

Fonctions du fichier de librairie `Echo.lib`

Dimitri OBEID

2021

Table des matières

1	Présentation	3
1.1	Présentation	3
1.2	Définitions des éléments mentionnés	3
1.2.1	Chemins de fichiers	3
1.2.2	Fonctions externes appelées	3
1.2.3	Variables globales externes et / ou variables d'environnement appelées	3
1.2.4	Variables globales externes et / ou variables d'environnement appelées	4
2	Fonctions - Section « PRINTING FORMATTED STRINGS »	4
2.1	Sous-section « DEBUGGING »	4
2.1.1	BU::Main::Echo::_EchoVoid	4
2.1.2	BU::Main::Echo::_EchoOutput	5
2.1.3	BU::Main::Echo::_EchoCPLS	7

1 Présentation

1.1 Présentation

Ce fichier source inclut des fonctions servant au traitement du texte, comprenant entre autre l’affichage, les redirections vers le fichier de logs et / ou le terminal, ainsi que la vérification de la casse.

1.2 Définitions des éléments mentionnés

1.2.1 Chemins de fichiers

Dossier parent	Nom du fichier
<code>\$_BU_MODULE_INSTALL/</code>	<code>Bash-utils-init.sh</code>

Dossier parent	Nom du fichier
<code>\$_BU_MODULE_UTILS_ROOT/config/modules/main</code>	<code>Colors.conf</code>
<code>\$_BU_MODULE_UTILS_ROOT/config/modules/main</code>	<code>Status.conf</code>
<code>\$_BU_MODULE_UTILS_ROOT/config/modules/main</code>	<code>Text.conf</code>

Dossier parent	Nom du fichier
<code>\$_BU_MAIN_ROOT_DIR_PATH/lib/functions/main</code>	<code>Checkings.lib</code>
<code>\$_BU_MAIN_ROOT_DIR_PATH/lib/functions/main</code>	<code>Decho.lib</code>
<code>\$_BU_MAIN_ROOT_DIR_PATH/lib/functions/main</code>	<code>Errors.lib</code>
<code>\$_BU_MAIN_ROOT_DIR_PATH/lib/functions/main</code>	<code>Headers.lib</code>
<code>\$_BU_MAIN_ROOT_DIR_PATH/lib/functions/main</code>	<code>Status.lib</code>

1.2.2 Fonctions externes appelées

Fonction	Fichier de définition
<code>BU : :ModuleInit : :Msg</code>	<code>Bash-utils-init.sh</code>
<code>BU : :Main : :Checkings : :CheckProjectRelatedFile</code>	<code>Checkings.lib</code>
<code>BU : :IsInt</code>	<code>Checkings.lib</code>
<code>BU : :Main : :Checkings : :CheckEcho</code>	<code>Checkings.lib</code>
<code>BU : :CheckProjectLogStatus</code>	<code>Checkings.lib</code>
<code>BU : :DechoHighlightFunction</code>	<code>Decho.lib</code>
<code>BU : :DechoHighlightVar</code>	<code>Decho.lib</code>
<code>BU : :Main : :Errors : :HandleSmallErrors</code>	<code>Errors.lib</code>
<code>BU : :HeaderError</code>	<code>Headers.lib</code>
<code>BU : :HeaderNewstep</code>	<code>Headers.lib</code>
<code>BU : :HeaderSuccess</code>	<code>Headers.lib</code>
<code>BU : :HeaderWarning</code>	<code>Headers.lib</code>
<code>BU : :Main : :Status : :CheckStatIsInitializing</code>	<code>Status.lib</code>
<code>BU : :Main : :Status : :CheckStatIsBackupEchoing</code>	<code>Status.lib</code>
<code>BU : :Main : :Status : :ChangeSTAT_ECHO</code>	<code>Status.lib</code>

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 :

Cette variable globale sert à mettre le script dans un état dit "stable", vérifiant si un texte peut être redirigé vers un fichier de logs sans provoquer de boucles infinies, dans le cas où un rappel de la même fonction dans l'état précédent peut causer ceci.

