

OS2Unit

COLLABORATORS

	<i>TITLE :</i> OS2Unit		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		February 8, 2023	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	OS2Unit	1
1.1	OS2Unit by Björn Schotte	1
1.2	1. Warum diese Unit?	1
1.3	2. Die Funktionen	2
1.4	ausfuellen	3
1.5	OS2-Unit/ActStringGad	3
1.6	OS2-Unit/ASLFileReq	4
1.7	OS2-Unit/CloseASL	5
1.8	OS2-Unit/CloseGadTools	6
1.9	OS2-Unit/ComputeFont	6
1.10	OS2-Unit/ComputeX	7
1.11	OS2-Unit/ComputeY	8
1.12	OS2-Unit/FS_BevelBox	8
1.13	OS2-Unit/GadSelect	9
1.14	OS2-Unit/GhostGadget	10
1.15	OS2-Unit/InitASLStruct	10
1.16	OS2-Unit/SensitivGadget	11
1.17	OS2-Unit/SetCheckBox	11
1.18	OS2-Unit/SetCycleGad	12
1.19	OS2-Unit/SetListViewGad	13
1.20	OS2-Unit/SetListViewList	14
1.21	OS2-Unit/SetMXGad	14
1.22	OS2-Unit/SetNumberGad	15
1.23	OS2-Unit/OpenASL	16
1.24	OS2-Unit/OpenGadTools	17
1.25	userreq	17
1.26	3. Hinweise	18
1.27	4. History	18
1.28	5. Adresse/Copyright	19

Chapter 1

OS2Unit

1.1 OS2Unit by Björn Schotte

OS2-Unit

©1994 by Björn Schotte

- GIFTWARE -

1. Warum diese Unit?
2. Die einzelnen Funktionen
3. Hinweise
4. History
5. Adresse/Copyright

1.2 1. Warum diese Unit?

Nun, da ich schon seit geraumer Zeit nur noch Programme entwickle, die ab OS2.0 funktionieren, wollte ich die Entwicklung noch etwas vereinfachen, indem ich mir einen Befehlssatz, der in (fast) jedem Programm vorkommt, in eine Unit auslagerte.

OS2.x stellt ja bekanntlich viele tolle Eigenschaften zur Verfügung, z.B. neue (einfache) Gadgets etc. Dies wird z.B. durch die gadtools.library zur Verfügung gestellt. Und genau mit dieser Library beschäftigt sich diese Unit...

Es werden z.B. auch Routinen vorgestellt, die das fontssensitive Programmieren von GUIs vereinfachen...

1.3 2. Die Funktionen

Alphabetische Liste aller Prozeduren/Funktionen, die von der Unit zur Verfügung gestellt werden:

ActStringGad

ASLFileReq

CloseASL

CloseGadTools

ComputeFont

ComputeX

ComputeY

FS_BevelBox

GadSelect

GhostGadget

InitASLStruct

OpenASL

OpenGadTools

SensitivGadget

SetCheckBox

SetCycleGad

SetListViewGad

SetListViewList

SetMXGad

SetNumberGad

UserReq

Zu beachten ist auch, daß im Interface-Teil einige Variablen definiert werden, die vom Programm ausgefüllt werden muß, damit

fontsensitives Programmieren möglich ist!

1.4 ausfuellen

```
ScreenW      : Breite des Screens, auf dem das Fenster
               geöffnet werden soll.
ScreenH      : Analog dazu die Höhe.
WBRight      : Breite des Rahmens auf der rechten Seite.
               (sp^.WBorRight);
WBBottom     : Höhe des Rahmens auf der unteren Seite.
               (sp^.WBorBottom);
XOff         : sp^.WBorLeft;
YOff         : sp^.BarHeight;
```

Beispiel:

```
|----- snipp -----|
...
VAR
  ps : p_Screen;

BEGIN
  ps := LockPubScreen("Workbench");
  IF ps = NIL THEN { Fehler }
  { Felder kopieren }
  ScreenW := ps^.Width;
  ScreenH := ps^.Height;
  WBRight := ps^.WBorRight;
  WBBottom := ps^.WBorBottom;
  XOff := ps^.WBorLeft;
  YOff := ps^.WBorTop + ps^.Font^.ta_YSize + 1;
  psfont := ps^.RastPort.Font;
  UnLockPubScreen(NIL, ps);
  ComputeFont(psfont, 0, 0);
...
|----- snipp -----|
```

1.5 OS2-Unit/ActStringGad

NAME

ActStringGad

SYNOPSIS

```
ActStringGad(wp, gad, req);
```

```
wp  : p_Window;
gad : p_Gadget;
req : p_Requester;
```

ERKLÄRUNG

Diese Prozedur aktiviert das Gadget im Window und, sofern nötig, im angegebenen Requester

RÜCKGABE

Keine.

