

Beschreibung zur Graphamp.Dll

Inhalt

- 1 . Wozu dient die Dll ?
- 2 . Internes
- 3 . Ressourcen
- 4 . Funktionen
- 5 . Filter
- 6 . Defintionen für Profan²
- 7 . Tips und Tricks
- 8 . Rechtliches
- 9 . Freischalten

Das ganze Paket beinhaltet folgende Dateien :

- a) Die Graphamp.dll
- b) Ein Testprogramm (PRF), um die Funktionen zu testen (Profan² Version 7 - ist aber leicht anzupassen)
- c) Eine Dll mit zwei Testbildern.
- d) Diese PDF-Datei
- e) Reg_Key.exe zum Freischalten der Dll
- f) Profan-Header-Datei
- g) Ein paar Beispiel - Quellcodes

Besonderheiten:

Die Dll ist auch in der Lage, die Bitmaps aus einem Modul zu laden (DLL oder EXE).

Diese Ressourcen sind ja mit dem Resourcehacker relativ leicht in eine Dll zu packen.

Beim Laden aus Modulen sind aber ein paar Kleinigkeiten zu beachten:

1. Der Ressourcenname muss, wenn die Resource nur eine Nummer hat folgendermaßen aussehen "#100".
2. Die Extension muss unbedingt richtig angegeben werden. Ist die Resource ein Bitmap und die Extension JPG dann kommt es garantiert zu einer Fehlermeldung.

Die Dll kann ein eigenes Bitmap-Format erzeugen und natürlich auch wieder einlesen.

Dieses Format kann nicht von anderen Grafikprogrammen gelesen werden.

Es eignet sich also ganz gut dazu seine Grafiken zu schützen oder als Resource in Programme einzubinden.

Ausserdem wird das Bitmap komprimiert.

Je nach Größe der Originals liegt der Komprimierungsfaktor zwischen 40% und 50%.

Die Endung ist PRB (eben für **Profan-Bitmap**).

Gravierende Änderungen zu Vorgängerversionen:

Stretch und SetLoadsize sind jetzt auch für Metafiles gültig.

Die Funktion P_LoadImage ist erweitert worden. Der letzte Parameter kann jetzt auch 3 oder 4 sein.

Ein Fehler bei den Clipboard-Funktionen wurde bereinigt.

Ein paar neue Funktionen.

Ein besonderer Dank geht an dieser Stelle an Bernhard Künzel für seine umfangreichen Tests, Fehlermeldungen und Verbesserungsvorschläge.

Wozu dient die DII ?

Einfach gesagt zum Laden, Speichern und Manipulieren von Pixel-Grafiken, die vom aufrufenden Programm nicht standardmässig unterstützt werden.

21 Effekte und 25 Filter können zur Manipulation der Bitmaps, sowie 7 Filter zum Resamplen der Bitmaps eingesetzt werden.

Ladbare Formate :

- 1 . BMP
- 2 . RLE
- 3 . JPG -progressiv / nicht progressiv
- 4 . TIF
- 5 . WMF
- 6 . EMF
- 7 . TGA
- 8 . PCX
- 9 . PCD
10. PNG
11. GIF (keine animierten)
12. PRB
13. Bitmaps vom Scanner(Twain)
14. ICO
15. CUR
16. ANI

Speicherbare Format :

- 1 . BMP
- 2 . JPG
- 3 . PNG
- 4 . PRB
- 5 . RLE

Internes

Die DLL muss beim ersten Aufruf initialisiert werden. Hier werden interne Arrays angelegt.

Die DLL kann bis 500 Bitmaps im Speicher halten. Es ist aber nicht unbedingt ratsam 500 grosse Bilder zu laden, das kann zum Systemabsturz führen. Bei kleineren Bitmaps sollte es aber keine Probleme geben. So kann man z.B. Animationen erzeugen.

Beim Programmende ist unbedingt darauf zu achten, dass die geladenen Bitmaps wieder freigegeben werden. Sonst werden die GDI-Ressourcen erst beim Herunterfahren von Windows freigegeben.

Die DLL gibt beim Laden von Bitmaps immer nur das Handle zurück. Was der Programmierer damit anfängt ist dann seine Sache.

Resourcen

Es gibt eine zusätzliche Resource in der Dll, und zwar einen Mauscursor, das konnte ich mir als alter Atarianer nicht verkneifen.

Funktionen

Insgesamt hat die Dll 78 Funktionen.
Im einzelnen sind das :

InitDLL

initialisiert die Dll

Parameter : Integer (Anzahl der möglichen Controls)

Rückgabe : Falsch oder Wahr (0 oder 1) --> Bool

IsInit

ist die Dll initialisieren ??

Rückgabe : Falsch oder Wahr (0 oder 1) --> Bool

ExitDll

Freigabe der Ressourcen, muss auf jeden Fall vor

Programm-Ende aufgerufen werden !! Es muss unbedingt

darauf geachtet werden, dass das Programm korrekt beendet wird !!

Rückgabe : Falsch oder Wahr (0 oder 1) --> Bool

CreateTempfilename

Temporären Filenamen ermitteln

Könnte man vielleicht für UNDO-Files benutzen ?

Parameter : Zeiger auf String (max. 256 Zeichen)

Rückgabe : Falsch oder Wahr (0 oder 1) --> Bool

GetDllInfo

Ausgabe von Informationen

Rückgabe : Falsch oder Wahr (0 oder 1) --> Bool

P_LoadImage

es können maximal 500 Handles im Speicher gehalten werden

Ladbare Formate :

BMP,RLE,JPG,GIF,TIF,PCX,TGA,PCD,WMF,EMF,PRB,ICO,CUR,ANI

Parameter : Elternfenster (%hwnd)

: Zeiger auf Dateinamen

: Index des Bitmaps

: Zeiger auf LongInt --> hier wird die Breite zurückgegeben

: Zeiger auf LongInt --> hier wird die Höhe zurückgegeben

: LongInt (Stretch to Window) --> passt das Bitmap proportional an das Elternfenster an

> 0 = Originalgrösse

> 1 = ins Elternfenster eingepasst

> 3 = ins Elternfenster gestreckt

> 4 = in ein Rechteck gestreckt siehe **P_SetLoadSize**

> mit den neuen Funktionen **P_SetMaxZoom** und **P_SetMaxStretch** kann das Ladeverhalten beeinflusst werden.