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 - Section « PRINTING FORMATTED STRINGS »

2.1 Sous-section « DEBUGGING »

2.1.1 BU : :Main : :Echo : :__EchoVoid

Fonction : **BU : :Main : :Echo : :__EchoVoid**

Description :

Cette fonction affiche un espace vide de 10 lignes de hauteur pour séparer toutes les sorties des commandes appelées lors de l'exécution des fonctions **Echo<...>** du reste des sorties du script, lors de l'exécution du script avec la commande **bash -x**, qui affiche tous.

Paramètres :

Aucun paramètre n'est supporté.

Fonctionnement :

La fonction **BU : :Main : :Echo : :__EchoVoid** crée un espace vide de 10 lignes en appelant la commande **printf** sur 10 lignes, le redirige vers un **fichier temporaire** (enregistré dans la variable globale de projet **\$_BU_MAIN_PROJECT_ECHO_OUTPUT_FILE_PATH**) pour ne pas l'afficher sur le terminal si la commande **bash -x** n'est pas utilisée pour exécuter le script principal, et affiche un message d'avertissement si le texte ne peut être redirigé.

Utilisation :

Il est recommandé de n'appeler cette fonction que dans la fonction **BU : :Main : :Echo : :__EchoOutput** définie à la suite.

2.1.2 BU : :Main : :Echo : :__EchoOutput

Fonction : **BU : :Main : :Echo : :__EchoOutput**

Description :

Selon les étapes des fonctions **Echo<...>** définies dans ce fichier de librairie, cette fonction appelle la fonction **BU : :Main : :Echo : :__EchoVoid** définie précédemment pour créer un espace vide sur 10 lignes redirigé vers le **fichier temporaire**, pour mieux observer le début et la fin de la sortie d'une des fonctions **Echo<...>** lors d'un déboguage avec la commande **bash -x**, à la condition de fixer la valeur de la variable globale de statut **\$_BU_MAIN_STAT_DEBUG** à **true**.

Paramètres :

p_string	String
p_context	String
p_type	Char

Fonctionnement :

La fonction **BU : :Main : :Echo : :__EchoOutput** vérifie en premier la valeur de l'argument **p_type**.

Si sa valeur n'est pas un simple caractère alphabétique, **alors** la valeur est assignée à une nouvelle variable nommée **v_type**, tout en ne récupérant que le premier caractère, avant de redéclarer la variable **p_type** et d'y assigner la valeur de la variable **v_type**.

Fin de la condition.

Si le script n'est pas en mode déboguage (la variable globale de statut **\$_BU_MAIN_STAT_DEBUG** est fixée à **false**), **alors** l'exécution de la fonction est interrompue, et cette dernière retourne la valeur 0.

Sinon, **si** le script est en mode déboguage, **alors** :

- Si le traitement de texte à afficher n'est pas en mode sécurisé (la valeur de la variable globale de statut `$_BU_MAIN_STAT_ECHO` est définie à `false`), alors une nouvelle variable locale nommée `v_BU_MAIN_STAT_ECHO_val` est créée pour y enregistrer la valeur actuelle de la variable globale de statut sus-mentionnée, avant qu'elle ne soit modifiée à `true`.

Fin de la condition.

La fonction `BU : :Main : :Checkings : :CheckProjectRelatedFile` est appelée pour vérifier la présence du **fichier temporaire** où la sortie de la fonction `BU : :Main : :Echo : :__EchoOutput` doit être redirigée, ou le créer le cas échéant. En cas d'échec, un message d'erreur est affiché sur le terminal.

- Si la valeur de l'argument `p_context` correspond à la chaîne de caractères `begin`, alors :
 - La fonction `BU : :Main : :Echo : :__EchoVoid` est appelée pour afficher l'espace vide parmi la masse d'informations retournées par la commande `bash -x` sur le terminal.
 - Si le caractère enregistré dans le paramètre `p_type` est égal au premier caractère du type de message à afficher (`E -> Error`, `m -> Msg`, `N -> Newstep`, `S -> Success`, `W -> Warning`), alors la commande `echo -e` est appelée et une substitution de commande `y` est passée en argument.

