FreeBSD Accessibility Handbook

Abstract

Welcome to the Accessibility Handbook. This book covers the assistive technologies available in FreeBSD.

Accessibility is an ongoing effort, not a fixed destination. As such, this handbook is a living document and continually evolving. Contributions, suggestions, and feedback are highly encouraged. If you are interested in contributing, please reach out to us via the FreeBSD accessibility mailing list.

You can download this book in various formats and compression options from the FreeBSD download server or one of its many mirror sites.

Table of Contents

Preface . . . 4
Intended Audience . . . 4
Organization of this Book . . . 4
Commands . . . 4
I: General . . . 6
1. Help . . . 7
1.1. Mailing Lists . . . 7
1.2. Chat . . . 7
1.3. Ports and Packages . . . 7
II: Vision . . . 8
2. Virtual Terminal . . . 9
2.1. Introduction . . . 9
2.2. Colors . . . 9
2.3. Resolution . . . 11
2.4. Font . . . 11
2.5. Bell . . . 12
3. High Contrast . . . 13
3.1. Introduction . . . 13
3.2. High Contrast Theme . . . 13
3.3. High Contrast Icons . . . 14
3.4. Color temperature . . . 15
3.5. Ports Collection . . . 19
3.6. Brightness . . . 20
3.7. Color Blindness . . . 22
4. Low Vision . . . 23
4.1. Introduction . . . 23
4.2. Desktop Size . . . 23
4.3. Font . . . 23
4.4. Magnify . . . 25
4.5. Mouse . . . 27
5. Blindness . . . 30
5.1. Introduction . . . 30
5.2. Screen Reader . . . 30
5.3. Speech Synthesizer . . . 31
5.4. Ports Collection . . . 33
5.5. Editor . . . 34
5.6. Internet and WEB . . . 34
5.7. Liblouis . . . 36

6. Development . . . 37
6.1. Introduction . . . 37
6.2. Colors . . . 37
6.3. Command Line Interface . . . 37
6.4. Libraries . . . 37
6.5. Tools . . . 39

Preface

Intended Audience

This handbook is intended for anyone interested in the assistive technologies available in the FreeBSD operating system. Users and administrators can learn how to configure the system for accessibility. Developers will find resources and guidance to help them create, improve, and test accessible software.

Organization of this Book

This book is organized into sections, chapters, and paragraphs, all of which can be read independently and out of order. The first section, *Part I. General*, provides guidance on how to seek help and includes various miscellaneous topics. The second section, *Part II. Vision*, focuses on assistive technologies for visual accessibility; It is intentionally free of images to maximize accessibility All sections and chapters are self-contained and can be read in any order.

Chapters:

Help

Explains where and how to seek support within the FreeBSD community.

Virtual Terminal

Highlights features of the system's virtual console for visual accessibility.

Colors

Describes how to customize color settings in graphical environments for better visibility.

Low Vision

Introduces tools and settings for users with low vision.

Blindness

Covers accessibility tools and configurations for blind users, including screen readers, text-based utilities, and relevant ports.

Development

Offers guidelines and resources for developers to enhance accessibility for users with visual impairments.

Commands

FreeBSD offers a powerful command line interface that enables users to execute a wide range of text commands to set up and use the operating system. This Handbook, like all official FreeBSD documentation, follows these command prompt conventions:

- A command preceded by `%` can be executed by any regular user.

- A command preceded by `#` must be executed by the `root` user (the system administrator).

Part I: General

The General Part includes miscellaneous information, settings, and tools that do not belong to a specific category.

Chapter 1. Help

1.1. Mailing Lists

Mailing lists are the primary communication channel for the FreeBSD community, covering a wide range of topics related to the project. Messages are typically written in plain text without graphical elements, making them well-suited for use with screen readers (both Braille displays and speech synthesizers). A complete list of available mailing lists can be found at: https://lists.freebsd.org/.

For accessibility issues, the main point of contact is the FreeBSD Accessibility Mailing List. To subscribe, browse the archive, send messages, or manage your subscription, visit: https://lists.freebsd.org/subscription/freebsd-accessibility

1.2. Chat

FreeBSD users and developers are also active on IRC (Internet Relay Chat), where real-time communication takes place. A complete list of servers and channels is available at: https://wiki.freebsd.org/IRC/Channels Messages are typically exchanged in plain text, making IRC well-suited for screen readers (both Braille displays and speech synthesizers).

1.3. Ports and Packages

If the issue pertains to an external tool installed through a port or package, consider including the port maintainer in the discussion to ensure proper attention and resolution.

To find out who maintains a specific port, navigate to the port's directory and run the command `make maintainer`. For example, to find the maintainer of the www/edbrowse port, run:

```
% cd /usr/ports/www/edbrowse
% make maintainer
```

To find the maintainer of an installed package, use the `pkg info -f package`. For example:

```
% pkg info -f edbrowse | grep Maintainer | awk '{print $3}'
```

Part II: Vision

Section for Assistive technologies for vision accessibility.

The operating system provides two main interfaces for user interaction: Virtual Terminal and several Desktop Environments.

Virtual Terminal, commonly referenced and documented as vt(4) is the system's built-in virtual console. It is installed by default, provides a command-line interface, and starts automatically at boot. Its features relevant to vision accessibility are detailed in the Virtual Terminal chapter.

Desktop environments provide graphical elements, such as windows, menus, buttons, and so on. They can be installed using ports and packages; refer to Chapters X Window System, Wayland, and Desktop Environments of the FreeBSD Handbook. This section explains how to install, configure, and use color schemes, themes, and visual accessibility features designed to support users with low vision or blindness in graphical environments.