Rückgabe : BitmapHandle, Cursor-Handle, Icon-Handle oder Metafile-Handle

P_LoadImageFromModul

lädt ein Bitmap aus einer DLL oder EXE

Ladbare Formate :

BMP,JPG,GIF,TIF,PCX,TGA,PCD,WMF,PRB

Parameter : Elternfenster (%hwnd)

: Index des Bitmaps

: Zeiger auf LongInt --> hier wird die Breite zurückgegeben

: Zeiger auf LongInt --> hier wird die Höhe zurückgegeben

: LongInt (Stretch to Window) --> passt das Bitmap proportional an das Elternfenster an

: Modulname (Zeiger auf String)

: ResourceName (Zeiger auf String)

: ResourceType (Zeiger auf String)

: Extension (ohne Punkt nur z.B. ".jpg")

Rückgabe : BitmapHandle

P_GetBitmapInfo

Liefert Informationen zum Bitmap

Parameter : Zeiger auf String (Filename)

: Zeiger auf LongInt (Breite)

: Zeiger auf LongInt (Höhe)

: Zeiger auf LongInt (Pixelformat)

P_FreeBitmap

Löscht ein Bitmap aus dem Speicher

Parameter : Index des Bitmaps

P_StretchToFit

Streckt das Bild ins Elternfenster

Es ist unbedingt darauf zu achten, dass der BildIndex stimmt, sonst wird das vorhandene Bild zerstört.

Parameter : HWND(LongInt)

: BildIndex(LongInt)

Rückgabe : Bitmaphandle(LongInt)

P_Greyscale

wandelt das Bitmap in Graustufen

Es ist unbedingt darauf zu achten, dass der BildIndex stimmt, sonst wird das vorhandene Bild zerstört.

Parameter : BildIndex(LongInt)

Rückgabe : Bitmaphandle(LongInt)

P_Negativ

kehrt die Farben um

Es ist unbedingt darauf zu achten, dass der BildIndex stimmt, sonst wird das vorhandene Bild zerstört.

Parameter : BildIndex(LongInt)

Rückgabe : Bitmaphandle(LongInt)

P_Contrast

Kontrast einstellen

Es ist unbedingt darauf zu achten, dass der BildIndex stimmt, sonst wird das vorhandene Bild zerstört.

Parameter : BildIndex(LongInt)

: Wert (LongInt)

Rückgabe : Bitmaphandle(LongInt)

P_Lightness

Helligkeit einstellen

Es ist unbedingt darauf zu achten, dass der BildIndex stimmt, sonst wird das vorhandene Bild zerstört.

Parameter : BildIndex(LongInt)

: Wert (LongInt)

Rückgabe : Bitmaphandle(LongInt)

P_Mirror

spiegelt das Bitmap horizontal

Es ist unbedingt darauf zu achten, dass der BildIndex stimmt, sonst wird das vorhandene Bild zerstört.

Parameter : BildIndex(LongInt)

Rückgabe : Bitmaphandle(LongInt)

P_Flip

spiegelt das Bitmap vertikal

Es ist unbedingt darauf zu achten, dass der BildIndex stimmt, sonst wird das vorhandene Bild zerstört.

Parameter : BildIndex(LongInt)

Rückgabe : Bitmaphandle(LongInt)

P_Resample

ändert die Bitmapgröße

Es ist unbedingt darauf zu achten, dass der BildIndex stimmt, sonst wird das vorhandene Bild zerstört.

Parameter : BildIndex(LongInt)

: neue Breite (LongInt)

: neue Höhe (LongInt)

: Filter (LongInt) *siehe Filter*

Rückgabe : Bitmaphandle(LongInt)

P_Crop

schneidet einen Bereich aus

Es ist unbedingt darauf zu achten, dass der BildIndex stimmt, sonst wird das vorhandene Bild zerstört.

Parameter : BildIndex(LongInt)

: Left (LongInt)

: Top (LongInt)

: Right (LongInt)

: Bottom (LongInt)

Rückgabe : Bitmaphandle(LongInt)

P_SetToClipboard

speichert das Bitmap in der Zwischenablage

Es ist unbedingt darauf zu achten, dass der BildIndex stimmt, sonst wird das vorhandene Bild zerstört.

Parameter : BildIndex(LongInt)

Rückgabe : Bitmaphandle(LongInt)

P_GetFromClipboard

holt das Bitmap aus der Zwischenablage

Es ist unbedingt darauf zu achten, dass der BildIndex stimmt, sonst wird das vorhandene Bild zerstört.

Parameter : BildIndex(LongInt)

Rückgabe : Bitmaphandle(LongInt)

P_SetJpgOptions

setzt die Jpeg-Optionen

Parameter : Speicherqualität (Longint) 0% - 100%

P_SaveBitmap

speichert das Bitmap

Parameter : Index

: Dateiname

P_AssignBitmap

Legt ein neues Bitmap im Array an und kopiert das Bitmap mit INDEX[] hinein. Vielleicht für UNDO-Funktion ??

Parameter : Index (alt)

: Index (neu)

Rückgabe : Bitmaphandle(LongInt)

P_ConvertTo256

Konvertiert das Bitmap mit INDEX in ein 256 Farben-Bitmap

Parameter : Index

Rückgabe : Bitmaphandle(LongInt)

P_AssignExtBitmap

Legt ein neues Bitmap im Array an und kopiert das Bitmap mit HANDLE hinein. Vielleicht für Scanmodule geeignet ??

