRun__Inspect__SkipFalseOrBlank=${True}
}

```

2.54.15. `run.obfuscate-string-value`

```

run.obfuscate-string-value ()
{
    local value="$1"
    local len="30"
    local sha=$(echo "${value}" | sha512sum | cut -d ' ' -f 1)
    printf "%s" "${sha:0:${len}}"
}

```

2.54.16. `run.on-error.ask-is-enabled`

```

run.on-error.ask-is-enabled ()
{
    [[ ${LibRun__AskOnError} -eq ${True} ]]
}

```

2.54.17. `run.post-command-with-output`

```
run.post-command-with-output ()  
{  
    local duration="$1"  
    if [[ ${LibRun__ShowCommand} -eq ${True} ]]; then  
        command-spacer  
        duration "${duration}" ${LibRun__LastExitCode}  
    fi  
}
```

2.54.18. `run.print-command`

```
run.print-command ()  
{  
    local command="$1"  
    local max_width=${2:-"150"}  
    local min_width=60  
    local w  
    w=$((screen-width) - 10))  
    [[ $w -gt ${max_width} ]] && w=${max_width}  
    export LibRun__AssignedWidth=$w  
    local prefix="${LibOutput__LeftPrefix}${clr}"  
    local ascii_cmd  
    local command_prompt="${prefix} " "  
    local command_width=$((w - 25))  
    [[ ${command_width} -lt ${min_width} ]] && command_width=${min_width}  
    ascii_cmd=$(printf "${command_prompt}${run.dry-run-prefix}%-.${command_width}s "  
    "${command:0:${command_width}}")  
    export LibRun__CommandLength=${#ascii_cmd}  
    if [[ "${LibRun__ShowCommand}" -eq ${False} ]]; then  
        printf -- "${prefix}${LibRun__PromptColor} ${LibRun__CommandColor} %-.  
    ${command_width}s " "$(.output.replicate-to " " ${command_width})"  
    else  
        printf -- "${prefix}${LibRun__PromptColor} ${LibRun__CommandColor} %-.  
    ${command_width}s " "${command:0:${command_width}}"  
    fi  
}
```

2.54.19. `run.print-command-full-screen`

```
run.print-command-full-screen ()  
{  
    run.print-long-command "$1" "$(screen.width)"  
}
```

2.54.20. `run.print-long-command`

```
run.print-long-command ()  
{  
    local command="$1"  
    local max_width=${2:-"150"}  
    local w  
    w=$((.output.screen-width - 10))  
    [[ ${w} -gt ${max_width} ]] && w=${max_width}  
    export LibRun__AssignedWidth=${w}  
    local prefix="${LibRun__PromptColor}${LibOutput__LeftPrefix}${clr}"  
    local ascii_cmd  
    local command_prompt="${prefix} ${run.dry-run-prefix}"  
    local command_width=$((w - 10))  
    printf "${prefix} ${bldylw}"  
    printf "${command}" | fold -s -w"${w}" | awk 'NR > 1 {printf "          "}; {  
        printf "%s\n", $0}'  
}
```

2.54.21. `run.print-obfuscated-vars`

```
run.print-obfuscated-vars ()  
{  
    for var in "${OBFUSCATED_VARIABLES[@]}"  
    do  
        run.inspect-variable "${var}"  
    done  
}
```

2.54.22. `run.print-variable`

```
run.print-variable ()  
{  
    run.inspect-variable "$1"  
}
```

2.54.23. `run.print-variables`

```
run.print-variables ()  
{  
    local title="${1}"  
    shift  
    hl.yellow "${title}"  
    for var in $@  
    do  
        run.print-variable "${var}"  
    done  
}
```

```
        done  
    }
```

2.54.24. `run.register-obfuscated-vars`

```
run.register-obfuscated-vars ()  
{  
    run.add-obfuscated-var password api_key secret token  
}
```

2.54.25. `run.ui.press-any-key`

```
run.ui.press-any-key ()  
{  
    local prompt="$*"  
    [[ -z ${prompt} ]] && prompt="Press any key to continue..."  
    br  
    printf "    ${txtgrn}${italic}${prompt} ${clr}  "  
    read -r -s -n1 key  
    cursor.rewind  
    printf ""  
    cursor.up 2  
    cursor.rewind  
    echo  
}
```

2.54.26. `run.variables-ending-with`

```
run.variables-ending-with ()  
{  
    local suffix="${1}"  
    env | grep -E -e ".*$suffix=.*\$" | grep -v -E "${SENSITIVE_VARS_REGEX}" | grep '=' | sedx 's/=.*//g' | sort  
}
```

2.54.27. `run.variables-starting-with`

```
run.variables-starting-with ()  
{  
    local prefix="${1}"  
    env | grep -E -e "^.${prefix}" | grep '=' | sedx 's/=.*//g' | grep -v -E "${SENSITIVE_VARS_REGEX}" | sort  
}
```

2.54.28. `run.was-successful`

```
run.was-successful ()  
{  
    [[ ${LibRun__LastExitCode} -eq 0 ]]  
}
```

2.54.29. `run.with.minimum-duration`

```
run.with.minimum-duration ()  
{  
    local min_duration=$1  
    shift  
    local command="$*"  
    local started=$(millis)  
    info "starting a command with the minimum duration of ${bldylw}${min_duration}  
seconds"  
    run "$command"  
    local result=$?  
    local now=$(millis)  
    local duration=$(ruby -e "puts (${now} - ${started})/1000.0")  
    if [[ ${result} -eq 0 && ${duration} -lt ${min_duration} ]]; then  
        local cmd=$(echo "${command}" | sed 's//g')  
        error "An operation finished too quickly. The threshold was set to ${bldylw}  
${min_duration} sec." "The command took ${bldylw}${duration}${txtred} secs."  
        ${bldylw}${cmd}${txtred}"  
        ((BASH_IN_SUBSHELL)) && exit 1 || return 1  
    else  
        if [[ ${duration} -gt ${min_duration} ]]; then  
            info "minimum duration operation ran in ${duration} seconds."  
        fi  
    fi  
    return ${result}  
}
```

2.54.30. `run.with.ruby-bundle`

```
run.with.ruby-bundle ()  
{  
    .run.bundle.exec "$@"  
}
```

2.54.31. `run.with.ruby-bundle-and-output`

```
run.with.ruby-bundle-and-output ()  
{
```

```
.run.bundle.exec.with-output "$@"
}
```

2.54.32. var.is-falsy

```
var.is-falsy ()
{
    local value="${1}"
    [[ -z "${value}" || "${value}" == "${False}" || "${value}" == "0" || ${value} ==
>false" ]]
}
```

2.54.33. var.is-truthy

```
var.is-truthy ()
{
    local value="${1}"
    [[ "${value}" == "${True}" || "${value}" == "1" || ${value} == "true" ]]
}
```

2.55. Module runtime-config

2.55.1. is.dry-run.off

```
is.dry-run.off ()
{
    ((LibRun__DryRun)) || return 0
    return 1
}
```

2.55.2. is.dry-run.on

```
is.dry-run.on ()
{
    ((LibRun__DryRun))
}
```

2.55.3. run.inspect

```
run.inspect ()
{
    if [[ ${#@} -eq 0 || ${array.has-element "config" "$@") == "true" ]]; then
```

```

run.inspect-variables-that-are starting-with LibRun
fi
if [[ ${#@} -eq 0 || ${array.has-element "totals" "$@"} == "true" ]]; then
    hl.subtle "TOTALS"
    info "${bldgrn}${commands_completed} commands completed successfully"
    [[ ${commands_failed} -gt 0 ]] && info "${bldred}${commands_failed} commands
failed"
    [[ ${commands_ignored} -gt 0 ]] && info "${bldylw}${commands_ignored} commands
failed, but were ignored."
    echo
fi
if [[ ${#@} -eq 0 || ${array.has-element "current" "$@"} == "true" ]]; then
    run.inspect-variables-that-are ending-with _LastExitCode
fi
reset-color
}

```

2.55.4. `run.set-all`

```

run.set-all ()
{
    ___run.configure all "$@"
}

```

2.55.5. `run.set-all.list`

```

run.set-all.list ()
{
    set | ${GrepCommand} '^___run.set.all' | awk 'BEGIN{FS="."){print $4}' | sedx
's/[() ]//g'
}

```

2.55.6. `run.set-next`

```

run.set-next ()
{
    ___run.configure next "$@"
}

```

2.55.7. `run.set-next.list`

```

run.set-next.list ()
{
    set | ${GrepCommand} '^___run.set.next' | awk 'BEGIN{FS="."){print $4}' | sedx
's/[() ]//g'
}

```

```
}
```

2.55.8. `set.dry-run.off`

```
set.dry-run.off ()  
{  
    export LibRun__DryRun=0  
}
```

2.55.9. `set.dry-run.on`

```
set.dry-run.on ()  
{  
    export LibRun__DryRun=1  
}
```

2.56. Module `sedx`

2.56.1. `sedx`

```
sedx ()  
{  
    [[ -z ${bashmatic__sed_command} ]] && sedx.cache-command  
    [[ -z ${bashmatic__sed_command} ]] && {  
        warning "Can't determine advanced sed location, using regular sed."  
        1>&2  
        export bashmatic__sed_command="/usr/bin/sed"  
    }  
    ${bashmatic__sed_command} -E "$@"  
}
```

2.56.2. `sedx.cache-command`

```
sedx.cache-command ()  
{  
    local sed_path=$(util.ensure-gnu-sed)  
    [[ -x "${sed_path}" ]] || sed_path=$(command -v gsed || command -v sed)  
    export bashmatic__sed_command="${sed_path}"  
}
```

2.56.3. `util.ensure-gnu-sed`