Chapter 2. Virtual Terminal

2.1. Introduction

Virtual Terminal, commonly referenced and documented as vt(4) is the system's built-in virtual console. It is installed by default, provides a command-line interface, and starts automatically at boot. This chapter presents an overview and examples of key features and utilities relevant to visual accessibility.

2.2. Colors

The Virtual Terminal supports color configuration both via configuration files and interactively through a dedicated utility.

2.2.1. Loader.conf

Colors can be set and saved using the loader.conf(5) file. To change a default color, add a line in the following format: `kern.vt.color.colornum.rgb="colorspec"`, where:

- `colornum` is the numeric identifier of the color to be customized (see the table below).
- `colorspec` is a specification of the color, either as a comma-separated RGB triplet (red, green, and blue, each value from 0 to 255) or as an HTML-style hexadecimal value.

Table 1. Default palette: Color, color number, default color in rgb, default color in html.

Color	*colornum*	**default** *colorspec* **RGB**	**default** *colorspec* **html**
Black	0	0,0,0	#000000
Dark red	1	127,0,0	#7f0000
Dark green	2	0,127,0	#007f00
Dark yellow	3	196,161,0	#c4a100
Dark blue	4	51,102,163	#3366a3
Dark magenta	5	127,0,127	#7f007f
Dark cyan	6	0,127,127	#007f7f
Light gray	7	191,191,191	#bfbfbf
Dark gray	8	115,130,138	#73828a
Light red	9	255,0,0	#ff0000
Light green	10	0,255, 0	#00ff00
Light yellow	11	255,255,0	#ffff00
Light blue	12	115,158,207	#739ecf
Light magenta	13	255,0,255	#ff00ff
Light cyan	14	0,255,255	#00ffff

Color	*colornum*	**default** *colorspec* **RGB**	**default** *colorspec* **html**
White	15	255,255,255	#ffffff

Example 1. Set background in white, text in black, boot messages in red.

/boot/loader.conf

```
# Background black to white
kern.vt.color.0.rgb="255,255,255"

# Text light gray to black
kern.vt.color.7.rgb="0,0,0"

# Boot white to red
kern.vt.color.15.rgb="255,0,0"
```

2.2.2. Vidcontrol

The vidcontrol(1) utility allows you to configure colors interactively. Use the `show` option to display the list of available colors:

```
% vidcontrol show
```

To set the foreground color, use `vidcontrol color`. For example, to set the text color to green:

```
% vidcontrol green
```

To set both foreground and background colors, use `vidcontrol foreground background`. For example, to set blue text on a green background:

```
% vidcontrol blue green
```

Configurations made with vidcontrol(1) are not persistent across shutdown. To make the settings permanent, add the following line to rc.conf(5): `allscreens_flags="-c foreground background"`.

Example 2. Set blue foreground and green background.

/etc/rc.conf

```
allscreens_flags="-c blue green"
```

Setting allscreens_flags="-c *colors*" in /etc/rc.conf takes effect after the system

boots. To change the colors during the boot process, edit /boot/loader.conf as previously described.

2.3. Resolution

The `kern.vt.fb.default_mode="<width>x<height>"` option in loader.conf(5) allows you to configure the display resolution. Technically, it specifies the display mode of the Virtual Terminal, allowing you to control the size of elements displayed on the screen.

Example 3. Sets the width to 800 and the height to 600.

/boot/loader.conf

```
kern.vt.fb.default_mode="800x600"
```

2.4. Font

Virtual Terminal allows you to configure the font in use, and the system supports installing and utilizing additional fonts.

2.4.1. Select a Font

The utilities vidfont(1) and vidcontrol(1) can load a font from /usr/share/vt/fonts/. To list the currently available fonts:

```
% ls /usr/share/vt/fonts
```

Usually a file name has the format `<name>-<width>x<height>`, where:

- `<name>` is the font name.
- `<width>x<height>` is the size.

vidfont(1) provides an interactive menu in a Text User Interface to select a font. Run the following command:

```
% vidfont
```

Alternatively, vidcontrol(1) provides a CLI, `vidcontrol -f font`. Example for `spleen-32x64.fnt`:

```
% vidcontrol -f spleen-32x64.fnt
```

Tip: You can omit the *.fnt* extension when specifying the file name. For example, to load `spleen-32x64.fnt`, use:

```
% vidcontrol -f spleen-32x64
```

Settings configured with either utility are not preserved after a reboot.

Example 4. Select the font to load at each startup.

/etc/rc.conf

```
allscreens_flags="-f spleen-32x64.fnt"
```

2.4.2. New Fonts

The vtfontcvt(8) utility allows you to convert a font into the .fnt format for use in the Virtual Terminal. The converted font must be copied to /usr/share/vt/fonts/ and selected as previously described.

2.5. Bell

The terminal bell can be activated by:

```
# sysctl kern.vt.enable_bell=1
```

To make the configuration persistent after reboot, save it in either loader.conf(5) or sysctl.conf(5).

Example 5. Active the bell at each startup.

/boot/loader.conf or /etc/sysctl.conf

```
kern.vt.enable_bell=1
```

Chapter 3. High Contrast

3.1. Introduction

This chapter explains how to configure colors for on-screen elements, including:

- Setting high-contrast color schemes in graphical environments.
- Adapting screen color temperature for optimal viewing in daylight and nighttime conditions.
- Selecting colors for configuration ports menus.

3.2. High Contrast Theme

Graphical Environments allow you to easily install and set up graphical themes. This paragraph describes how to configure high contrast themes for windows.

3.2.1. KDE Plasma

Launch the `Global Theme` module:

```
% kcmshell6 kcm_lookandfeel
```

The window shows the current installed themes. Click on the "Get New..." icon in the top right corner. A new window will show additional themes available for installation. In the search field at the top right, you can look for new themes to install. You can search for "high contrast" or "Acrilic Dark".