Parameter : HWND

: Index (alt)

: Handle (neu)

: LongInt (Stretch to Window) --> passt das Bitmap proportional an das Elternfenster an

Rückgabe : Bitmaphandle(LongInt)

P_CreateEmptyBitmap

Legt ein neues Bitmap (leer)im Array an.

Mit P_AssignExtBitmap vielleicht für Scanmodule geeignet ??

Parameter : Index (alt)

Rückgabe : Bitmaphandle(LongInt)

P_Rotate

Rotiert das Bitmap mit INDEX und GRAD

Parameter : INDEX

: GRAD

: Hintergrundfarbe

Rückgabe : Bitmaphandle

P_Blend

Blendet 2 Bitmap ineinander

ist nur sinnvoll wenn beide Bitmaps gleich gross sind.

sonst wird Bitmap2 an die Groesse von Bitmap 1 angepasst.

Parameter : Index 1

: Index 2

: MaxBlendValue(LongInt)

: Multiplikator(LongInt)

Rückgabe : Bitmaphandle(LongInt)

P_Filter

wendet einen Filter an

Parameter : Index 1

: Effekt (1-25) siehe **Filter**

Rückgabe : Bitmaphandle(LongInt)

P_IsTwain

fragt ab, ob ein Twain-Modul installiert ist

Rückgabe : Bool

P_ScanToBitmap

scannt ein Bitmap ein.

Parameter : HWND

: Index

: Quelle wählen oder nicht (1 oder 0)

: Ohne Manager (1 oder 0)

: Clipboard (1 oder 0)

Rückgabe : Bitmaphandle(LongInt)

P_Gamma

Gamma-Korrektur

Parameter : Index

: Wert (Vorkomma)

: Wert (Nachkomma)

Rückgabe : Bitmaphandle(LongInt)

P_Colorize

Kolorierung

Parameter : Index

: Wert (Hue)

: Wert (Saturation)

: Wert (Lightness)

Rückgabe : Bitmaphandle(LongInt)

P_Solarize

Solarisation

Parameter : Index

: Wert (Solarisation)

Rückgabe : Bitmaphandle(LongInt)

P_Spray

Sprühdose

Parameter : Index

: Wert

Rückgabe : Bitmaphandle(LongInt)

P_IncDecRGB

Farben einzeln korrigieren

Parameter : Index

: Wert (Rot)

: Wert (Grün)

: Wert (Blau)

Rückgabe : Bitmaphandle(LongInt)

P_PrintBitmap

Druck das Bitmap mit INDEX auf DC

Das kann entweder %HDC,%HDC2,ein DruckerDeviceContext oder iergendein anderer mit GetDC ermittelter Device-Context sein. X,Y,W,H bezeichnen das Rechteck in das gedruckt wird.

Parameter : Index

: DC

: x

: y

: w

: h

: Flag fuer Transparenz (-1 = transparent, 0 = normal)

Rückgabe : Bitmaphandle

P_Arithmetic

verbindet 2 Bitmaps Mit INDEX unter Verwendung von FILTER

Parameter : Index 1

: Index 2

: FILTER (1-12) **siehe Filter**

Rückgabe : Bitmaphandle(LongInt)

P_AddNoise

gibt ein Rauschen über das Bitmap

Parameter : Index

: Wert

: MONO (1 oder 0) wahr oder falsch

Rückgabe : Bitmaphandle(LongInt)

P_AntiAlias

entfernt Moire-Effekte

Parameter : Index

: R (1 oder 0)

: G (1 oder 0)

: B (1 oder 0)

Rückgabe : Bitmaphandle(LongInt)

P_Mosaic

erzeugt ein Mosaic aus dem Bitmap

Parameter : Index

: Breite(Integer)

: Höhe(Integer)

Rückgabe : Bitmaphandle(LongInt)

P_GetLoadFilename

interner Laden-Dialog mit Bildvorschau

Parameter : Zeiger auf String, der den Filenamen zurückgibt

Rückgabe : Bool

P_GetSaveFilename

interner Speichern-Dialog mit Bildvorschau

Parameter : Zeiger auf String, der den Filenamen zurückgibt

Rückgabe : Bool

P_SetJpgOptions

setzt die Jpeg-Optionen

Standard 24Bit-nicht progressiv

Parameter : Speicherqualität (Longint) 0% - 100%

P_SetExtJpgOptions

setzt Lade und Speicheroptionen für das JPG-Format

			Laden	Speichern
Parameter :	BitTiefe	8 = 8Bit 24 = 24Bit	X	X
	Qualität	0 = Speed 1 = Qualität	X	
	Size	0 = Full 1 = Half	X	
		2 = Viertel 3 = Achtel	X	
	Graustufen	0 = nein 1 = ja	X	
	Weich	0 = nein 1 = ja	X	
	Progressiv	0 = nein 1 = ja	X	
	Kompressionsstufe	1% - 100%		X

Besonderheiten bei Size:

Das Bitmap wird je nach Angabe geladen, also

Full = Originalgrösse

Half = Halbe-Originalgrösse

usw.

'je kleiner Size ist desto schneller wird das Bitmap geladen

'eigentlich ideal um Thumbnails anzulegen

P_GetLoadedBitmapInfo

Liefert Informationen zum Bitmap

Parameter : Index des geladenen Bitmaps

: Zeiger auf LongInt (Breite)

: Zeiger auf LongInt (Höhe)

: Zeiger auf LongInt (Pixelformat)

P_GetDPI

Parameter : Zeiger auf String (Filename)

Es werden folgende Formate unterstützt : BMP,RLE,JPG,PCX

Rückgabe : DPI (Integer)

P_SetRLECompressValue

Parameter : Setzt die Kompressionsrate beim Speichern als RLE

je kleiner desto schlechter die Bildqualität

mögliche Werte: 1-100

Beim Speichern muss darauf geachtet werden, dass hier nur 256Farben_Bitmaps unterstützt werden. Da sollte man eventuell eine Abfrage einbauen, sonst kommt garantiert eine Fehlermeldung wenn das Format nicht stimmt !