BUGS

Keine bekannt.

1.6 OS2-Unit/ASLFileReq

NAME

ASLFileReq

SYNOPSIS

```
LONG = ASLFileReq(ASLStruct);
```

```
ASLStruct : ASLFileStruct;
```

```
ASLFileStruct = RECORD
```

```

    left, top,           { Linke Ecke, rechte Ecke,      }
    width,              { Breite und                    }
    height      : INTEGER; { Höhe d. FileReqs (optional)  }
    titel       : STRING[80]; { Titel des FileReqs           }
    pfad,       { Hier werden Pfad und Datei    }
    datei,     { von der Prozedur abgelegt.  }
    initp,     { Vorgegebener Pfad/Datei, der }
    initd,     { als erstes im Req dargestellt }
              { werden soll.                }
    filename   : STRING[256]; { Hier wird von der Prozedur ein }
              { vollständiger Filename erstellt }
              { also z.B. "Work:xx/yy"        }
    pattern    : STRING[80]; { Muster (z.B. "~(#?.info)" (optional) }
    display_pat: BOOLEAN;  { Soll ein String-Gadget im Req }
              { dargestellt werden, damit man }
              { das Pattern ändern kann ?    }
    win        : p_Window; { Zugehöriges Window des Reqs (opt.) }
    winsleep,  { Soll Wartezeiger + unsichtb. Req }
              { auf <win> geöffnet werden ? }
    canceled   : BOOLEAN;  { Wurde der Requester gecanceled ? }
    negativ,   { Negativer Gadget-Text (opt.) }
    positiv    : STRING;   { Positiver Gadget-Text (opt.) }

```

```
END;
```

ERKLÄRUNG

Diese mächtige Funktion stellt einen ASLFile-Requester dar. Zu beachten ist, daß die "asl.library" mit der Funktion

OpenASL

zu öffnen ist, denn die Prozedur tut das nicht selber !! Auch die "dos.library" muß geöffnet werden, da die Prozedur eine Funktion aus der Library verwendet.

Als erstes muß das Programm, daß die Funktion aufruft, mit der Prozedur

InitASLStruct

sicherstellen, daß alle Variablen der Struktur initialisiert werden.

Als nächstes initialisiert man im Programm die Struktur mit seinen eigenen Werten, bei den meisten braucht man, wenn man das gewünschte Feature nicht haben will, nichts eintragen. O.g. Variablen haben in der Erklärung ein "opt.", wenn diese Variable nicht zwingend ausgefüllt werden muß.

Man füllt also diese Struktur aus und übergibt Sie der Funktion. Die Funktion retourniert 0, wenn alles klar ist, ansonsten `FREQ_NOALLOC` (`=-100`). Dies bedeutet, daß die Allokation per `AllocASLRequest()` fehlgeschlagen ist!

In der eigenen Struktur, die man `ASLFileReq()` übergeben hat, findet sich unter `<filename>` der vollständige Dateiname inkl. "/" oder ":" bei Verzeichnis-/Disknamen. Unter `<pfad>` und `<datei>` findet man jeweils den einzelnen Pfad bzw. die einzelne Datei. Unter `pattern` findet man entweder das alte Pattern bzw. ein neues, falls der Benutzer im Filerequester ein neues eingegeben hat. `<initp>` und `<initd>` erhalten die Werte von `<pfad>` und `<datei>`. `<anceled>` ist `TRUE`, wenn der Benutzer den Filerequester gecancel hat.

Ich hoffe, daß damit alles klar ist!

RÜCKGABE

0, wenn alles klar ist, ansonsten `FREQ_NOALLOC` (`=-100`), wenn das Allokieren der FileRequester-Struktur per `AllocASLRequest()` fehlgeschlagen ist.

BUGS

Keine bekannt.

