AMOSPro_GadTools.Include

Harald Wagner

COLLABORATORS			
	TITLE : AMOSPro_GadTools.Include		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Harald Wagner	January 2, 2023	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

AMOSPro_GadTools.Include

1.1 AMOSPro GadTools-Include Instructions

```
                    AMOSPro GadTools.Inc Version 1.0 - Copyright © 1995
-------------------------------------------------------
              Written by Harald Wagner
     Ein Erweiterungs-Tool für AMOS-Professional

                     (FreeWare)



        Einleitung
      Was sind die AMOSPro GadTools?

        Kurz-Index
      Kurz Übersicht aller GadTools Befehle.

      Befehls-Index
      Alle 32 GadTools Befehle im Überblick.

      Vorgehensweise
      So benutzen sie die GadTools.

        Beispiele
      Die Beispiel-Listings.

     Einschränkungen
      Reservierte Lokale Variablen & Bänke.

       Anregungen
      Anregungen für weitere versionen.

       Copyright
      Allgemeine Infos zu den GadTools.

       Bug Report
      Bug Reports bitte an den Autor senden.
```

1.2 Was sind die AMOSPro GadTools?

```
Was sind die AMOSPro GadTools?
------------------------------

Die AMOSPro GadTools sind ein Include-File für AMOS Professional.
D.H. es wird mit der 'Include' anweisung von AMOSPro an das eigentliche
Hauptprogramm angehangen. Danach stehen dem Anwender 32 Befehle für
die verschiedensten GadGets zur verfügung. Bei der programmierung der
GadTools ging es mir aber nicht nur darum irgendwelche GadGets auf
dem Bildschirm zu setzen, sondern vielmehr auch darum dem Anwender ein
packet anzubieten, mit dem man auf bequemer weise screens und Masken
erstellen kann, die direkt im OS2.0 Look erscheinen.
```

1.3 Kurz Übersicht aller GadTools Befehle.

```
Kurz Übersicht aller GadTools Befehle.
--------------------------------------

_PushButton[Id,X,Y,Color1,Color2,Color3,Color4,"Button-Name"]
_Del_PushButton[Id]
_PushButton_Hit
_CheckBox[Id,X,Y,Color1,Color2,Color3,Color4,On/Off]
_Set_CheckBox_State[Id,On/Off]
_Get_CheckBox_State[Id]
_Del_CheckBox[Id]
_CheckBox_Hit
_RadioButton[Id,X,Y,Color1,Color2,Color3,On/Off]
_Set_RadioButton_State[Id,On/Off]
_Get_RadioButton_State[Id]
_Del_RadioButton[Id]
_RadioButton_Hit
_CycleButton[Id,X,Y,Color1,Color2,Color3,Color4,"Text1|Text2..."]
_Set_CycleButton_State[Id,TextNumber]
_Get_CycleButton_State[Id]
_Del_CycleButton[Id]
_CycleButton_Hit
_ToggleButton[Id,X,Y,Color1,Color2,Color3,Color4,On/Off,"Button-Name"]
_Set_ToggleButton_State[Id,On/Off]
_Get_ToggleButton_State[Id]
_Del_ToggleButton[Id]
_ToggleButton_Hit
_StringGadget[Id,X,Y,Color1,Color2,Color3,Color4,Color5,GadLen,MaxTextLen]
_Set_StringGadget_State[Id,String$]
_Get_StringGadget_State[Id]
_Del_StringGadget[Id]
_StringGadget_Hit
_TextGadget[Id,X,Y,Color1,Color2,Color3,Color4,MaxTextLen]
_Set_TextGadget_Text[Id,String$]
_Make_Mask[X1,Y1,X2,Y2,Color1,Color2,Color3]
_Screen_Open[Screen,0-2 {Lores/Hires/Interlace} ]
```