```

util.ensure-gnu-sed ()
{
    local sed_path
    local gsed_path
    util.os
    case "${BASHMATIC_OS}" in
        darwin)
            gsed_path=$(command -v gsed 2> /dev/null)
            if [[ -z "${gsed_path}" ]]; then
                echo
                h3 "Please wait while we install gnu-sed using Brew..." "It's a
required dependency for many key features." 1>&2
                ( brew install gnu-sed --force --quiet && brew unlink gnu-sed
                  brew link gnu-sed --overwrite ) 1>&2 > /dev/null
                hash -r 2> /dev/null
                gsed_path=$(command -v gsed 2> /dev/null)
                [[ -z ${gsed_path} && -x /usr/local/bin/gsed ]] && gsed_path
                ="/usr/local/bin/gsed"
            fi
            [[ -n "${gsed_path}" && -x "${gsed_path}" ]] || {
                error "Can't find GNU sed even after installation." 1>&2
            }
            sed_path="${gsed_path}"
        ;
        *)
            sed_path=$(command -v sed)
        ;
    esac
    echo -n "${sed_path}"
}

```

2.57. Module serial

2.57.1. serial.baud

```

serial.baud ()
{
    if [[ -n $1 ]]; then
        is.number "$1" && export BashmaticSerial__DefaultBaud="$1"
    fi
    h2 "Current Baud Rate:" "${bldylw}${BashmaticSerial__DefaultBaud} Bits/Second"
}

```

2.57.2. serial.device

```
serial.device ()
```

```

{
    if [[ -n $1 && $1 =~ ^/dev ]]; then
        export BashmaticSerial__DefaultDevice="$1"
    else
        export BashmaticSerial__DefaultDevice=$(ls -1 /dev/cu.* | grep -vi bluetooth | tail -1)
    fi
    h3 "Current Device is:" "${bldylw}${BashmaticSerial__DefaultDevice}"
}

```

2.58. Module set

2.58.1. set-e-restore

```

set-e-restore ()
{
    [[ -f ${__bash_set_errexit_status} ]] && {
        error "You must first save it with the function:s ${bldgrn}set-e-save"
        return 1
    }
    local status=$(cat "${__bash_set_errexit_status}" | tr -d '\n')
    if [[ ${status} != 'on' && ${status} != 'off' ]]; then
        error "Invalid data in the set -e tempfile:" "$(cat \
${__bash_set_errexit_status})"
        return 1
    fi
    set -o errexit "${status}"
    rm -f "${__bash_set_errexit_status}" 2> /dev/null
}

```

2.58.2. set-e-save

```

set-e-save ()
{
    export __bash_set_errexit_status=$(mktemp -t 'errexit')
    rm -f "${__bash_set_errexit_status}" 2> /dev/null
    set-e-status > "${__bash_set_errexit_status}"
}

```

2.58.3. set-e-status

```

set-e-status ()
{
    set -o | grep errexit | awk '{print $2}'
}

```

2.59. Module shasum

2.59.1. sha

```
sha ()  
{  
    if output.has-stdin; then  
        shasum -a 256 "$@" | cut -d ' ' -f 1  
    else  
        shasum -a 256 "$@" | cut -d ' ' -f 1  
    fi  
}
```

2.59.2. shasum.all-files

```
shasum.all-files ()  
{  
    shasum.sha "$@" | awk '{print $2 " " $1}' | sort | .sha-only  
}
```

2.59.3. shasum.all-files-in-dir

```
shasum.all-files-in-dir ()  
{  
    local dir="$1"  
    shift  
    local name_pattern="$1"  
    shift  
    [[ -n ${name_pattern} ]] && name_pattern="-name ${name_pattern}"  
    shasum.sha $(eval "find ${dir} -type f ${name_pattern}") | awk '{print $2 " "  
$1}' | sort | .sha-only  
}
```

2.59.4. shasum.set-algo

```
shasum.set-algo ()  
{  
    local algo="${1:-1}"  
    export __bashmatic__sha_command="${__default_bashmatic__sha_command} -a ${algo}"  
}
```

2.59.5. shasum.set-command

```
shasum.set-command ()  
{  
    export __bashmatic__sha_command="$*"  
}
```

2.59.6. shasum.sha

```
shasum.sha ()  
{  
    .sha "$@" 2> /dev/null  
}
```

2.59.7. shasum.sha-only

```
shasum.sha-only ()  
{  
    .sha-only "$@" 2> /dev/null  
}
```

2.59.8. shasum.sha-only-stdin

```
shasum.sha-only-stdin ()  
{  
    echo "$*" | eval "${__bashmatic__sha_command}" | cut -d' ' -f 1  
}
```

2.59.9. shasum.to-hash

```
shasum.to-hash ()  
{  
    local hash_name="$1"  
    shift  
    local index=0  
    local last_sha  
    for file in $(shasum.sha "$@")  
    do  
        index=$((index + 1))  
        if [[ $((index % 2)) == 0 ]]; then  
            is-dbg 8&& info "${last_sha} <- ${bldpur}${file}"  
            eval "${hash_name}[${file}]=${last_sha}"  
        else  
            last_sha="${file}"  
        fi  
    done
```

```

is-dbg && {
    local count=$((index / 2))
    eval "local size=\${#\${hash_name}[@]}"
    h.yellow "Total ${count} files, ${size} hash entries in a Hash Variable '"
    ${hash_name}'"
}
return
}

```

2.60. Module shdoc

2.60.1. gawk.install

```

gawk.install () {
    local gawk_path="$(command -v gawk 2> /dev/null)"
    [[ -n "${gawk_path}" && -x "${gawk_path}" ]] || brew.install.package gawk
}

```

2.61. Module shell-set

2.61.1. save-restore-x

```

save-restore-x () {
    shell-set.pop-stack x
}

```

2.61.2. save-set-x

```

save-set-x () {
    shell-set.push-stack x
}

```

2.61.3. shell-set.init-stack

```

shell-set.init-stack () {
    unset SetOptsStack
    declare -a SetOptsStack=()
    export SetOptsStack
}

```

2.61.4. shell-set.is-set

```
shell-set.is-set ()  
{  
    local v="$1"  
    local is_set=${-//[^${v}]/}  
    if [[ -n ${is_set} ]]; then  
        return 0  
    else  
        return 1  
    fi  
}
```

2.61.5. shell-set.pop-stack

```
shell-set.pop-stack ()  
{  
    local value="$1"  
    local len=${#SetOptsStack[@]}  
    local last_index=$((len - 1))  
    local last=${SetOptsStack[$last_index]}  
    if [[ ${last} != "-$value" && ${last} != "+$value" ]]; then  
        error "Can not restore ${value}, not the last element in ${SetOptsStack[*]}  
stack."  
        return 1  
    fi  
    local pop=(${last})  
    export SetOptsStack=("${SetOptsStack[@]}/${pop}/")  
    [[ -n ${BASHMATIC_DEBUG} ]] && shell-set-show  
    eval "set ${last}"  
}
```

2.61.6. shell-set.push-stack

```
shell-set.push-stack ()  
{  
    local value="$1"  
    local is_set=${-//[^${value}]/}  
    shell-set.is-set "${value}" && export SetOptsStack=(${SetOptsStack[@]} "-${value}")  
    shell-set.is-set "${value}" || export SetOptsStack=(${SetOptsStack[@]} "+${value}")  
    [[ -n ${BASHMATIC_DEBUG} ]] && shell-set-show  
}
```

2.61.7. shell-set.show-stack

```
shell-set.show-stack ()  
{  
    info "Current Shell Set Stack: ${bldylw}[${SetOptsStack[*]}]"  
}
```

2.62. Module ssh

2.62.1. ssh.key.copy

```
ssh.key.copy ()  
{  
    cat ~/.ssh/id_rsa.pub | pbcopy  
}
```

2.62.2. ssh.key.filenames

```
ssh.key.filenames ()  
{  
    local name="$1"  
    is.not-blank "${name}" && name="_${name}"  
    export __bm__private_key_path="${HOME}/.ssh/id_rsa${name}"  
    export __bm__public_key_path="${HOME}/.ssh/id_rsa${name}.pub"  
    export __bm__ssh_folder="${HOME}/.ssh"  
}
```

2.62.3. ssh.keys.generate

```
ssh.keys.generate ()  
{  
    local name="$1"  
    local code=0  
    local email  
    local date="$(time.now.db)"  
    ssh.key.filenames "$@"  
    if is.a-non-empty-file "${__bm__private_key_path}"; then  
        warning "Private key already exists at the path: ${bldred}  
        }${__bm__private_key_path}"  
        is.blank "${name}" && info "NOTE You can pass an optional name argument to  
        this function"  
        info "so that the key file will be unique."  
        ( run.ui.ask "Replace the existing key (previous key will be backed up)??" )  
        code=$?  
        ((code)) && return 1  
    fi  
}
```

```

fi
is.a-function user.gitconfig.email && email="${user.gitconfig.email}"
is.blank "${email}" && run.ui.ask-user-value email "Please enter the email address
for this key:"
if [[ -f "${__bm__private_key_path}" ]]; then
    for file in "${__bm__private_key_path}" "${__bm__public_key_path}"
    do
        [[ -f ${file} ]] && run "mv ${file} ${file}.backup.${date}"
    done
fi
run.set-next show-output-on
run "ssh-keygen -t rsa -b 4096 -C ${email}"
}

```

2.62.4. ssh.load-keys

