

# **GUICreatorWB**

Markus Hillenbrand

**COLLABORATORS**

	<i>TITLE :</i> GUICreatorWB		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Markus Hillenbrand	December 8, 2024	

**REVISION HISTORY**

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>GUICreatorWB</b>	<b>1</b>
1.1	Inhaltsverzeichnis . . . . .	1
1.2	Vorwort . . . . .	1
1.3	1.2 Warum sollte ich dieses Programm benutzen? . . . . .	2
1.4	1.3 Systemvoraussetzungen . . . . .	2
1.5	1.4 Installation . . . . .	3
1.6	1.5 Autor . . . . .	3
1.7	2.3 Registrierung . . . . .	4
1.8	Konzept . . . . .	5
1.9	2.2 Rechtliches . . . . .	6
1.10	2.4 Preferencesprogramm . . . . .	7
1.11	2.5 Hauptprogramm . . . . .	7
1.12	2.5 Hauptprogramm . . . . .	8
1.13	2.6 Programmiertips . . . . .	11
1.14	2.7 Gesucht... . . . .	11
1.15	2.8 Ausblick . . . . .	12
1.16	EBNF der Meta-Sprache . . . . .	12

# Chapter 1

## GUICreatorWB

### 1.1 Inhaltsverzeichnis

Inhaltsverzeichnis

Version 1.3 (21. Aug 95)

#### I. Einleitung

- 1.1 Vorwort
- 1.2 Warum sollte ich dieses Programm benutzen?
- 1.3 Systemvoraussetzungen
- 1.4 Installation
- 1.5 Autor

#### II. Programmbeschreibung"

- 2.1 Das Programmkonzept
- 2.2 Rechtliches
- 2.3 Registrierung
- 2.4 Preferencesprogramm
- 2.5 Hauptprogramm
- 2.6 Programmierhinweise
- 2.7 Gesucht...
- 2.8 Ausblick

### 1.2 Vorwort

#### I. Einleitung

##### 1.1 Vorwort

Was? Noch ein Programm zur Erzeugung von Benutzeroberflächen?

Eigentlich könnte man meinen, daß es schon genug Programme zur Generierung ↔  
von  
Benutzeroberflächen gibt. Doch keines dieser bisher existierenden Programme ↔  
stellte

---

mich vollends zufrieden. Die einen sind zu verspielt (was auf die Geschwindigkeit geht), andere werden nicht mehr weiterentwickelt und können so die neuen Betriebssystemfunktionen ab OS 3.0 nicht nutzen. Und der Hauptgrund zur Entwicklung dieses Programmes war, daß fast alle dieser Generatoren eine externe Library benötigen und man sich für diese oft auch noch registrieren lassen muß.

### 1.3 1.2 Warum sollte ich dieses Programm benutzen?

1.2 Warum sollte ich dieses Programm benutzen?

Oder anderes gefragt: Was hat dieses Programm denn besonderes zu bieten gegenüber anderen GUI-Generatoren?

Nun, GUI-Creator benutzt ausschließlich die Funktionen der GadTools.library v39. So- mit gibt es keine externe Library, welche Funktionen zur Verfügung stellen müßte.

Weiterhin erzeugt GUI-Creator einen gut lesbaren und leicht zu verändernden Code, der ohne GLOBALE Variablen auskommt.

### 1.4 1.3 Systemvoraussetzungen

1.3 Systemvoraussetzungen

Das Programm benötigt mindestens OS 3.0 (spätere Versionen laufen evtl. auch unter OS 2.04. Sofern Sie auf die Verwendung von Size-Gadgets, Short-Cuts für Fenster bzw. Gadgets verzichten und die SleepWindow() bzw. WakenWindow() Funktionen umschreiben, so sollte der erzeugte Code (nicht das Hauptprogramm !) auch eigentlich schon jetzt unter OS 2.04 laufen.

Ein schneller Prozessor ist empfehlenswert, auch sollten mindestens 2 MB freies RAM vorhanden sein, da bei größeren Projekten das RAM sehr schnell knapp wird.

Ach ja, eine Festplatte ist zwar nicht erforderlich, aber ratsam. (Eigentlich sollte jeder, der programmiert, eine Festplatte haben.)