P_GetWidth

Ermittelt die Breite des geladenen Bitmaps

Parameter : Bitmap-Index

P_GetHeight

Ermittelt die Höhe des geladenen Bitmaps

Parameter : Bitmap-Index

P_GetColorBit

Ermittelt die Farbtiefe des geladenen Bitmaps

Parameter : Bitmap-Index

P_SetLoadSize

Setzt die maximale Ladebreite und Ladehöhe der Bitmaps (global)

Parameter : Breite (LongInt)

: Höhe (LongInt)

Rückgabe : LongInt (LoWord = Breite, HiWord = Höhe)

P_GetLoadSize

Ermittelt die maximale Ladebreite der Bitmaps (global)

Rückgabe : LongInt (LoWord = Breite, HiWord = Höhe)

P_PlayMeta

Zeichnet MetaFiles auf DC (WMF und EMF)

Parameter : DC (Zeiger auf LongInt)

: Filename (Zeiger auf String)

: X

: Y

: Breite

: Höhe

Rueckgabe: Bool

P_GetMetaDims

Ermittelt Breite und Höhe von MetaFiles

Parameter : Filename (Zeiger auf String)

Rückgabe : LongInt

LoWord = Breite

HiWord = Höhe

P_FreeMeta

Löscht ein Metafile aus dem Speicher

Parameter : Index des Metafiles

P_FreeIcon

Löscht ein Icon aus dem Speicher

Parameter : Index des Icons

P_FreeCursor

Löscht einen Cursor dem Speicher

Parameter : Index des Cursors

P_IsBitmap

Stellt fest ob das Handle-Format ein Bitmap ist

Parameter : Index

Rueckgabe : Bool

P_IsIcon

Stellt fest ob das Handle-Format ein Icon ist

Parameter : Index

Rueckgabe : Bool

P_IsCursor

Stellt fest ob das Handle-Format ein Cursor ist

Parameter : Index

Rueckgabe : Bool

P_IsMeta

Stellt fest ob das Handle-Format ein Metafile ist

Parameter : Index

Rueckgabe : Bool

P_GetMetaFilename

interner Laden-Dialog fuer Metafiles mit Bildvorschau

Parameter : Zeiger auf String, der den Filenamen zurückgibt

Rückgabe : Bool

P_GetMultiLoadFilename

interner Laden-Dialog mit Mehrfachauswahl

Parameter : Zeiger auf String, der die Filenamen(Komma getrennt) zurückgibt

es ist darauf zu achten, dass der Rueckgabe-Bereich gross genug ist

Rückgabe : Anzahl der gewaehlten Dateien oder -1

Trennzeichen ist die PIPE (|)

P_GetMultiMetaFilename

interner Laden-Dialog fuer Metafile mit Mehrfachauswahl

Parameter : Zeiger auf String, der die Filenamen(Komma getrennt) zurückgibt

es ist darauf zu achten, dass der Rueckgabe-Bereich gross genug ist

Rückgabe : Anzahl der gewaehlten Dateien oder -1

Trennzeichen ist die PIPE (|)

P_ConvertMetaToBitmap

konvertiert ein Metafile-Bild in Bitmap

Parameter : Index

Rueckgabe : Bitmaphandle

Das Metafilehandle wird dabei aus dem Speicher geloescht

LoadSize wird beruecksichtigt !

P_GetUserLoadFilename

benutzerdefinierter Lade-Dialog

Mehrfachauswahl oder Einfachauswahl

Eigener Titel

Eigene Dateiauswahl

Parameter : Zeiger auf String der die Rueckgabe aufnimmt

Es ist darauf zu achten das der Rueckgabepuffer

gross genug dimensioniert wird damit er bei

Mehrfachauswahl alle Strings aufnehmen kann

Die gewählten Einträge werden Komma-getrennt zurückgegeben

und sind dementsprechend auszulesen (Substr(...))

: Zeiger auf String der den Titel aufnimmt

: Zeiger auf String der die Auswahl festlegt

Beispiel :

Choose\$ = "Alle unterstützten Dateien |*.bmp;*.jpg;*.prb|"

Choose\$ = Choose\$+"Bitmap |*.bmp|"

Choose\$ = Choose\$+"Jpg |*.jpg|"

: LongInt 1 oder 0 (Mehrfachauswahl(1) oder Einfachauswahl(0))

: LongInt bezeichnet den voreingestellten FilterIndex

Rueckgabe : LongInt der die Anzahl der gewählten Einträge aufnimmt

Trennzeichen ist die PIPE (|)

P_GetUserSaveFilename

benutzerdefinierter Speichern-Dialog

Eigener Titel

Eigene Dateiauswahl

Parameter : Zeiger auf String der die Rueckgabe aufnimmt

: Zeiger auf String der den Titel aufnimmt

: Zeiger auf String der die Auswahl festlegt

Beispiel :

Choose\$ = "Alle unterstützten Dateien |*.bmp;*.jpg;*.prb|"

Choose\$ = Choose\$+"Bitmap |*.bmp|"

Choose\$ = Choose\$+"Jpg |*.jpg|"

: Vorgabe-String fuer Dateinamen

: LongInt bezeichnet den voreingestellten FilterIndex

Rueckgabe : 1 oder -1

P_QuickLoadJPG

schnelles Laden von JPG's, speziell fuer Thumbnails

Das Jpg wird in ein Rechteck eingepasst

Breite und Hoehe des Rechtecks werden bestimmt durch P_SetLoadSize(..)