```

ssh.load-keys ()
{
    local pattern="$1"
    ssh.key.filenames "$@"
    local dir="${__bm__ssh_folder}"
    local regex="id_*${pattern}*" 
    info "Loading keys from ${bldylw}${dir}${txt-info}, matching regex: ${bldred}[${regex}]"
    find "${dir}" -type f -name "${regex}" -and -not -name '*.pub' -exec ssh-add {} \
}

```

2.63. Module subshell

2.63.1. bashmatic.detect-subshell

```

bashmatic.detect-subshell ()
{
    bashmatic.subshell-init
    [[ -n ${BASH_SUBSHELL_DETECTED} && -n ${BASH_IN_SUBSHELL} ]] && return "${BASH_IN_SUBSHELL}"
    unset BASH_IN_SUBSHELL
    export BASH_SUBSHELL_DETECTED=true
    local len="#${BASH_SOURCE[@]}"
    local last_index=$((len - 1))
    [[ -n ${BASHMATIC_DEBUG} ]] && {
        echo "BASH_SOURCE[*] = ${BASH_SOURCE[*]}" 1>&2
        echo "BASH_SOURCE[$last_index] = ${BASH_SOURCE[$last_index]}" 1>&2
        echo "\$0      = $0" 1>&2
    }
    if [[ -n ${ZSH_EVAL_CONTEXT} && ${ZSH_EVAL_CONTEXT} =~ :file$ ]] || [[ -n
${BASH_VERSION} && "$0" != "${BASH_SOURCE[$last_index]}" ]]; then

```

```

        export BASH_IN_SUBSHELL=0
    else
        export BASH_IN_SUBSHELL=1
    fi
    return ${BASH_IN_SUBSHELL}
}

```

2.63.2. `bashmatic.run-if-subshell`

```

bashmatic.run-if-subshell ()
{
    local current_shell=$(ps -p $$ -o comm | grep -v COMM | sed 's/-//g')
    set +e
    if [[ ${current_shell} == "bash" ]]; then
        local len="${#BASH_SOURCE[@]}"
        local last_index=$((len - 1))
        local last_script="${BASH_SOURCE[$last_index]}"
        [[ ${last_index} -lt 0 ]] && last_index=0
        is-dbg && dbg "Detected BASH, last script name is [${last_script}], sourcing
in [$0], is it sourced in? "
        if [[ -n ${BASH_VERSION} && "$0" != "${BASH_SOURCE[$last_index]}" ]]; then
            is-dbg && dbg "YES"
            return 0
        else
            is-dbg && dbg "NO"
            eval "$*"
        fi
    else
        if [[ ${current_shell} == "zsh" ]]; then
            is-dbg && dbg "Detected ZSH, ZSH_EVAL_CONTEXT = [${ZSH_EVAL_CONTEXT}], is
it sourced in? "
            if [[ -n ${ZSH_EVAL_CONTEXT} && ${ZSH_EVAL_CONTEXT} =~ :shfunc$ ]]; then
                is-dbg && dbg "YES"
                return 0
            else
                is-dbg && dbg "NO"
                eval "$*"
            fi
        else
            error "SHELL ${current_shell} is not supported."
            return 1
        fi
    fi
}

```

2.63.3. `bashmatic.subshell-init`

```

bashmatic.subshell-init ()

```

```
{  
    export BASH_SUBSHELL_DETECTED=  
}
```

2.63.4. `bashmatic.validate-sourced-in`

```
bashmatic.validate-sourced-in ()  
{  
    bashmatic.detect-subshell  
    [[ ${BASH_IN_SUBSHELL} -eq 0 ]] || {  
        echo "This script is meant to be sourced in, not run in a subshell." 1>&2  
        return 1  
    }  
    return 0  
}
```

2.63.5. `bashmatic.validate-subshell`

```
bashmatic.validate-subshell ()  
{  
    bashmatic.detect-subshell  
    [[ ${BASH_IN_SUBSHELL} -eq 1 ]] || {  
        echo "This script is meant to be run, not sourced-in" 1>&2  
        return 1  
    }  
    return 0  
}
```

2.64. Module `sym`

2.64.1. `decrypt.secrets`

```
decrypt.secrets ()  
{  
    ./bin/decrypt  
    local code=$?  
    [[ ${code} != 0 ]] && {  
        error "bin/decrypt returned non-zero exit status ${code}"  
        echo  
        exit ${code}  
    }  
}
```

2.64.2. dev.crypt.chef

```
dev.crypt.chef ()  
{  
    sym -ck APP_CHEF_SYM_KEY "$*"  
}
```

2.64.3. dev.decrypt.file

```
dev.decrypt.file ()  
{  
    [[ -f ${1} ]] || {  
        error 'usage: dev.decrypt.file <filename.enc>'  
        return  
    }  
    sym -ck APP_SYM_KEY -n "${1}"  
}
```

2.64.4. dev.decrypt.str

```
dev.decrypt.str ()  
{  
    [[ -z ${1} ]] && {  
        error 'usage: dev.decrypt.str "string to decrypt"'  
        return  
    }  
    sym -ck APP_SYM_KEY -d -s "$*"  
}
```

2.64.5. dev.edit.file

```
dev.edit.file ()  
{  
    [[ -f ${1} ]] || {  
        error 'usage: dev.edit.file <filename>'  
        return  
    }  
    sym -ck APP_SYM_KEY -t "${1}"  
}
```

2.64.6. dev.encrypt.file

```
dev.encrypt.file ()  
{
```

```
[[ -f ${1} ]] || {
    error 'usage: dev.encrypt.file <filename>'
    return
}
sym -ck APP_SYM_KEY -e -f "${1}" -o "${1}.enc"
}
```

2.64.7. `dev.encrypt.str`

```
dev.encrypt.str ()
{
    [[ -z "${1}" ]] && {
        error 'usage: dev.encrypt.str "string to encrypt"'
        return
    }
    sym -ck APP_SYM_KEY -e -s "$*"
}
```

2.64.8. `dev.sym`

```
dev.sym ()
{
    sym -cqk APP_SYM_KEY "$*"
}
```

2.64.9. `sym.dev.configure`

```
sym.dev.configure ()
{
    export SYMIT__KEY="APP_SYM_KEY"
}
```

2.64.10. `sym.dev.files`

```
sym.dev.files ()
{
    find . -name '*.enc' -type f
}
```

2.64.11. `sym.dev.have-key`

```
sym.dev.have-key ()
{
```

```

sym.dev.configure
if [[ -z ${CI} ]]; then
    [[ -z "$(keychain ${SYMIT_KEY} find 2> /dev/null)" ]] || printf "yes"
else
    [[ -n "${APP_SYM_KEY}" ]] && print "yes"
fi
}

```

2.64.12. sym.dev.import

```

sym.dev.import ()
{
    local skip_instructions=${1:-0}
    if [[ ${BASHMATIC_OS} != 'darwin' ]]; then
        error 'This is only meant to run on Mac OS-X'
        return
    fi
    sym.dev.configure
    sym.install.symit
    [[ -f ~/.sym.symit.bash ]] && source ~/.sym.symit.bash
    h2 'Encryption Key Import'
    info "Checking for the existence of the current key..."
    if [[ -n "$(sym.dev.have-key)" ]]; then
        info: "Key ${SYMIT_KEY} is already in your OS-X Key Chain."
        run.ui.ask "Would you like to re-import it?"
        [[ $? != 0 ]] && return
    fi
    if [[ ${skip_instructions} == ${false} ]]; then
        hr
        echo
        info "1. Please open 1Password App and search for 'Encryption Key'"
        echo
        info "2. Once you find the entry, it will contain two items: encryption key"
        info "      and password. Start by copying the key to the clipboard."
        echo
        info "3. You will need to paste the key first, and then copy/paste"
        info "      the key password (also in 1Password)"
        echo
        info "4. As a final setup, you will be asked to create a new password."
        info "      It must be at least 7 characters long, and will be used to
encrypt"
        info "      the key locally on your machine."
        echo
        echo
        run.ui.ask "Ready?"
        [[ $? != 0 ]] && return
    fi
    echo
    hr
}

```

```

sym -iqpx APP_SYM_KEY
code=$?
[[ ${code} != 0 ]] && {
    error "Sym exited with error code ${code}"
    return ${code}
}
hr
echo
info "Key import was successful, great job! ${bldylw} "
info "You can test that it works by encrypting, and decrypting a string,"
echo
info "\$ ${bldylw}source bin/bash"
info "\$ ${bldylw}dev.encrypt.str hello"
info "\$ ${bldylw}dev.decrypt.str \$(dev.encrypt.str hello )"
echo
info "Or a file:"
info "\$ ${bldylw}dev.decrypt.file config/application.dev.yml.enc"
echo
info "You can edit the file as if it wasn't encrypted:"
info "\$ ${bldylw}dev.edit.file config/application.dev.yml.enc"
echo
}

```

2.64.13. sym.dev.install-shell-helpers

```

sym.dev.install-shell-helpers ()
{
    local found=
    declare -a init_files=(${util.shell-init-files})
    for file in "${init_files[@]}"
    do
        f="${file}"
        [[ ! -f "${f}" ]] && continue
        [[ -n $(grep sym.symit "${f}") ]] && {
            found=${f}
            break
        }
    done
    if [[ -z ${found} ]]; then
        for file in "${init_files[@]}"
        do
            f="${file}"
            if [[ -f "${f}" ]]; then
                run "sym -B ${f} 1>/dev/null"
                return $?
            fi
        done
    else
        run "sym -B ${found} 1>/dev/null"
    fi
}

```