Zuguterletzt sei noch erwähnt, daß ein Bildschirmformat von mindestens 640x400 Pixel

benötigt wird. Um anständig mit den vielen Fenstern arbeiten zu können sei 800 ←  
x600  
empfohlen (-> Grafikkarte).

Erwähnt sei auch noch, daß Sie einen Compiler benötigen, der den erzeugten ←  
Quellcode  
übersetzen kann.

Zur Zeit werden folgende Compiler direkt unterstützt:

- Maxon C/C++
- SAS/C
- GNU C
- DICE

In Vorbereitung sind die Generatoren für:

- Oberon
- Modula-2

## 1.5 1.4 Installation

### 1.4 Installation

Die Installation wird mittels des Installers von Commodore erledigt.

Ist die Installation beendet, rufen Sie zur Anpassung an Ihr System bitte ←  
das  
Preferences-Programm auf.

## 1.6 1.5 Autor

### 1.5 Autor

Adresse:

Markus Hillenbrand  
Hauptstraße 6  
36124 Eichenzell-Löschenrod  
( Deutschland )  
Tel.: 06659/3176 (Sa-So) und 0631/99658 (Mo-Fr)

EMail: m\_hillen@informatik.uni-kl.de  
( während der Semesterferien kaum zu erreichen )

Bankverbindung:  
Raiffeisenbank Eichenzell (BLZ 53060180)  
Konto-Nummer: 2103850

Das Programm wurde geschrieben auf einem A4000/040.

Ich bin sehr dankbar für:

---

- Geschenke
- SCSI-Festplatten
- SCSI-Streamer
- Lottogewinne

und natürlich

- Bugreports
- Verbesserungsvorschläge
- Kommentare

und vor allem

- REGISTRIERUNGEN

## 1.7 2.3 Registrierung

### 2.3 Registrierung

Die Registrierungsgebühr beträgt 30,- DM oder 30 US Dollar.

----> Formular drucken ---> (deutsch) (englisch)

Sollten Sie daran interessiert sein, weitere Programmversionen per Post zu erhalten, so kostet jede weitere Lieferung 7,50 DM. (Diese Gebühr ist sofort nach Erhalt der Lieferung auf mein Konto zu überweisen, siehe Kapitel Autor.) Sollten Sie daran interessiert sein, weitere Programmversionen per E-Mail zu erhalten, so kostet jede weitere Lieferung 4,- DM. (Diese Gebühr ist sofort nach Erhalt der Lieferung auf mein Konto zu überweisen, siehe Kapitel Autor.)

Wer GUI-Creator über eine Testphase (ca. 2 Wochen) hinaus benutzt, ohne sich registrieren zu lassen, verstößt gegen die Nutzungsbedingungen und verletzt so mein Copyright! Die unregistrierte Version ist zwar uneingeschränkt funktionsfähig, doch erscheinen von Zeit zu Zeit "Nerv-Requester", die an die Registrierung erinnern sollen. Beim Testen von GUI-Creator kann man ja bereits feststellen, ob das Programm auf der eigenen Rechnerkonfiguration zufriedenstellend funktioniert. Ist dies nicht der Fall, löscht man GUI-Creator einfach wieder und vergißt die ganze Sache (einschließlich Registrierung natürlich) oder man schreibt an den Autor des Programmes (nur per e-mail:m\_hillen@informatik.uni-kl.de) und bittet um Hilfe.

Bedenken Sie bitte, daß in GUI-Creator viele Monate Arbeit stecken, für die Regi-

strierungsgebühr meines Erachtens wirklich nicht zu viel sind, zumal Porto und Verpackung im Preis inbegriffen sind. Außerdem entscheiden auch Rückmeldungen der Nutzer in Form von Registrierungen über mögliche Weiterentwicklungen und Fehlerbeseitigungen von GUI-Creator.

Durch die Registrierung bekommt man ein persönliches "Keyfile", daß einen zur dauerhaften Benutzung von GUI-Creator und zweier eventueller zukünftiger Major-Updates berechtigt. Ich kann aber natürlich nicht zusichern, ob und wie viele Updates noch geben wird. Nach der Installation des Keyfiles verschwinden natürlich die Info-Fenster und Nerv-Requester! Spätere Major-Updates (nach den 2 freien Major-Updates) werden dann für bereits registrierte Nutzer zu fairen Preisen angeboten.