1.7 OS2-Unit/CloseASL

NAME

CloseASL

SYNOPSIS

CloseASL

ERKLÄRUNG

Schliesst die evtl. geöffnete "asl.library".

RÜCKGABE

Keine.

BUGS

Keine bekannt.

SIEHE AUCH

OpenASL

1.8 OS2-Unit/CloseGadTools

NAME

CloseGadTools

SYNOPSIS

CloseGadTools;

ERKLÄRUNG

Schliesst die evtl. geöffnete "gadtools.library".

RÜCKGABE

Keine

BUGS

Keine bekannt

SIEHE AUCH

OpenGadTools

1.9 OS2-Unit/ComputeFont

NAME

ComputeFont

SYNOPSIS

ComputeFont(TheFont, width, height);

```
TheFont : p_TextFont;  
width,  
height  : INTEGER;
```

ERKLÄRUNG

Hauptprozedur für fontsensitive GUIs.
Übergeben wird ein Font und Breite und Höhe bei topaz/8-Font (z.B. fürs Fenster-Öffnen).
Die Prozedur berechnet als erstes die Breite und Höhe bei TheFont und überprüft dann, ob diese auf den Screen passen.

Folgende Variablen müssen "von Hand" initialisiert werden:

ScreenW, ScreenH, WBRight, WBBottom, XOff, YOff

RÜCKGABE

Keine.

BUGS

Keine bekannt.

SIEHE AUCH

ComputeX

ComputeY

1.10 OS2-Unit/ComputeX

NAME

ComputeX

SYNOPSIS

```
LONG = ComputeX(value);
```

```
value : INTEGER;
```

ERKLÄRUNG

Übergeben wird die Breite bei topaz/8-Font.
ComputeX berechnet dann die Breite bei TheFont von

ComputeFont

.

RÜCKGABE

Breite bei TheFont.

BUGS

Keine bekannt.

SIEHE AUCH

ComputeFont

1.11 OS2-Unit/ComputeY

NAME

ComputeY

SYNOPSIS

```
LONG = ComputeY(value);
```

```
value : INTEGER;
```

ERKLÄRUNG

Übergeben wird die Höhe bei topaz/8-Font.
ComputeY berechnet dann die Höhe bei TheFont
von

ComputeFont

.

RÜCKGABE

Höhe bei TheFont.

BUGS

Keine bekannt.

SIEHE AUCH

ComputeFont

1.12 OS2-Unit/FS_BevelBox

NAME

FS_BevelBox

SYNOPSIS

```
FS_BevelBox(wp,vi,x,y,b,h,recessed);

wp      : p_Window;
vi      : PTR; { VisualInfo }
x,y     : INTEGER; { Linke, obere Ecke }
b,h     : INTEGER; { Breite und Höhe }
recessed : BOOLEAN; { Rahmen "eingedrückt" ?? }
```

ERKLÄRUNG

Zeichnet eine fontsensitive BevelBox.

RÜCKGABE

Keine

BUGS

Keine bekannt.

1.13 OS2-Unit/GadSelect

NAME

GadSelect

SYNOPSIS

```
GadSelect(wp,gad);
```

```
wp : p_Window;
gad : p_Gadget;
```

ERKLÄRUNG

Per GT_Underscore kann man bei einem Gadget bestimmen, welcher Buchstabe unterstrichen werden muss. Dies bedeutet, daß man das Gadget auch über die Tastatur anwählen kann. Um dies sichtbar zu machen, existiert diese Prozedur. Ihr wird das Fenster und das Gadget übergeben.

ACHTUNG: Die Tastendrucke, die vom Programm empfangen werden, müssen vom Typ IDCMP_RAWKEY sein !!

Wenn man also im Programm eine Message vom Typ IDCMP_RAWKEY empfangen hat, muss man den Tastendruck abfragen. Es muss aber nur der msg^.Code beim Drücken der Taste abgefragt werden, damit die Prozedur auf eine Release warten kann!!

RÜCKGABE

Keine.

BUGS

Keine bekannt.

