NAME

blight – GUI frontend to busylight

SYNOPSIS

[**wish8.6**] **blight.tcl**

DESCRIPTION

The **blight.tcl** script is a simple GUI front-end to the **busylight**(1) program. It provides an on-screen indication of the light status (helpful if you can't physically see the LEDs) and a row of buttons to click on to easily change the light status.

All of its operations are carried out by running the **busylight** program to do the work.

Every 5 minutes, it will query the state of the lights and update its on-screen indicator accordingly, but it will also do this when you manually change the status or click the **refresh** button.

Buttons are provided for each of the defined status values in your **~/.busylight/config.json** file (except the **start** and **stop** statuses), as well as a series of buttons labelled **server mute**, **server open**, etc., which allow you to set the daemon's state.

ACTIVITIES

Blight adds another feature not otherwise provided by **busylight**. It allows for activity tracking. To use this feature, you will need to create a file called **~/.busylight/activities.json** which contains a JSON array of objects, each of which has the following fields:

Name This should be a short (preferably one-word) name of the activity to be tracked.

Status This is a JSON array of string values. Each value is a set of arguments to **busylight** to set the lights appropriately for this activity. The words in the string value will be added to the command line; if more than one string is given, **busylight** will be invoked once for each string.

Elapsed An integer value giving the number of minutes spent so far on that activity.

An example **activities.json** file might look like this:

```
[
 {"Name":"Meetings", "Status":["-open"], "Elapsed":0},
 {"Name":"Idle", "Status":["-status idle"], "Elapsed":12},
 {"Name":"Games", "Status":["-status busy","-status lowpri"],
  "Elapsed":120}
]
```

The **blight** script will update this file with new **Elapsed** values while activities are active.

To start an activity, click on its button. That will set the lights for the activity and start the timer to track time for that activity. Clicking other buttons will change lights but not change the running timer. Clicking another activity will stop the current one before starting the new activity. Clicking the **(stop activity)** button will stop the activity timer without starting a new one.

AUTHOR

Steve Willoughby <*steve@madscience.zone*>