

Ribbon.dll

Version 0.2.53 vom 25.08.2021

Befehlsreferenz

Alle Befehle auf einen Blick:

<code>ActivateRibbonCategory()</code>	<code>LinkRibbonPopup()</code>
<code>AddPopupText()</code>	<code>MoveControlToRibbonPopup()</code>
<code>AddRibbonItem()</code>	<code>MovePopupText()</code>
<code>AddRibbonItemStatus()</code>	<code>RemovePopupText()</code>
<code>CheckRibbonActivity()</code>	<code>RemoveRibbon()</code>
<code>CloseRibbonPopup()</code>	<code>RemoveRibbonItem()</code>
<code>CreateRibbon()</code>	<code>RenderRibbon()</code>
<code>EditPopupText()</code>	<code>ResizeRibbonPopup()</code>
<code>FlushRibbonActivity()</code>	<code>RibbonVersion()</code>
<code>GetActiveRibbonCategory()</code>	<code>SetCallbackMessage()</code>
<code>GetRibbonItemStatus()</code>	<code>SetRibbonCheckboxImage()</code>
<code>GetRibbonItemImage()</code>	<code>SetRibbonFont()</code>
<code>GetRibbonItemPosition()</code>	<code>SetRibbonItemCheckStatus()</code>
<code>GetRibbonItemStatus()</code>	<code>SetRibbonItemImage()</code>
<code>GetRibbonItemText()</code>	<code>SetRibbonItemPosition()</code>
<code>GetRibbonItemType()</code>	<code>SetRibbonItemStatus()</code>
<code>GetRibbonMetric()</code>	<code>SetRibbonItemText()</code>
<code>GetRibbonPopupHandle()</code>	<code>SetRibbonMetric()</code>
<code>HideRibbon()</code>	<code>SubRibbonItemStatus()</code>
	<code>UnlinkRibbonPopup()</code>

Allgemeine Hinweise:

- Bei der Übergabe von Strings an die DLL wird generell die Adresse des Strings übergeben, also statt `text$` wird `Addr(text$)` benutzt. Diese Parameter sind hier mit einem vorangestellten @ gekennzeichnet.
- Das Gleiche gilt für die Rückgabe von Strings (z.B. `GetRibbonItemText()`): Hier wird ein String mit ausreichender Länge dimensioniert (z.B. `text$=Space$(255)`) und anschließend die Adresse übergeben (`Addr(text$)`). Ein Beispiel findet sich bei `GetRibbonItemText()`. Diese Parameter sind ebenfalls mit einem vorangestellten @ gekennzeichnet.

Die Befehle im Einzelnen:

`ActivateRibbonCategory(id)`

id...ID der zu aktivierenden Category

Rückgabe: ID der zuletzt aktiven Category dieses Ribbons, -1 kein Ribbon eine Category mit der ID „id“ besitzt

Aktiviert die Category mit der ID „id“. Die Category wird sichtbar gemacht und das zugehörige Ribbon neu gerendert.

AddPopupText(popup, x, y, @text)

popup...Handle eines Popupfensters

x, y...Koordinaten

text...String mit dem Text, der angezeigt werden soll

Rückgabe: ID des Textes (z.B. um ihn nachträglich zu ändern)

Fügt dem Popup mit der ID „id“ an Position x,y den Text „text“ hinzu.

AddRibbonItem(parent, id, typ, @text, image, status)

parent...ID des übergeordneten Elements oder Handle des Ribbons (je nachdem, was für ein Item erstellt werden soll)

id...ID des neuen Items

text...String mit Text auf dem Item

image...Handle eines Icons

status...Kombination von Statusflags

Fügt ein neues Item in ein Ribbon ein. parent ist die ID des übergeordneten Items, im Fall von Categories, HeadItems und RightHeadItems ist es das Handle des Ribbons. typ gibt die Art des Steuerelementes an (siehe Tabelle). image ist ein Handle eines geladenen Icons (z.B. mittels `Create („hIcon“, ...)` oder im Fall eines Images

`Create („hPic“, ...)`).