Parameter : Index

Zeiger auf String (Filename)

Rueckgabe : Bitmaphandle

P_IsJpgProgressiv

ermittelt ob ein Jpg progressiv gespeichert ist

Parameter : Zeiger auf String (Dateiname)

Rückgabe : Bool (Ja = 1 , Nein = 0)

P_SetMaxZoom

hiermit kann der maximale Zoom für das zu ladende Bild angegeben werden.

Hat nur Auswirkung wenn bei P_LoadImage der letzte Parameter 3 oder 4 ist

Das Bitmap wird nur dann ins Ziel eingepasst wenn innerhalb der Prozentangabe die angegebene maximale Ladegrösse erreicht wird !

Parameter : Prozentangabe (LongInt)

Rückgabe : LongInt

P_SetMaxStretch

hiermit kann der Streckungsfaktor für das zu ladende Bild angegeben werden.

Hat nur Auswirkung wenn bei P_LoadImage der letzte Parameter 3 oder 4 ist

Das Bitmap wird nur dann ins Ziel gestreckt wenn innerhalb der Prozentangabe ein vorhandener freier Rand ausgefüllt werden kann.

Parameter : Prozentangabe (LongInt)

Rückgabe : LongInt

P_GetMaxZoom

maximalen Zoom-Faktor ermitteln

Rückgabe : LongInt

P_GetMaxStretch

maximalen Stretch-faktor ermitteln

Rückgabe : LongInt

P_PlayHMeta

Zeichnet ein geladenes Metafile ins angegebene Fenster wobei X,Y,Breite und Hoehe ein Rechteck bestimmen.

Parameter : 1 = Index
 2 = Fensterhandle
 3 = X
 4 = Y
 5 = Breite
 6 = Höhe

P_CheckFiles

Da bei fehlerhaften Dateien das Programm abraucht kann man mit dieser Funktion testen ob das Bild geladen werden kann. Das kann je nach Grösse des Bildes allerdings entsprechend lange dauern. Bei normalgrossen Bildern (JPG 1024 X 768) ist allerdings kaum ein Geschwindigkeitsverlust zu bemerken.

Parameter : Zeiger auf String (Filename)
Rueckgabe : Bool (ja oder nein)

P_RegisterClipMessage

Hiermit kann eine Usermessage bestimmt werden die beim Füllen der Zwischenablage an des Profan-Fenster abgeschickt wird.

Parameter : 1 = Fensterhandle
 2 = Message

P_HasClipBitmap

die Funktion ermittelt ob dei Zwischenablage ein Bitmap enthält
Rückgabe : Bool (ja oder nein)

Filter

Insgesamt gibt es 25 Filter-Funktion mit denen das Bitmap manipuliert werden kann.

Im einzelnen :

- 1 . EmbossColor
- 2 . EmbossLight
- 3 . EmbossMedium
- 4 . EmbossDark
- 5 . Enhance
- 6 . BlurBartlett
- 7 . BlurGaussian
- 8 . Negative
- 9 . Average
- 10 . Blur
- 11 . BlurSoftly
- 12 . BlurMore
- 13 . Prewitt
- 14 . TraceContour
- 15 . Sharpen
- 16 . SharpenMore
- 17 . SharpenLess
- 18 . UnSharpMask
- 19 . EdgesStrong
- 20 . EdgesWeak
- 21 . Etch
- 22 . LaplacianHV
- 23 . LaplacianOmni
- 24 . SharpenDirectional
- 25 . SobelPass

Arithmetic-Filter :

- 1 . None;
- 2 . Add;
- 3 . Subtract;
- 4 . Multiply;
- 5 . Divide;
- 6 . Darkest;
- 7 . Lightest;
- 8 . Difference;
- 9 . BinaryOr;
- 10 . BinaryAnd;
- 11 . Average;

Resample-Filter :

- 1 . SplineFilter
- 2 . BellFilter
- 3 . TriangleFilter
- 4 . BoxFilter
- 5 . HermiteFilter
- 6 . Lanczos3Filter
- 7 . MitchellFilter

