

```
/*  
*   module   : Tmessage.odt  
*   projet   : documentation de l'objet TMessage  
*  
*   version  : 5.1  
*   auteur   : laporte  
*   création : 23/12/94  
*   modif    : 6/01/00  
*/
```

Table des matières

1. Introduction.....	2
2. Liste des méthodes de la classe.....	2
3. Description des méthodes.....	2

1. Introduction

La librairie TMessage a été conçue dans le but de soulager le programmeur lors du développement d'applications ou dans le cadre d'une internationalisation de logiciels.

Les messages destinés à être affichés à l'écran sont placés dans un fichier texte à inclure dans le code PHP.

Il est ainsi plus aisé de localiser les différents messages dans un code source lors de modifications ou de maintenance avec une séparation nette entre les couches métiers.

[\[sommaire\]](#)

2. Liste des méthodes de la classe

TMessage	- Constructeur de la classe
read	- Retourne un message formaté
count	- Retourne le nombre de messages contenus dans le fichier texte

[\[sommaire\]](#)

3. Description des méthodes

Nom

TMessage - Constructeur de la classe

Syntaxe

```
$objet = TMessage($file)  
$objet = TMessage($file, $rootDir)
```

Description

Le premier constructeur ouvre un fichier messages \$file. Ce fichier doit se trouver dans le répertoire où se trouve le code.

Le deuxième constructeur ouvre un fichier messages \$file qui se trouve dans le dossier \$rootDir.

Valeurs retournées

Objet TMessage.

Exemple

```
require "./inc/Tmessage.php"; // description objet TMessage  
  
$msg = new TMessage("./msg/fr/home.php"); // appel au constructeur
```

Nom

read - Retourne un message formaté

Syntaxe

```
$string = TMessage->read($IDmsg)  
$string = TMessage->read($IDmsg, $arg)  
$string = TMessage->read($IDmsg, $arg, $bool)
```

Description

La méthode read recherche le texte correspondant au numéro de message \$IDmsg.
Le 2ème argument \$arg est utilisé pour mettre à jour une variable dans le texte identifiée par %1 (variable n° 1). Si le texte contient plusieurs variables, il est possible de passer un tableau en argument qui affectera chaque valeur au n° de variable correspondant.
Le 3ème argument booléen indique si le pluriel dans le texte, identifié par %s, doit être pris en compte.

Valeurs retournées

Le texte correspondant au message numéro \$IDmsg, remplacé par les éventuels arguments arg ou la chaîne vide "" si aucune correspondance n'a été trouvée.

Exemple

```
require "./inc/Tmessage.php"; // description objet Tmessage  
require "./msg/home.php"; // description des n° de messages  
$msg = new TMessage("./msg/fr/home.php"); // appel au constructeur  
  
$title = $msg->read($MSG_TITLE); // lecture du 1er message  
$title = $msg->read(0); // équivalent au code ci-dessus  
$cpb = $msg->read($MSG_CPB, 10); // message avec un argument (10)  
$time = $msg->read($MSG_TIME, 22, true); // message avec un argument (22)  
// avec prise en compte du pluriel  
$range = $msg->read($MSG_RANGE, array(0, 6)); // message avec deux arguments (0 et 6)  
  
// le fichier "./msg/home.php"  
static $MSG_TITLE = 0;  
static $MSG_CPB = 1;  
static $MSG_TIME = 2;  
static $MSG_RANGE = 3;  
  
// le fichier "./msg/fr/home.php" correspondant  
static $message = Array(  
    "Connectez-vous avec TOR",  
    "%1 cpb",  
    "%1 minute%s",  
    "plage %1-%2 heures"  
);
```

Nom

count - Retourne le nombre de messages

Syntaxe

```
$integer = TMessage->count()
```

Description

La méthode count() indique le nombre de messages disponibles dans le fichier messages.

Valeurs retournées

Le nombre de messages disponibles dans le fichier messages.

Exemple

```
require "./inc/Tmessage.php"; // description objet TMessage  
$msg = new TMessage("./msg/fr/home.php"); // appel au constructeur  
  
$total = $msg->count()
```

[\[sommaire\]](#)