Fonctions du fichier de librairie Headers.lib

Dimitri OBEID

2021

Table des matières

1 Présentation

1.1 Description

Ce fichier source inclut des fonctions servant à la création de headers, séparant ce qu'un développeur de scripts Bash peut imaginer comme étant deux grandes étapes différentes d'un script.

1.2 Définitions des éléments mentionnés

1.2.1 Chemins des fichiers

Dossier parent (dossier des configurations)	Nom du fichier
$__BU_MODULE_UTILS_ROOT/config/modules/main	Colors.conf
$__BU_MODULE_UTILS_ROOT/config/modules/main	Status.conf
$__BU_MODULE_UTILS_ROOT/config/modules/main	Text.conf

Dossier parent (dossier de la librairie)	Nom du fichier
$__BU_MAIN_ROOT_DIR_PATH/lib/functions/main	Checkings.lib
$__BU_MAIN_ROOT_DIR_PATH/lib/functions/main	Decho.lib
$__BU_MAIN_ROOT_DIR_PATH/lib/functions/main	Echo.lib
$__BU_MAIN_ROOT_DIR_PATH/lib/functions/main	Errors.lib

1.2.2 Fonctions externes appelées

Fonction	Fichier de définition
BU : :IsAlphaNumChar	Checkings.lib
BU : :IsInt	Checkings.lib
BU : :IsIntpos	Checkings.lib
BU : :DechoBGColor	Decho.lib
BU : :DechoHighlightFunction	Decho.lib
BU : :EchoMsg	Echo.lib
BU : :Newline	Echo.lib
BU : :Main : :Errors : :HandleSmallErrors	Errors.lib

1.2.3 Variables globales externes et / ou variables d'environnement appelées

Variable : $__BU_MAIN_COLOR_BG_...

Description :

Chacune des variables globales de couleur de fond commençant par ce nom retourne le résultat de la fonction BU : :Main : :ModConfig : :Colors : :SetBGColor lors de l'appel d'une de ces variables, pour colorer l'arrière plan du texte selon le code couleur.

Fichier de définition : Colors.conf

1.2.4 Variables globales externes et / ou variables d'environnement appelées

Variable : $__BU_MAIN_COLOR_TXT_...

Description :

Chacune des variables globales de couleur de texte commençant par ce nom retourne le résultat de la fonction **BU : :Main : :ModConfig : :Colors : :SetTextColor** lors de l'appel d'une de ces variables, pour colorer le texte selon le code couleur.

Fichier de définition : Colors.conf

Variable : $__BU_MAIN_STAT_ECHO

Description :

Fichier de définition : Status.conf

Variable : $__BU_MAIN_TXT_CHAR_HEADER_LINE

Description :

Cette variable globale enregistre le caractère par défaut à afficher sur chaque colonne d'une ligne d'un header.

Fichier de définition : Text.conf

---

2 Fonctions

2.1 Création de base d'un header

2.1.1 BU : :Main : :Headers : :DrawLine

Fonction : BU : :Main : :Headers : :DrawLine

Description :

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

Paramètres :

Paramètre	Type
p_lineChar	Caractère
p_lineColor	Substitution de commande
p_strlen	Nombre entier
p_bgColor	Substitution de commande

Fonctionnement :

La fonction **BU : :Main : :Headers : :DrawLine** vérifie tout d'abord que la valeur du second paramètre

p_lineChar soit bien un simple caractère (alphabétique ou numérique) via la fonction **IsChar**, définie dans le fichier **Checkings.lib**. Si ce n'est pas le cas, ce paramètre est redéfini, et seul le premier caractère de la chaîne est récupéré pour l'affichage de la ligne.

Si une substitution de commande est passée au paramètre **p_lineColor**, **alors** la fonction actuelle l'exécute. Cette substitution **DOIT** être un appel de la fonction **BU : :Main : :ModConfig : :Colors : :SetTextColor**.

Une fois la commande exécutée pendant la comparaison, la commande **echo** est appelée avec ses options **-ne** pour encoder la couleur de la ligne de caractère à dessiner.

Fin de la condition.

Une fois la vérification de coloration de la ligne effectuée, la fonction **en fait de même avec la couleur de fond**, dont la valeur a été passée au paramètre **p_bgColor** via une substitution de commande, qui **DOIT** être un appel de la la fonction **BU : :Main : :ModConfig : :Colors : :SetBGColor**.

Une fois les deux condition précédentes vérifiées, la fonction vérifie si une valeur est passée au paramètre **p_strlen**, pour créer une ligne d'une longueur définie.