```
    fi
}
```

2.64.14. `sym.install.symit`

```
sym.install.symit ()
{
    if [[ ! -f config.ru ]]; then
        error "Please run this command from the RAILS_ROOT folder"
        return 1
    fi
    [[ -n "$(which sym 2> /dev/null)" && -f ~/.sym.symit.bash ]] && return
    local symit_source="/tmp/sym.symit.bash.$$$"
    trap "rm -f ${symit_source}; " EXIT
    local symit_url
    ="https://raw.githubusercontent.com/kigster/sym/master/bin/sym.symit.bash"
    local cmd="curl -fsSL ${symit_url} -o ${symit_source}"
    export LibRun__AbortOnError=${True}
    run "${cmd}"
    if [[ ! -f ${symit_source} ]]; then
        err "unable to find downloaded file ${symit_source}"
        return 1
    fi
    source ${symit_source}
    rm -f ${symit_source}
    run "symit install"
    sym.dev.install-shell-helpers
}
```

2.65. Module `text`

2.65.1. `text.chr`

```
text.chr ()
{
    [ "$1" -lt 256 ] || return 1
    printf "\$(printf '%03o' \"$1\")"
}
```

2.65.2. `text.markdown-to-asciidoc`

```
text.markdown-to-asciidoc ()
{
    local file="$1"
    shift
    local default_flags="--imagesdir=/assets/images --no-html-to-native"
```

```

[[ -n $(command -v kramdoc) ]] || gem.install "kramdown-asciidoc"
if [[ -z "${file}" ]]; then
    usage.set-min-flag-len 1
    usage-box "text.markdown-to-asciidoc markdown-file [ flags ] © Converts a
markdown doc to asciidoc using the kramdown-asciidoc ruby gem" " " "Default flags:
${bldcyn}${default_flags}" " " "To override pass any flags that are supported by
${bldred}kramdoc${bldylw}, see below:"
    printf "\n${txtblu}"
    kramdoc --help | tail -16
    printf "${clr}\n"
    return 0
fi
[[ -f ${file} && ${file}.extension "${file}" == "md" ]] || {
    error "File ${file} either does not exist, or is not markdown."
    run.set-all on-decline-return
    run.ui.ask "Convert anyway?"
}
local target="${file}.extension.replace adoc "${file}""
file.ask.if-exists "${target}" || {
    info "Aborting conversion, leaving ${target} in place."
    return 1
}
gem.install "kramdown-asciidoc"
local args
if [[ -z "$*" ]]; then
    args="--auto-ids --auto-id-prefix=_ --auto-id-separator=_"
--imagesdir=/assets/images --no-html-to-native"
else
    args="$*"
fi
run "kramdoc -o ${target} ${args} ${file}"
}

```

2.65.3. `text.ord`

```

text.ord ()
{
    LC_CTYPE=C printf '%d' "'$1"
}

```

2.66. Module `time`

2.66.1. `date.now.with-time`

```

date.now.with-time ()
{
    date '+%F' | tr -d '\n'
}

```

```
        time.now.with-ms  
    }
```

2.66.2. date.now.with-time.and.zone

```
date.now.with-time.and.zone ()  
{  
    ( date '+%F '  
    time.now.with-ms  
    date '+ %z' ) | tr -d '\n'  
}
```

2.66.3. epoch

```
epoch ()  
{  
    date +%s  
}
```

2.66.4. millis

```
millis ()  
{  
    .run.millis  
}
```

2.66.5. time.a-command

```
time.a-command ()  
{  
    local start="$(millis)"  
    eval "$*"  
    local end="$(millis)"  
    local ruby_expr="secs=(0.0 + ${end} - ${start}).to_f/1000.0; mins=secs/60; secs=(  
secs - secs/60 ) if mins > 0 ; printf('%d minutes %.3f seconds', mins, secs)"  
    local duration=$(ruby -e "${ruby_expr}")  
    echo -en "${duration}"  
}
```

2.66.6. time.date-from-epoch

```
time.date-from-epoch ()  
{
```

```

local epoch_ts
epoch_ts="$1"
printf "date --date='@$epoch_ts' "
}

```

2.66.7. time.duration.humanize

```

time.duration.humanize ()
{
    local seconds=${1}
    local hours=$((seconds / 3600))
    local remainder=$((seconds - hours * 3600))
    local mins=$((remainder / 60))
    local secs=$((seconds - hours * 3600 - mins * 60))
    local prefixed=0
    [[ ${hours} -gt 0 ]] && {
        printf "%02dh:" ${hours}
        prefixed=1
    }
    [[ ${mins} -gt 0 || ${prefixed} == 1 ]] && {
        printf "%02dm:" ${mins}
        prefixed=1
    }
    {
        printf "%02ds" ${secs}
    }
}

```

2.66.8. time.duration.millis-to-secs

```

time.duration.millis-to-secs ()
{
    local duration="$1"
    local format="${2:-"%d.%d\"}"
    local seconds=$((duration / 1000))
    local leftover=$((duration - 1000 * seconds))
    printf "${format}" ${seconds} ${leftover}
}

```

2.66.9. time.epoch-to-iso

```

time.epoch-to-iso ()
{
    local epoch_ts=$1
    eval "$(time.date-from-epoch "${epoch_ts}") -u \"+%Y-%m-%dT%H:%M:%S%z\" | sed
's/0000/00:00/g'

```

```
}
```

2.66.10. `time.epoch-to-local`

```
time.epoch-to-local ()  
{  
    local epoch_ts=$1  
    [[ -z ${epoch_ts} ]] && epoch_ts=$(epoch)  
    eval "$(time.date-from-epoch "${epoch_ts}") "+%m/%d/%Y, %r""  
}
```

2.66.11. `time.epoch.minutes-ago`

```
time.epoch.minutes-ago ()  
{  
    local mins=${1}  
    [[ -z ${mins} ]] && mins=1  
    local seconds=$((mins * 60))  
    local now_epoch=$(epoch)  
    echo $((now_epoch - seconds))  
}
```

2.66.12. `time.now.db`

```
time.now.db ()  
{  
    date '+%F.%T.%S' | tr -d '\:\-\.'  
}
```

2.66.13. `time.now.file-extension`

```
time.now.file-extension ()  
{  
    time.now.db  
}
```

2.66.14. `time.now.with-ms`

```
time.now.with-ms ()  
{  
    util.os  
    local date_runnable  
    date_runnable='date'
```

```

if [[ "${BASHMATIC_OS}" == "darwin" ]]; then
    [[ -z $(command -v gdate) ]] && .time.osx.coreutils
    [[ -n $(command -v gdate) ]] && date_runnable='gdate'
fi
${date_runnable} '+%T.%3N'
}

```

2.66.15. `time.with-duration`

```

time.with-duration ()
{
    local quiet=false
    local verbose=false
    local secret=false
    [[ "$1" == quiet ]] && {
        shift
        quiet=true
    }
    [[ "$1" == verbose ]] && {
        shift
        verbose=true
    }
    [[ "$1" == secret ]] && {
        shift
        secret=true
    }
    local -a command=("$@")
    local marker=$(util.random-string 10)
    time.with-duration.start "${marker}"
    local cmd="${command[*]}"
    ${quiet} && cmd+=">/dev/null 2>&1"
    if ${secret}; then
        ${verbose} && inf "⌚️ Running Command: ${txtblk}${bakblu}[REDACTED]"
    else
        ${verbose} && inf "⌚️ Running Command: ${txtblk}${bakblu}${cmd}"
    fi
    set +e
    local code=0
    /usr/bin/env bash -c "${cmd}"
    code=$?
    if ${verbose}; then
        if [[ ${code} -eq 0 ]]; then
            ok:
            hr
            info "⌚️ Total time taken: ${bldgrn}$(time.with-duration.end "${marker}")",
            command successful."
        else
            not-ok:
            hr
        fi
    fi
}

```

```

        info "Total time taken: ${bldred}$(time.with-duration.end "${marker}")",
exit code: ${bldred}${code}"
    fi
    fi
    return ${code}
}

```

2.66.16. `time.with-duration.clear`

```

time.with-duration.clear ()
{
    local name="$1"
    eval "unset __bashmatic_with_duration_ms${name}"
}

```

2.66.17. `time.with-duration.end`

```

time.with-duration.end ()
{
    local name="$1"
    shift
    if [[ -z ${name} ]]; then
        name="_default"
    else
        name="${name/_/}"
    fi
    local var="__bashmatic_with_duration_ms${name}"
    local started=$(.subst ${var})
    [[ -z ${started} ]] && started=${!var}
    [[ -z ${started} ]] && {
        error "No start time recorded for namespace ${name}."
        return 1
    }
    local finished="$(millis)"
    local duration=$((finished - started))
    duration="$(time.duration.millis-to-secs "${duration}")"
    printf -- "$* %s\n" "${duration} sec"
}

```

2.66.18. `time.with-duration.start`

```

time.with-duration.start ()
{
    local name="$1"
    if [[ -z ${name} ]]; then
        name="_default"

```

```

    else
        name="${name/_/_}"
    fi
    eval "export __bashmatic_with_duration_ms${name}=\$(millis)"
}

```

2.66.19. today

```

today ()
{
    date +'%Y-%m-%d'
}

```

2.67. Module trap

2.67.1. trap-setup

```

trap-setup ()
{
    .trap-remove
    local signal="${1:-"SIGINT"}"
    trap '.trap-catch' "${signal}"
    export __int_signal__="${signal}"
}

```

2.67.2. trap-was-fired