Art des Controls	Parent	Text	Image	Checked
~Ribbon_Type_Button	ButtonContainer/Container/Group	✓/✗	✓*	✗
~Ribbon_Type_ButtonContainer	Group	✗	✗	✗
~Ribbon_Type_Category	Ribbon	✓	16*16	✗
~Ribbon_Type_Checkbox	Container/Group	✓	✗	✓
~Ribbon_Type_Container	Group	✗	✗	✗
~Ribbon_Type_Group	Category	✓	✗	✗
~Ribbon_Type_HeadButton	Ribbon	✗	16*16	✗
~Ribbon_Type_HeadPushButton	Ribbon	✗	16*16	✓
~Ribbon_Type_HeadSeparator	Ribbon	✗	✗	✗
~Ribbon_Type_Image	Group	✗	✓**	✗
~Ribbon_Type_PushButton	ButtonContainer/Container/Group	✓/✗	✓*	✓
~Ribbon_Type_RightHeadButton	Ribbon	✗	16*16	✗
~Ribbon_Type_RightHeadPushButton	Ribbon	✗	16*16	✓
~Ribbon_Type_RightHeadSeparator	Ribbon	✗	✗	✗
~Ribbon_Type_RightHeadText	Ribbon	✓	✗	✗
~Ribbon_Type_RightHeadTextButton	Ribbon	✓	✗	✗
~Ribbon_Type_Separator	ButtonContainer/Container/Group	✗	✗	✗
<i>*In Group: 48*48, in Container: 16*16, in ButtonContainer: 24*24</i>				
<i>**Bildgröße wird verzerrungsfrei auf eine Höhe von 70 Pixel angepasst, Bilder können nicht gehovert oder angeklickt werden.</i>				

Folgende Statusflags sind vorhanden, werden aber nicht auf allen Controls dargestellt:

Konstante	Wert	Beschreibung
Ribbon_Status_Standard	0	Standard
Ribbon_Status_Deactivated	1	Item ist deaktiviert. Es erzeugt keine Ereignisse und nimmt keine Ereignisse entgegen. Es hovert nicht.*
Ribbon_Status_Checked	2	Item hat den Status "aktiviert". Checkboxen bekommen einen Haken, PushButtons sind "eingedrückt".
Ribbon_Status_Hidden	4	Item ist versteckt. Es wird nicht gerendert, kann aber weiterhin verwendet werden.
Ribbon_Status_UseOriginalImage	8	NICHT in der DLL verwendet (funktioniert nur mit PureBasic und ist für die DLL nicht relevant)!
Ribbon_Status_Inbetween	16	Item hat den Status "teilweise aktiviert". Nur für Checkboxen.

*Ein deaktiviertes Item deaktiviert auch alle untergeordneten Items. Das heißt, dass diese zwar ihren eigenen Status behalten, aber behandelt werden, als wären sie inaktiv. Sie können nicht angeklickt oder gehovert werden.

Die Statusflags sind in der Ribbon.PH gespeichert und können nachträglich mittels `AddRibbonItemStatus()`, `SetRibbonItemStatus()` und `SubRibbonItemStatus()` geändert sowie mittels `GetRibbonItemStatus()` gelesen werden. Es können mehrere Flags miteinander addiert werden.

AddRibbonItemStatus(id, status)

id...ID des neuen Items

status...Zu addierendes Statusflag

Setzt ein bestimmtes Statusflag eines Ribbon-Items. Es wird dafür gesorgt, dass das Statusflag aktiviert ist. Sollte es bereits aktiviert sein ändert sich nichts. Die einzelnen Statusflags können bei **AddRibbonItem()** nachgeschlagen werden.

AddRibbonItemStatus(activity)

activity...Bereichsvariable, die die Informationen der nächsten Aktivität aufnimmt. Muss mit der Struktur „Ribbon“ initialisiert sein

Rückgabe: Anzahl anliegender Aktivitäten, einschließlich der gerade abgerufenen.

Diese Funktion ruft das nächste Aktivität ab. Liegt keine Aktivität an (Mausklick etc.) wird der Inhalt des Bereichs nicht verändert, ansonsten wird der Bereich mit den entsprechenden Informationen gefüllt. Der Rückgabewert repräsentiert die Anzahl anliegender Ereignisse, einschließlich des gerade abgerufenen (wird eine 0 zurückgegeben muss der Bereich also nicht ausgewertet werden). Die einzelnen Elemente des Bereichs enthalten die Informationen zum Ereignis: Ribbon&...Das Ribbon auf dem das Ereignis stattfand. In der Regel nicht nötig, aber wichtig um die System-Buttons auszuwerten, da diese auf jedem Ribbon die gleich ID haben.

Event&...Das ausgelöste Event (vergleiche dazu die Header-Datei)

id&...Die ID des Elementes, das das Event ausgelöst hat

controlx&, controly&...Die linke obere Ecke des Elementes innerhalb des Ribbons

controldx&, controldy&...Die Größe des Elementes

mousex&, mousey&...die Position der Maus, als das Ereignis ausgelöst wurde, wieder innerhalb des Ribbons

CloseRibbonPopup()

Schließt ein eventuell offenes Popupfenster. Popupfenster schließen sich von allein, wenn sie den Fokus verlieren. Hiermit kann das Schließen erzwungen werden.