Es gibt zwei Möglichkeiten, sich registrieren zu lassen:

a) E-Mail:

Sie überweisen die Registrierungsgebühr auf mein Konto (siehe Kapitel Autor) und schreiben mir dann eine Nachricht, daß Sie sich für GUI-Creator registrieren lassen möchten und wann Sie das Geld überwiesen haben. Nachdem das Geld bei mir angekommen ist, werde ich das Keyfile per E-Mail zurücksenden. Da bekanntermaßen durchaus einmal eine Nachricht verloren gehen kann, schreiben Sie mir ruhig noch einmal, wenn meine Antwort länger als drei Wochen aussteht. Ich werde dann natürlich noch einmal das Keyfile absenden. Sollte es so gar nicht funktionieren, dann schicken Sie mir einfach einen Brief mit der Post und ich sende Ihnen eine Diskette mit Keyfile zurück. Um Pannen zu vermeiden ist es ganz wichtig, daß Sie im "Verwendungszweck" des Überweisungsvordrucks Ihre E-Mail- und am besten auch reale Adresse angeben - dann kann eigentlich nichts mehr passieren!

b) Post

Sie überweisen die Gebühr auf mein Konto (siehe Kapitel Autor) und schicken mir einen Brief, daß Sie sich für GUI-Creator registrieren lassen möchten und wann Sie das Geld überwiesen haben. Sie können anstatt der Überweisung auch einfach das Geld in bar oder Euroscheck dem Brief beilegen. Egal wie - sobald das Geld bei mir angekommen ist, werde ich Ihnen per Post die Diskette mit Keyfile zuschicken.

Wer sich jetzt registrieren lassen möchte, füllt am besten das mitgelieferte Bestellformular (deutsch od. englisch) aus, auf dem alles wesentliche noch einmal steht...

Vielen Dank für die Unterstützung des Shareware-Gedankens!

## 1.8 Konzept

---

## II. Programmbeschreibung

### 2.1 Das Programmkonzept

... ist relativ neu. Das Programm arbeitet nach einem Rasterprinzip, das heißt, die Gadgets und Fenster eines Projektes werden nicht nach Pixeln berechnet, sondern nach der Fontgröße des System-Default-Fonts. Das bedeutet, daß ein Fenster z.B. ein solches Raster hat.

## 1.9 2.2 Rechtliches

### 2.2 Rechtliches

GUICreator © 1995 Markus Hillenbrand.

GUI-Creator ist Shareware. Der Autor behält sich das Copyright für das Programm vor. Wer GUI-Creator einigermaßen ernsthaft bzw. über längere Zeit hinweg benutzen möchte, muß sich registrieren lassen. Die Weitergabe der unregistrierten Version bleibt erlaubt, soweit damit keine kommerziellen Interessen verbunden sind. Konkret darf der Preis einer Diskette, die GUI-Creator enthält, 5.- DM nicht übersteigen. Eine Aufnahme in PD-Serien ist hiermit unter diesen Bedingungen explizit erlaubt, ebenso die Verbreitung mittels DFÜ. Bei jeder Art der Weitergabe müssen alle zum Originalarchiv von GUI-Creator gehörenden Dateien unverändert und gemeinsam übernommen werden. Der Autor gewährt keine Garantie für die Richtigkeit und Funktionsfähigkeit von GUI-Creator und übernimmt keinerlei Haftung für irgendwelche negativen Folgen, die durch GUI-Creator entstehen könnten. Spätere Updates und Fehlerbeseitigungen werden nicht garantiert. GUI-Creator wurde mit dem Maxon C/C++ V3.0 der Firma Maxon Computer, Eschborn erstellt. Die Oberfläche von GUI-Creator wurde mit dem Programm selbst erstellt und erlaubt somit einen ersten Eindruck des erzeugten Quellcodes. MagicWB und somit auch einige der mitgelieferten Icons von GUI-Creator stehen unter Copyright von Martin Huttenloher.

## 1.10 2.4 Preferencesprogramm

### 2.4 Preferencesprogramm

Zur Einstellung von Parametern existiert ein externes Preferencesprogramm. Folgende Optionen bieten es an:

#### - Screenmode Preferences

Wählen Sie zuerst, ob Sie auf der Workbench arbeiten wollen, oder ob ein eigener Screen geöffnet werden soll. Den Bildschirmmodus des eigenen Screens können Sie mit Hilfe des Buttons 'Screenmode' einstellen.

