

# Fonctions du fichier de librairie Errors.lib

Dimitri OBEID

2021

## Table des matières

|          |  |          |
|----------|--|----------|
| <b>1</b> | <b>Présentation</b>  | <b>2</b> |
| 1.1      | Présentation . . . . .   | 2        |
| 1.2      | Définitions des éléments mentionnés . . . . .                                    | 2        |
| 1.2.1    | Chemins de fichiers . . . . .  | 2        |
| 1.2.2    | Fonctions . . . . .  | 2        |
| 1.2.3    | Variables globales externes et / ou variables d'environnement appelées . . . . . | 2        |
| <b>2</b> | <b>Fonctions</b>   | <b>2</b> |
| 2.1      | Sous-section CLASSIC ERRORS HANDLING . . . . .                                   | 2        |
| 2.1.1    | BU : :Main : :Errors : :HandleSmallErrors . . . . .                              | 2        |

## 1 Présentation

### 1.1 Présentation

Ce fichier source inclut des fonctions servant à la gestion d'erreurs.

---

### 1.2 Définitions des éléments mentionnés

#### 1.2.1 Chemins de fichiers

Nom du fichier :

Dossier parent :

\* : Voir la variable `$_BU_MAIN_ROOT_DIR_PATH/lib/functions/main` dans la sous-sous-catégorie [Variables](#).

---

#### 1.2.2 Fonctions

Fonction :

Fichier de définition :

---

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

Variable :

Description : .

Fichier de définition :

---

Variable :

Description : .

Fichier de définition :

---

## 2 Fonctions

### 2.1 Sous-section CLASSIC ERRORS HANDLING

#### 2.1.1 BU : :Main : :Errors : :HandleSmallErrors

Description :

Cette fonction affiche un simple message d'avertissement ou d'erreur, selon le contexte choisi, contrairement à la fonction `BU : :Main : :Errors : :HandleErrors`, qui fait un rapport complet de l'erreur, et qui est à privilégier pour de plus grosses erreurs.

**Paramètres :**

- **p\_type** Ce paramètre attend soit le caractère **W** (warning), soit le paramètre **E** (error)
- **p\_string** Ce paramètre attend la chaîne de caractère à afficher
- **p\_return** Ce paramètre attend les mêmes caractères que le paramètre **p\_type**
- **p\_cpls** Ce paramètre attend la chaîne de caractère **CPLS**, pour changer la valeur de la variable globale de statut **\$\_BU\_MAIN\_STAT\_ECHO**.
- **Note :** **CPLS** sont les initiales de la fonction **BU : :CheckProjectLogStatus**, qui vérifie si et comment un message peut être redirigé vers un fichier de logs, et où un rappel d'une fonction **BU : :Echo<...>** avec la valeur de la variable globale de statut mentionnée ci-dessus à **false** provoquerait une boucle infinie.

**Fonctionnement :**

Cette fonction vérifie en premier lieu si la valeur **CPLS** est passée en quatrième argument, pour sauvegarder la valeur actuelle de la variable globale de statut **\$\_BU\_MAIN\_STAT\_ECHO**, et la mettre à **true**.

Passer cette valeur à ce paramètre est absolument nécessaire dans les situations où l'appel des fonctions **BU : :Echo<...>()** peut provoquer une boucle infinie.

Ensuite, elle vérifie quelle valeur a été passée en premier argument, puis affiche la chaîne de caractères passée en deuxième argument selon la valeur du premier (W = warning, donc appel de la fonction **BU : :EchoWarning**, et E = error, donc appel de la fonction **BU : :EchoError**).

Si la valeur du premier argument n'est ni renseignée, ni attendue, alors une de ces fonctions est aléatoirement appelée pour afficher cette chaîne de caractères.

Une fois la chaîne de caractère affichée, la fonction vérifie quelle valeur a été passée en troisième argument (**p\_return**), pour simplement continuer l'exécution de votre script ou pour directement arrêter son exécution.

En cas d'absence de valeur ou de présence d'une valeur inattendue, un message d'avertissement s'affiche sur le terminal, avant que l'exécution du script ne soit définitivement interrompue.

Enfin, si la valeur **CPLS** a été passée en quatrième argument, l'ancienne valeur est automatiquement réassignée à la variable globale de statut **\$\_BU\_MAIN\_STAT\_ECHO**.

**Utilisation :**