Si une valeur est passée au paramètre **p_strlen**, **alors** :

- **Si** cette valeur est un nombre entier, **alors** une boucle **for** est exécutée pour afficher **n** caractères selon le nombre de colonnes défini via ce paramètre.
  - **Pour** un nombre **i** allant de **1** au nombre enregistré dans le paramètre **p_strlen**, on affiche **i** fois, sans retour à la ligne, le caractère passé en deuxième argument (**$p_lineChar**).

    Fin de la boucle « pour ».

  Sinon, si cette valeur n'est pas un nombre entier, alors un message d'avertissement est affiché via un appel de la fonction **BU : :Main : :Errors : :HandleSmallErrors**.

  Fin de la condition « si ».

Sinon, si aucune valeur n'est passée au paramètre **p_strlen**, **alors** une boucle **for** est exécutée pour afficher **n** caractères selon la longueur du mode texte.

- **Pour** un nombre **i** allant de **1** au nombre de colonnes présentes dans le mode texte du terminal (nombre obtenu grâce à la commande **eval echo**), on affiche **i** fois, sans retour à la ligne, le caractère passé en deuxième argument (**$p_lineChar**).

  Fin de la boucle « pour ».

Fin de la condition « si ».

Enfin, une fois l'une des deux boucles exécutée, **si** les paramètres **p_lineColor OU p_bgColor** contiennent une valeur, **alors** la fonction ré-encode la couleur d'affichage du texte **ET** celle de coloration du fond, selon la couleur par défaut du terminal. **Fin de la condition « si »**.

Utilisation :

Appelez cette fonction pour dessiner une ligne devant remplir la totalité des colonnes d'une ligne du terminal **ou** une ligne au nombre de colonnes bien défini.

2.1.2 BU : :HeaderBase

Fonction : BU : :HeaderBase

Description :

Cette fonction affiche un header complet, avec ses deux lignes entourant une chaîne de caractères.

Paramètres :

p_lineColor	Substitution de commande
p_lineChar	Caractère
p_stringColor	Substitution de commande
p_stringTxt	Chaîne de caractères
p_strlen	Chaîne de caractères « strlen »
p_bgColorCol	Substitution de commande
p_bgColorPos	Nombre entier « 1 », « 2 » et / ou « 3 »

Variables :

v_stat_time_newline	Nombre décimal
v_stat_time_txt	Nombre décimal
v_strlen	Nombre entier

Fonctionnement :

Tout d'abord, sachant que le paramètre **p_strlen** attend précisément la chaîne de caractères **strlen**, la fonction se doit de récupérer la longueur de la chaîne de caractères principale précédemment passée en argument (paramètre **p_stringTxt**).

Il en est ainsi pour éviter que l'utilisateur n'ait à retaper la même chaîne de caractères ou à définir une nouvelle variable pour afficher cette dernière, puis récupérer son nombre de caractères via un enchaînement de processus avec un tube (**echo "$var" | wc -c**) (méthode utilisée par la fonction actuelle) ou de cette manière : **echo "${#var}"**.

Dans un soucis de temps de pause cohérent, la fonction **BU : :HeaderBase** change la valeur des variables globales de statut **$__BU_MAIN_STAT_TIME_NEWLINE** et **$__BU_MAIN_STAT_TIME_TXT**, mettant ainsi le temps de pause du script de ces variable à 0 secondes.

Pour laisser le choix aux développeurs quant à la ligne à colorer (première ou deuxième ligne de caractères, ou la chaîne de caractères du milieu), le paramètre **p_bgColorPos** accepte un nombre comprenant entre un et chiffres, correspondants à la ligne à colorer, et dont les valeurs acceptées sont **1**, **2** et / ou **3**.

Si aucune valeur n'est passée au paramètre **p_bgColorPos** lors de l'appel de la fonction, **alors** cette dernière appelle la fonction **BU : :Main : :Headers : :DrawLine** en y passant en arguments la valeur des paramètres et variables **p_lineChar**, **p_lineColor**, **v_strlen** et **p_bgColor** (ces deux derniers arguments seront traités par la fonction appelée s'ils contiennent une valeur).

Sinon, si une valeur est passée au paramètre **p_bgColorPos** lors de l'appel de la fonction, **alors** :

- Pour connaitre la valeur de chaque chiffre enregistré dans ce paramètre, Un tableau est créé et prend en valeurs chacun des chiffres.

Si un nombre entier positif est passé à ce même paramètre **ET** que ce nombre entier soit plus grand ou égal à 1 **ET** qu'il soit plus petit ou égal à 3, **alors** :

Pour un nombre **i** de valeurs dans l'intervalle autorisée :

- **Si** l'un des chiffres de ce nombre à trois chiffres est égal à **1**, **alors** la fonction **BU : :DechoBGColor** est appelée pour colorier le fond de la zone de texte où la première ligne de caractères est affichée.