Definitionen für Profan²

```
DEF Init_dll(1) ! "Graphamp.dll", "InitDll"
DEF Is_Init(0) ! "Graphamp.dll", "IsInit"
DEF Exit_Dll(0) ! "Graphamp.dll", "ExitDll"
DEF Create_Tempfilename(1) ! "Graphamp.dll", "CreateTempfilename"
DEF GetDllInfo(0) ! "Graphamp.dll", "GetDllInfo"
DEF P_LoadImage(6) ! "Graphamp.dll", "P_LoadImage"
DEF P_LoadImageFromModul(9) ! "Graphamp.dll", "P_LoadImageFromModul"
DEF P_GetBitmapInfo(4) ! "Graphamp.dll", "P_GetBitmapInfo"
DEF P_FreeBitmap(1) ! "Graphamp.dll", "P_FreeBitmap"
DEF P_StretchToFit(2) ! "Graphamp.dll", "P_StretchToFit"
DEF P_Greyscale(1) ! "Graphamp.dll", "P_Greyscale"
DEF P_Negativ(1) ! "Graphamp.dll", "P_Negativ"
DEF P_Contrast(2) ! "Graphamp.dll", "P_Contrast"
DEF P_Lightness(2) ! "Graphamp.dll", "P_Lightness"
DEF P_Mirror(1) ! "Graphamp.dll", "P_Mirror"
DEF P_Flip(1) ! "Graphamp.dll", "P_Flip"
DEF P_Resample(4) ! "Graphamp.dll", "P_Resample"
DEF P_Crop(5) ! "Graphamp.dll", "P_Crop"
DEF P_SetToClipboard(1) ! "Graphamp.dll", "P_SetToClipboard"
DEF P_GetFromClipboard(1) ! "Graphamp.dll", "P_GetFromClipboard"
DEF P_SaveBitmap(2) ! "Graphamp.dll", "P_SaveBitmap"
DEF P_AssignExtBitmap(4) ! "Graphamp.dll", "P_AssignExtBitmap"
DEF P_CreateEmptyBitmap(1) ! "Graphamp.dll", "P_CreateEmptyBitmap"
DEF P_Rotate(3) ! "Graphamp.dll", "P_Rotate"
DEF P_Blend(4) ! "Graphamp.dll", "P_Blend"
DEF P_Filter(2) ! "Graphamp.dll", "P_Filter"
DEF P_ScanToBitmap(5) ! "Graphamp.dll", "P_ScanToBitmap"
DEF P_Gamma(3) ! "Graphamp.dll", "P_Gamma"
DEF P_Solarize(2) ! "Graphamp.dll", "P_Solarize"
DEF P_IncDecRGB(4) ! "Graphamp.dll", "P_IncDecRGB"
DEF P_PrintBitmap(7) ! "Graphamp.dll", "P_PrintBitmap"
DEF P_Arithmetic(3) ! "Graphamp.dll", "P_Arithmetic"
DEF P_ConvertTo256(1) ! "Graphamp.dll", "P_ConvertTo256"
DEF P_AddNoise(3) ! "Graphamp.dll", "P_AddNoise"
DEF P_AntiAlias(4) ! "Graphamp.dll", "P_AntiAlias"
DEF P_Mosaic(3) ! "Graphamp.dll", "P_Mosaic"
DEF P_IsTwain(0) ! "Graphamp.dll", "P_IsTwain"
DEF P_GetLoadFilename(1) ! "Graphamp.dll", "P_GetLoadFilename"
DEF P_GetSaveFilename(1) ! "Graphamp.dll", "P_GetSaveFilename"
DEF P_SetPngOptions(3) ! "Graphamp.dll", "P_SetPngOptions"
DEF P_SetJpgOptions(1) ! "Graphamp.dll", "P_SetJpgOptions"
DEF P_SetExtJpgOptions(7) ! "Graphamp.dll", "P_SetExtJpgOptions"
DEF P_GetLoadedBitmapInfo(4) ! "Graphamp.dll", "P_GetLoadedBitmapInfo"
DEF P_GetDPI(1) ! "Graphamp.dll", "P_GetDPI"
DEF P_SetRLECompressValue(1) ! "Graphamp.dll", "P_SetRLECompressValue"
DEF P_GetWidth(1) ! "Graphamp.dll", "P_GetWidth"
DEF P_GetHeight(1) ! "Graphamp.dll", "P_GetHeight"
DEF P_GetColorBit(1) ! "Graphamp.dll", "P_GetColorBit"
DEF P_SetLoadSize(2) ! "Graphamp.dll", "P_SetLoadSize"
DEF P_GetLoadSize(0) ! "Graphamp.dll", "P_GetLoadSize"
DEF P_PlayMeta(6) ! "graphamp.dll", "P_PlayMeta"
DEF P_GetMetaDims(1) ! "graphamp.dll", "P_GetMetaDims"
DEF P_FreeMeta(1) ! "graphamp.dll", "P_FreeMeta"
DEF P_Freelcon(1) ! "graphamp.dll", "P_Freelcon"
DEF P_FreeCursor(1) ! "graphamp.dll", "P_FreeCursor"
DEF P_IsBitmap(1) ! "graphamp.dll", "P_IsBitmap"
DEF P_IsIcon(1) ! "graphamp.dll", "P_IsIcon"
```

```
DEF P_IsCursor(1) ! "graphamp.dll", "P_IsCursor"
DEF P_IsMeta(1) ! "graphamp.dll", "P_IsMeta"
DEF P_GetMetaFilename(1) ! "graphamp.dll", "P_GetMetaFilename"
DEF P_GetMultiLoadFilename(1) ! "Graphamp.dll", "P_GetMultiLoadFilename"
DEF P_GetMultiMetaFilename(1) ! "Graphamp.dll", "P_GetMultiMetaFilename"
DEF P_ConvertMetaToBitmap(1) ! "Graphamp.dll", "P_ConvertMetaToBitmap"
DEF P_GetUserLoadFilename(5) ! "Graphamp.dll", "P_GetUserLoadFilename"
DEF P_GetUserSaveFilename(5) ! "Graphamp.dll", "P_GetUserSaveFilename"
DEF P_QuickLoadJPG(2) ! "Graphamp.dll", "P_QuickLoadJPG"
DEF P_IsJpgProgressiv(1) ! "Graphamp.dll", "P_IsJpgProgressiv"

DEF P_SetMaxZoom(1) ! "Graphamp.dll", "P_SetMaxZoom"
DEF P_SetMaxStretch(1) ! "Graphamp.dll", "P_SetMaxStretch"
DEF P_GetMaxZoom(0) ! "Graphamp.dll", "P_GetMaxZoom"
DEF P_GetMaxStretch(0) ! "Graphamp.dll", "P_GetMaxStretch"
DEF P_PlayHMeta(6) ! "Graphamp.dll", "P_PlayHMeta"
DEF P_CheckFiles(1) ! "Graphamp.dll", "P_CheckFiles"
DEF P_RegisterClipMessage(2) ! "Graphamp.dll", "P_RegisterClipMessage"
DEF P_HasClipBitmap(0) ! "Graphamp.dll", "P_HasClipBitmap"
```

Tips und Tricks

Bilder bearbeiten

Normalerweise lädt man das Bild so ein, dass man es auch gut ansehen kann, also an das Elterncontrol angepasst. Das hat aber zur Folge, dass es nicht mehr unbedingt die Originalausmaße besitzt, sondern verkleinert wurde. Wenn ich das Bild jetzt mit einem Filter bearbeite kann ich nur das verkleinerte Bild abspeichern.

Abhilfe : Das Bild zweimal laden. Einmal zum anschauen und einmal nur ins Speicherarray. Alles, was ich mit dem "Vorschaubild" mache, kann ich im Speicher mit dem 2.Bild proportional auch machen.