#### - Code Generator Options

Hiermit stellen Sie einige Default-Werte für das Programm ein:

- a) Code Generator: Der Shell-Befehl, mit dem der Code erzeugt werden soll. Diese Befehle befinden sich im Verzeichnis "C" des Hauptprogrammes.
- b) Create Main Part: Die Voreinstellung, ob ein Hauptprogramm im Code erzeugt werden soll, welches alle benötigten Libraries öffnet und ebenso das erste Fenster anzeigt.
- c) Create all Function-Templates: Hier legen Sie fest, ob sogenannte Templates erzeugt werden sollen, also Funktionsrumpfe, die Sie später nur noch ausfüllen müssen.
- d) Create a Makefile: Hier wird festgelegt, ob ein Makefile (es ist kein echtes Makefile) erzeugt werden soll. Dieses Makefile ist ein Shell-Skript, welches Sie nur ausführen müssen, um den Code zu generieren.
- e) Create Foldmarks: Hier stellen Sie ein, ob Faltmarken für die Funktionen generiert werden sollen, z.Zt. werden nur '////' unterstützt (-> GoldED).

#### - Save und Cancel

sorgen dafür, daß die eingestellten Parameter gespeichert werden oder das Voreinstellungs-Programm beendet wird. Beenden Sie das Programm, so werden keine Modifikationen übernommen. Auch beeinflussen die neuen Werte nicht das laufende Programm; um sie zu aktivieren, müssen Sie das Programm erneut laden.

## 1.11 2.5 Hauptprogramm

Window Prefs:

- Width : Die Breite des Fensters als Vielfaches der Breite des Default-Fonts
- Height: Die Höhe des Fensters als Vielfaches der Höhe des Default-Fonts

- Name : Der Name des Fensters, der den Funktionen im Sourcecode vorangestellt wird ←  
(und Sie die Fenster somit unterscheiden können)
- Title : Der Fenster-Titel
- Close : Wenn Sie hier CLOSEWINDOW eingeben, wird das Fenster einfach geschlossen, ←  
sollten Sie etwas anderes eingeben, so wird eine BOOL-Funtion ←  
aufgerufen,  
die Sie selbst schreiben müssen. Gibt diese Funktion TRUE zurück, so ←  
wird  
das Fenster geschlossen, bei FALSE bleibt es geöffnet.
- SizeXX: Soll das Fenster ein Size-Gadget haben (unten oder rechts oder beides ?)
- Setup : Diese Funktion wird aufgerufen, nachdem das Fenster geöffnet und ←  
sämtliche  
Gadgets gezeichnet wurden.
- Refr. : Eine Funktion, die aufgerufen wird, wenn das Sizegadget des Fensters ←  
betä-  
tigt wurde (und Sie somit irgendetwas neu zeichnen müssen oder so)

## 1.12 2.5 Hauptprogramm

### 2.5 Hauptprogramm

Nach dem Start des Programmes sehen Sie zwei Gadgets, mit denen Sie entweder ←  
eine  
vorhandene Datei laden oder eine neue erstellen können. Da das Laden wohl klar ←  
ist,  
betrachten wir nun das Erstellen einer neuen GUI:

- Name of GUI: Der Name der GUI, der beim Erzeugen des Source-Codes ←  
vorangestellt  
wird, damit sich verschiedene Projekte nicht gegenseitig ←  
überschrei-  
ben.
- Path of GUI: Der Pfad, in dem der Source-Code später erzeugt werden soll. Man ←  
kann  
hier entweder einen Pfad per Hand eingeben oder das ←  
nebenstehende  
Symbol anklicken, um einen ASL-Requester zu bekommen.
- Picture : Der Name des Bildes, das angezeigt wird, wenn der Benutzer des ←  
späte-  
ren Programmes "About" aus dem Menü wählt. Dieser Pfad muß ←  
entweder  
absolut oder relativ zum Programmpfad des späteren Programms ←  
sein.  
(Am besten ist hier PROGDIR>About.IFF oder PROGDIR:Bilder/Über.IFF ←  
o.  
ähnliches).
- About Text : Wenn Sie hier Edit anklicken, so erhalten Sie eine Reihe von ←  
String-  
Gadgets, welche einen About-Text aufnehmen, also einen Text zu ←  
dem  
obigen Bild. Das kann zum Beispiel ein Hinweis auf das Programm, ←  
den  
Autor und das Copyright sein.

