

ToolType

COLLABORATORS

	TITLE : ToolType		
ACTION	NAME	DATE	SIGNATURE
WRITTEN BY		January 23, 2025	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	ToolType	1
1.1	ToolType V1.00	1
1.2	freetooltype	1
1.3	getnexttooltypestring	2
1.4	getnumberoftooltypes	2
1.5	gettooltypevalue	2
1.6	inittooltype	2
1.7	matchtooltype	3
1.8	matchtooltypestring	3
1.9	readtooltypediskinfo	4
1.10	writetooltypediskinfo	4

Chapter 1

ToolType

1.1 ToolType V1.00

PureBasic - ToolType V1.00

Les "ToolTypes" sont les paramètres contenus au sein d'une icône du Workbench. Vous pouvez vous en servir pour stocker la configuration de votre programme, sans utiliser un fichier supplémentaire. De plus l'utilisateur pourra changer la configuration à son aise, grâce à la fenêtre 'Icône/Information' située dans le menu du Workbench.

Commands summary:

```
FreeToolType
GetNextToolTypeString
GetNumberOfToolTypes
GetToolTypeValue
InitToolType
MatchToolType
MatchToolTypeString
ReadToolTypeDiskInfo
WriteToolTypeDiskInfo
```

ToolType Demo

1.2 freetooltype

Syntaxe

```
FreeToolType (#Objet)
```

Résumé

Libère un objet 'ToolType' qui avait été initialisé par la commande `ReadToolTypeDiskInfo()`..

#Objet: Identifiant numérique de l'objet à libérer.

1.3 getnexttooltypestring

Syntaxe

```
String$ = GetNextToolTypeString(#Obj)
```

Résumé

Retourne la contenu et sa valeur (ou ses valeurs, si le 'ToolType' en accepte plusieurs) du prochain 'ToolType'.

Une fois que la fin des ToolTypes est atteinte, le prochain ToolType lu sera le premier (ça boucle). Pour savoir combien de 'ToolType' sont présent, utilisez la commande `GetNumberOfToolTypes()`

La longueur d'un 'ToolType' est limitée à 128 octets.

#Objet: Identifiant numérique de l'object à utiliser.

String: Chaîne de caractère contenant le 'ToolType' lu comme, par exemple, ceci: "AMIGA=1200|4000".

1.4 getnumberoftooltypes

Syntaxe

```
Result.w = GetNumberOfToolTypes(#Objet)
```

Résumé

Renvoie le nombre de 'ToolType' contenu dans cet objet (c'est à dire dans l'icone correspondante). Si la valeur renvoyée est '-1', alors il n'y a pas de 'ToolType' dans cette icône

#Objet: Identifiant numérique de l'object à utiliser.

1.5 gettooltypevalue

Syntaxe

```
String$ = GetToolTypeValue(#Obj.w, ToolName$)
```

Résumé

Retourne une chaîne de caractère contenant la valeur du 'ToolType' spécifié, s'il existe, sous la forme "A1200|A4000"

#Objet: Identifiant numérique de l'object à utiliser.

ToolName\$: Nom du 'ToolType' à examiner.

1.6 inittooltype

Syntaxe

```
Result.l = InitToolType(Objects.l,*WBStartup)
```

Résumé

Initialise l'environnement nécessaire à la gestion future des 'ToolTypes'. Si la valeur retournée est NULLE, alors l'initialisation a échoué. Cette fonction doit être appelée avant toutes les autres fonctions de cette bibliothèque. L'objet '0' est automatiquement créé et initialisé avec les paramètres de l'icône du programme lancé. Le répertoire courant de l'application est également changé par le répertoire qui contient le programme lancé (très pratique).

Objects:: Nombre maximum d'object (c'est à dire d'icônes) à gérer.

*WBStartup: Pointeur vers la structure WBStartupMessage, retournée par la fonction WBStartup(). Le programme doit être lancé du workbench (via une icône) pour que cette commande s'exécute).

1.7 matchtooltype

Syntaxe

```
Result.w = MatchToolType(#Obj.w,ToolName$,Value$)
```

Résumé

Regarde si le ToolType spécifié existe, et s'il est égal à la valeur demandée. Cette commande est très utile pour examiner les 'ToolTypes'. Si le résultat est '-1' alors le ToolType est introuvable. Si le résultat est '0' alors le ToolType est trouvé, mais la valeur demandée n'est pas valable. Si le résultat est '1' alors le ToolType est trouvé, et la valeur est présente.

ToolName: Nom du 'ToolType' à examiner.

Value: Valeur du 'ToolType'

1.8 matchtooltypestring

Syntaxe

```
Result.w = MatchToolTypeString(String$,ToolName$,Value$)
```

Résumé

Recherche dans une chaîne de caractères un 'ToolType' spécifique ainsi que la valeur demandée. Si le résultat est '-1' alors le ToolType est introuvable.

Si le résultat est '0' alors le ToolType est trouvé, mais la valeur demandée n'est pas valable. Si le résultat est '1' alors le ToolType est trouvé, et la valeur est présente.

String: Chaîne de caractères à examiner.

ToolName: Nom du 'ToolType' à rechercher.

Value: Valeur du 'ToolType'

1.9 readtooltypediskinfo

Syntaxe

```
Result.l = ReadToolTypeDiskInfo(#Obj.w, IconName$)
```

Résumé

Initialise l'objet spécifié par le contenu de l'icône donnée. Une fois qu'un objet a été initialisé, il faut utiliser la commande `FreeToolType()` pour pouvoir réutiliser cet objet. Si le résultat est NUL, alors l'icône ne peut pas être lue.

#Obj: Identifiant numérique de l'objet

IconName: Nom de l'icône (".info" est automatiquement ajouté à la fin du nom).

1.10 writetooltypediskinfo

Syntaxe

```
Result.w = WriteToolTypeDiskInfo(#Obj.w, Array(), IconName$)
```

Résumé

Écrit dans l'icône les paramètres contenu dans le tableau.

#Obj: Identifiant numérique de l'objet à utiliser.

Array(): Tableau contenant les chaînes de caractères à écrire dans l'icône.

IconName: Nom de l'icône (".info" est automatiquement ajouté à la fin du nom).

Les nouveaux 'ToolTypes' écrits écrasent entièrement les anciens, donc si vous voulez en ajouter un, il faut d'abord lire les anciens puis ajouter une entrée au tableau.
