

builder

COLLABORATORS

	<i>TITLE :</i> builder		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		July 26, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	builder	1
1.1	MUIBuilder Anleitung	1
1.2	COPYRIGHT	1
1.3	Einführung	2
1.4	MUI	2
1.5	Vorteile	3
1.6	Die Bedienung von MUI-Builder	3
1.7	Grundlegende Prinzipien	4
1.8	Objekte	4
1.9	Einstellungen	5
1.10	Speichern der Oberfläche	6
1.11	Gewichtung	6
1.12	Applikation	6
1.13	Fenster	7
1.14	Zwischengespeicherte Objekte	8
1.15	Gruppe	8
1.16	bouton	9
1.17	Liste	9
1.18	Verzeichnisliste	10
1.19	Stringgadget	10
1.20	Marke	11
1.21	Blättersymbol	11
1.22	Schalterleiste	11
1.23	Bild	11
1.24	Zwischenraum	12
1.25	Häkchen	12
1.26	Schieberegler	12
1.27	Wertanzeiger	12
1.28	Skala	13
1.29	Textfeld	13

1.30 Schieber	13
1.31 'Special Characters'	14
1.32 Quelltext	14
1.33 Optionen	15
1.34 Applikations Quelltext	16
1.35 Objekt Quelltext	16
1.36 Entferne eine Kennung	16
1.37 Füge eine Kennung hinzu	16
1.38 Lokalisation	16
1.39 Katalog	17
1.40 Die Sprache C	17
1.41 gencodec_old	18
1.42 gencodec	18
1.43 Die Sprache E	19
1.44 HILFE Text	20
1.45 Registrieren	20
1.46 Zukünftige Erweiterungen	21
1.47 Grüße & Dank	21

Chapter 1

builder

1.1 MUIBuilder Anleitung

MUI-BUILDER
V1.1

Geschrieben von Eric Totel
1994

Deutsche Anleitung und Catalog von
Christian Brandel

Einführung
Vorteile von MUIBuilder
Bedienung von MUIBuilder
Quelltexterstellung
AmigaGuide-Erstellung
Zukünftige Erweiterungen
Copyright
Registrieren
Grüße & Dank

1.2 COPYRIGHT

(C) Copyright 1993 Eric Totel. Alle Rechte vorbehalten.

Sie können mich unter folgenden Adressen erreichen:

E-Mail : totel@laas.fr

oder

Eric Totel
5 rue Riquet
31000 Toulouse

Dieses Programm darf frei weitergegeben werden, solange nicht mehr

als normale Kopier- und Versandgebühren dafür verlangt werden. Jede andere Art des Vertriebs muß vom Autor ausdrücklich erlaubt werden.

Dieses Programm darf in Freeware Sammlungen aufgenommen werden, solange die obengenannten Bedingungen erfüllt sind.

Dieses Programm wird ohne jegliche Garantiezusagen zur Verfügung gestellt. Der Autor kann keinesfalls, direkt oder indirekt, für eventuelle Schäden, die sich aus Fehlern im Programm oder seiner Dokumentation herleiten, verantwortlich gemacht werden. Der Benutzer sei hiermit vor der Möglichkeit derartiger Schäden gewarnt.

1.3 Einführung

Danke, daß Sie MUI-Builder ausprobieren!!!

MUI-Builder ist ein Hilfsprogramm, von dem ich hoffe, daß Sie es einfach bedienen können. Trotzdem ist es noch weit von der Perfektion entfernt.

Senden Sie mir bitte ohne Bedenken alle Ihre Ideen und was Sie an diesem Programm mögen ... damit es in Ihrem Sinne weiterentwickelt wird.

Mit MUI-Builder, werden Sie in der Lage sein MUI Programme zu schreiben, ohne haufenweise Quelltext schreiben zu müssen oder etwas über die Syntax der @{ " MUI " link BUILD-2 } Funktionen zu wissen (auch wenn MUI selbst recht einfach ist).

