

ToolType

COLLABORATORS

	<i>TITLE :</i> ToolType		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		January 19, 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	3
1.7	matchtooltype	3
1.8	matchtooltypestring	4
1.9	readtooltypediskinfo	4
1.10	writetooltypediskinfo	5

Chapter 1

ToolType

1.1 ToolType V1.00

PureBasic - ToolType V1.00

Tooltypes ermöglichen Ihnen, an die - im Icon des gestarteten Programms enthaltenen - Informationen zu gelangen. Mit dieser Library können Sie ganz einfach alle Tooltypes eines Icons lesen, verändern und hinzufügen

Befehlsübersicht:

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

ToolType Demo

1.2 freetooltype

SYNTAX

```
FreeToolType (#Obj.w)
```

STATEMENT

Dieser Befehl gibt ein Tooltype Objekt wieder frei, das mit ReadToolTypeDiskInfo() initialisiert wurde. Der Befehl bleibt ohne Wirkung, wenn das angegebene Objekt gar nicht initialisiert wurde.

#Obj

Das freizugebende Objekt.

1.3 getNexttooltypestring

SYNTAX

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

FUNCTION

Diese Funktion gibt das nächste Tooltype und alle zugehörigen Werte in einem String zurück.

Nachdem das letzte Tooltype erreicht wurde, wird als nächstes Tooltype wieder das Erste zurückgegeben. Um herauszufinden wieviele Tooltype Strings existieren, benutzen Sie den Rückgabewert der Funktion `GetNumberOfToolTypes()`.

Ein Tooltype ist in seiner Länge durch das OS auf 128 Bytes begrenzt.

#Obj

Das zu benutzende Objekt.

String

Der zurückgegebene String beinhaltet das Tooltype und alle damit verbundenen Werte, wie z.B. "AMIGA=1200|4000".

1.4 getnumberoftooltypes

SYNTAX

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

FUNCTION

Diese Funktion ermittelt, wieviele Tooltypes in der .info Datei des Icons enthalten sind. Sie sollte benutzt werden, um einfach die richtige Anzahl der Tooltype Strings herauszufinden.

#Obj

Das zu benutzende Objekt.

Result

Jeder positiver Wert ist die Anzahl der in der .info Datei des Icons enthaltenen Tooltypes. Ergibt der Rückgabewert dagegen -1 zeigt dies, dass keine Tooltypes in der .info Datei des Icons enthalten sind.

1.5 gettooltypevalue

SYNTAX

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

FUNCTION

Diese Funktion gibt die - zum angegebenen Tooltype gehörenden - Werte in einem String zurück.

#Obj

Das zu benutzende Objekt.

String

Der zurückgegebene String beinhaltet die Werte des Tooltypes, wie z.B. "1200|4000". Wenn der angegebene Tooltype-Eintrag in der .info Datei des Icons dagegen nicht existiert, ist der Rückgabestring leer.

1.6 inittooltype

SYNTAX

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

FUNCTION

Diese Funktion richtet einige Grundlagen für die Handhabung von ToolTypes ein.

Dies ist die Initialisierung-Routine und sollte immer vor irgendeiner ToolType Funktion aufgerufen werden. Zur Zeit kann sie nur einmal aufgerufen werden, wenn dabei ein Fehler auftritt, sollte das Programm immer beendet werden.

Objects

Dieser Parameter gibt an, wieviele ToolType Objekte benötigt werden. Jedes Objekt kann die Daten zu einer .info Datei beinhalten.

Die maximale Zahl an Objekten ist 2046.

*WBStartup

Dies ist ein Zeiger auf eine WBStartupMessage, diese Mitteilung (Message) wird von der WorkBench an das Programm übergeben. *WBStartup ist der Rückgabewert von der Funktion WBStartup(), welche die Message einholt.