Wenn Sie nun Continue anklicken oder auch wenn Sie eine GUI geladen haben, ←  
erscheint  
das Layout-Fenster, mit dem Sie folgende Aktionen machen können:

- Application:

Ein Fenster, das Sie schon kennen, erscheint.

- Window :

Klicken Sie hier, um Fenster hinzuzufügen, auszuwählen, zu  
löschen oder um die Reihenfolge der Fenster zu verändern (denn das erste ←  
Fenster  
wird bei Erzeugung eines Makefile zuerst geöffnet).

- Menus :

Klicken Sie hier, um für das aktuelle Fenster Menüeinträge zu definieren. Die ←  
Ein-  
gabe von Menüs erfolgt mit einem komfortablen Requester, der intuitiv ←  
bedienbar  
ist. Achten Sie auf die Pfeile im oberen Teil des Fensters: Diese geben an, ←  
in  
welchem Modus Sie sich zur Zeit befinden (Menu, Item oder Subitem). Wenn Sie ←  
als  
Funktion 'About' angeben, so wird die Standardfunktion About() aufgerufen, die ←  
bei  
den Versionen 1.0 bis 1.2 standardmäßig an jedem Fenster hing. Sie können also ←  
nun  
selbst entscheiden, wo dieses Fenster geöffnet wird. Sollten Sie das ←  
Aboutfenster  
nicht benutzen wollen, so geben Sie in den Application-Prefs unter Aboutpic ←  
bitte  
einen leeren String ein, denn so wird die Funktion erst gar nicht erzeugt (nun ←  
ja,  
ein guter Compiler bzw. Linker sollte unbenutzte Funktionen sowieso verwerfen).

Bemerkung: sich gegenseitig ausschließende Menüpunkte (mutually exclusive menus)  
werden noch nicht unterstützt.

- Button :

Geben Sie hier die Parameter ein, die das Aussehen des Gadgets beeinflussen.

- String :

Geben Sie hier die Parameter ein, die das Aussehen des Gadgets beeinflussen.

- Integer :

Geben Sie hier die Parameter ein, die das Aussehen des Gadgets beeinflussen.

- Text :

Geben Sie hier die Parameter ein, die das Aussehen des Gadgets beeinflussen.

---

- Number :  
Geben Sie hier die Parameter ein, die das Aussehen des Gadgets beeinflussen.
  - Listview :  
Geben Sie hier die Parameter ein, die das Aussehen des Gadgets beeinflussen.
  - Checkbox :  
Geben Sie hier die Parameter ein, die das Aussehen des Gadgets beeinflussen.
  - Cycle :  
Geben Sie hier die Parameter ein, die das Aussehen des Gadgets beeinflussen.
  - Radio :  
Geben Sie hier die Parameter ein, die das Aussehen des Gadgets beeinflussen.
  - Scroller :  
Geben Sie hier die Parameter ein, die das Aussehen des Gadgets beeinflussen.
  - Slider :  
Geben Sie hier die Parameter ein, die das Aussehen des Gadgets beeinflussen.
  - Palette :  
Geben Sie hier die Parameter ein, die das Aussehen des Gadgets beeinflussen.
  - Label :  
Geben Sie hier die Parameter ein, die das Aussehen des Gadgets beeinflussen.
  - Frame :  
Geben Sie hier die Parameter ein, die das Aussehen des Gadgets beeinflussen.
  - Line :  
Geben Sie hier die Parameter ein, die das Aussehen des Gadgets beeinflussen.
  - GadgetList :  
Dies dient zum Umändern von Gadgetdaten.
  - Save :  
Zum Speichern der GUI erscheint ein Requester.
  - Source :  
Hier können Sie die Parameter ändern (siehe Prefs-Programm) und den Source- ↔ Code
-

generieren lassen. Die Applikation schläft, bis der Source-Code generiert wurde. ←  
 Eine Abfrage, ob alles positiv verlief, gibt es (noch) nicht.