```

trap-was-fired ()
{
    if [[ -n "${__int_marker__}" && -f "${__int_marker__}" ]]; then
        rm -f "${__int_marker__}"
        return 0
    fi
    return 1
}

```

2.67.3. trapped

```

trapped ()
{
    if [[ ${__int_flag__} -eq 1 ]]; then
        unset __int_flag__
        return 0
    fi
}

```

```
    return 1
}
```

2.68. Module url

2.68.1. url.cert.domain

```
url.cert.domain ()
{
    url.cert.info "$@" | grep 'subject: CN=' | cut -d '=' -f 2
}
```

2.68.2. url.cert.info

```
url.cert.info ()
{
    local url="$1"
    local quiet="${2:-false}"
    [[ ${url} =~ https:// ]] || url="https://${url}"
    url.host.is-valid "${url}" || return 1
    curl --insecure -vvI "${url}" 2>&1 | awk 'BEGIN { cert=0 }
        /^\* SSL connection/ { cert=1 } /^\*/ { if (cert) print }'
}
```

2.68.3. url.cert.is-valid

```
url.cert.is-valid ()
{
    local url="$1"
    shift
    [[ ${url} =~ https:// ]] || url="https://${url}"
    curl -L -q -I "${url}" "$@" > /dev/null 2>&1
}
```

2.68.4. url.downloader

```
url.downloader ()
{
    local downloader=
    if [[ -z "${LibUrl__Downloader}" ]]; then
        [[ -z "${downloader}" && -n $(which curl) ]] && downloader=$(which curl)
        ${LibUrl__CurlDownloaderFlags}
        [[ -z "${downloader}" && -n $(which wget) ]] && downloader=$(which wget)
        ${LibUrl__WgetDownloaderFlags}
    fi
}
```

```

[[ -z "${downloader}" ]] && {
    error "Neither Curl nor WGet appear in the \$PATH... HALP?"
    return 1
}
export LibUrl__Downloader="${downloader}"
fi
printf "${LibUrl__Downloader}"
}

```

2.68.5. url.host.is-valid

```

url.host.is-valid ()
{
    local host="$1"
    local host="${host/https:\//}"
    host "${host}" > /dev/null || {
        error "${host} does not appear to be a valid host." 1>&2
        return 255
    }
}

```

2.68.6. url.http-code

```

url.http-code ()
{
    local url="$1"
    local quiet="${2:-false}"
    [[ -z $(which wget) ]] && {
        echo 1>&2
        err "This function currently only supports ${bldylw}wget.\n" 1>&2
        echo 1>&2
        return 100
    }
    url.is-valid "$url" || {
        echo 1>&2
        err "The URL provided is not a valid URL: ${bldylw}${url}\n" 1>&2
        echo 1>&2
        return 101
    }
    local result=$(wget -v --spider "${url}" 2>&1 | ${GrepCommand} "response" | awk
'{print $6}' | tr -d '\n' | tail -1)
    export LibUrl__LastHttpCode="${result}"
    if [[ ${quiet} == true ]]; then
        if [[ ${result} -gt 199 && ${result} -lt 210 ]]; then
            return 0
        else
            return 1
    fi
}

```

```

    else
        [[ -n "${result}" ]] && printf "${result}" || printf "404"
    fi
}

```

2.68.7. url.is-valid

```

url.is-valid ()
{
    local url="$1"
    if [[ $(url.valid-status "$url") = "ok" ]]; then
        return 0
    else
        return 1
    fi
}

```

2.68.8. url.shorten

```

url.shorten ()
{
    local longUrl="$1"
    error "This function used Bitly API V3 which is now defunct."
    return 1
    if [[ -z "${BITLY_LOGIN}" || -z "${BITLY_API_KEY}" ]]; then
        printf "${longUrl}"
    else
        export BITLY_LOGIN=$(printf '%s' "${BITLY_LOGIN}" | tr -d '\r' | tr -d '\n')
        export BITLY_API_KEY=$(printf '%s' "${BITLY_API_KEY}" | tr -d '\r' | tr -d '\n')
        if [[ -n $(which ruby) ]]; then
            longUrl=$(ruby -e "require 'uri'; str = '${longUrl}'".force_encoding('ASCII-8BIT'); puts URI.encode(str))
        fi
        bitlyUrl="http://api.bit.ly/v3/shorten?login=${BITLY_LOGIN}&apiKey=${BITLY_API_KEY}&format=txt&longURL=${longUrl}"
        debug "BITLY_LOGIN : ${clr}${bldylw}${BITLY_LOGIN}" 1>&2
        debug "BITLY_LOGIN : ${clr}${bldgrn}${BITLY_API_KEY}" 1>&2
        debug "BITLY_API_URL: ${clr}${undblu}${bitlyUrl}${clr}" 1>&2
        local output=$(url.downloader "${bitlyUrl}" 2>&1)
        if [[ "${output}" =~ "INVALID" || "${output}" =~ "Server Error" ]]; then
            error "${output}"
            return 1
        else
            printf "%s" "${output}" | tr -d '\n' | tr -d ' '
            return 0
        fi
    fi
}

```

```
}
```

2.68.9. url.valid-status

```
url.valid-status ()  
{  
    local url="$1"  
    echo "${url}" | ruby -ne '  
        require "uri"  
        u = URI.parse("#{$_}.chomp")  
        if u && u.host && u.host&.include?(".") && u&.scheme =~ /^http/  
            print "ok"  
        else  
            print "invalid"  
        end'  
}
```

2.69. Module usage

2.69.1. help-command

```
help-command ()  
{  
    printf "${bldylw}\$ ${*}\n"  
}
```

2.69.2. help-comment

```
help-comment ()  
{  
    printf "${txtblk}# ${*}\n"  
}
```

2.69.3. help-details

```
help-details ()  
{  
    printf "${txtblu}${*}\n"  
}
```

2.69.4. help-example

```
help-example ()  
{  
    printf "    ${bldgrn}\$ $*\n"  
}
```

2.69.5. help-name

```
help-name ()  
{  
    box.white-on-green "$@"  
}
```

2.69.6. help-section

```
help-section ()  
{  
    printf "\n${bldgrn}$(echo "$*" | tr '[[:lower:]]' '[[:upper:]]')\n"  
}
```

2.69.7. usage-box

```
usage-box ()  
{  
    local backup=".usage-cache-file"  
    if [[ "${EXPIRE_USAGE_CACHE}" -eq 0 && -s "${backup}" ]]; then  
        cat "${backup}"  
    else  
        .usage.box "$@" | tee "${backup}"  
    fi  
}
```

2.69.8. usage-box.section

```
usage-box.section ()  
{  
    printf "${__color_headers}"  
    .usage.hdr "$*"  
}
```

2.69.9. usage-box.sub-section

```
usage-box.sub-section ()
```

```
{
    .output.box-separator "${__color_bdr}"
    .outputboxed-text "${__color_bdr}" "${__color_sub_headers}" "${(.usage.hdr "$1")}"
}
```

2.69.10. usage-widget

```
usage-widget () {
    local width="$1"
    local cache_wipe=0
    [[ ${width} =~ ^- ]] && {
        cache_wipe=1
        width=${width:1}
    }
    is.numeric "${width}" && {
        shift
        bashmatic.set-widget-width-to "${width}"
    }
    ((cache_wipe)) && rm -f "${(.usage-cache-file)}"
    local -a args=("$@")
    is-debug && {
        h1 "Got total of ${#args[@]} arguments."
    }
    local -a details
    local left_space
    left_space=$(cursor.right 1)
    if [[ ${#args[@]} -gt 2 ]]; then
        for i in $(seq 2 50)
        do
            [[ -z ${args[$i]} ]] && break
            if [[ ${args[$i]} =~ ^@ ]]; then
                details+=("${cursor.left 4}${(.usage-box.section "${args[$i]}/@/")}")
                details+=(" ")
            else
                details+=("${left_space}")
                details+=("${cursor.left 6}${args[$i]}")
            fi
        done
    fi
    usage-box "${args[0]} @ ${args[1]}" "${(cursor.up 1; cursor.left 5)} "
    ${cursor.right 10} "${details[@]}"
}
```

2.69.11. usage.set-min-flag-len

```
usage.set-min-flag-len () {

```

```
    export LibUsage__MinFlagLen="${1}"
}
```

2.70. Module user

2.70.1. user

```
user () {
    local user
    user=$(user.finger.name)
    [[ -z "${user}" ]] && user=$(user.gitconfig.name)
    [[ -z "${user}" ]] && user=$(user.gitconfig.email)
    [[ -z "${user}" ]] && user=$(user.username)
    echo "${user}"
}
```

2.70.2. user.current-shell

```
user.current-shell () {
    /bin/ps -p $$ -o comm | tail -1 | sed -E 's/-//g'
```

2.70.3. user.current-shell-init-file

```
user.current-shell-init-file () {
    declare -a shell_files=(${util.shell-init-files "$(user.current-shell)"})
    .user.pick-shell-init-file "${shell_files[@]}"
```

2.70.4. user.finger.name

```
user.finger.name () {
    [[ -n $(which finger) ]] && finger "${USER}" | head -1 | sedx 's/.*Name: //g'
```

2.70.5. user.first

```
user.first ()
```

```
{  
    user | tr '\n' ' ' | ruby -ne 'puts $_.split(/ /).first.capitalize'  
}
```

2.70.6. `user.gitconfig.email`

```
user.gitconfig.email ()  
{  
    if [[ -s ${HOME}/.gitconfig ]]; then  
        grep email "${HOME}/.gitconfig" | sedx 's/.*=\\s?//g'  
    fi  
}
```

2.70.7. `user.gitconfig.name`