3.2.2. XFCE

Download from High Contrast Windows 11 Themes:

- *Adwaita-Dusk.tar.bz2*
- *Adwaita-Desert.tar.bz2*
- *Adwaita-Aquatic.tar.bz2*
- *Adwaita-NightSky.tar.bz2*

The example is for *NightSky*, but of course you can choose whichever theme you prefer.

```
% tar xzvf Adwaita-NightSky.tar.bz2 ①
% mkdir ~/.themes ②
% cp -a Adwaita-NightSky ~/.themes/ ③
% rm Adwaita-NightSky.tar.bz2 ④
% xfconf-query -c xfwm4 -p /general/theme -s 'Adwaita-NightSky' ⑤
% xfconf-query -c xsettings -p /Net/ThemeName -s 'Adwaita-NightSky' ⑥
```

① Extract the downloaded theme archive.

② Create the ~/.themes directory if it does not already exist.

③ Copy the extracted themes into ~/.themes.

④ Optionally, delete the original compressed file to save space.

⑤ Apply the Adwaita-NightSky theme for the XFCE desktop environment.

⑥ Apply the Adwaita-NightSky theme for GTK applications.

The theme selection steps (5 and 6) can also be performed using the graphical configuration dialogs in XFCE:

```
% xfwm4-settings
% xfce4-appearance-settings
```

3.3. High Contrast Icons

Graphical Environments allow you to easily install and set up icon themes. This paragraph describes how to configure high contrast icons.

3.3.1. KDE Plasma

Open the `Icons` module:

```
% kcmshell6 kcm_icons
```

The window displays the currently installed icon themes. To install a new theme, click the `Get New···` icon in the top right corner. A new window will appear, allowing you to browse and install additional icon themes. As a suggestion, you can search for `Treepata` using the search bar at the top right. Although originally designed for XFCE and later adapted for KDE Plasma, Treepata was created as a high-contrast accessibility theme.

3.3.2. XFCE

Run the following commands to use the *Treepata - High Contrast* icons theme with XFCE.

Download *Treepata.zip* from Treepata - High contrast. Then execute:

```
# pkg search zip ①
% unzip Treepata.zip ②
% mkdir ~/.icons ③
% cp -a Treepata ~/.icons/ ④
% rm Treepata.zip ⑤
% xfconf-query -c xsettings -p /Net/IconThemeName -s Treepata ⑥
```

① Install the package archivers/zip:

② Extract the downloaded archive file.

③ Create the ~/.icons directory if it does not already exist.

④ Copy the extracted theme into ~/.icons.

⑤ Optionally, delete the downloaded archive file to free up space.

⑥ Set the *Treepata* icon theme.

The icon theme (step 6 above) can also be set graphically via the `Appearance` dialog under the `Icons` tab. To do so, open the dialog and select your preferred icon theme:

```
% xfce4-appearance-settings
```

If you have icon theme problems, install misc/hicolor-icon-theme and graphics/gtk-update-icon-cache:

```
# pkg install hicolor-icon-theme gtk-update-icon-cache
```

Run gtk-update-icon-cache(1) to create or update the icon cache:

```
% gtk-update-icon-cache -f -t ~/.icons/treepata
```

3.4. Color temperature

This paragraph explains various utilities that adjust screen colors based on temperature, day/night cycles, and gamma settings. These utilities enable both interactive color adjustments and automatic changes based on the local time and geographical location. Some utilities are compatible with the X Window System, others support Wayland, and some work on both display servers.

Table 2. Color Temperature Utilities

Name	Auto Update	Environment	Package
darkman	Yes	X11	accessibility/darkman
gammastep	Yes	X11	accessibility/gammastep
Night Light	Yes	KDE Plasma	x11/kde
redshift	Yes	X11, Wayland	accessibility/redshift
sct	No	X11	accessibility/sct
sctd	Yes	X11	accessibility/sctd
wlsunset	Yes	Wayland	accessibility/wlsunset

The rest of this section details the steps to install, configure, and launch the utilities.

3.4.1. darkman

darkman(1) is a daemon that automatically manages transitions between dark mode and light mode. It operates primarily in the background, switching the screen's color scheme based on the local time of the current location. To install, run:

```
# pkg install darkman
```

Create the file ~/.config/darkman/config.yaml and write your latitude and longitude. Example:

```
lat: 37.52
lng: 122.16
```

The utility does not directly change the screen colors; instead, it calls external utilities to perform the color transitions. Add scripts to be executed at sundown by placing them in $XDG_DATA_DIRS/dark-mode.d/ and scripts to be executed at sunrise by placing them in XDG_DATA_DIRS/light-mode.d/. Several example scripts are available in the project repository.

To start darkman manually, run:

```
% darkman run &
```

The utility also offers some interactive options; for more details refer to the darkman(1) manual page.

3.4.2. gammastep

The gammastep(1) utility adjusts your screen's color temperature automatically based on your location. To install, run:

```
# pkg install gammastep
```

The utility works out of the box without any configuration. Simply start it by running:

```
% gammastep &
```

If you encounter any issues, customize the configuration by adapting the gammastep.conf.sample file and save it as ~/.config/gammastep/conf.ini. Refer to the README for troubleshooting.

3.4.3. KDE Plasma

KDE allows you to adjust the screen color temperature based on geographic location and local time. Launch the `Night Light` module:

```
% kcmshell6 kcm_nightlight
```

Enable automatic color adjustment by selecting the desired option from the `Switching times` drop-down menu.

3.4.4. redshift

The redshift(1) utility find your position and sets the color temperature accordingly. To install, execute:

```
# pkg install redshift
```

The redshift utility runs without any configuration. To start it, simply execute:

```
% redshift &
```

The utility might hang while trying to determine the location or the correct method to set the color. In this case, specify them explicitly using `-l latitude:longitude` and `-m method`. The available methods can be listed by running: `% redshift -m list`.