Slideshow

Es ist ein Leichtes mit der DLL eine Slideshow zu programmieren. Einfach alle Files die man anschauen möchte in eine Liste einlesen und diese abarbeiten. Es gibt allerdings etwas zu beachten.

Beachte : Normalerweise wird die Slideshow in einer Schleife angezeigt. Und hier sollte eine Pause eingebaut werden, da es bei grossen Bildern zu Verzögerungen beim Laden kommen kann. Das gibt dann unschöne Effekte. Die Pause darf aber auf keinen Fall SLEEP sein, da dann auch keine Bilder geladen werden. Besser ist es einen Timer zu benutzen.

Beispiel :

While

Bild laden und anzeigen

Settimer 1000

Waitinput

Killtimer

Wend

Macht mindestens 1 Sekunde Pause !

Filter mischen

Durch geschicktes Mischen der Filter kann man sich schöne Effekte selber zusammenstellen.
Einen Zeichen z.B. durch TraceContour und Überblenden.

Rechtliches

Freeware - DLL für Freeware-Programme !

Diese DLL darf nur für Freeware-Programm eingesetzt werden. Jegliche Form der kommerziellen Nutzung ist untersagt.

Autoren:

Andreas Miethe -> GRAPHAMP.DLL -> basierend auf

Filter und Effekte > Andreas Moser.

Png > Jack Goman und Edmund H. Hand

Gif > Anders Melander

Beachte:

JEDE VERWENDUNG DER "GRAPHAMP.DLL" ERFOLGT AUF EIGENE GEFAHR.

DER FREISCHALT-CODE DARF NICHT WEITERGEGEBEN WERDEN !

DIE DLL DARF NICHT VERKAUFT, ODER SONSTWIE ZU KOMMERZIELLEN ZWECKEN GENUTZT WERDEN. SIE DARF AUSSCHLIESSLICH FÜR FREEWARE-PROGRAMME GENUTZT WERDEN.

Andreas Miethe im September 2001

Freischalten

Die Dll ist zwar Freeware, aber um zu sie voll d.h. ohne Nagscreen nutzen zu können, muss sie registriert werden. Die Registrierung erfolgt mit dem Programm "Reg_Key.exe", das zum Paket gehört.

Registriernamen und Schlüssel kann über die Homepage des Authors angefordert werden.

[Homepage](#)

```
{*****}
{      Header-Datei für graphamp.dll      }
{      Andreas Miethe                     }
{      Dezember 2001                     }
{*****}
```

```
P_MaxIndex          = 500;
{*****}
```

```
{ Load-Filter }
{*****}
```

```
StretchToSize       = 2;
StretchToFit        = 1;
Normal              = 0;
Transparent          = -1;
FullColor           = 0;
{*****}
```

```
{ Resample-Filter }
{*****}
```

```
SplineFilter        = 1;
BellFilter           = 2;
TriangleFilter       = 3;
BoxFilter            = 4;
HermiteFilter        = 5;
Lanczos3Filter       = 6;
MitchellFilter       = 7;
```

```
{*****}
```

```
{ Arithmetic-Filter }
{*****}
```

```
None                = 1;
Add                 = 2;
Substract           = 3;
Multiply            = 4;
Divide              = 5;
Darkest             = 6;
Lightest            = 7;
Difference          = 8;
BinaryOr            = 9;
BinaryAnd           = 10;
Average             = 11;
```

```
{*****}
```

```
{ Graphic-Filter }
{*****}
```

```
EmbossColor         = 1;
EmbossLight         = 2;
EmbossMedium        = 3;
EmbossDark          = 4;
Enhance             = 5;
BlurBartlett        = 6;
BlurGaussian        = 7;
Negative            = 8;
Average             = 9;
Blur                = 10;
BlurSoftly          = 11;
BlurMore            = 12;
Prewitt             = 13;
TraceContour        = 14;
Sharpen             = 15;
```

```
SharpenMore          = 16;
SharpenLess          = 17;
UnSharpMask          = 18;
EdgesStrong          = 19;
EdgesWeak            = 20;
Etch                 = 21;
LaplacianHV          = 22;
LaplacianOmni        = 23;
SharpenDirectional   = 24;
SobelPass            = 25;
```

```
{*****}
{ Scanner }
{*****}
```

```
No_Source           = 0;
Source              = 1;
No_Manager           = 1;
Manager             = 0;
ToClip              = 1;
ToBitmap            = 0;
```

```
{*****}
{ Print - Ausgabe auf DC's }
{*****}
Transparent          = 1;
Normal               = 0;
```

```
{*****}
{ PNG - Optionen }
{*****}
Interlace            = 1;
NoInterlace          = 0;
```

```
{*****}
{ JPG - Optionen }
{-----}
{Bit-Tiefe}
{-----}
24Bit                = 24;
8Bit                 = 8;
{-----}
{Ladegeschwindigkeit}
{-----}
Speed                = 0;
Quality              = 1;
{-----}
{Ladegroesse}
{-----}
Fullsize             = 0;
HalfSize             = 1;
QuarterSize          = 2;
EighthSize           = 3;
{-----}
{Ladefarbe}
{-----}
Grayscale            = 1;
Color                = 0;
{-----}
{Anzeige}
```

```

{-----}
Smooth                = 1;
NoSmooth              = 0;
{-----}
{Progressiv speichern oder nicht}
{-----}
Progressiv            = 1;
NoProgressiv          = 0;

{*****}
{DLL-Funktionen}
{*****}
Init_Dll( = External("graphamp.dll", "InitDll",;
Is_Init( = External("graphamp.dll", "IsInit",;
Exit_Dll( = External("graphamp.dll", "ExitDll",;
Create_Tempfilename( = External("graphamp.dll", "CreateTempfilename",;
P_LoadImage( = External("graphamp.dll", "P_LoadImage",;
P_LoadImageFromModul( = External("graphamp.dll", "P_LoadImageFromModul",;
P_GetBitmapInfo( = External("graphamp.dll", "P_GetBitmapInfo",;
P_FreeBitmap( = External("graphamp.dll", "P_FreeBitmap",;
P_StretchToFit( = External("graphamp.dll", "P_StretchToFit",;
P_Greyscale( = External("graphamp.dll", "P_Greyscale",;
P_Negativ( = External("graphamp.dll", "P_Negativ",;
P_Contrast( = External("graphamp.dll", "P_Contrast",;
P_Lightness( = External("graphamp.dll", "P_Lightness",;
P_Mirror( = External("graphamp.dll", "P_Mirror",;
P_Flip( = External("graphamp.dll", "P_Flip",;
P_Resample( = External("graphamp.dll", "P_Resample",;
P_Crop( = External("graphamp.dll", "P_Crop",;
P_SetToClipboard( = External("graphamp.dll", "P_SetToClipboard",;
P_GetFromClipboard( = External("graphamp.dll", "P_GetFromClipboard",;
P_SaveBitmap( = External("graphamp.dll", "P_SaveBitmap",;
P_AssignBitmap( = External("graphamp.dll", "P_AssignBitmap",;
P_AssignExtBitmap( = External("graphamp.dll", "P_AssignExtBitmap",;
P_CreateEmptyBitmap( = External("graphamp.dll", "P_CreateEmptyBitmap",;
P_Rotate( = External("graphamp.dll", "P_Rotate",;
P_Blend( = External("graphamp.dll", "P_Blend",;
P_Filter( = External("graphamp.dll", "P_Filter",;
P_ScanToBitmap( = External("graphamp.dll", "P_ScanToBitmap",;
P_Gamma( = External("graphamp.dll", "P_Gamma",;
P_Colorize( = External("graphamp.dll", "P_Colorize",;
P_Solarize( = External("graphamp.dll", "P_Solarize",;
P_Spray( = External("graphamp.dll", "P_Spray",;
P_IncDecRGB( = External("graphamp.dll", "P_IncDecRGB",;
P_PrintBitmap( = External("graphamp.dll", "P_PrintBitmap",;
P_Arithmetic( = External("graphamp.dll", "P_Arithmetic",;
P_ConvertTo256( = External("graphamp.dll", "P_ConvertTo256",;
P_AddNoise( = External("graphamp.dll", "P_AddNoise",;
P_AntiAlias( = External("graphamp.dll", "P_AntiAlias",;
P_Mosaic( = External("graphamp.dll", "P_Mosaic",;
P_IsTwain( = External("graphamp.dll", "P_IsTwain",;
P_GetLoadFilename( = External("graphamp.dll", "P_GetLoadFilename",;
P_GetSaveFilename( = External("graphamp.dll", "P_GetSaveFilename",;
P_SetPngOptions( = External("graphamp.dll", "P_SetPngOptions",;
P_SetJpgOptions( = External("graphamp.dll", "P_SetJpgOptions",;
P_SetExtJpgOptions( = External("graphamp.dll", "P_SetExtJpgOptions",;
P_GetLoadedBitmapInfo( = External("graphamp.dll", "P_GetLoadedBitmapInfo",;
P_GetDPI( = External("Graphamp.dll", "P_GetDPI",;
P_SetRLECompressValue( = External("Graphamp.dll", "P_SetRLECompressValue",;