1.14 OS2-Unit/GhostGadget

NAME

GhostGadget

SYNOPSIS

```
GhostGadget (wp, gad, req, dis);
```

```
wp      : p_Window;  
gad     : p_Gadget;  
req     : p_Requester; (optional)  
dis     : BOOLEAN;
```

ERKLÄRUNG

Übergeben wird das Fenster, das Gadget und, wenn nötig, der Requester, in dem das Gadget "liegt". Weiterhin wird entweder TRUE oder FALSE übergeben, jenachdem, wie man das Gadget haben will: TRUE, um es zu "ghosten", ansonsten FALSE.

RÜCKGABE

Keine.

BUGS

Keine bekannt.

1.15 OS2-Unit/InitASLStruct

NAME

InitASLStruct

SYNOPSIS

```
InitASLStruct (ASLStruct);  
ASLStruct : ASLFileStruct;
```

ERKLÄRUNG

Diese Funktion steht in Verbindung mit
ASLFileReq()

Sie muß vom aufrufenden Programm vor (!) dem ersten Benutzen von ASLFileReq() aufgerufen werden. Mit ihr wird die übergebene ASLFileStruct initialisiert.

RÜCKGABE

Keine.

BUGS

Keine bekannt.

SIEHE AUCH

ASLFileReq

1.16 OS2-Unit/SensitivGadget

NAME

SensitivGadget

SYNOPSIS

```
SensitivGadget (ng);
```

```
ng : NewGadget;
```

ERKLÄRUNG

Verändert die Koordinaten der NewGadget-Struktur so, daß sie sich fontsensitiv anpassen...

ACHTUNG: In der NewGadget-Struktur müssen die Koordinaten so eingetragen sein, als ob sie dem topaz/8-Font entsprechen würden !!

RÜCKGABE

Keine.

BUGS

Keine bekannt.

1.17 OS2-Unit/SetCheckBox

NAME

SetCheckBox

SYNOPSIS

```
SetCheckBox(wp, gad, req, flag);
```

```
wp      : p_Window;  
gad     : p_Gadget;  
req     : p_Requester;  
flag    : BOOLEAN;
```

ERKLÄRUNG

Stellt ein CheckBox-Gadget mit Häkchen (=aktiviert) oder ohne dar.

RÜCKGABE

Keine.

BUGS

Keine bekannt.

SIEHE AUCH

```
SetCycleGad  
SetListViewGad  
SetListViewList  
SetMXGad  
SetNumberGad
```

1.18 OS2-Unit/SetCycleGad

NAME

```
SetCycleGad
```

SYNOPSIS

```
SetCycleGad(wp, gad, req, active);
```

```
wp      : p_Window;  
gad     : p_Gadget;  
req     : p_Requester;  
active  : LONG;
```

ERKLÄRUNG

Setzt den CycleGadget-Text Nr. <active> (von 0 beginnend!)

RÜCKGABE

Keine.

BUGS

Keine bekannt.

SIEHE AUCH

SetCheckBox

SetListViewGad

SetListViewList

SetMXGad

SetNumberGad

1.19 OS2-Unit/SetListViewGad

NAME

SetListViewGad

SYNOPSIS

```
SetListViewGad(wp, gad, req, active, top);
```

```
wp      : p_Window;  
gad     : p_Gadget;  
req     : p_Requester;  
active,  
top     : LONG;
```

ERKLÄRUNG

Setzt den aktiven Eintrag eines ListView-Gadgets. Ausserdem wird noch das oberste sichtbare Listen-Element mit angegeben (top).

RÜCKGABE

Keine.

BUGS

Keine bekannt.

SIEHE AUCH

SetCheckBox
SetCycleGad
SetListViewList
SetMXGad
SetNumberGad

1.20 OS2-Unit/SetListViewList

NAME

SetListViewList

SYNOPSIS

```
SetListViewList (wp, gad, req, NeueList);
```

```
wp      : p_Window;  
gad     : p_Gadget;  
req     : p_Requester;  
NeueList : p_List;
```

ERKLÄRUNG

Setzt die Liste eines ListView-Gadgets neu.

RÜCKGABE

Keine.

BUGS

Keine bekannt.

SIEHE AUCH

SetCheckBox
SetCycleGad
SetListViewGad
SetMXGad
SetNumberGad

1.21 OS2-Unit/SetMXGad

NAME

SetMXGad

SYNOPSIS

```
SetMXGad(wp, gad, req, active);
```

```
wp      : p_Window;  
gad     : p_Gadget;  
req     : p_Requester;  
active  : LONG;
```

ERKLÄRUNG

Setzt das aktive MX-Element.