Das Ziel von MUI-Builder ist es, Sie Ihre graphische Oberfläche ohne technische Probleme und ohne grösseren Aufwand erstellen zu lassen und dabei nicht das Endresultat aus den Augen zu verlieren.

Anfänglich schrieb ich dieses Programm nur für meinen eigenen Gebrauch und um den Umgang mit MUI zu lernen. MUI ist ein wunderbares Hilfsprogramm von Stefan Stuntz (dem hiermit gedankt sei).

Ich hoffe, Sie finden dieses Programm genauso nützlich, wie ich.

1.4 MUI

This application uses

MUI - MagicUserInterface

(c) Copyright 1993 by Stefan Stuntz

MUI is a system to generate and maintain graphical user interfaces. With the aid of a preferences program, the user of an application has the

ability to customize the outfit according to his personal taste.

MUI is distributed as shareware. To obtain a complete package containing lots of examples and more information about registration please look for a file called "muiXXusr.lha" (XX means the latest version number) on your local bulletin boards or on public domain disks.

If you want to register directly, feel free to send

DM 20.- or US\$ 15.-

to

Stefan Stuntz
Eduard-Spranger-Straße 7
80935 München
GERMANY

1.5 Vorteile

Viele Leute werden sich fragen ob dieses Programm überhaupt Sinnvoll ist, weil MUI relativ einfach zu programmieren ist.

Hier sind all die Vorteile, die ich und meine Beta-tester beim Benutzen von MUI-Builder gefunden haben:

1. MUI-Builder ist eine einfache Methode, schnell das Programmieren mit MUI zu lernen, indem man den erzeugten Quellcode liest.
2. MUI-Builder ist eine Art MUI-Interpreter. Man kann das GUI eines Programmes testen und sehen, wie es am Ende wirklich aussieht.
3. MUI-Builder bietet dem Benutzer eine große Flexibilität bei der Quellcode-Erzeugung. Es besteht die Möglichkeit exakt auszuwählen, zu welchem Objekt Code erzeugt werden soll, ob mit oder ohne IDCMP Schleife, mit oder ohne Deklarationen und Initialisierungen.
4. Die Online Hilfe kann direkt in Verbindung mit der grafischen Benutzeroberfläche aus MUI-Builder heraus erzeugt werden.
5. Schlußendlich ist es der erste Designer, der eine einfache Anpassung an verschiedenen Sprachen (LOCALE) bietet !

1.6 Die Bedienung von MUI-Builder

Das Prinzip, nach dem MUI-Builder benutzt wird, kann in folgende Gruppen unterteilt werden. :

Grundlegende Prinzipien
Objekte
Einstellungen
Speichern der Oberfläche

Drücken Sie einfach 'HELP' wann immer Sie im Programm Hilfe benötigen.

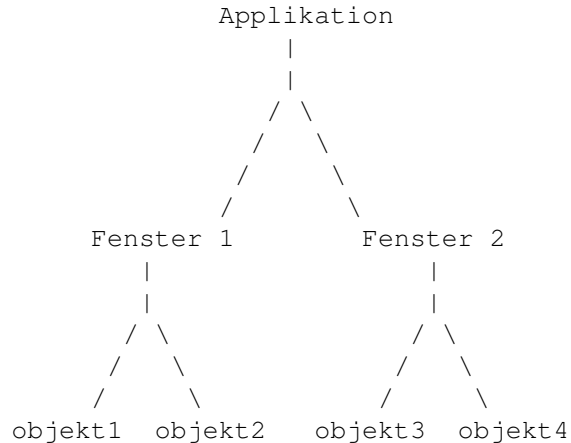
1.7 Grundlegende Prinzipien

Die Erstellung einer Oberfläche mit MUI-Builder fußt auf der gleichen Grundidee wie bei MUI, weshalb die beiden Programme in starkem Zusammenhang zueinander stehen.