```



```
P_GetWidth( = External("Graphamp.dll", "P_GetWidth",;
P_GetHeight( = External("Graphamp.dll", "P_GetHeight",;
P_GetColorBit( = External("Graphamp.dll", "P_GetColorBit",;
P_SetLoadSize( = External("Graphamp.dll", "P_SetLoadSize",;
P_GetLoadSize( = External("Graphamp.dll", "P_GetLoadSize",;
```

'Neue Funktionen

```
P_PlayMeta( = External("Graphamp.dll", "P_PlayMeta",;
P_GetMetaDims( = External("Graphamp.dll", "P_GetMetaDims",;
P_FreeMeta( = External("Graphamp.dll", "P_FreeMeta",;
P_Freelcon( = External("Graphamp.dll", "P_Freelcon",;
P_FreeCursor( = External("Graphamp.dll", "P_FreeCursor",;
P_ConvertMetaToBitmap( = External("Graphamp.dll", "P_ConvertMetaToBitmap",;
P_QuickLoadJPG( = External("Graphamp.dll", "P_QuickLoadJPG",;
P_IsJpgProgressiv( = External("Graphamp.dll", "P_IsJpgProgressiv",;
```

'Neue Kontrollen

```
P_IsBitmap( = External("Graphamp.dll", "P_IsBitmap",;
P_Islcon( = External("Graphamp.dll", "P_Islcon",;
P_IsCursor( = External("Graphamp.dll", "P_IsCursor",;
P_IsMeta( = External("Graphamp.dll", "P_IsMeta",;
```

'neue Dialoge

```
P_GetMetaFilename( = External("Graphamp.dll", "P_GetMetaFilename",;
P_GetMultiLoadFilename( = External("Graphamp.dll", "P_GetMultiLoadFilename",;
P_GetMultiMetaFilename( = External("Graphamp.dll", "P_GetMultiMetaFilename",;
P_GetUserLoadFilename( = External("Graphamp.dll", "P_GetUserLoadFilename",;
P_GetUserSaveFilename( = External("Graphamp.dll", "P_GetUserSaveFilename",;
```

```
P_CheckFormat( = External("Graphamp.dll", "P_CheckFormat",;
GetDllInfo( = External("Graphamp.dll", "GetDllInfo",;
```

```
P_SetMaxZoom( = External("Graphamp.dll", "P_SetMaxZoom",;
P_SetMaxStretch( = External("Graphamp.dll", "P_SetMaxStretch",;
```

```
P_PlayHMeta( = External("Graphamp.dll", "P_PlayHMeta",;
{'*****}
```