Example of running redshift with an explicit location and method:

```
% redshift -l 37.86:-122.27 -m randr &
```

To make this setting permanent, add the following lines to ~/.config/redshift/redshift.conf file:

```
[redshift]
adjustment-method=randr
location-provider=manual

[manual]
lat=37.86
lon=-122.27
```

The project provides a default redshift.conf.sample configuration file. For troubleshooting and additional information, please refer to the README.

3.4.5. sct

The sct(1) utility is a simple command-line tool for controlling screen color temperature. It does not update automatically and must be run repeatedly to adjust the temperature over time. To install, execute:

```
# pkg install sct
```

The utility takes temperature values between 1000 and 10000, 6500 is the default. To run sct, type: `sct value`:

```
% sct 8000
```

The utility suggests 3 styles:

1) Campfire style

```
% sct 4500
```

2) Dust storm on Mars style

```
% sct 2000
```

3) Coffee free all nighter style

```
% sct 8000
```

3.4.6. sctd

The sctd utility calculates sunrise and sunset times based on latitude and longitude, and adjusts the screen's color temperature accordingly. To install, run:

```
# pkg install sctd
```

To run sctd, specify your location using the `--latitude` and `--longitude` options:

```
% sctd --latitude 40.55 --longitude 14.34 &
```

To reset the screen color to default, run:

```
% sctd --reset
```

3.4.7. wlsunset

The wlsunset(1) utility is a Day/Night gamma adjustments for Wayland compositors.

To install, execute:

```
# pkg install wlsunset
```

To start the utility type `wlsunset -l latitude -L longitude`. Example:

```
% wlsunset -l 14.34 -L 40.55 &
```

3.5. Ports Collection

FreeBSD provides the Ports Collection, which offers a simple way to install applications. Some ports allow you to configure options before building and installation. By default, configuration is handled through a TUI menu built by the ports-mgmt/portconfig. The colors can be customized in several ways:

1. Terminal color capabilities.
2. The $NO_COLOR environment variable.
3. The $PORTCONFIG_THEMEFILE environment variable.
4. The global /usr/local/etc/portconfig/theme.conf file.

These methods are listed in order of decreasing priority and are mutually exclusive, meaning only one can be active at a time.

1. Terminal Colors

portconfig(1) automatically detects whether the terminal supports colors. If not, it defaults to a black-and-white theme.

2. NO_COLOR

If the `$NO_COLOR` environment variable is set and not empty, portconfig forces the black-and-white theme

Example 6. Example persistent setting `$NO_COLOR`

/etc/make.conf

```
export NO_COLOR=YES
```

3. PORTCONFIG_THEMEFILE

If the `$PORTCONFIG_THEMEFILE` environment variable is set and points to a valid theme file, it will be used. Otherwise, the default theme is applied.

Example 7. .Example persistent setting `$PORTCONFIG_THEMEFILE`

/etc/make.conf

```
export PORTCONFIG_THEMEFILE=/home/foo/mytheme.conf
```

To create a new theme file, run:

```
% /usr/local/bin/bsddialog --save-theme mytheme.conf --infobox "Saving
theme..." 0 0
```

4. Global theme.conf

If the file /usr/local/etc/portconfig/theme.conf exists and is valid, it will be used. If not, the default theme is applied.

Sample themes are included in /usr/local/etc/portconfig/:

- blackwhite.conf
- default.conf
- red-green.conf
- yellow-blue.conf

To use the yellow-blue.conf theme:

```
# ln -s /usr/local/etc/portconfig/yellow-blue.conf
/usr/local/etc/portconfig/theme.conf
```

3.6. Brightness

Screen brightness can be adjusted in various ways. The available methods depend heavily on the hardware, so it's recommended to try the features described in the following sections.

3.6.1. backlight

The backlight(8) utility is included with the operating system. It is used to set the brightness of devices located under /dev/backlight/.

The command `backlight -f device value` sets the brightness of *device* to *value*, where *value* is an integer between 0 (dim) and 100 (bright). If no device is specified, the default device /dev/backlight/backlight0 is considered.

Example for setting the brightness of the default device to 50%:

```
% backlight 50
```

Unless the user belongs to the "video" group, the command needs to run as root.

3.6.2. acpi_video

The acpi_video(4) driver, included with the operating system, uses the ACPI Video Extensions to control display switching and backlight brightness. To load the kernel module, run:

```
# kldload acpi_video
```

Brightness settings can be configured using sysctl(8). The following parameters are available:

- hw.acpi.video.*device*.levels: List of supported brightness levels.
- hw.acpi.video.*device*.brightness: Current brightness level of the device.

device is a possible display device. Example to set the brightness to 50% for the lcd0 device:

```
% sysctl hw.acpi.video.lcd0.brightness ①
hw.acpi.video.lcd0.brightness: 90
% sysctl hw.acpi.video.lcd0.levels ②
hw.acpi.video.lcd0.levels: 90 60 2 4 6 8 10 12 14 16
18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50 52
54 56 58 60 62 64 66 68 70 72 74 76 78 80 82 84 86 88
90 92 94 96 98 100
# sysctl hw.acpi.video.lcd0.brightness=50 ③
hw.acpi.video.lcd0.brightness: 90 -> 50
```

① `sysctl hw.acpi.video.device.brightness` to view the current brightness level.

② `sysctl hw.acpi.video.device.levels` to view the available levels.

③ `sysctl hw.acpi.video.device.brightness=50` to set the brightness to 50.

acpi_video(4) should be loaded after any of the DRM kernel modules.