1.4 Alle 32 GadTools Befehle im Überblick.

```
Alle 32 GadTools Befehle im Überblick.
--------------------------------------

Nachdem man mit >>>Include "Device+Pfad/GadTools.Inc"<<< die GadTools
am ende des Hauptprogramms angehangen hat, stehen einem folgende befehle
zur verfügung.:
------------------------------------------------------------------------

_Screen_Open[ScreenNummer,Value]

Mit dieser anweisung wird ein Screen eröffnet das im OS2.0 Look erscheint.
Der erste Parameter is eine ScreenNummer im bereich von 0-7. Der zweite
Parameter (Value) bestimmt welchen Modi sie für den Screen benutzen
möchten. Dabei kann ein wert von 0-2 angegen werden.

Value 0 = Lores (320 x 256 x 4)
Value 1 = Hires (640 x 256 x 4)
Value 2 = Laced (640 x 512 x 4)

Ein somit geöffneter Screen enthält danach automatisch die farben
im OS2.0 Look.

------------------------------------------------------------------------

_Make_Mask[X1,Y1,X2,Y2,Color1,Color2,Color3]

Mit dieser anweisung wird eine Maske im Screen eröffnet (Panel) das im
OS2.0 Look dargestellt wird. Dabei wird die Maske an den Coordinaten
X1,Y1,X2,Y2 gezeichnet. Color1 bestimmt das FarbAtribut für den oberen
und linken teil des Borders und Color2 für den rechten und unteren teil
des Borders. Mit Color3 wird das FarbAtribut für die farbe der Maske
selbst bestimmt. Man sollte den _Make_Mask befehl immer im zusammenhang
mit _Screen_Open benutzen.

------------------------------------------------------------------------

_PushButton[Id,X,Y,Color1,Color2,Color3,Color4,"Button-Name"]

Mit dieser anweisung wird ein PushButton gezeichnet. Dabei wird das
Button mit der Id-Nummer 'Id' an den Coordinaten X,Y gezeichnet. Mit
Color1 wird das FarbAtribut für den oberen und linken teil des Borders
gewählt und mit Color2 das FarbAtribut für den unteren und rechten teil
des Borders. Mit Color3 wird das FarbAtribut für das Button selbst
bestimmt und Color4 setzt das FarbAtribut für den Text im Button.
Mit Button-Name wird der name des Buttons festgelegt. Insgesammt kann
man 30 (1-30) PushButtons definieren. Die angabe der 'Id-Nr.' darf also
nur im bereich von 1-30 liegen.

------------------------------------------------------------------------

_Del_PushButton[Id]

Mit dieser anweisung wird ein PushButton mit der Id-Kennung 'Id' aus
dem speicher entfernt. Danach lässt sich dieses Button nicht mehr
```

abfragen. Die grafische darstellung bleibt jedoch auf dem Screen erhalten. Diese anweisung ist sehr wichtig, wenn man z.b. einen zweiten screen öffnet in dem sich ein option-panel aufbaut. Damit die Buttons aus dem vorherigen screen nicht mit abgefragt werden. Möchte man das somit glöschte Button wieder aktivieren, so muss man dieses wieder mit der _PushButton anweisung neu definieren. (Siehe unter _PushButton)

_PushButton_Hit

Mit dieser anweisung werden alle bis dato definierten PushButtons abgefragt. Wurde eines betätigt, so wird die Id-Nummer in der variable 'Param' zurückgegeben. (z.b. '1' wenn PushButton 1 angeklickt wurde.) Wurde kein Button angeklickt, so wird '0' in der variable 'Param' zurückgegeben. Diese anweisung sollte immer mit in der hauptschleife des hauptprogramms ausgeführt werden.

_CheckBox[Id,X,Y,Color1,Color2,Color3,Color4,On/Off]

Mit dieser anweisung wird eine CheckBox mit der Id-Nummer 'Id' an den Coordinaten X,Y gezeichnet. Color1 bestimmt dabei das FarbAtribut für den oberen und linken teil des Borders und Color2 den unteren und rechten teil des Borders. Mit Color3 wird das FarbAtribut für die CheckBox selbst festgelegt und Color4 bestimmt das FarbAtribut für das häckchen das bei aktiviertem zustand angezeigt wird. Mit On/Off wird ein Value von 0-1 angegeben, wobei mit '1' sofort ein häckchen in der Box gezeichnet wird. Mit '0' wird die CheckBox inaktiviert gezeichnet. Auch hier lassen sich maximal 30 CheckBoxen definieren. Die Id-Kennung darf also von 1-30 gehen.

_Set_CheckBox_State[Id,On/Off]

Mit dieser anweisung wird die CheckBox mit der Id-Kennung 'Id' aktiviert bzw. inaktiviert. Mit On/Off wird ein Value von 0-1 angegeben. Dabei bedutet ein wert von '0' das die CheckBox inaktiviert wird (häckchen wird aus der CheckBox entfernt.) und ein wert von '1' aktiviert die angegebene CheckBox. (häckchen wird sichtbar.)