```
user.gitconfig.name ()  
{  
    if [[ -s ${HOME}/.gitconfig ]]; then  
        grep name "${HOME}/.gitconfig" | sedx 's/.*=\\s?//g'  
    fi  
}
```

2.70.8. `user.host`

```
user.host ()  
{  
    local host=  
    host=$(user.my.reverse-ip)  
    [[ -z ${host} ]] && host=$(user.my.ip)  
    printf "${host}"  
}
```

2.70.9. `user.login-shell`

```
user.login-shell ()  
{  
    basename "$(user.login-shell-path)"  
}
```

2.70.10. `user.login-shell-init-file`

```
user.login-shell-init-file ()  
{
```

```
declare -a shell_files=(${util.shell-init-files})
.user.pick-shell-init-file "${shell_files[@]}"
}
```

2.70.11. `user.login-shell-path`

```
user.login-shell-path ()
{
    if [[ -n $(command -v finger 2> /dev/null) ]]; then
        finger "${USER}" | grep Shell: | sed 's/^.*Shell: //g'
    else
        if grep -q "${USER}" /etc/passwd 2> /dev/null; then
            grep "${USER}" /etc/passwd | sed 's/.*://g'
        else
            command -v "$(user.current-shell)"
        fi
    fi
}
```

2.70.12. `user.my.external-ip`

```
user.my.external-ip ()
{
    ( curl -s http://checkip.dyndns.org/ | sed 's/[a-zA-Z<>/ :]//g' | sed -E
's/^[\d\.]/g; s/\r/g;' 2>&1 )
}
```

2.70.13. `user.my.ip`

```
user.my.ip ()
{
    user.my.external-ip | tr -d '\n'
}
```

2.70.14. `user.my.local-ip`

```
user.my.local-ip ()
{
    ifconfig -l | xargs -n1 ipconfig getifaddr
}
```

2.70.15. `user.my.reverse-ips`

```

user.my.reverse-ips ()
{
    local ip=$(user.my.ip)
    local output=$(curl -s "https://api.hackertarget.com/reverseiplookup/?q=${ip}"
2>&1)
    if [[ ${output} =~ Membership ]]; then
        error "You have exceeded the number of free API calls to determine your
reverse IP." "Please visit https://hackertarget.com/scan-membership/ if you wish to "
"increase your limit." 1>&2
        return 1
    else
        echo
    fi
}

```

2.70.16. user.pairs.email

```

user.pairs.email ()
{
    [[ ! -s "${bashmatic_git_pairs}" || -z "$1" ]] && return
    local username=$(user.pairs.username "$1")
    local domain=$(grep domain "${bashmatic_git_pairs}" | sed 's/.*/domain:/g' | tr
-d ' ')
    [[ -n ${username} && -n "${domain}" ]] || {
        error "Couldn't determine username or domain from ${bashmatic_git_pairs} file
for input ${bldwht}${*}"
        return 1
    }
    echo "${username}@${domain}"
}

```

2.70.17. user.pairs.firstname

```

user.pairs.firstname ()
{
    [[ ! -s "${bashmatic_git_pairs}" || -z "$1" ]] && return
    grep -i "$1" "${bashmatic_git_pairs}" | head -1 | awk '{print $2}' | tr -d ';'
}

```

2.70.18. user.pairs.lastname

```

user.pairs.lastname ()
{
    [[ ! -s "${bashmatic_git_pairs}" || -z "$1" ]] && return
    grep -i "$1" "${bashmatic_git_pairs}" | head -1 | awk '{print $3}' | tr -d ';'
}

```

```
}
```

2.70.19. user.pairs.set-file

```
user.pairs.set-file ()  
{  
    [[ -s "$1" ]] || {  
        error "Please pass a valid path to the .pairs file, typically in your home.  
You passed: [$1]"  
        return 1  
    }  
    export bashmatic_git_pairs="$1"  
}
```

2.70.20. user.pairs.username

```
user.pairs.username ()  
{  
    [[ ! -s "${bashmatic_git_pairs}" || -z "$1" ]] && return  
    grep -i "$1" "${bashmatic_git_pairs}" | head -1 | awk '{print $4}' | tr -d ';'
```

2.70.21. user.username

```
user.username ()  
{  
    echo "${USER:-$(whoami)}"
```

2.71. Module util

2.71.1. is-func

```
is-func ()  
{  
    is.a-function "$@"
```

2.71.2. pause

```
pause ()  
{
```

```
    sleep "${1:-1}"
}
```

2.71.3. pause.long

```
pause.long ()
{
    sleep "${1:-10}"
}
```

2.71.4. pause.medium

```
pause.medium ()
{
    sleep "${1:-0.3}"
}
```

2.71.5. pause.short

```
pause.short ()
{
    sleep "${1:-0.1}"
}
```

2.71.6. system.save-os-name

```
system.save-os-name ()
{
    local _os=""
    _os=$(($system.uname) -s | tr '[:upper:]' '[:lower:]' | tr -d '\n')
    [[ -z "${BASHMATIC_OS_NAME}" ]] && export BASHMATIC_OS_NAME="${_os}" > /dev/null
2>&1
    export BASHMASTIC_OS="${BASHMATIC_OS_NAME}"
}
```

2.71.7. system.uname

```
system.uname ()
{
    command -v uname && return 0
    [[ -x ${__bashmatic_uname_binary} && -x ${__bashmatic_uname_binary} ]] && {
        echo ${__bashmatic_uname_binary}
        return 0
    }
}
```

```

}
local -a uname_options=("/bin/uname" "/usr/bin/uname" "/sbin/uname"
"/usr/sbin/uname")
local binary
for binary in ${uname_options[@]}
do
  [[ -x ${binary} ]] && {
    export __bashmatic_uname_binary=$(printf -- "%s" "${binary}")
    printf "%s" "$__bashmatic_uname_binary"
    return 0
  }
done
return 1
}

```

2.71.8. util.append-to-init-files

```

util.append-to-init-files ()
{
  local string="$1"
  local search="${2: -$1}"
  is_installed=
  declare -a shell_files=(${util.shell-init-files})
  for init_file in "${shell_files[@]}"
  do
    file="${init_file}"
    [[ -f ${file} ]] && grep -q "${search}" "${file}" && {
      is_installed=${file}
      break
    }
  done
  if [[ -z "${is_installed}" ]]; then
    for init_file in "${shell_files[@]}"
    do
      file="${init_file}"
      [[ -f ${file} ]] && {
        echo "${string}" >> "${file}"
        is_installed="${file}"
        break
      }
    done
  fi
  printf "${is_installed}"
}

```

2.71.9. util.arch

```

util.arch ()

```

```
{  
    echo -n "${BASHMATIC_OS}-$(($system.uname -m))-$(($system.uname -p))" |  
    /usr/bin/tr '[[:upper:]]' '[[:lower:]]'  
}
```

2.71.10. util.call-if-function

```
util.call-if-function ()  
{  
    local func="$1"  
    shift  
    util.is-a-function "${func}" && {  
        ${func} "$@"  
    }  
}
```

2.71.11. util.dev-setup.update

```
util.dev-setup.update ()  
{  
    run "rm -f ${BASHMATIC_HOME}/bin/.dev-setup"  
    run "dev-setup -N -h > /tmp/a"  
    run "mv /tmp/a ${BASHMATIC_HOME}/bin/.dev-setup"  
    run "cd ${BASHMATIC_HOME} && git add bin/.dev-setup"  
    run "cd -"  
}
```

2.71.12. util.eval-function-body

```
util.eval-function-body ()  
{  
    local name="$1"  
    shift  
    type "${name}" | grep -qi function || {  
        error "${name} is not a function."  
        return 1  
    }  
    eval "$(type "${name}" | sedx '1,3d;$d')" "$@"  
}
```

2.71.13. util.functions-matching

```
util.functions-matching ()  
{  
    local prefix="${1}"
```

```

local extra_command=${2:-"cat"}
declare -f | ${GrepCommand} "^\${prefix}" | sedx 's/[\\()]///g;' | /usr/bin/tr -d '
' | /usr/bin/tr '\n' ''
}

```

2.71.14. util.functions-matching.diff

```

util.functions-matching.diff ()
{
    while true; do
        [[ -z "$1" ]] && break
        for e in $(util.functions-matching "${1}")
        do
            printf "${e/$1/}\n"
        done
        shift
    done
    return 0
}

```

2.71.15. util.functions-starting-with

```

util.functions-starting-with ()
{
    local prefix="${1}"
    local extra_command=${2:-"cat"}
    set | ${GrepCommand} '^[^ ].* \\($' | ${GrepCommand} "^\${prefix}" | sedx
's/[\\()]///g;' | ${extra_command} | /usr/bin/tr '\n' ''
}

```

2.71.16. util.functions-starting-with-csv

```

util.functions-starting-with-csv ()
{
    local prefix="$1"
    util.functions-starting-with "${prefix}" | sedx "s/${prefix}/./\\.//g; s/\\s+\\w/,\\1/g;""
}

```

2.71.17. util.functions-starting-with-lines

```

util.functions-starting-with-lines ()
{
    local prefix="${1}"
    local extra_command=${2:-"cat"}

```

```
    set | ${GrepCommand} '^[^ ].* \\(\) \$' | ${GrepCommand} "^\${prefix}" | sedx
's/[\\(\)]//g;' | ${extra_command}
}
```

2.71.18. `util.generate-password`

