Darkman for Emacs

Taha Aziz Ben Ali

March 17, 2024

Contents

This is the manual for `darkman.el` version 1.1.0.

- Homepage: `https://darkman.grtcdr.tn`
- Repository: `https://git.sr.ht/~grtcdr/darkman.el`
- Mailing list: `https://lists.sr.ht/~grtcdr/pub`
- Issue tracker: `https://todo.sr.ht/~grtcdr/darkman.el`

1 GNU Free Documentation License

2 Installation

This package is available from MELPA provided you've added that to the list of package archives to fetch from, install it by evaluating the following:

```
(package-install 'darkman)
```

For a manual installation, begin by cloning the repository:

```
git clone --branch 1.0.3 https://github.com/grtcdr/darkman.el darkman
```

Next, add the package to your load path:

```
(add-to-list 'load-path "darkman")
```

Finally, require the package like so:

```
(require 'darkman)
```

3 Usage

In your `init.el` file, define which theme is associated with which mode. By default, `modus-themes` will be used, but we'll choose the `tango` variants in this example.

```
(setq darkman-themes '(:light tango :dark tango-dark))
```

You can also use `M-x customize-group darkman RET` to customize the variable.

`darkman` can listen for any signals to change the theme without requiring you to manually create any scripts. You can enable this behavior by enabling the minor mode:

```
(darkman-mode)
```

You should not call `load-theme` in your configuration as `darkman-mode` when enabled will do that for you, even on startup.

3.1 Other functions

`M-x darkman-toggle` will toggle the mode of the Darkman service from within Emacs.

Additionally, you can use the `darkman-get` and `darkman-set` functions to manually get and set the mode of the service respectively.

4 Tips

4.1 Working alongside the built-in safety features

`load-theme` is part of the `custom` library and it comes with certain security features that you may regard as inconvenient, but they're absolutely necessary to preserve the security of your system. We have decided to work hand in hand with these pre-existing safety mechanisms in lieu of implementing our own.

Problems may arise [1] if this package is loaded in a server setting and too early during the initialization phase, i.e. before `custom-file`, you can prevent this from happening by activating the mode **only** when the session has been fully initialized:

```
(when (daemonp)
  (add-hook 'server-after-make-frame-hook #'darkman-mode)
  (advice-add 'darkman-mode
              :after
              (lambda ()
                (remove-hook 'server-after-make-frame-hook
                             #'darkman-mode))))
```

[1] This is known to happen in Doom Emacs.

4.2 Working alongside the third-party `spacious-padding` package

`spacious-padding` significantly improves the aesthetics of our beloved Emacs by padding the windows and mode line. I've personally encountered an issue that made `spacious-padding` behave incorrectly when (1) used in conjuction with `darkman-mode` and (2) when Emacs is started as a daemon, resulting in window/mode line borders being assigned the wrong color.

I found, through trial and error, that simply delaying the activation of `spacious-padding` when started in daemon mode until the first frame has appeared solved the problem.

```
(if (daemonp)
    (add-hook 'server-after-make-frame-hook #'spacious-padding-mode)
  (spacious-padding-mode))
```

4.3 Disabling existing themes

Emacs does not by default disable the current theme when another one is loaded, and while this behavior might seem counter-intuitive, we can always advise `darkman-set` (and by extension `darkman-toggle`), or even `load-theme` if you prefer, to disable any existing themes first.

```
(defadvice darkman-set (before no-theme-stacking activate)
  "Disable the previous theme before loading a new one."
  (mapc #'disable-theme custom-enabled-themes))
```

5 Debugging

5.1 Is D-Bus running?

Use your service manager to verify whether the `darkman` service is running, here's an example using `systemd`:

```
systemctl status dbus
```

```
dbus.service - D-Bus System Message Bus
   Loaded: loaded
   Active: active (running)
```

5.2 Is Darkman running?

Use your service manager to verify whether the `darkman` service is running, here's an example using `systemd`:

```
systemctl status --user darkman
```

```
darkman.service - Framework for dark-mode and light-mode transitions.
   Loaded: loaded
   Active: active (running)
```

5.3 Is Emacs built with D-Bus support?

Usually it is, unless you're building from source, in which case you can verify whether or not Emacs was built with D-Bus support using `C-h v system-configuration-` RET which should list DBUS.