CreateRibbon(hWnd, color, backcolor, style, flags)

hwnd...Das Fenster auf dem das Ribbon erstellt wird

color...Die Hintergrundfarbe der Steuerelemente

backcolor...Die Hintergrundfarbe des Ribbons an sich

style...Stil des Ribbons

flags...Flags

Rückgabe: Handle des Ribbons

Diese Funktion erzeugt ein neues Ribbon auf dem Fenster hWnd. Die Hintergrundfarbe kann frei definiert werden oder mittels ~Ribbon_Color_Auto auf die aktuell in Windows eingestellte Hintergrundfarbe festgelegt werden. Der Style gibt an, wie sich Texte und Rahmen vom Hintergrund abheben (siehe PH-Datei). Über die Flags lassen sich diverse Eigenschaften des Ribbon einstellen (siehe ebenfalls die PH-Datei).

EditPopupText(id, tid, @text)

id...ID des Elements auf einem Ribbon, dem das Steuerelement zugewiesen wurde

tid...Handle des Steuerelements, dessen Text geändert werden soll

text...neuer Text

Ändert den Text auf einem Steuerelement, das einem Popup eines Elementes zugewiesen wurde. Entspricht im Prinzip dem XProfan-Befehl SetText.

FlushRibbonActivity()

Löscht den Puffer mit allen Aktivitäten auf allen Ribbons. Alle Ereignisse, die noch nicht abgerufen wurde, werden verworfen.

GetActiveRibbonCategory(ribbon)

ribbon...Handle eines Ribbons

Ermittelt die ID der aktuell aktiven Kategorie des gewünschten Ribbons.

GetRibbonItemCheckStatus(id)

id...ID eines Elements

Ermittelt, ob ein Element gecheckt ist. Rückgabewerte können Ribbon_Status_Checked und Ribbon_Status_Unchecked (sowie Ribbon_Status_Inbetween bei Checkboxes) sein. Es ist nicht vorgesehen, dass ein Element, was keine Checkbox ist (z.B. ein PushButton), den Status Ribbon_Status_Inbetween erhält. Allerdings ist es möglich diesen Status per Befehl zuzuweisen, was aber nicht empfohlen ist. Sollten mehrere Statusflags aktiviert sein, so ist die Prioritätenreihenfolge (von oben nach unten): Ribbon_Status_Checked, Ribbon_Status_Inbetween, Ribbon_Status_Unchecked.

GetRibbonItemImage(id)

id...ID eines Elements

Ermittelt das Handle des Bildes eines Elements.

GetRibbonItemPosition(id)

id...ID eines Elements

Ermittelt die Position eines Elements innerhalb seiner Gruppe. Das erste Element hat die Position 0.

GetRibbonItemStatus(id)

id...ID eines Elements

Ermittelt die aktuelle Kombination von Statusflags eines Elements. Die Flags stehen in der PH-Datei (~Ribbon_Status_...).

GetRibbonItemStatusFlag(id, flag)

id...ID eines Elements

flag...Das abzufragende Flag

Ermittelt den Status eines Flags des Elements (aktiv/inaktiv). Die Flags stehen in der PH-Datei (~Ribbon_Status_...).

GetRibbonItemText(id, @text)

id...ID eines Elements

text...Adresse eines Bereichs oder Strings, der ausreichend dimensioniert ist

Rückgabe: Anzahl Zeichen des Rückgabestrings. Es wird ein Zeichen mehr am Ende des Strings geschrieben (Chr\$(0)).

Ermittelt den Text des Elements. Der Text wird an die Adresse @text geschrieben. Der Bereich oder String muss vor der Übergabe ausreichend dimensioniert sein.

GetRibbonMetric(handle, metric)

handle...Handle eines Ribbons

metric...Die zu ermittelnde Eigenschaft

Rückgabe: Aktueller Zustand der Eigenschaft

Ermittelt den Zustand einer Eigenschaft eines Ribbons (siehe PH-Datei unter ~Ribbon_Metric_...).

GetRibbonPopupHandle(id)

id...ID eines Elements

Rückgabe: Handle des zu diesem Element zugeordneten Pop-upfensters (oder 0, falls keins zugeordnet ist)

Ermittelt das Handle des einem Element zugeordneten Pop-upfensters. Ist dem Element kein Fenster zugeordnet wird 0 zurückgegeben.

HideRibbon(handle)

handle...Handle eines Ribbons

Macht ein Ribbon unsichtbar. Es wird nicht neu gezeichnet, bis es wieder sichtbar gemacht wird.

LinkRibbonPopup(id, @title, x, y)

id...ID eines Elements

title...Titel des Fensters

x,y...Größe des Fensters

Rückgabe: ID des Popupfensters

Ordnet einem Element ein Popupfenster zu. Dieses Fenster kann nachher mit allerlei Steuerelemente gefüllt werden (siehe `MoveControlToRibbonPopup()`).