Dans cette substitution de commande, une fonction d'affichage de header coloré (correspondant au code couleur associé au type du message traité) est appelée, et la chaîne de caractère « **BEGIN DEBUGGING TEXT :** » est passée avec la valeur du paramètre `p_string`, correspondant à la chaîne de caractères de la fonction `Echo<...>` d'origine.

Ensuite, la sortie de la fonction d'affichage de headers est redirigée vers le **même fichier temporaire** en écrasant le contenu déjà écrit pour ne pas alourdir l'espace disque. Le cas échéant, un message d'avertissement est affiché à l'écran.

Fin de la condition.

- Si la variable locale `v_BU_MAIN_STAT_ECHO_val` contient la valeur précédente de la variable globale de statut `$_BU_MAIN_STAT_ECHO`, alors la valeur enregistrée dans la variable locale est réassignée à la variable globale, avant que celle enregistrée dans la variable locale ne soit effacée.

Fin de la condition.

- Sinon, si la valeur de l'argument `p_context` correspond à la chaîne de caractères `end`, alors :
 - Le code exécuté dans **cette nouvelle condition** est sensiblement le même que dans la condition précédente, où la l'argument `p_context` correspond à la chaîne de caractères `begin`, à deux exceptions près :
 1. Le mot « **BEGIN** » de la chaîne de caractères passée en argument de la fonction d'affichage de headers est remplacé par le mot « **END** ».
 2. La fonction `BU : :Main : :Echo : :__EchoVoid` est appelée en tant que toute dernière instruction, au lieu d'être appelée en tout premier.

Fin de la condition.

Fin de la condition.

Fin de la condition.

Utilisation :

Appelez cette fonction dans toute fonction d'affichage de texte que vous souhaitez créer. Inspirez-vous de fonctions existantes, comme la fonction **BU : :EchoMsg**.

2.1.3 BU : :Main : :Echo : :__EchoCPLS

Fonction : **BU : :Main : :Echo : :__EchoCPLS**

Description :

Cette fonction remplit le même rôle que la fonction **BU : :Main : :Checkings : :CheckProjectLogStatus**, à la différence qu'elle ne sert que de fonction de secours lorsque la valeur de la variable globale de statut **\$_BU_MAIN_STAT_ECHO** est fixée à **true**, dans les situations où le rappel de cette même fonction pour gérer l'affichage du texte crée une boucle infinie.

Paramètres :

p_string	Chaîne de caractères
p_option	Caractère O U chaîne de caractères
pa_extraArgs	Tableau

Fonctionnement :

Pour chaque valeur enregistrée dans le tableau **pa_extraArgs** :

- **Si** la valeur correspond à la chaîne de caractères **nodate**, **alors** :
- **Dans le cas où** la valeur du paramètre **p_option** correspond à la chaîne de caractères **-n** (avec ou sans le tiret) :
- **Si** le script est en pleine étape d'initialisation des modules (renseignement obtenu via la valeur de retour de la fonction **BU : :Main : :Status : :CheckStatsInitializing**), **alors** la fonction **BU : :ModuleInit : :Msg** est appelée en passant la chaîne de caractères enregistrée dans le paramètre **p_string**, et la valeur du paramètre **p_option** y est passée en second argument pour ne pas sauter de ligne.

Sinon, la commande **echo -ne** est appelée pour afficher la chaîne de caractères enregistrée dans le paramètre **p_string**.

Fin de la condition.

- **Sinon, si** aucune valeur n'est passée au paramètre **p_option**, **alors** :
- **Si** le script est en pleine étape d'initialisation des modules, **alors** la fonction **BU : :ModuleInit : :Msg** est appelée en passant la date de log, puis la chaîne de caractères enregistrée dans le paramètre **p_string**, et la valeur du paramètre **p_option** y est passée en argument pour ne pas sauter de ligne.
- Sinon**, la commande **echo -ne** est appelée pour afficher la chaîne de caractères enregistrée

dans le paramètre **p_string**, toujours avec la date de log écrite auparavant.

Fin de la condition.

- **Sinon, si** une valeur non-supportée est passée au paramètre **p_option**, **alors** :

—