RÜCKGABE

Keine

BUGS

Keine bekannt

SIEHE AUCH

SetCheckBox

SetCycleGad

SetListViewGad

SetListViewList

SetNumberGad

1.22 OS2-Unit/SetNumberGad

NAME

SetNumberGad

SYNOPSIS

```
SetNumberGad(wp, gad, req, nummer);
```

```
wp      : p_Window;  
gad     : p_Gadget;  
req     : p_Requester;  
nummer  : LONG;
```

ERKLÄRUNG

Setzt ein Number-Gadget mit dem Wert <nummer>.

RÜCKGABE

Keine

BUGS

Keine bekannt

SIEHE AUCH

SetCheckBox

SetCycleGad

SetMXGad

SetListViewGad

SetListViewList

1.23 OS2-Unit/OpenASL

NAME

OpenASL

SYNOPSIS

```
BOOLEAN = OpenASL(version);
```

```
version : INTEGER;
```

ERKLÄRUNG

Versucht, die "asl.library" mit der Version <version> oder höher, zu öffnen.

RÜCKGABE

TRUE, falls die Library geöffnet werden konnte, ansonsten FALSE.

BUGS

Keine bekannt.

SIEHE AUCH

CloseASL

1.24 OS2-Unit/OpenGadTools

NAME

OpenGadTools

SYNOPSIS

```
BOOLEAN := OpenGadTools(version);
```

```
version : INTEGER;
```

ERKLÄRUNG

Versucht, die "gadtools.library" mit der Version <version> oder höher, zu öffnen.

RÜCKGABE

TRUE, falls die Library erfolgreich geöffnet werden konnte, ansonsten FALSE.

BUGS

Keine bekannt

SIEHE AUCH

CloseGadTools

1.25 userreq

NAME

UserReq

SYNOPSIS

```
LONG := UserReq(win,tit,txt,gad);
```

```
win : p_Window;
```

```
tit,tit,txt,gad : STR;
```

ERKLÄRUNG

Erstellt einen EasyRequester auf dem Window win. win kann auch NIL sein. tit ist der Titel des Requesters, txt der Requester-Text und gad der GadgetText, z.B. "Quit|Quit & Save|Abort" für 3 Auswahlgadgets.

RÜCKGABE

0 entspricht dem am weitesten rechts liegenden Aktionssymbol, während die übrigen Aktionssymbole von links beginnend von 1 an durchnummeriert sind.

BUGS

Keine bekannt.

1.26 3. Hinweise

Wichtig:

Das Programm, das diese Unit einbindet, muß (!) mit der Funktion

```
OpenGadTools()
    dafür sorgen, daß die "gadtools.library"
geöffnet wird!
```

1.27 4. History

```
-----
- Revision: V0.040 -
-----
```

Funktion UserReq() hinzugefügt.
Guide-Datei erweitert.

```
-----
- Revision: V0.030 -
-----
```

ASLFileReq() hinzugefügt. Mächtige Funktion
zum Auswählen von Dateien.

```
-----
- Revision: V0.020 -
-----
```

SetXXXGad-Prozeduren erweitert;
GhostGadget(), GadSelect hinzugefügt.

```
-----
- Revision: V0.010 -
-----
```

SetXXXGad-Prozeduren hinzugefügt.

```
-----
- Revision: V0.000 -
```

--- Initial release ---

1.28 5. Adresse/Copyright

Diese Unit ist GIFTWARE, d.h. wenn es Ihnen gefällt, können (!) Sie mir etwas schicken, was ich als Spende auffassen werde, z.B.: Geld, PD-DisX, Pascal-Sources, Schokolade....

Das ist nicht zwingend, aber: Wer die Möglichkeit hat, EMail zu benutzen, der sollte (!) mir wenigstens eine nette Mail schreiben...

Meine Adresse (bitte Rückporto beilegen!):

Björn Schotte
Am Burkardstuhl 45
D-97267 Himmelstadt
Germany

EMail: bjoern@bomber.mayn.sub.de

(Pascal-FTP-Server: [ftp@bomber.mayn.sub.de](ftp://ftp@bomber.mayn.sub.de), Subject: HELPALL)

Ich wünsche Ihnen viel Spass mit dieser Unit!