_Get_CheckBox_State[Id]

Mit dieser anweisung wird der status einer CheckBox ausgelesen. Dabei wird in der variable 'Param' ein wert von '0' zurückgegeben wenn die CheckBox inaktiviert ist und '1' wenn sie aktiviert ist. Wurde ein wert von '-1' zurückgegeben, so ist ein fehler aufgetreten. Entweder existiert die CheckBox noch nicht oder es wurde die grenze der maximalen Id-Nummern von 1-30 überschritten.

_Del_CheckBox[Id]

Das selbe wie _Del_PushButton[Id] nur hier für CheckBoxen. (Siehe unter _Del_PushButton.)

_CheckBox_Hit

Das selbe wie _PushButton_Hit nur hier für CheckBoxen. (Siehe unter _PushButton_Hit.)

_RadioButton[Id,X,Y,Color1,Color2,Color3,On/Off]

Mit dieser anweisung wird das RadioButton mit der Id-Kennung 'Id' an den Coordinaten X,Y gezeichnet. Color1 bestimmt dabei das FarbAtribut für das Button selbst. Color2 bestimmt das FarbAtribut für die aktivierte darstellung und Color3 für die inaktivierte darstellung. Mit On/Off wird ein Value von 0-1 angegeben, wobei das Button mit '0' inaktiviert und mit '1' aktiviert gezeichnet wird. Auch hier ist eine grenze von 1-30 gesetzt. (D.H. Max. 30 Button können definiert werden.)

_Set_RadioButton_State[Id,On/Off]

Das selbe wie _Set_CheckBox_State nur hier für RadioButtons. (Siehe unter _Set_CheckBox_State.)

_Get_RadioButton_State[Id]

Das selbe wie _Get_CheckBox_State nur hier für RadioButtons. (Siehe unter _Get_CheckBox_State.)

_Del_RadioButton[Id]

Das selbe wie _Del_PushButton nur hier für RadioButtons. (Siehe unter _Del_PushButton.)

_RadioButton_Hit

Das selbe wie _PushButton_Hit nur hier für RadioButtons. (Siehe unter _PushButton_Hit.)

_CycleButton[Id,X,Y,Color1,Color2,Color3,Color4,"Text1|Text2|Text3..."]

Mit dieser anweisung wird das CycleButton mit der Id-Kennung 'Id' an den Coordinaten X,Y gezeichnet. Color1 bestimmt dabei das FarbAtribut für den

oberen und linken teil des Borders und Color2 den unteren und rechten
teil des Borders. Mit Color3 wird das FarbAtribut für das Button selbst
festgelegt und Color4 bestimmt die TextFarbe. Die eingegebenen Texte
werden hier durch einen senkrechten strich '|' abgetrennt. Wie immer
lassen sich bis zu 30 Buttons definieren. (Id = Max. 1-30)

_Set_CycleButton_State[Id,TextNumber]

Mit dieser anweisung wird dem CycleButton mit der Id-Kennung 'Id' einen
neuen status zugewiesen. D.H.: Wenn man z.b. 3 Texte in dem CycleButton
mit der Id-Kennung '1' definiert hat und mit _Set_CycleButton_State[1,2]
den status verändert, so würde dieses CycleButton den zweiten Text dar-
stellen.

_Get_CycleButton_State[Id]

Mit dieser anweisung wird der status eines CycleButtons ausgelesen. Das
ergebnis wird dan in der variable 'Param' zurückgegeben. Ist in dem
angegebenen Button z.b. gerade der 2 text sichtbar, so wird in der
variable 'Param' der wert '2' zurückgegeben. Wurde eine zu hohe (Max. 1-30)
Id-Kennung angegeben oder ein Button das noch nicht existiert wird '-1'
zurückgegeben, das in einem solchen fall auf einen fehler hinweist.

_Del_CycleButton[Id]

Das selbe wie _Del_PushButton nur hier für CycleButtons. (Siehe unter
_Del_PushButton.)

_CycleButton_Hit

Das selbe wie _PushButton_Hit nur hier für CycleButtons. (Siehe unter
_PushButton_Hit.)

_ToggleButton[Id,X,Y,Color1,Color2,Color3,Color4,On/Off,"Button-Name"]

Mit dieser anweisung wird das ToggleButton mit der Id-Kennung 'Id' an den
Coordinaten X,Y gezeichnet. Mit Color1 wird das FarbAtribut für den
oberen und linken teil des Borders gesetzt und mit Color2 das für den
unteren und rechten teil des Borders. Mit Color3 wird das FarbAtribut für
das Button selbst festgelegt und mit Color4 das für den Buton-Name.
Mit On/Off wird ein value von 0-1 angegeben, wobei das Button mit '0'
inaktiviert und mit '1' aktiviert dargestellt wird. Mit Button-Name wird
der name des Buttons festgelegt der im Button angezeigt wird. Auch hier
sind wieder Max. 30 Buttons definierbar. (Id-Kennung Max. 1-30)

Set_ToggleButton_State[Id,On/Off]

Das selbe wie _Set_CheckBox_State nur hier für ToggleButtons. (Siehe unter _Set_CheckBox_State.)