Deshalb ist mein erster Rat der, die MUI Dokumentation zu lesen, um zu erfahren, was diese Grundidee beinhaltet.

Mit MUI-Builder können Sie die vollständige Graphische Oberfläche eines Programmes erstellen. Zum Beispiel sind alle Fenster dieses Programmes mit MUI Builder erstellt worden.

Jedes Programm besteht aus einem Baum von Objekten. Das folgende Beispiel zeigt eine Applikation mit zwei Fenster, von denen jedes zwei weitere Objekte enthält.



Die Idee hinter MUI-Builder ist es, diesen Baum einfach mit Hilfe einer (wie ich hoffe) benutzerfreundlichen grafischen Oberfläche erstellen zu können.

1.8 Objekte

Mit MUI-Builder können Sie alle Objekte erstellen, die Sie in Ihrem Programm verwenden wollen.

Das sind folgende Objekte :

	Applikation		
	Fenster		
Gruppe	Group	Schalter	Button
Liste	List	Verzeichnisliste	DirList
Stringgadget	String	Marke	Label
Cyclegadget	Cycle	Radio Button	Radio
Bild	Image	Zwischenraum	Space
Häkchen	CheckMark	Slider	Slider
Gauge	Gauge	Skala	Scale
Text	Text	Propgadget	Prop

Ein Objekt, das schon erstellt wurde, kann durch einfaches Doppelklicken in der Liste verändert werden.

Einige Objekte haben ein Attribut für 'Gewichtung', das verändert werden kann.

1.9 Einstellungen

Das Einstellungsfenster bietet eine Vielzahl von Möglichkeiten einzelne Programmparameter einzustellen. Alle diese Einstellungen werden in einer Environmentvariablen (ENV:MUIBuilder.env) gespeichert, die während der Installation erzeugt wird.

Sie haben folgende Möglichkeiten:

- Nur ein Fenster : ist dieser Schalter angewählt, erscheint nie mehr als ein Fenster gleichzeitig auf dem Schirm, vielmehr schliessen sich die momentan nicht benötigten. Das kann sehr nützlich sein, wenn Sie nicht all zu viele Fenster gleichzeitig auf dem Schirm haben wollen, oder wenn Sie eine relativ langsame Maschine haben (z.B. 68000).
- Piktogramm : wenn Sie diesen Schalter wählen, werden die Dateien, die MUIBuilder erzeugt mit einem Standardpiktogramm versehen. Dieses befindet sich unter dem Namen 'def_MUIBuilder.info' im Verzeichnis 'ENV:sys/'. Es wird normalerweise bei der Installation dorthin kopiert.
- Aktionen bestätigen : zeigt an, ob Sie jede 'gefährliche' Aktion in einem Requester bestätigen wollen (z.B. vor dem Löschen eines Objektes).
- Quellcode : Zeigt an, welchen Quellcodegenerator Sie verwenden wollen.
- Editor : Geben Sie hier Ihren Lieblingseditor an! Vergewissern Sie sich, daß der Editor 'synchron' gestartet wird. (d.h. daß er sich, vom CLI gesteuert, nicht selbst löst, sondern das Prompt erst nach Beenden des Editors zurückkehrt !)

Folgende Editoren sollten ohne Probleme laufen:

- Vim (v2.0) : geben Sie die Option '-x' mit an
- CygnusEd : mit der Option '-keepio'
- TurboText : mit der Option WAIT
- GoldEd : mit der Option 'STICKY'
- Ed : der voreingestellte Editor

- GetString : ACHTUNG das hier ist WICHTIG !!!
Wenn Sie hier einen String eingeben, wird er zur Bildung des Namens der Funktion 'GetString' benutzt, die zur Lokalisation benötigt wird. Dieser String wird in das entsprechende Gadget des 'Quellcode' Fensters kopiert, wenn diese zum ERSTEN Mal geöffnet wird (nur einmal, damit Sie auch einen andere als den voreingestellten angeben können). Wird dieser String nicht angegeben, so wird er aus dem Namen der Applikation gebildet ('GetApplikationsnameString').

