



# 1 Présentation

---

## 2 Fonctions

### 2.1 Création de base d'un header

#### Fonction : **DrawLine**

##### Description :

Cette fonction dessine une ligne en remplissant chaque colonne du terminal avec un caractère choisi et passé en second argument.

##### Paramètres :

**p\_lineColor** -> Ce paramètre attend un code couleur \*.

**p\_lineChar** -> Ce paramètre attend un seul caractère, qui sera affiché sur chaque colonne.

\* Chaque code couleur est défini dans le fichier **Colors.conf**.

##### Fonctionnement :

La fonction **DrawLine** vérifie d'abord qu'une valeur soit passée en premier argument (paramètre **p\_lineColor**). Si c'est le cas, la commande **echo** est appelée avec ses options **-ne** pour afficher.

##### Utilisation :

Appelez cette fonction pour dessiner une ligne devant remplir la totalité des colonnes d'une ligne du terminal.

---

#### Fonction : **HeaderBase**

##### Description :

Cette fonction affiche un header complet.

##### Paramètres :

**p\_lineColor** -> Ce paramètre attend un code couleur.

**p\_lineChar** -> Ce paramètre attend un seul caractère.

**p\_stringColor** -> Ce paramètre attend un code couleur.

**p\_stringTxt** -> Ce paramètre attend une chaîne de caractères.

##### Fonctionnement :

Tout d'abord, la fonction **HeaderBase** vérifie l'état de la variable de statut **\_\_BU\_MAIN\_STAT\_ECHO**, vérifiant si un texte peut être redirigé vers un fichier de logs sans provoquer de boucles infinies en appelant certaines fonctions.

En adaptant la commande d'affichage de texte à appeler, cette fonction appelle la fonction **DrawLine** en y passant en arguments ses deux premiers paramètres.

Une fois ceci fait, un saut de ligne est effectué, et un message est affiché, avec la couleur choisie encodée en premier (troisième paramètre) et le texte passé en quatrième argument.

La commande de ré-initialisation de décoration du terminal (définie dans le fichier) est ensuite appelée (elle est définie dans les fichiers **Colors.conf** et **Text.conf**) pour réinitialiser la couleur d'affichage de texte selon la configuration du terminal.

#### Utilisation :

Appelez cette fonction pour afficher un texte en majuscules entre deux lignes, et pour davantage personnaliser cet affichage si vous ne trouvez pas votre compte parmi les fonctions des sections **Headers unicolores** et **Headers multicolores**.

---

## 2.2 Headers unicolores

#### Description :

Chacune des fonctions de cette catégorie appellent la fonction **HeaderBase**, en lui passant en premier et troisième arguments une même couleur (définie dans le fichier **Colors.conf**), un caractère défini par la variable **\$\_BU\_MAIN\_TXT\_PRINT\_HEADER\_LINE\_CHAR** en deuxième argument, puis la chaîne de caractères à afficher en quatrième argument (seul paramètre que prennent ces fonctions).

#### Paramètres :

**\$1** -> Ce paramètre attend la chaîne de caractère à afficher.

---

## 2.3 Headers bicolores

Pour faciliter davantage la tâche des développeurs, des fonctions de création de headers bicolores existent.

Elles fonctionnent chacune de la même manière que les précédentes, à deux exceptions prêtes :

- le nom : chaque fonction porte le nom de la couleur à afficher sur chaque ligne, puis le nom de la couleur à appliquer au texte.
  - le nom (bis) : Il n'y a pas de fonctions portant deux fois le nom de la même couleur, car le rendu serait le même que celui des fonctions de la catégorie précédente.
- 
- 

## 2.4 Headers tricolores

Étant donné la charge de travail nécessaire pour créer ces headers, il serait beaucoup trop fastidieux de créer tous ces headers prédéfinis.

Pour vous donner une idée, prenez la formule mathématique suivante :

$$nbcol * (nbcol - 1) * nbcol$$

Où nbcol = nombre de codes couleurs disponibles

Au moment où j'ai écrit cette partie de la documentation, il existait 12 codes couleurs disponibles. En effectuant le calcul précédent, nous obtenons **12 \* 11 \* 12 = 1 584 combinaisons possibles**, sans parler de l'ajout d'éventuels nouveaux codes couleurs.