Get_ToggleButton_State[Id]

Das selbe wie _Get_CheckBox_State nur hier für ToggleButtons. (Siehe unter _Get_CheckBox_State.)

Del_ToggleButton[Id]

Das selbe wie _Del_PushButton nur hier für ToggleButtons. (Siehe unter _Del_PushButton.)

_ToggleButton_Hit

Das selbe wie _PushButton_Hit nur hier für ToggleButtons. (Siehe unter _PushButton_Hit.)

_StringGadget[Id,X,Y,Color1,Color2,Color3,Color4,Color5,GadLen,MaxTexLen]

Mit dieser anweisung wird das StringGadget mit der Id-Kennung 'Id' an den Coordinaten X,Y gezeichnet. Mit Color1 wird das FarbAtribut für den oberen un linken teil des Borders gesetzt und mit Color2 das für den unteren und rechten teil des Borders. Mit Color3 wird das FarbAtribut für das StringGadget selbst gesetzt und Color4 das für den Text der im StringGadget dargestellt wird. Mit Color5 wird das FarbAtribut für den Cursor im StringGadget gesetzt. Mit GadLen wird die länge des StringGadgets in zeichen angegeben und mit MaxTexLen die maximale länge an Text die das StringGadget aufnehmen kann. Wie immer sind auch hier maximal 30 StringGadgets definierbar. (Id = Max. 1-30)

_Set_StringGadget_State[Id,String$]

Mit dieser anweisung wird das StringGadget mit der Id-Kennung 'Id' den Text 'String$' zugewiesen.

_Get_StringGadget_State[Id]

Mit dieser anweisung wird der inhalt des StringGadgets mit der Id-Kennung 'Id' ausgelesen. Der inhalt wird dabei dann in der variable 'Param$' zurückgegeben.

```
_Del_StringGadget[Id]

Das selbe wie _Del_PushButton nur hier für StringGadgets. (Siehe unter
_Del_PushButton.)

------------------------------------------------------------------------

_StringGadget_Hit

Das selbe wie _PushButton_Hit nur hier für StringGadgets. (Siehe unter
_PushButton_Hit.)

------------------------------------------------------------------------

_TextGadget[Id,X,Y,Color1,Color2,Color3,Color4,MaxTexLen]

Mit dieser anweisung wird das TextGadget mit der Id-Kennung 'Id' an den
Coordinaten X,Y gezeichnet. Color1 bestimmt dabei das FarbAtribut für den
oberen und linken teil des Borders und Color2 das für den unteren und
rechten teil des Borders. Mit Color3 wird das FarbAtribut für das Gadget
selbst festgelegt und mit Color4 das für den Text der im TextGadget zu
stehen kommt. Mit MaxTexLen wird die grösse des TextGadgets in zeichen
angegeben.

Hinweis:
--------
Das TextGadget ist kein Button das man anklicken kann, sondern es dient
mehr dazu um Texte anzuzeigen. (z.b. für status-meldungen auszugeben.)

------------------------------------------------------------------------

_Set_TextGadget_Text[Id,String$]

Mit dieser anweisung wird dem TextGadget mit der Id-Kennung 'Id' dem
string 'String$' zugewiesen. Ist der angegebene String länger als das
TextGadget, so wird der String automatisch rechts abgeschnitten.

------------------------------------------------------------------------
```

1.5 So benutzen sie die GadTools.

```
So benutzen sie die GadTools.
-----------------------------

Um die GadTools-Befehle auch einsetzen zu können, müssen sie das Include-
File erst noch an ihr Hauptprogramm anhängen. Dazu dient die 'Include'
anweisung von AMOSPro V2.00. Am besten sie fahren mit dem Cursor auf
die letzte zeile ihres Sources und tragen dort dann folgendes ein:

Include "Device+Pfad/GadTools.Inc"

Für Device+Pfad müssen sie noch das entsprechende verzeichnis eintragen
unter dem sich das GadTools-Include befindet.
```

