

Tint2, une barre de tâches

- Objet : Tint2 - Installation.
- Niveau requis :
[débutant](#), [avisé](#)
- Commentaires : *La barre de tâche très légère.*
- Débutant, à savoir : [Utiliser GNU/Linux en ligne de commande, tout commence là !](#) 😊
- Suivi :
[à-tester](#)
 - Création par [titia](#) le 06/05/2011.
 - Testé par le
- Commentaires sur le forum : [C'est ici](#)¹⁾

Présentation

Tint2 se présente sous forme d'une barre de tâches très légère et facile à configurer. Il inclut une zone de notification (systray). Son installation et sa configuration sont faciles à réaliser en éditant son fichier "tint2rc". Cependant, un utilitaire graphique permet de le faire [tintwizard](#).

Installation

Rien de plus facile que de l'installer avec votre gestionnaire de paquets habituel

```
apt-get update && apt-get install tint2
```

Voici la liste de ses dépendances :

- Dépend: libatk1.0-0
- Dépend: libc6
- Dépend: libcairo2
- Dépend: libfontconfig1
- Dépend: libfreetype6
- Dépend: libglib2.0-0
- Dépend: libgtk2.0-0
- Dépend: libimlib2
- Dépend: libpango1.0-0
- Dépend: libx11-6
- Dépend: libxcomposite1
- Dépend: libxdamage1
- Dépend: libxfixes3
- Dépend: libxinerama1
- Dépend: libxrandr2
- Dépend: libxrender1

- Recommande: python
- Recommande: python-gtk2

Vous pouvez lancer en console par

```
tint2 &
```

Le plus simple étant de le lancer au démarrage. Il suffit alors d'ajouter une ligne "tint2" dans les programmes au démarrage.

Lancement au démarrage

Openbox

Dans le fichier ".config/openbox/autostart.sh", on ajoute :

```
tint2 &
```

Gnome

Dans Système > Préférences > Applications au démarrage, vous cliquez sur le bouton «ajouter». Une fenêtre s'affiche. Dans le champ «Nom» : Tint, dans le champ «Commande» : "/usr/bin/tint2", dans le champ «Commentaire» : lancement de tint2.

Configuration

Tint2 est très simple à configurer, cela se fait dans un seul et unique fichier "./config/tint2/tint2rc"

la partie «background and border»

```
#-----  
# BACKGROUND AND BORDER  
#-----  
rounded = 7  
border_width = 2  
background_color = #ff0080 60  
border_color = #ff0000 80
```

```
rounded = 10  
border_width = 1  
background_color = #000000 30  
border_color = #ffffff 60
```

```
rounded = 10
```

```
border_width = 1
background_color = #000000 10
border_color = #939393 40
```

Dans cette partie, on va présenter les différentes décorations “background_id” que l'on veut. On peut en mettre autant que l'on veut. Le premier sera identifié par “1”, le deuxième “2”. “0” donne la transparence totale. Étonnant, non?

“rounded” sert à donner la forme arrondie. “0” donnera un coin carré. On renseigne ensuite la largeur du contour avec “border_width”, la couleur du fond et la couleur de la bordure. Dans ces deux cas, le chiffre qui suit la couleur permet la transparence, “0” pour entièrement transparente et “100” pour l'opacité totale.

la partie « panel »

```
#-----
# PANEL
#-----
panel_monitor = all
panel_position = bottom center
panel_size = 94% 30
panel_margin = 0 0
panel_padding = 7 0
font_shadow = 0
panel_background_id = 1
wm_menu = 0
```

Le “panel_monitor” définit sur quel écran apparaît tint2, “all” permet de l'avoir sur chaque écran.



Attention, je n'ai pas dit « bureau » mais écran. On parle de « moniteur ».

On renseigne ensuite sa position par “panel_position”. “bottom center” est un exemple pour avoir la barre centrée en bas. “panel_position” renseigne la taille. Elle se compose d'un pourcentage par rapport à la largeur du bureau et on lui affecte une hauteur.

On fixe ensuite les marges intérieures et extérieures dans “panel_margin” et “panel_padding”. On peut ombrer le texte “font_shadow” avec “1” pour oui, “0” pour non. Booléen, non !

On déclare quel “background” il utilise dans “panel_background_id”. Avec “wm_menu”, on autorise ou non le suivi de la souris dans la wm. Vous pourrez, par exemple, avoir le menu par clic droit sur la barre.

la partie « taskbar »

```
#-----
# TASKBAR
#-----
#taskbar_mode = multi_desktop
taskbar_mode = single_desktop
taskbar_padding = 2 3 2
```

```
taskbar_background_id = 0
```

“taskbar_mode” définit sur quel bureau vont apparaître les taches de la barre tint2. “single” permet d'avoir les taches du bureau concerné. “multi” va lister dans la barre toutes les tâches lancées sur tous les bureaux.

“taskbar_padding” permet de définir les marges des fenêtres des applications dans la barre des tâches.