  Sinon, si l'un des chiffres de ce nombre à trois chiffres est égal à **2**, **alors** la fonction **BU : :DechoBGColor** est appelée pour colorier le fond de la zone de texte où la chaîne de caractères principale est affichée.

  Sinon, si l'un des chiffres de ce nombre à trois chiffres est égal à **3**, **alors** la fonction **BU : :DechoBGColor** est appelée pour coloriee le fond de la zone de texte où la seconde ligne de caractères est affichée.

  Sinon, si l'un des chiffres de ce nombre à trois chiffres n'est pas égal à l'une des trois valeurs attendues, **Sinon, si** un message d'avertissement est affiché sur la zone de texte.

  Fin de la condition « si ».

Fin de la boucle « pour ».

- **Sinon, si** un nombre entier positif est passé à ce paramètre **ET** que sa longueur est supérieure à trois chiffres, **alors** un message d'avertissement est affiché sur la zone de texte.

  Sinon, si aucune valeur n'est passée à ce paramètre, **alors** un message d'avertissement est affiché sur la zone de texte.

  Sinon, si une valeur inattendue est passée à ce paramètre, **alors** un message d'avertissement est affiché sur la zone de texte.

  Fin de la condition « si ».

Fin de la condition « si ».

Une fois le header (ou un message d'avertissement affiché), un saut de ligne est effectué, puis le script se met en pause pendant le nombre de secondes défini par la valeur enregistrée dans la variable globale de statut **$__BU_MAIN_STAT_TIME_HEADER**.

Finalement, les valeurs des deux variables globales de statut **$__BU_MAIN_STAT_TIME_NEWLINE** et **$__BU_MAIN_STAT_TIME_TXT**, modifiées au début de la fonction, sont restaurées.

Utilisation :

Appelez cette fonction pour afficher un texte entre deux lignes, pour marquer une séparation entre plusieurs étapes de votre script, et pour davantage personnaliser cet affichage si vous ne trouvez pas votre compte parmi les fonctions proposées dans les sous-sections suivantes : **Headers unicolores** et **Headers multicolores**.

2.2 Headers unicolores

Description : Chacune de ces fonctions appelle la fonction **BU : :HeaderBase**, en lui passant ses trois premiers arguments prédéfinis.

Paramètres positionnels :

$1	Chaîne de caractères
$2	Chaîne de caractères « strlen »
$3	Substitution de commande
$4	Nombre entier « 1 », « 2 » et / ou « 3 »

Fonctionnement :

Chacune des fonctions de cette catégorie appelle la fonction **BU : :HeaderBase**, en lui passant en premier et troisième arguments une même substitution de commande (**$__BU_MAIN_COLOR...**), puis un caractère défini par la variable **$__BU_MAIN_TXT_CHAR_HEADER_LINE** en deuxième argument.

Vient enfin le passage des valeurs des paramètres positionnels : la chaîne de caractères à afficher (**$1**), la chaîne de caractères **strlen** (**$2**), la substitution de commande colorant l'arrière plan du texte (**$3**) et la position de la ligne à colorer (**$4**).

Utilisation :

Appelez une de ces fonctions pour marquer une séparation entre plusieurs étapes de votre script, tout en affichant un header unicolore sans avoir à passer ses trois premiers arguments.

2.3 Headers bicolores

Note : En raison du trop grand nombre de fonctions à déclarer, plus aucune nouvelle couleur ne sera officiellement ajoutée parmi les 23 couleurs actuelles (qui représentent en tout plus de 500 fonctions déclarées rien que dans ce fichier de librairie).

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

Chaque fonction porte le nom de la couleur à afficher sur chaque ligne, puis le nom de la couleur à appliquer au texte.

Elles fonctionnent chacune de la même manière que les fonctions de la catégorie précédente, à deux exceptions près :

- **le nom des fonctions et des variables :** Il n'y a pas de fonction portant deux fois le nom de la même couleur, ni d'appels double des mêmes variables, car le résultat et le rendu seraient le même que celui des fonctions de la sous-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 de la quantité de travail, 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 (dimanche 29 août 2021), il existait 12 codes couleurs disponibles.

En effectuant le calcul précédent, nous obtenons donc **12 * 11 * 12 = 1 584 combinaisons possibles**, sans parler de l'ajout d'éventuels nouveaux codes couleurs.

Il faudra donc se contenter d'appeler la fonction **BU : :HeaderBase**, suivie de ses 4 arguments, dont chaque valeur sera différente des autres.