1.6 Die Beispiel-Listings.

```
Die Beispiel-Listings.
----------------------

In dem verzeichnis ':Beispiele/' befinden sich zu allen Buttons ein
beispiel-listing. Dabei liegen diese für alle ScreenModi zur verfügung.
(Lores/Hires/Interlace) Alle beispiele sind direkt als ASC-Files
abgelegt und können direkt über einen TextED betrachtet werden, oder
eben direkt über 'Merge ASC' in dem AMOSPro-Interpreter eingeladen werden.
Evtl. muss aber noch in der letzten zeile der pfad richtig eingestellt
werden, damit die listings das GadTools-Include auch finden und beiladen
können. Ich habe bewusst darauf verzichtet grosse beispiel-listings zu
schreiben, da das alles nur unübersichtlich machen würde. Jedes listing
beinhaltet also wirklich nur das nötigste, damit es auch leicht ver-
ständlich bleibt.
```

1.7 Reservierte Lokale Variablen & Bänke.

```
Reservierte Lokale Variablen & Bänke.
-------------------------------------

Da es sich bei den GadTools um ein Include-File (Proceduren) handelt und
nicht um eine echte AMOS-Extension, existieren in diesem File auch
variablen und Bänke die sich im eigenen Hauptprogramm nicht überschneiden
sollten. Deshalb hier eine kurze auflistung aller Variablen & Bänke die
das GadTools-Include benötigt.:

Variablen : A-I, X-Z, A$, B$, C$
AMOS-Bänke: 501-531,601-631,701-731,801-831,901-931,1001-1031,1101-1131

Da Ich keine Globalen variablen benutzt habe, könnte man die variablen
auch ohne weiteres im Hauptprogramm einsetzen, solange sie Lokal bleiben,
oder man diese als die sogenannten schmiervariablen benutzt. Eigentlich
sollte man die variablen A-Z immer als schmiervariablen verwenden und
nicht als standfeste werte für irgendwelche routinen. Erfahrende
programmierer wissen sicherlich wovon ich rede. ;-)  Was die AMOS-Bänke
betrifft.: Diese enthalten alle Button-Daten wie Position, farben etc.
Dabei ist z.b. Bank 501 PushButton-1, Bank 502 PushButton-2 u.s.w.
Alle diese aufgeführten Bänke sollten/dürfen nicht vom Hauptprogramm
verändert, gelöscht, oder neu reserviert werden. Sonst dürfte es
probleme geben. ;-)
```

1.8 Anregungen für weitere versionen.

```
Anregungen für weitere versionen.
---------------------------------

Wenn ihr eine idee oder eben einen verbesserungsvorschlag habt, dann
lasst es mich wissen. Ich bin gerne bereit diese entgegen zu nehmen.
Ihr könnt mich unter folgender adresse erreichen.:
```

Harald Wagner
Nordring 135
52531 Übach-Palenberg

EMail: Harald@Headless.tng.oche.de

1.9 Allgemeine Infos zu den GadTools.

```
Allgemeine Infos zu den GadTools.
---------------------------------

Das GadTools-Include und alles dazugehöhrige ist FreeWare. Es ist ohne
jede einschränkung erlaubt die GadTools in eigene programme einzubringen.
D.H.: Sie können die GadTools in ihrem programm weitergeben ob als Source
oder als Compilat. Es ist jedoch NICHT erlaubt veränderungen an den
GadTools (Proceduren) ohne schriftliche genemigung des Autors vorzunehmen.
Von den Autoren die das GadTools-Include einsetzen, erwarte ich zumindest
das sie in ihren DOC-Files die herkunft der GadTools erwähnen. Dies
gebietet schliesslich schon der anstand. Es ist allersings keine Pflicht
dies zu tun, wer das nicht möchte. Aber Ich setze mal das vertrauen in den
Autoren. ;-)

Ein wort zu den Distributoren:
------------------------------

Wer das GadTools-Packet gerne auf seine serie nehmen möchte kann dies
ohne jede einschränkung tun. Ich habe da allerdings eine Bitte an allen
Distributoren.: Bitte Nehmt das ganze Packet mit auf eurer serie so wie
es ist. D.H.: Nicht nur das Include-File, sondern auch die kompletten
DOCS sowie die beispiel-listings. Weiterhin ist das verbreiten auf CDs
sehr erwünscht.
```

1.10 Bug Reports bitte an den Autor senden.

```
Bug Reports bitte an den Autor senden.
--------------------------------------

Wenn ihr irgendwelche fehler gefunden habt, dann bitte ich euch mir
diese mitzuteilen, damit ich diese schnell beseitigen kann. Meine
adresse könnt ihr unter 'Anregungen' finden.
```