3.6.3. xbrightness

The x11/xbrightness utility can change the brightness via the X Window System.

```
# pkg install xbrightness
```

The xbrightness(1) utility allows to set a value between 0 (darkness) and 65535 (full brightness). Example for setting brightness to 50%:

```
% xbrightness 32767
```

3.6.4. xrandr

The x11/xrandr package is an extension of the X11 server that provides the xrandr(1) utility. It allows you to configure various display properties, including brightness. To install execute:

```
# pkg install xrandr
```

The command `xrandr --output monitor --brightness brightness` allows you to set the brightness for monitor to a value between 0 and 1.0. To list the available monitors, run:

```
% xrandr --listmonitors
Monitors: 2
 0: +HDMI-1 1920/598x1080/337+0+0  HDMI-1
 1: +eDP-1 1920/344x1080/194+1920+0  eDP-1
```

Example for setting brightness to 50% for HDMI-1:

```
% xrandr --output HDMI-1 --brightness 0.5
```

3.7. Color Blindness

This section describes how to use utilities designed for color blindness.

3.7.1. KDE Plasma

KDE allows you to filter colors for graphical elements. Launch the `Accessibility` module:

```
% kcmshell6 kcm_access
```

In the navigation pane, select `Color Blindness Correction`. Enable the `Color blindness correction` option, then choose the appropriate color transformation mode from the `Mode` drop-down menu.

Chapter 4. Low Vision

4.1. Introduction

This chapter outlines accessibility features available in graphical desktop environments designed for users with low vision, such as virtual magnifiers, easy-to-read fonts, and screen scaling.

4.2. Desktop Size

Graphical environments allow users to adjust the desktop display size, a feature commonly known as screen scaling or HiDPI scaling. The goal is to enlarge desktop elements such as icons, text, and windows to improve visibility and accessibility.

4.2.1. KDE Plasma

Open the `Display Configuration` module:

```
% kcmshell6 kcm_kscreen
```

Select the scaling via the `Global Scale` bar. The change will take effect after a KDE Plasma restart.

4.2.2. XFCE

To set a custom scaling factor, for example 4, run the following command:

```
% xfconf-query -c xsettings -p /Gdk/WindowScalingFactor -s 4
```

Alternatively, HiDPI scaling can be enabled with a fixed 2x factor using the graphical interface. This option is available in the `Appearance` dialog, under the `Settings` tab as `Window Scaling`. To open the `Appearance` dialog, execute:

```
% xfce4-appearance-settings
```

4.3. Font

The x11-fonts/atkinson-hyperlegible provides the *atkinson-hyperlegible* font. It is designed for good readability, the letter shapes aim to increase character recognition. To install, execute:

```
# pkg install atkinson-hyperlegible
```

The remainder of this section describes how to set up fonts in various graphical environments using *atkinson-hyperlegible* as an example.

4.3.1. KDE Plasma

KDE allows selecting a font and its size via the Fonts module. Run:

```
% kcmshell6 kcm_fonts
```

Click on the Adjust All Fonts··· button to select the font and its size for all graphical elements. Alternatively, you can choose the font and size for individual elements using the forms below.

At the bottom of the window, you can also select options such as Anti-Aliasing, Sub-pixel Rendering, Hinting, and Force Font DPI. Each option has an icon on the right to open a help dialog.

4.3.2. XFCE

XFCE allows you to select the font and size for window titles and content text, as well as the font size for icon labels.

```
% xfconf-query -c xfwm4 -p /general/title_font -s "Atkinson Hyperlegible 15" ①
% xfconf-query -c xsettings -p /Gtk/FontName -s "Atkinson Hyperlegible 15" ②
% xfconf-query -c xsettings -p /Gtk/MonospaceFontName -s "Atkinson Hyperlegible 15" ③
% xfconf-query -c xfce4-desktop -p /desktop-icons/use-custom-font-size -s true ④
% xfconf-query -c xfce4-desktop -p /desktop-icons/font-size -s 15.0 ⑤
```

① Set the window title font to *Atkinson Hyperlegible*, size 15.

② Set the window content font to *Atkinson Hyperlegible*, size 15.

③ Set the monospace font within windows to *Atkinson Hyperlegible*, size 15.

④ Enable custom font sizing for icons.

⑤ Set the icon font size to 15.0.

Alternatively, you can configure the same settings using the graphical XFCE configuration dialogs.

To select the font for window titles, open the Window Manager dialog:

```
% xfwm4-settings
```

To choose the font used inside windows, open the Appearance dialog and go to the Fonts tab:

```
% xfce4-appearance-settings
```

In the Fonts tab, you can also adjust options such as anti-aliasing, Hinting, Sub-pixel order, and Custom DPI. For more details, see the XFCE documentation.

To set the font size for icons, open the Desktop dialog and go to the Desktop Icons tab where you can adjust the icon font size:

```
% xfdesktop-settings
```

4.4. Magnify

This section describes various utilities that magnify portions of the screen, acting as virtual magnifying lenses. Some tools continuously enlarge the area beneath the mouse cursor, while others do not. Certain utilities automatically move the magnified window to follow the cursor, whereas others keep it fixed in place.

Table 3. Screen magnifier utilities.

Name	Auto Update	Auto Move Lens	Package
lupe	Yes	No	x11/lupe
kmag	Yes	No	accessibility/kmag
xlupe	Yes	No	x11/xlupe
xmag	No	No	x11/xmag
xzoom	No	No	x11/xzoom

The rest of this section details the steps to install, configure, and launch the utilities.

4.4.1. lupe

The lupe(1) utility is a screen magnifier for the X Window System that updates the magnified area in real time. To install it, run:

```
# pkg install lupe
```

By default, lupe magnifies the screen content beneath the mouse pointer within a circular area (25×25 pixels) displayed at the center of the screen. To start lupe, run:

```
% lupe
```

To exit lupe, press the `q` key.