Mit diesem Schritt wird das Objekt 0 mit den Daten der .info Datei des doppelt angeklickten Icons immer automatisch initialisiert. Das aktuelle Verzeichnis wird ebenfalls auf das Verzeichnis des Icons gesetzt.

Ist dieser Parameter kein Zeiger, stattdessen NULL, hat keine der vorgenannten Aktionen stattgefunden.

Result

Wenn ein Fehler auftrat, ergibt das Resultat FALSE und es gibt keinen Grund, irgendeine weitere ToolType Funktion aufzurufen. Sollte immer überprüft werden.

1.7 matchtooltype

SYNTAX

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

FUNCTION

Diese Funktion überprüft, ob das angegebene Tooltype existiert und wenn ja, wird es auf den angegebenen Wert gesetzt.

Keines der ToolTypes muß für diese Funktion in PureBasic Strings eingelesen werden, es ist einfacher, wenn ToolTypes nur überprüft werden müssen.

ToolName

Dies ist das ToolType, nach dem gesucht wird.

Value

Dies ist der Wert, nach dem gesucht wird. Kann auch ein Leerstring sein.

Result

Ergibt dieses -1, dann ist das ToolType in der .info Datei des Icons nicht vorhanden. Ist das Ergebnis gleich 0, dann wurde das ToolType gefunden, aber es beinhaltet nicht den gesuchten Wert. Ist das Ergebnis schließlich 1, dann wurde das ToolType gefunden und es hat einen Wert, der mit dem dritten Parameter übereinstimmt.

1.8 matchtooltypestring

SYNTAX

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

FUNCTION

Diese Funktion überprüft, ob der PureBasic String das angegebene ToolType und den passenden Wert beinhaltet.

String

Dies ist der zu überprüfende String.

ToolName

Dies ist das zu überprüfende ToolType.

Value

Dies ist der zu überprüfende Wert, kann auch ein Leerstring sein.

Result

Ergibt dieses -1, dann ist das ToolType im angegebenen PureBasic String nicht vorhanden. Ist das Ergebnis gleich 0, dann wurde das ToolType gefunden, aber es beinhaltet nicht den gesuchten Wert. Ist das Ergebnis schließlich 1, dann wurde das ToolType gefunden und es hat einen Wert, der mit dem dritten Parameter übereinstimmt.

1.9 readtooltypediskinfo

SYNTAX

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

FUNCTION

Diese Funktion liest die .info Datei des angegebenen Icons ein und initialisiert das ToolType Objekt.

Ist das Objekt bereits initialisiert, macht dies der Funktion nichts aus, sie erstellt einfach ein neues Objekt, ohne das alte zu löschen. Dann

gibt es allerdings keinen möglichen Weg mehr, irgendwelche Aktionen am alten Objekt auszuführen.

Die maximale Zahl an Tooltypes, die verwaltet werden können, ist 32767, beschränkt durch diese PureBasic Library.

#Obj

Das zu benutzende Objekt.

IconName

Der Name des Icons, ".info" wird automatisch am Ende des Namens angefügt.

Result

Ergibt dieses FALSE, konnte die .info Datei des Icons nicht eingelesen werden.

1.10 writetooltypediskinfo

SYNTAX

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

FUNCTION

Diese Funktion schreibt die .info Datei des Icons zurück auf Disk.

#Obj

Das zu benutzende Objekt.

Array()

Dieses Array enthält die Strings, welche die zu sichernden neuen ToolTypes beinhalten.

Die neuen ToolTypes haben eine höhere Priorität als alte ToolTypes. Wenn also nur ein ToolType hinzugefügt wurde, müssen erst alle ToolTypes mit der Funktion GetNextToolTypeString() in PureBasic Strings eingelesen werden.

Die maximale Anzahl an ToolTypes, die verwaltet werden können, ist 32767, beschränkt durch diese PureBasic Library.

IconName

Der Name des Icons, ".info" wird automatisch am Ende des Namens angefügt.

Result

Ist dieses TRUE, war das Abspeichern erfolgreich.
