

**ToolType**

<b>COLLABORATORS</b>
----------------------

	<i>TITLE :</i> ToolType		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		January 23, 2025	

<b>REVISION HISTORY</b>
-------------------------

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>ToolType</b>	<b>1</b>
1.1	ToolType . . . . .	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

PureBasic - ToolType V1.01

Die 'Tooltypes' befinden sich in einem Workbench Icon. Es sind einige Strings, die Sie einfach lesen, verändern und hinzufügen können, um z.B. Ihre Programmkonfiguration abzuspeichern. Dies ist sehr nützlich, da Sie somit keine separate Datei erstellen müssen und der User die Tooltypes einfach selbst über das Menü 'Piktogramm/Informationen anzeigen' der Workbench verändern kann.

Befehlsübersicht:

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

ToolType Demo

### 1.2 freetooltype

SYNTAX

```
FreeToolType(#Info.w)
```

STATEMENT

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

#Info

Das freizugebende Objekt.

---

## 1.3 getnexttooltypestring

### SYNTAX

```
String$ = GetNextToolTypeString(#Info.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.

**#Info**

Das zu benutzende Info-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(#Info.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.

**#Info**

Die zu benutzende Info-Objekt.

**Result**

Jeder Wert größer als null gibt die Anzahl der in der .info Datei des Icons enthaltenen Tooltypes wieder. Ergibt der Rückgabewert dagegen 0 zeigt dies, dass keine Tooltypes in der .info Datei des Icons enthalten sind.

## 1.5 gettooltypevalue

### SYNTAX

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

### FUNCTION

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

**#Info**

---

Das zu benutzende Info-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(#Infos.l)
```

FUNCTION

Dies ist die Initialisierung-Routine und sollte immer vor irgendeiner ToolType Funktion aufgerufen werden. Wenn bei diesem Befehl ein Fehler auftrat, sollte keine andere Funktion aufgerufen werden.

#Infos

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

Die maximale Zahl an Objekten ist 2046.

Spezielle Anmerkung:

Der zweite Parameter, benutzt in der Version 1.00, welcher einen Wert 0 oder den Rückgabewert von WBStartup() enthalten konnte, ist weggefallen. Dies wird jetzt automatisch erledigt, mit der Einschränkung, dass WBStartup() vor InitToolType() aufgerufen werden muß.

Wenn InitToolType() feststellt, dass das Programm von der Workbench gestartet wurde, dann wird #Info 0 automatisch mit den Daten der .info Datei des doppelt angeklickten Icons initialisiert. Ebenso wird als aktuelles Verzeichnis das Verzeichnis des angeklickten Icons festgelegt.

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(#Info.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.

#Info

Das zu benutzende Info-Objekt.

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(#Info.w,IconName\$)

FUNCTION

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

Ist das Info-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.

#Info

Das zu benutzende Info-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(#Info.w,Array(),IconName\$)

FUNCTION

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

#Info

Das zu benutzende Info-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 und dann zurückgeschrieben 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.

---