## 1.13 2.6 Programmertips

### 2.6 Programmertips

- Das About-Bild muß eine Größe von 200 x 100 Pixeln haben und sollte nicht mehr als 8 Farben verwenden (am besten geht es wohl mit den vier Workbench-Farben). Sollten Sie einen eigenen Screen öffnen, auf dem die Fenster sein sollen, so sollten Sie ein Bild in der Palette des Screens malen. ←
- Die Parameter "APTR userdata" sind für Sie gedacht. Diese werden einfach durchgereicht. ←
- Die Parameter der Funktion Handle<WindowName>() sind ein Screen, auf dem die Fenster geöffnet werden sollen, die linke und obere Position der Fenster (-1 für vermitteln) sowie userdata (siehe oben). ←
- Soll ein Fenster geschlossen werden, so können Sie das auch wie folgt machen:  

```
void CancelClicked( ... )
{
    Signal(FindTask(NULL), SIGBREAKF_CTRL_C);
}
```
- Das erzeugte Makefile der GUICreator\_C hat am Anfang ein "cc" stehen. Wenn Sie hierauf ein Alias machen, so können Sie die Programme einfach kompilieren. Beim MAXON C/C++ Compiler kann das zum Beispiel so aussehen: ←  

```
Alias cc MCPP:Bin/MCPPC3 -w1500 -pc -i MCPP:Include|T: -j MCPP:Include|T: []
```
- Der Stack eines Programmes sollte auf 50.000 Byte gesetzt werden, damit es zu keinem Stack-Überlauf kommt. ←

## 1.14 2.7 Gesucht...

### 2.7 Gesucht...

Gesucht werden:

- Verbesserungsvorschläge für die Code-Generierung und die Handhabung des Hauptpro- ←

grammes.

- Hinweise auf Fehler im Programm oder in der Dokumentation.
- Programmierer, die Source-Code Generatoren für andere Sprachen (Oberon, E, Modula, Lisp :-) usw. schreiben möchten. Sollten Sie Interesse daran haben und per E-Mail erreichbar sein, so winkt Ihnen ein kostenloses Keyfile für GUI-Creator. Schreiben Sie mir, und ich sende Ihnen den Source meines C-Generators als Richtlinie zu. Und hier ist schon einmal die EBNF der Meta-Sprache der GUI-Projekte: ->
- Vielschreiber, die daran Interesse haben, die Dokumentation ins Englische, Französische, Italienische, Chinesische :-) zu übersetzen. Schicken Sie mir die Übersetzung zu, und Sie erhalten ein kostenloses Keyfile (Aber vorher fragen, ob diese schon in Arbeit ist!)

## 1.15 2.8 Ausblick

### 2.8 Ausblick

Für die Zukunft ist folgendes geplant:

- Menü-Unterstützung
- neue Gadget-Typen
- Screen-Support
- Notifications
- HELP-Taste Unterstützung via Amiga-GUIDE und Multiview
- Generierung von OS2.04 Code
- integrierter Locale-Support
- freie Fontwahl
- eingebauter Exec-List Support für Listviews und sonstiges

Alle diese Dinge sollen irgendwann einmal realisiert werden. Die Reihenfolge der Auflistung hat nichts mit deren Implementierung zu tun. Schreiben Sie auf Ihre Registrierung, was am wichtigsten ist. Und so werde ich (bei entsprechend vielen Mitteilungen) diese Sachen implementieren.

## 1.16 EBNF der Meta-Sprache

Achtung: DIES IST EINE ALTE VERSION. DIE NEUE VERSION IST WESENTLICH FLEXIBLER, DOCH HABE ICH NOCH KEINE ZEIT GEFUNDEN, DIESE IN EBNF SCHREIBWEISE ZU NOTIEREN.