```
util.generate-password ()
{
    local len=\$1:-32
    local val=\$((\$((date '+%s') - 100000 * RANDOM)))
    [[ \$val:0:1 ] == "-" ]] && val=\$val/-/
    printf "\$(echo \$val | shasum -a 512 | awk '{print \$1}' | base64 | head -c \"\${
len\"))"
}
```

2.71.19. `util.i-to-ver`

```
util.i-to-ver ()
{
    version=\$1
    /usr/bin/env ruby -e "ver='\$version'; printf %Q{\%d.\%d.\%d}, ver[1..2].to_i,
ver[3..5].to_i, ver[6..8].to_i"
}
```

2.71.20. `util.install-direnv`

```
util.install-direnv ()
{
    "\${BASHMATIC_HOME}/bin/install-direnv"
```

2.71.21. `util.invoke-if-function`

```
util.invoke-if-function ()
{
    local namespace=\$1
    shift
    local action=\$1
    shift
    local func="\${namespace}.\${action}"
    util.is-a-function "\${func}" || return 255
    \${func} "\$@"
}
```

2.71.22. `util.is-a-function`

```
util.is-a-function ()  
{  
    is.a-function "$@"  
}
```

2.71.23. `util.is-numeric`

```
util.is-numeric ()  
{  
    is.numeric "$1"  
}
```

2.71.24. `util.is-variable-defined`

```
util.is-variable-defined ()  
{  
    local var_name="$1"  
    [[ -n ${!var_name+x} ]]  
}
```

2.71.25. `util.lines-in-folder`

```
util.lines-in-folder ()  
{  
    local folder=${1%.*}  
    find "${folder}" -type f -exec wc -l {} \; | awk 'BEGIN{a=0}{a+=1}END{print a}'  
}
```

2.71.26. `util.os`

```
util.os ()  
{  
    export BASHMATIC_OS="${BASHMATIC_OS_NAME:=$(uname -s | tr '[:upper:]' '[:lower:]'  
    | tr -d '[:space:]')}"  
    export BASHMATIC_OS_NAME  
}
```

2.71.27. `util.random-number`

```
util.random-number ()
```

```
{
    local limit="${1:-"1000000}"
    printf $((($RANDOM % limit)))
}
```

2.71.28. util.random-string.of-length

```
util.random-string.of-length ()
{
    util.generate-password "$@"
}
```

2.71.29. util.remove-from-init-files

```
util.remove-from-init-files ()
{
    local search="${1}"
    local backup_extension="${2}"
    [[ -z ${backup_extension} ]] && backup_extension="${epoch}.backup"
    [[ -z ${search} ]] && return
    declare -a shell_files=(${util.shell-init-files})
    local temp_holder=$(mktemp)
    for init_file in "${shell_files[@]}"
    do
        run.config.detail-is-enabled && inf "verifying file ${init_file}..."
        file="${init_file}"
        if [[ -f ${file} ]] && grep -q "${search}" "${file}"; then
            run.config.detail-is-enabled && ui.closer.ok:
            local matches=$(grep -c "${search}" "${file}")
            run.config.detail-is-enabled && info "file ${init_file} matches with
${bldylw}${matches} matches"
            run "grep -v \\"${search}\\" ${file} > ${temp_holder}"
            if [[ -n "${backup_extension}" ]]; then
                local backup="${file}.${backup_extension}"
                run.config.detail-is-enabled && info "backup file will created in
${bldylw}${backup}"
                [[ -n "${do_backup_changes}" ]] && "mv ${file} ${backup}"
            fi
            run "cp -v ${temp_holder} ${file}"
        else
            run.config.detail-is-enabled && ui.closer.not-ok:
        fi
    done
    return "${LibRun__LastExitCode}"
}
```

2.71.30. `util.rot13`

```
util.rot13 ()
{
    local value="$*"
    echo "$value" | util.rot13-stdin
}
```

2.71.31. `util.rot13-stdin`

```
util.rot13-stdin ()
{
    ruby -e 'puts STDIN.read.
    tr(
        "0123456789",
        "5678901234").
    tr(
        "abcdefghijklmnopqrstuvwxyz",
        "nopqrstuvwxyzabcdefghijklmnopqrstuvwxyz").
    tr(
        "ABCDEFGHIJKLMNOPQRSTUVWXYZ",
        "NOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ")'
}
```

2.71.32. `util.shell-init-files`

```
util.shell-init-files ()
{
    local shell_function="${1:-"user.login-shell"}"
    local shell_name=$( ${shell_function} )
    if [[ ${shell_name} == "bash" ]]; then
        echo "${HOME}/.bash_profile ${HOME}/.bash_login ${HOME}/.bashrc ${HOME}
${HOME}/.profile"
    else
        if [[ ${shell_name} == "zsh" ]]; then
            echo "${HOME}/.zshrc"
        else
            error "Shell ${shell_name} is not supported."
            return 1
        fi
    fi
}
```

2.71.33. `util.ver-to-i`

```
util.ver-to-i ()
```

```
{  
    version=${1}  
    echo "$version" | awk 'BEGIN{FS="."){ printf "1%02d%03.3d%03.3d", $1, $2, $3}'  
}
```

2.71.34. var.expand

```
var.expand ()  
{  
    .subst "$@"  
}
```

2.71.35. watch.command

```
watch.command ()  
{  
    [[ -z "$1" ]] && return 1  
    trap "return 1" SIGINT  
    while true; do  
        clear  
        hr.colored "${txtblu}"  
        printf " █ Command: ${bldgrn}*$${clr} █ ${txtblu}$(date)${clr} █ Refresh:  
${bldcyn}${LibUtil__WatchRefreshSeconds}${clr}\n"  
        hr.colored "${txtblu}"  
        eval "$*"  
        hr.colored "${txtblu}"  
        printf "To change refresh rate run ${bldylw}watch.set-refresh <seconds>${clr}  
}\n\n"  
        sleep "${LibUtil__WatchRefreshSeconds}"  
    done  
}
```

2.71.36. watch.ls-al

```
watch.ls-al ()  
{  
    while true; do  
        ls -al  
        sleep "${LibUtil__WatchRefreshSeconds}"  
        clear  
    done  
}
```

2.71.37. `watch.set-refresh`

```
watch.set-refresh ()  
{  
    export LibUtil__WatchRefreshSeconds="${1:-"0.5"}"  
}
```

2.72. Module `video`

2.72.1. `video.encode`

```
video.encode ()  
{  
    set +e  
    local file="$1"  
    shift  
    local algo="${1:-"11"}"  
    shift  
    local output="${1}"  
    shift  
    [[ -z "${output}" ]] && output=$(video.filename.encoded "${file}" "${algo}")  
    output="${output%.*}\.mkv"  
    [[ "${file}" == "${output}" ]] && output="${output%.mkv/-converted.mkv}"  
    is.an-existing-file "${output}" && {  
        info "File ${output} already exists, making a backup..."  
        local t="$(time.now.db | tr -d ' ')"  
        local backup_output=$(echo "${output}" | sed "s/\.\(\.*\)\$/-\${t}\.\1/g")  
        info "Backing up a name clashing file to ${backup_output}..."  
        mv -v "${output}" "${backup_output}"  
    }  
    h2 "Starting \\"$ffmpeg_binary\\" conversion, source file size is ${bldred}  
    ${file.size.mb} \"${file}\"" " Source:      [${file}]" " Destination: [${output}]" "  
    Algorithm:   [${algo}]"  
    video.install.dependencies  
    local func=".video.convert.compress-${algo}"  
    is.a-function "${func}" || {  
        error "${func} is not a valid function name."  
        return 1  
    }  
    arrow.blk-on-ylw "Conversion Function: ${func}"  
    arrow.blk-on-blu "Source File:          \"${file}\""  
    arrow.blk-on-grn "Destination File:    \"${output}\""  
    local token=$(echo "${file}" | shasum.sha | cut -f 1 -d ' ')  
    time.with-duration.start "${token}"  
    info "Please wait while we compress this file... (set DEBUG=1 to see the output)"  
    echo  
    run.set-all show-output-on abort-on-error  
    ${func} "${file}" "${output}"
```

```

run.set-all show-output-off
local before=$(file.size "${file}")
local after=$(file.size "${output}")
local reduction=
local duration=$(time.with-duration.end "${token}")
if [[ ${before} -lt ${after} ]]; then
    reduction=$((100 * (after - before) / before))
    warning "${output} was generated with ${reduction}%% increase in file size"
"from ${before} to ${after}" "and took ${duration}"
else
    reduction=$((100 * (before - after) / before))
    success "${output} was generated with ${reduction}%% reduction in file size"
"from ${before} to ${after}" "and took ${duration}"
fi
return 0
}

```

2.72.2. `video.filename.encoded`

```

video.filename.encoded ()
{
    local source="$1"
    local algo="${2:--compressed}"
    local dest
    dest=$(echo "${source}" | tr ' ' '-' | tr '[[:upper:]]' '[[:lower:]]' | sed -E
's/\.(.*$/-"${algo}"\.\1/g' | tr -d '-')
    printf -- "%s" "${dest}"
}

```

2.72.3. `video.install.dependencies`

```

video.install.dependencies ()
{
    brew.cache-reset
    run.set-all verbose-on show-output-on
    set +e
    for package in "${required_packages[@]}"
    do
        brew.package.is-installed "${package}" > /dev/null || brew.install.package "${package}"
    done
    command -v ffmpeg > /dev/null || brew.install.package ffmpeg
    command -v npm > /dev/null || brew.install.package node
    command -v ffmpeg-bar > /dev/null || run "npm install -g ffmpeg-progressbar-cli"
    if ( command -v ffmpeg-bar > /dev/null ); then
        export ffmpeg_binary="ffmpeg-bar"
    else
        export ffmpeg_binary="ffmpeg-y -loglevel error -stats "
    fi
}

```