“taskbar_background_id” nous permet de choisir quelle décoration appliquer.

la partie « tasks »

```
#-----  
# TASKS  
#-----  
task_icon = 1  
task_text = 1  
task_width = 140  
task_centered = 1  
task_padding = 6 3  
task_font = Gentium 10  
task_font_color = #939393 70  
task_active_font_color = #ffffff 85  
task_background_id = 3  
task_active_background_id = 2
```

On va renseigner l'allure de la tâche. Rien de bien difficile à comprendre dans cette partie. Icône ou pas, texte ou pas, choix cornélien. “task_width” définit la taille de l'espace qu'aura la tâche dans la barre. Le “1” pour “task_centered” positionne au centre la décoration de la tâche. On indique ensuite les marges puis la police d'écriture.

“task_font_color” et “task_active_font_color” allouent respectivement la couleur pour les tâches inactives et actives.

“task_background_id” et “task_active_background_id” allouent respectivement la décoration aux tâches inactives et actives.

la partie « systraybar »

```
#-----  
# SYSTRAYBAR  
#-----  
systray_padding = 0 4 5  
systray_background_id = 0
```

Cette partie concerne la zone de notification. C'est là que les applications qui tournent vont s'icônifier. Dans un premier temps, on définit les marges. Ensuite, on applique une décoration.

la partie « clock »

```
#-----  
# CLOCK  
#-----  
time1_format = %H:%M  
time1_font = Gentium 10  
time2_format = %A %d %B  
time2_font = Gentium 9  
clock_font_color = #9a0f2d 100  
clock_padding = 1 0  
clock_background_id = 0
```

Nous allons renseigner quel format d'affichage de l'heure nous voulons. Pour cela, je vous renvoie aux manpages de “date”. Dans cet exemple, les heures et minutes s'affichent au-dessus du jour, numéro du jour et mois.

On choisit ensuite la couleur et l'opacité avec “clock_font_color”. Pour “padding” et “background_id”, vous êtes rodés, je pense. ;)

la partie « battery »

```
#-----  
# BATTERY  
#-----  
battery = 1  
battery_low_status = 10  
battery_low_cmd = notify-send "batterie faible"  
bat1_font = Gentium 9  
bat2_font = Gentium 8  
battery_font_color = #ffffff 76  
battery_padding = 1 0  
battery_background_id = 0
```

Cette partie n'est pas utile sur une tour.

“battery” positionnée sur “1” permet l'affichage de l'état de charge.

“battery_low_status” règle le seuil d'alerte de la batterie qui envoie le message “batterie faible grâce à “battery_low_cmd”.

On choisit ensuite la police d'écriture par “bat1_font”, la couleur, les marges et la décoration.

la partie « mouse on task »

```
#-----  
# MOUSE ACTION ON TASK  
#-----  
mouse_middle = none  
mouse_right = close  
mouse_scroll_up = toggle
```

```
mouse_scroll_down = iconify
```

Un petit passage en revue de l'effet de la souris sur les tâches. “mouse_middle”, “mouse_righ”, “mouse_scroll_up”, “mouse_scroll_down” représentent respectivement le clic sur la molette du milieu, le clic droit, la molette montante, la molette descendante avec les actions suivantes : aucun effet, fermer, agrandir, icônifier.

On atteint ici les limites de “tint2”. Pas de possibilité d'avoir un menu en cliquant droit sur la tâche. Ça viendra peut-être, patience.

la partie « tooltip »

```
#-----  
# TOOLTIP  
#-----  
tooltip = 1  
tooltip_padding = 2 2  
tooltip_show_timeout = 0.7  
tooltip_hide_timeout = 0.3  
tooltip_background_id = 1  
tooltip_font_color = #000000 80  
tooltip_font = Gentium 10
```

Cette partie concerne les infos bulles des tâches.

“tooltip_show_timeout” détermine le temps que l'info bulle va mettre à apparaître.

“tooltip_hide_timeout” détermine le temps que l'info bulle va mettre à disparaître.

la partie « autohide options »

```
#-----  
# AUTOHIDE OPTIONS  
#-----  
autohide = 1  
autohide_show_timeout = 0.3  
autohide_hide_timeout = 2  
autohide_height = 4  
strut_policy = minimum
```

On a la possibilité de masquer cette barre en donnant “1” à “autohide”.

On renseigne ensuite les temps d'affichage et de masquage.

“autohide_height” permet de voir une petite partie de la barre afin de pouvoir pointer dessus.



Le masquage ne fonctionne pas bien chez moi.

Prendre en compte les modifications

Si vous voulez voir les modifications que vous avez apportées, vous pouvez relancer “tint2” en console

```
killall tint2 && tint2 &
```

Exemples



Cet exemple est très basique. Vous remarquerez que “taskbar_background_id = 0”. La largeur est de 94 % du bureau.



La barre est de 50 % du bureau, pour sa décoration “rounded = 0”. Le programme “vlc” est présent dans la zone de notification. On voit l’affichage de l’état de la batterie, actuellement de “100 %”. Le format de l’heure est différent de celui de l’autre exemple.



La barre prend toute la largeur du bureau. Elle est opaque et les tâches sont dans des fenêtres rectangulaires.

Liens

Liens utiles vers :

- [Projet Tint2](#)
- [tintwizard](#)
- [Pour lenny voir sur le forum là](#)

1)

N'hésitez pas à y faire part de vos remarques, succès, améliorations ou échecs !

From:

<http://debian-facile.org/> - **Documentation - Wiki**

Permanent link:

<http://debian-facile.org/doc:environnements:x11:tint2>

Last update: **27/10/2016 15:33**