MoveControlToRibbonPopup(popup, handle, x, y)

popup...ID eines Popupfensters

handle...Handle des Steuerelements, das verschoben werden soll

x,y...Neue Position des Steuerelements innerhalb des Popupfensters

Verschiebt ein Steuerelement in ein Popupfenster. Die Schriftart und -größe wird automatisch an das neue Fenster angepasst.

MovePopupText(id, tid, x, y)

id...ID des Popups

tid...ID des Text innerhalb des Popups

x,y...neue Position

Verschiebt einen Text innerhalb eines Popups.

RemovePopupText(id, tid)

id...ID eines Popupfensters

tid...ID eines Textes im Popupfenster

Entfernt einen Text aus einem Popupfenster.

RemoveRibbon(ribbon)

ribbon...Handle eines Ribbons

Entfernt ein Ribbon.

RemoveRibbonItem(id)

id...ID eines Elements

Entfernt ein Ribbon-Element.

RemoveRibbonItem(ribbon)

ribbon...Handle eines Ribbons

Zeichnet ein Ribbon neu.

ResizeRibbonPopup(id, x, y)

id...ID des Popups

x,y...neue Größe

Weist einem Ribbon eine neue Größe zu.

RibbonVersion(type, @text)

type...Typ der zu ermittelnden Eigenschaft

text...Ergebnisstring (oder Bereich)

Rückgabe: Länge des Textes

Ermittelt die Version der Ribbon.dll. Type gibt dabei an, was ermittelt werden soll (siehe PH-Datei unter `Ribbon_Version_...`). Auch hier muss der Text vorher entsprechend dimensioniert werden.

SetCallbackMessage(message)

message...Message, die bei einem Ereignis gesendet wird.

Ändert die Message, die bei einem Ereignis gesendet wird. Standardmäßig ist es 10000, im Bedarfsfall kann sie hiermit angepasst werden.

SetRibbonCheckboxImage(ribbon, type, image)

ribbon...Handle eines Ribbons

type...gibt an welche Checkbox ersetzt werden soll

image...Handle eines Bildes, das verwendet werden soll

Weist den Checkboxen eines Ribbons ein neues Bild zu. Mit type kann genau festgelegt werden, welches Bild ersetzt werden soll (siehe PH-Datei unter Ribbon_CheckboxImage_...). Das Originalbild wird nicht verändert und kann nach diesem Befehl freigegeben werden.

SetRibbonFont(ribbon, @font)

ribbon...Handle eines Ribbons

font...Name der neuen Schriftart

Ändert die Schriftart eines Ribbons. Dieser Befehl ist nicht empfohlen, da die Darstellung des Ribbons gestört werden kann. Eventuell wird in der Zukunft noch daran gearbeitet, dass weniger Darstellungsfehler auftreten.

SetRibbonItemImage(id, status)

id...ID eines Elements

status...Checkstatus des Elements

Setzt den Checkstatus des Elements. Es wird auch sichergestellt, dass nur ein Flag aktiv ist (selbst wenn status mehrere Flags enthält). Zur Priorität und den möglichen Werten siehe GetRibbonItemImage().

SetRibbonItemImage(id, image)

id...ID eines Elements

image...Handle eines Bildes

Setzt ein neues Bild für ein Element.

SetRibbonItemPosition(id, position)

id...ID eines Elements

position...neue Position

Verschiebt ein Element an eine neue Position innerhalb seiner Hierarchie. Das erste Element hat die Position 0.

SetRibbonItemStatus(id, status)

id...ID eines Elements

status...neuer Status

Setzt eine neue Kombination von Statusflags eines Elements. Die Flags stehen in der PH-Datei (~Ribbon_Status_...).

SetRibbonItemText(id, @text)

id...ID eines Elements

text...Adresse eines Bereichs oder Strings, der ausreichend dimensioniert ist

Setzt einen neuen Text für ein Element.

SetRibbonMetric(handle, metric)

handle...Handle eines Ribbons

metric...Die zu setzende Eigenschaft

Rückgabe: Aktueller Zustand der Eigenschaft

Setzt den Zustand einer Eigenschaft eines Ribbons (siehe PH-Datei unter ~Ribbon_Metric_...).

SubRibbonItemStatus(id, status)

id...ID des neuen Items

status...Zu subtrahierendes Statusflag

Löscht ein bestimmtes Statusflag eines Ribbon-Items. Es wird dafür gesorgt, dass das Statusflag deaktiviert ist. Sollte es bereits deaktiviert sein ändert sich nichts. Die einzelnen Statusflags können bei [AddRibbonItem\(\)](#) nachgeschlagen werden.

UnlinkRibbonPopup(id)

id...ID eines Elements

Rückgabe: 0 bei Misserfolg, 1 bei Erfolg

Entfernt ein einem Steuerelement zugeordnetes Popup-Fenster.