ICH HABE EINEN SCANNER GESCHRIEBEN, DER DIE ZEICHEN ALS EINZELNE TOKEN ←  
 AUS-  
 WERTET UND EINEN SCANNER, DER DIESE DATEN AUSGEWERTET ALS BAUM AN DAS ←  
 PRO-  
 GRAMM SENDET.

```

boolean    = "TRUE" | "FALSE".
digit      = "1"|"2"|"3"|"4"|"5"|"6"|"7"|"8"|"9"|"0".
letter     = "a"|"b" | ... |"z"|"A"|"B" | ... |"Z".
allletters = letter|"ä"|"ü" |...|"ß".
dosstring  = letter* [":"] (letter* | "/" )*.

meta       = "GUICreator" version NEWLINE comment definition.
version    = "V" digit+.digit+.
comment    = (letter|digit)*.

NEXT       = NEWLINE TAB|SPACE.
NEXT2      = NEWLINE TAB|SPACE TAB|SPACE.

definition = "DEFINITION " letter+ NEXT path NEXT aboutpic NEXT abouttext NEXT
              "BEGIN" windowdef* NEWLINE "END" letter+.
path        = "PATH=" dosstring.
aboutpic    = "ABOUTPIC=" dosstring.
abouttext   = allletters "|" allletters "|" allletters "|" allletters "|"
              allletters "|" allletters "|" allletters "|" allletters "|"
              allletters "|" allletters "|" allletters "|" allletters "|"
              allletters "|" allletters "|".

windowdef   = "WINDOW " letter+ NEXT2 name NEXT2 title NEXT2 height NEXT2 width
              NEXT2 close NEXT2 size1 NEXT2 size2 NEXT2 setup NEXT2 refresh NEXT2 ←
              gadgetdef* NEXT letter+.
name        = "NAME=" letter+.
title       = "TITLE=" allletters*.
height      = "HEIGHT=" digit+.
width       = "WIDTH=" digit+.
close       = "CLOSEWINDOW" | letter+.
size1       = "SIZEBBOTTOM=" boolean.
size2       = "SIZEBBRIGHT=" boolean.
setup       = "SETUPFUNC=" boolean.
refresh     = "REFRESHFUNC=" boolean.

gadgetdef   = (bt | lv | st | cb | cy | in | mx | nm | sc | sl | tx | pa | la | li
              | fr)*.

position    = left "," top "," width "," height.
rest        = name "," gadgetid "," disabled "," shortcut.

left        = digit+.
top         = digit+.
width       = digit+.
height      = digit+.

name        = letter+.
gadgetid    = letter+.
disabled     = boolean.
shortcut     = letter | "".
  
```

```
text      = allletters*.

bt        = "BUTTON "   position "," rest "," text "," function.
readonly  = boolean.
spacing   = digit+.
string    = allletters*.
justify   = "GACT_STRINGLEFT" | "GACT_STRINGRIGHT" | "GACT_STRINGCENTER".
nmjustify = "GTJ_LEFT" | "GTJ_RIGHT" | "GTJ_CENTER".
replace   = boolean.
maxchars  = digit+.
checked   = boolean.
labels    = allletters "|" (allletters "|")*.
active    = digit+.
number    = digit*.
border    = boolean.
arrows    = digit+.
top       = digit+.
total     = digit+.
visible   = digit+.
freedom   = "LORIENT_HORIZ" | "LORIENT_VERT".
min       = digit+.
max       = digit+.
level     = digit+.
firstcolor = digit+.
depth     = digit+.
color     = digit+.

lv        = "LISTVIEW " position "," rest "," readonly "," spacing "," function.

st        = "STRING "   position "," rest "," maxchars "," string "," justify ←
            "," replace "," function.

cb        = "CHECKBOX " position "," rest "," checked "," function.

cy        = "CYCLE "    position "," rest "," labels "," active "," function.

in        = "INTEGER "  position "," rest "," maxchars "," number "," justify ←
            "," replace "," function.

mx        = "RADIO "    position "," rest "," labels "," active "," function.

nm        = "NUMBER "   position "," rest "," number "," border "," nmjustify ←
            "," function.

sc        = "SCROLLBAR " position "," rest "," arrows "," top "," total "," ←
            visible "," freedom "," function.

sl        = "SLIDER "   position "," rest "," level "," min "," max "," freedom ←
            "," function.

tx        = "TEXT "     position "," rest "," text "," border "," function.

pa        = "PALETTE "  position "," rest "," active "," firstcolor "," depth ←
            "," function.

la        = "LABEL "    position "," gadgetid "," text "," border "," function.
```

---

---

```
li      = "LINE "      position "," color "," function.  
fr      = "FRAME "     position "," text "," color "," function.
```