Start lupe with the `-noshape` option to display the magnified area in a separate, movable window. Without this option, the magnifier is locked to the center of the screen.

4.4.2. kmag

The kmag(1) utility provides a screen magnifier that displays a zoomed view of the area under the mouse cursor in a dedicated window. The magnifier window includes a menu with various options and settings for customization.

To install kmag, run:

```
# pkg install kmag
```

To launch kmag, execute:

```
% kmag
```

4.4.3. xlupe

The x11/xlupe utility is a screen magnifier for the X Window System that updates the magnified view in real time. To install it, run:

```
# pkg install xlupe
```

To launch xlupe, execute:

```
% xlupe
```

The utility does not include a manual page, but it starts in a graphical window with buttons that allow you to configure its options interactively.

4.4.4. xmag

The xmag(1) utility is a basic screen magnifier for the X Window System. Unlike other magnifiers, it does not update the magnified area in real time.

To install it, run:

To install, execute:

```
# pkg install xmag
```

To launch xmag, execute:

```
% mag
```

xmag does not display a magnified area immediately upon launch. Click anywhere on the screen with the left mouse button to select a region, and xmag will open a window showing a magnified view of that area. The application window includes a few buttons offering limited functionality.

4.4.5. xzoom

The xzoom(1) utility is a screen magnifier for the X Window System that updates the magnified view in real time. To install it, run:

```
# pkg install xzoom
```

To launch xzoom, execute:

```
% xzoom
```

Unlike other magnifiers, xzoom does not automatically follow the mouse cursor or magnify the area beneath it in its own window. Interaction is keyboard-driven, refer to xzoom(1) for available commands and usage details.

4.5. Mouse

4.5.1. Appearance

Some desktop environments allow you to change the size, color, and other graphical properties of the mouse cursor.

KDE Plasma

KDE Plasma provides the `Cursors` module to select the mouse cursor theme and size. To launch it, run:

```
% kcmshell6 kcm_cursortheme
```

Select an installed cursor theme in the window. There is a `Size` drop-down menu in the top left corner to choose the mouse cursor size, available sizes depend on the selected theme.

Click the `Get New···` icon in the top right corner to install a new theme. You can install new themes from the new window. As a suggestion, search for `Large Mouse Cursors`, a black and white high contrast theme.

XFCE

XFCE allows you to set the size and color of the mouse pointer via `Mouse and Touchpad` dialog. To run the dialog start:

```
% xfce4-mouse-settings
```

Then click on the `Theme` tab to select a size and a style.

Some themes, included the *Default*, do not support cursor size customization.

Alternatively, you can achieve the same result via the `xfconf-query` command. The following example sets the current theme cursor to size 50:

```
% xfconf-query -c xsettings -n -p /Gtk/CursorThemeSize -s 50
```

4.5.2. Find Cursor

Some desktop environments provide visual feedback of the mouse pointer's current position.

KDE Plasma

KDE Plasma draws two circular lines moving around the mouse cursor. Open the `Desktop Effects` module:

```
% kcmshell6 kcm_kwin_effects
```

Enable the `Track Mouse` checkbox. Draw circular lines by pressing `Meta` + `Ctrl` keys. You can change the shortcut via the `Configure` button related to `Track Mouse`.

On keyboards where the `Meta` key is not present, it is usually replaced by the `Super` key. Refer to https://en.wikipedia.org/wiki/Meta_key and https://en.wikipedia.org/wiki/Super_key_(keyboard_button) for more information.

XFCE

Xfce draws large red circles around the mouse pointer. Type:

```
% xfce4-find-cursor
```

The above command draws circles for a few seconds.

Keyboard Shortcut

XFCE allows to add a keyboard shortcut. Circles are drawn for as long as the shortcut keys are pressed. To use the shortcut before enable the feature then choose the key combination to bind to `xfce4-find-cursor`.

Example to enable and bind `Ctrl` + `Super` + `k`:

```
xfconf-query -c accessibility -p /FindCursor -s true
xfconf-query -c xfce4-keyboard-shortcuts -n -t 'string' -p
'/commands/custom/<Ctrl><Super>k' -s xfce4-find-cursor
```

Keyboard Shortcut via Dialogs

Alternatively, you can achieve the same result via the XFCE configuration dialogs. First step, open the XFCE `Accessibility` dialog:

```
% xfce4-accessibility-settings
```

Click on `Mouse` tab and enable the `Show location of pointer on keypress` checkbox.

Second step, open the XFCE `Keyboard` dialog:

```
% xfce4-keyboard-settings
```

Click on `Application Shortcuts` tab. Click on `+ Add` button, it opens the `Shortcut Command` dialog. Write *xfce4-find-cursor* in the form, then click on the `OK` button. Choose a key combination to bind to `xfce4-find-cursor` in the new dialog.

Chapter 5. Blindness

5.1. Introduction

This chapter describes assistive technologies for users who are blind, focusing primarily on screen readers and tools designed to work in conjunction with them. The chapter deliberately contains no images and aims to minimize the use of elements other than plain text.

5.2. Screen Reader

Currently, screen readers function only within the X Window System and are not supported in the Virtual Terminal. FreeBSD does not currently support Braille displays, so they rely exclusively on speech synthesizers.

5.2.1. Orca

Orca is a widely used and well-known open source screen reader. It offers a comprehensive set of features and is available on FreeBSD through the accessibility/orca package.

Installation

To install it, execute:

```
# pkg install orca
```

Launch

Orca can be launched in several ways. From a terminal, type:

```
% orca
```

Some desktop environments (KDE Plasma, XFCE, GNOME) provide a keyboard shortcut to launch Orca: `Super` + `Alt` + `s`.