```

    fi
    printf "%s" "${ffmpeg_binary}"
    return 0
}

```

2.72.4. video.make.mp4

```

video.make.mp4 ()
{
    local file="$1"
    local target="${2}"
    [[ -z "${target}" ]] && target="CC-${file/.MOV/.mp4}"
    [[ -s "${file}" ]] || {
        error "File [${file}] does not exist."
        return 1
    }
    [[ "${file}" == "${target}" ]] && target="COMPRESSED-${target}"
    h1bg "${file} -> ${target}"
    run.set-next show-output-on continue-on-error
    run "ffmpeg -n -loglevel error -stats -i \"${file}\" -c:v libx265 -b:v 4M -x265
-params pass=1 -f null /dev/null"
    run "ffmpeg -n -loglevel error -stats -i \"${file}\" -c:v libx265 -b:v 4M -x265
-params pass=2 \"${target}\""
}

```

2.72.5. video.shrink

```

video.shrink ()
{
    [[ -z "$*" ]] && {
        printf -- "${bldgrn}USAGE:\n      ${bldylw}[ DEBUG=1 ] video-shrink *.mp4${clr
}\n"
        return 0
    }
    for file in "$@"
    do
        [[ -s "${file}" ]] || {
            warning "File ${file} does not exist..."
            continue
        }
        dest="$(.destination-file-name "${file}")"
        h1 "Compressing \"${file}\" To \"${dest}\""
        video.encode "${file}" "shrinkwrap" "${dest}"
        done
    }

```

2.72.6. `video.squeeze`

```
video.squeeze ()  
{  
    [[ -z "$*" ]] && {  
        printf -- "${bldgrn}USAGE:\n${bldlw}[ DEBUG=1 ] video-squeeze *.mp4  
*.m4v${clr}\n"  
        return 0  
    }  
    for file in "$@"  
    do  
        [[ -s "${file}" ]] || {  
            warning "Skipping ${file}..."  
            continue  
        }  
        dest=".destination-file-name ${file}"  
        arrow.blk-on-blu "Compressing \"${file}\""  
        video.encode "${file}" "11" "${dest}"  
    done  
}
```

2.73. Module `vim`

2.73.1. `gvim.off`

```
gvim.off ()  
{  
    vim.gvim-off  
}
```

2.73.2. `gvim.on`

```
gvim.on ()  
{  
    vim.gvim-on  
}
```

2.73.3. `vim.gvim-off`

```
vim.gvim-off ()  
{  
    vim.setup  
    [[ "$EDITOR" == "vim" ]] && return 0  
    local regex_from='^export EDITOR=.*$'  
    local regex_to='export EDITOR=vim'
```

```

file.gsub "${LibVim__initFile}" "${regex_from}" "${regex_to}"
file.gsub "${LibVim__initFile}" '^gvim.on$' 'gvim.off'
${GrepCommand} -q "${regex_from}" "${LibVim__initFile}" || echo "${regex_to}" >>
"${LibVim__initFile}"
${GrepCommand} -q "^gvim\.o" "${LibVim__initFile}" || echo "gvim.off" >> "
${LibVim__initFile}"
eval "
[[ -n '${BASHMATIC_DEBUG}' ]] && set -x
export EDITOR=${LibVim__editorGvimOff}
unalias ${LibVim__editorVi} 2>/dev/null
unalias ${LibVim__editorGvimOff} 2>/dev/null
"
}

}

```

2.73.4. vim.gvim-on

```

vim.gvim-on ()
{
    vim.setup
    [[ "$EDITOR" == "gvim" ]] && return 0
    local regex_from='^export EDITOR=.*$'
    local regex_to='export EDITOR=gvim'
    file.gsub "${LibVim__initFile}" "${regex_from}" "${regex_to}"
    file.gsub "${LibVim__initFile}" '^gvim.off$' 'gvim.on'
    ${GrepCommand} -q "${regex_from}" "${LibVim__initFile}" || echo "${regex_to}" >>
"${LibVim__initFile}"
    ${GrepCommand} -q "^gvim\.o.*" "${LibVim__initFile}" || echo "gvim.on" >> "
${LibVim__initFile}"
    eval "
[[ -n '${BASHMATIC_DEBUG}' ]] && set -x
export EDITOR=${libVim__editorGvimOn}
alias ${LibVim__editorVi}=${LibVim__editorGvimOn}
alias ${LibVim__editorGvimOff}=${LibVim__editorGvimOn}
"
}

```

2.73.5. vim.setup

```

vim.setup ()
{
    export LibVim__initFile="${HOME}/.bash_profile"
    export LibVim__editorVi="vi"
    export LibVim__editorGvimOn="gvim"
    export LibVim__editorGvimOff="vim"
}

```

2.74. Module `yaml`

2.74.1. `yaml-diff`

```
yaml-diff ()  
{  
    yaml.diff "$@"  
}
```

2.74.2. `yaml-dump`

```
yaml-dump ()  
{  
    yaml.dump "$@"  
}
```

2.74.3. `yaml.diff`

```
yaml.diff ()  
{  
    local f1="$1"  
    shift  
    local f2="$1"  
    shift  
    [[ -f "$f1" && -f "$f2" ]] || {  
        h2 "USAGE: ${bldylw}yaml-diff file1.yml file2.yml [ ydiff-options ]"  
        return 1  
    }  
    [[ -n $(which ${BashMatic__DiffTool}) ]] || brew.package.install  
    ${BashMatic__DiffTool}  
    local t1="/tmp/${RANDOM}.$(basename "${f1}").$$.yml"  
    local t2="/tmp/${RANDOM}.$(basename "${f2}").$$.yml"  
    yaml.expand-aliases "$f1" > "$t1"  
    yaml.expand-aliases "$f2" > "$t2"  
    run.set-next show-output-on  
    hr  
    run "ydiff $* ${t1} ${t2}"  
    hr  
    run "rm -rf ${t1} ${t2}"  
}
```

2.74.4. `yaml.dump`

```
yaml.dump ()  
{
```

```

local f1="$1"
shift
[[ -f "$f1" ]] || {
    h2 "USAGE: ${bldylw}yaml-dump file.yaml" 1>&2
    return 1
}
[[ -n $(which ${BashMatic__DiffTool}) ]] || brew.package.install
${BashMatic__DiffTool}
local t1=$(mktemp)
export yaml_dump_file="$t1.$(basename "$f1")"
yaml.expand-aliases "$f1" > "${yaml_dump_file}"
cat "${yaml_dump_file}"
}

```

2.74.5. `yaml.edit`

```

yaml.edit ()
{
    yaml.dump "$@"
    vim "${yaml_dump_file}"
}

```

2.74.6. `yaml.expand-aliases`

```

yaml.expand-aliases ()
{
    local file="${1}"
    local temp=$(mktemp)
    cat <<-RUBY > "${temp}"
    require 'yaml';
    puts YAML.dump(YAML.load(File.read('${file}'), aliases: true))
RUBY

    ruby "${temp}"
    rm -f "${temp}"
}

```

2.75. Module `yarn`

2.75.1. `yarn_install`

```

yarn_install ()
{
    command -v yarn > /dev/null || npm install -g yarn
    if [[ ! -f .yarn.sha || "$(cat .yarn.sha)" != "$(yarn_sha)" ]]; then
        set -x

```

```
yarn install
set +x
yarn_sha > .yarn.sha
fi
}
```

2.75.2. `yarn_sha`

```
yarn_sha () {
    [[ -f package.json && -f yarn.lock ]] && cat package.json yarn.lock | sha
}
```

2.76. Module zsh

2.76.1. `zsh.install`

```
zsh.install () {
    zsh.install.plugins
}
```

2.76.2. `zsh.install.oh-my-zsh`

```
zsh.install.oh-my-zsh () {
    if [[ -d ${HOME}/.oh-my-zsh/ ]]; then
        info "oh-my-zsh is already installed, updating..."
        run "cd ${HOME}/.oh-my-zsh"
        run "git pull || true"
    else
        info "Installing oh-my-zsh..."
        run "sh -c \"\$(curl -fsSL
https://raw.githubusercontent.com/ohmyzsh/ohmyzsh/master/tools/install.sh)\""
    fi
    if [[ -f ${HOME}/.zshrc ]]; then
        run "sed -E -i '' 's/robbyrussell/agnoster/g' ${HOME}/.zshrc"
        run "sed -E -i '' 's/^plugins=.*$/plugins=(git wd golang osx aws brew zsh-
completions)/g' ${HOME}/.zshrc"
    fi
}
```

2.76.3. zsh.install.plugins

```
zsh.install.plugins ()  
{  
    zsh.install.oh-my-zsh  
    local dest="${ZSH_CUSTOM:=${HOME}/.oh-my-zsh/custom}/plugins"  
    run "mkdir -p ${dest}"  
    [[ -d ${dest}/zsh-completions ]] || {  
        run "git clone https://github.com/zsh-users/zsh-completions ${dest}/zsh-  
completions"  
    }  
    [[ -d ${dest}/brew ]] || {  
        run "mkdir ${dest}/brew"  
        run "curl -fsSL  
https://raw.githubusercontent.com/ohmyzsh/ohmyzsh/master/plugins/brew/brew.plugin.zsh  
> ${dest}/brew/brew.plugin.zsh"  
    }  
}
```

Chapter 3. Copyright

© 2016-2022 Konstantin Gredeskoul, All rights reserved, MIT License.