The Super key is usually labeled as Windows icon, Command icon, or "Super". Refer to https://en.wikipedia.org/wiki/Super_key_(keyboard_button) for more information.

Usage

Orca provides a manual page: orca(1). However, most of the documentation is available online. The main reference is the starting point in the Orca Users Guide, which includes:

- Getting Started
- Reading Documents and Web Pages

- Reviewing and Interacting with Screen Contents

The screen reader includes many keyboard commands for navigation and interaction. Refer to the Orca's Commands page for a full list.

Configuration

Orca can be configured through a graphical settings interface. To open it, run:

```
% orca -s
```

Configuration options are documented on the Orca's Preferences Dialogs page.

5.2.2. yasr

The accessibility/yasr utility is a lightweight screen reader that operates exclusively in a terminal environment. Yasr requires a speech synthesizer to work. Its default configuration file is set up to use accessibility/eflite.

To install both yasr and eflite, run:

```
# pkg install yasr eflite
```

To start the screen reader, execute:

```
% yasr
```

The global configuration file for yasr is located at /usr/local/share/yasr/yasr.conf. To create a user-specific configuration file, copy the global one to your home directory:

```
% cp /usr/local/share/yasr/yasr.conf ~/.yasr.conf
```

For details on configuration options and keyboard shortcuts, refer to yasr(1).

5.3. Speech Synthesizer

The purpose of the speech synthesizer is to provide audio output for screen readers by converting on-screen text to speech. This section explains how to install speech synthesizers, perform a basic functionality test, and locate their documentation for further configuration.

Information about the FreeBSD audio subsystem can be found in Multimedia and in the sound(4) manual page.

5.3.1. eSpeak NG

The audio/espeak-ng package provides espeak-ng(1), a multilingual speech synthesizer.

To install, run:

```
# pkg install espeak-ng
```

To test, execute the following command:

```
% espeak-ng "Hello World, FreeBSD!"
```

eSpeak NG is a versatile and feature-rich utility. For more information, consult its manual page, espeak-ng(1), and the official online documentation.

5.3.2. Flite

audio/flite is a speech synthesizer that is part of the FestVox project. It is designed as a small and fast alternative to Festival, built using the FestVox suite. To install, run:

```
# pkg install flite
```

To test, execute the following commands:

```
% flite "Hello world, BSD!"
% flite_time 10:30
The time is now, exactly half past ten, in the morning.
```

The project does not provide a manual page. Documentation is available in /usr/local/share/doc/flite/README.md and online.

5.3.3. Festival

The audio/festival package provides festival(1), a multilingual speech synthesizer. To install, run:

```
# pkg install festival
```

Festival needs at least one festvox-*voice* package to generate a synthetic voice. Run the following command to list available voice packages:

```
% pkg search festvox
```

Then install a voice package. For example, to install the American English male voice festvox-kal16-

1.4.0, run:

```
# pkg install festvox-kal16
```

To test, execute:

```
% echo "Hello world, BSD!" | festival --tts
```

In the case of the error `Can't access NAS server`, example:

```
% echo "Hello world, BSD!" | festival --tts
Can't access NAS server %
```

Add the following line to /usr/local/share/festival/lib/siteinit.scm:

```
(Parameter.set 'Audio_Method 'freebsd16audio)
```

For more information about Festival, consult its manual page, festival(1), and the official online documentation.

5.3.4. Speech Dispatcher

The Speech Dispatcher project provides a high-level device independent layer for access to speech synthesis. To install accessibility/speech-dispatcher, run:

```
# pkg install speech-dispatcher
```

To test, execute:

```
% spd-say "Hello world, FreeBSD!"
```

Speech Dispatcher provides two manual pages, spd-say(1) and speech-dispatcher(1), as well as online documentation.

The audio/festival-freebsoft-utils package provides additional features for Festival to interact with Speech Dispatcher. Refer to the online documentation to know more.

5.4. Ports Collection

FreeBSD provides the Ports Collection, which offers a simple way to install applications. Some ports allow you to configure options before building and installation. By default, configuration is handled

through a TUI menu not accessible by a screen reader.

The ports-mgmt/portoptscli utility is a text-only tool specifically designed to configure ports using a screen reader. To install it, run:

```
# pkg install portoptscli
```

Then add to /etc/make.conf:

```
DIALOG4PORTS=/usr/local/bin/portoptscli
```

From now on, the Ports framework will automatically invoke portoptscli whenever a port offers configurable options.

For detailed information about its features and usage, refer to the manual page portoptscli(1) and consult the online README.

5.5. Editor

5.5.1. ed

The ed(1) utility is a line-oriented text editor used to create, view, and modify text files. It is included by default in a standard FreeBSD installation. To start the editor, run:

```
% ed
```

ed is a powerful editor. Refer to its manual page ed(1) for a complete overview of its features and command syntax.

ed does not support multibyte characters and may not handle non-ASCII text correctly. Alternatively, consider using edbrowse, which offers robust text editing features in addition to its browsing capabilities.

5.6. Internet and WEB

5.6.1. Edbrowse

Edbrowse is a text-based browser, editor, IRC chat client, and mail client, specifically designed for screen reader users. Its interface is similar to ed(1), but with extended functionality.

Installation

To install www/edbrowse, execute:

```
# pkg install edbrowse
```

Usage

To launch Edbrowse, run:

```
% edbrowse
```

The first time you run edbrowse, it will not find the configuration file ~/.ebrc. It will automatically create a default configuration file, display a message, and exit:

```
% edbrowse
Your edbrowse config file is missing; a default file has been created for you.
Before running edbrowse again, take the time to personalize your config file:
/home/yournickname/.ebrc
edbrowse -c to edit
%
```

You can now relaunch Edbrowse; it will use the default configuration file. To customize it, either run:

```
% edbrowse -c
```

Or use your preferred text editor, for example:

```
% ed ~/.ebrc
```

Example configuration files are provided in /usr/local/share/doc/edbrowse/. These include:

- sample.ebrc: with comments in English
- sample_fr.ebrc: with comments in French
- sample_it.ebrc: with comments in Italian

Documentation

The utility provides a manual page: edbrowse(1). However, the main documentation is available online in the User's Guide.

The package also installs a local copy of the documentation at /usr/local/share/doc/edbrowse/usersguide.html. To view it using edbrowse, run:

```
% edbrowse /usr/local/share/doc/edbrowse/usersguide.html
```

Translated versions of the user guide are available in the same directory:

- usersguide_es.html: Spanish
- usersguide_fr.html: French
- usersguide_pt_br.html: Brazilian Portuguese

5.6.2. Web Apps

Some web applications can use APIs to send output to audio devices. Refer to Speech Synthesizer to install a utility and the corresponding library.

The Firefox browser can report the error message: `You can't use speech synthesis because the Speech Dispatcher library is missing` or other errors. To solve, install Speech Dispatcher.

5.7. Liblouis

devel/liblouis is a translator for many languages to and from Braille. It provides a large number of conversion tables. To install, execute:

```
# pkg install liblouis
```

The package installs both libraries and utilities. The translation utility is lou_translate(1).

The following example demonstrates how to transcribe the text `FreeBSD Accessibility Handbook` into Braille with English Grade 2 translation table and the unicode.dis display table. The final output is Braille encoded as Unicode dot patterns.

```
% echo "FreeBSD Accessibility Handbook" | lou_translate -f unicode.dis,en-us-g2.ctb
⠠⠋⠗⠑⠑⠃⠎⠙⠀⠠⠁⠉⠉⠑⠎⠎⠊⠃⠊⠇⠊⠞⠽⠀⠠⠓⠁⠝⠙⠃⠕⠕⠅
```

Documentation and additional examples for this utility are available on the project website, specifically in the Testing Translation Tables interactively section of the official documentation.

Chapter 6. Development

6.1. Introduction

This chapter provides resources for developers seeking to make their programs accessible, test accessibility, and improve the accessibility of their software.

6.2. Colors

Do not assume that users can distinguish all (or even any of) the colors used in your program. While official statistics on the prevalence of color blindness are lacking, it is estimated that approximately 5% of the population is affected by some form of it, whether partial or total.

6.2.1. Colors Guidelines

Avoid using color as the sole means of conveying information. If color must be used, provide a straightforward and well-documented method for configuring color settings. Where possible, ensure support for the `$NO_COLOR` environment variable to accommodate user preferences.

6.2.2. Colors Checker

The graphics/kontrast package provides the Kontrast GUI utility to check if the color combinations are distinct enough to be readable and accessible. To install:

```
# pkg install kontrast
```

6.3. Command Line Interface

The CLI is generally considered accessible by a screen reader. However, it is important to understand some concepts, follow best practices, and avoid certain mistakes to ensure an accessible UI.

Dahlke, Karl. *Command Line Programs for the Blind*. Available at: http://www.eklhad.net/philosophy.html. This article discusses the concept of presenting information in one dimension and addresses the accessibility challenges inherent in programs designed exclusively for sighted users.

Sampath, H., Merrick, A., & Macvean, A. (2021). Accessibility of Command Line Interfaces. In Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '21). ACM. Available at: https://dl.acm.org/doi/fullHtml/10.1145/3411764.3445544. This paper presents guidelines and common pitfalls to avoid when designing command-line interfaces that are accessible to screen reader users.

6.4. Libraries

6.4.1. Gnome

AT-SPI (Assistive Technology Service Provider Interface), accessibility/at-spi2-core, is a framework that establishes a communication channel between applications and assistive technologies. To install, execute:

```
# pkg install at-spi2-core
```

AT-SPI offers a wide range of features. For more information, refer to the following resources: README, Atk wiki, AT SPI Wiki, and GNOME Accessibility Team Wiki.

Some wrappers are available: accessibility/rubygem-atk (Ruby), accessibility/py-atspi (Python), and accessibility/atkmm (C++).

6.4.2. QT

accessibility/libqaccessibilityclient establishes a communication channel between applications and assistive technologies. In particular, it provides a bridge between applications built with Qt and the AT-SPI framework previously described. For full documentation, refer to the API Reference and the official repository. To install:

```
# pkg install libqaccessibilityclient
```

Libraries accessibility/qt5-speech and accessibility/qt6-speech provide access to a Text-To-Speech functionality (specifically to accessibility/speech-dispatcher). Documentation is available at https://doc.qt.io/archives/qt-5.15/qtspeech-index.html and https://doc.qt.io/qt-6/qttexttospeech-index.html. To install, run:

```
# pkg install qt5-speech
# pkg install qt6-speech
```

6.4.3. Speech Synthesis

accessibility/speech-dispatcher provides a simple, high level, and device independent layer for access to Speech Synthesizers. To install, run:

```
# pkg install speech-dispatcher
```

For detailed documentation, refer to the official Speech Dispatcher website: https://freebsoft.org/speechd.

A Python wrapper for the library is provided by the accessibility/py-speech-dispatcher.

6.4.4. Liblouis

devel/liblouis provides an API for translating many languages to and from Braille. To install, execute:

```
# pkg install liblouis
```

The library provides many features, documented on the liblouis project website.

Liblouis provides also a Python API, Python bindings.

6.5. Tools

accessibility/accerciser provides an accessibility inspection tool for applications developed for the GNOME desktop environment. To install it, run:

```
# pkg install accerciser
```

For detailed usage and features, refer to accerciser(1). Additional documentation is available in the README.md and the Accerciser User Guide.