Haben Sie die passenden Einstellungen gewählt, können diese mit dem entsprechenden Schalter abspeichern !

1.10 Speichern der Oberfläche

Die SAVE und LOAD Schalter speichern bzw. laden die Benutzeroberfläche einer Applikation, die Sie erstellen.

Mit ihrer Hilfe können unfertige Applikationen weiterbearbeitet oder bestehende verändert werden.

Beachten Sie, daß das MUI-BUILDER Fenster ein 'AppWindow' ist. Eine MUI-Builder Applikationsdatei kann also durch einfaches Ziehen ihres Piktogrammes in das Fenster geladen werden.

1.11 Gewichtung

Das Attribut für 'Gewichtung' ist eines der interessantesten eines Objektes. Es ermöglicht es, die Ausmaße eines Objektes relativ zu denen anderer zu definieren.

Jedes Objekt hat eine voreingestellte 'Gewichtung' von 100. Das bedeutet, daß jedem Objekt der gleiche Anteil an Platz in einem Fenster zukommt.

Wird dieser Wert verändert, so ändert sich die Größe einer Gruppe oder eines Objektes relativ zu anderen Gruppen oder Objekten. Wenn Sie z.B. zwei Objekte haben und eines dreimal so groß machen wollen wie das andere, würde ein Objekt die Gewichtung 75 bekommen, das andere die Gewichtung 25.

Auf diese Art und Weise kann man vermeiden, daß Schalter größer werden, als sie tatsächlich sollen.

Eine Gewichtung von NULL läßt das Objekt immer in den kleinstmöglichen Ausmaßen erscheinen, was immer auch passieren sollte.

1.12 Applikation

Die Applikation ist der Wurzeleintrag in dem Baum, der die komplette grafische Oberfläche darstellt.

Nach dem Anwählen des 'Appli'-Schalters im Hauptfenster können folgende Parameter eingestellt werden:

- Die 'Base' der Applikation d.h. den Namen der AREXX Adresse des Programms
- den Namen des Programmautors
- die Programmversion
- den Copyrightvermerk
- eine kurze Beschreibung des Programmes

Die Applikation kann nur eine Art von Kind haben : das Fenster .

1.13 Fenster

Wir alle wissen was ein Fenster ist! Aber das Problem ist, wie man eines mit MUI-Builder erstellt...

Zwei Pages teilen sich den Platz : eine mit Namen 'Aussehen', eine andere namens 'Eigenschaften' :

Die Page 'Aussehen' ist wiederum in zwei Teile aufgeteilt :

1. Einen Textbalken, in dem man die Art der Beziehung zwischen den angewählten Objekten erkennen kann.
2. Und schließlich eine Liste der Kinder, Enkelkinder usw. des Fensters. Diese sind in zwei Gruppen unterteilt :
 - 2.1 Die linke Liste :
Gruppen, die mit dem Fenster verbunden sind (inklusive der Haupt oder Wurzelgruppe, die das einzige direkte Kind des Fensters ist)
 - 2.2 Die mittlere Liste :
Die Kinder der (in der linken Liste) angewählten Gruppe .
 - 2.3 Die rechte Liste :
Eine Liste zwischengespeicherter Objekte (svw. tmp list).

Gruppen sind die grundlegenden Objekte, in die die anderen Objekte eingefügt werden.

Jedesmal, wenn ein Objekt erstellt wird, muß es als das Kind einer Gruppe definiert werden. Diese Gruppe kann durch anwählen in der Liste bestimmt werden.

Auf der Page 'Eigenschaften' befinden sich :

- Die Kennung des Fensters sowie sein Titel.
- die Attribute des Fensters :
 - AppWindow : Das Fenster soll ein AppWindow sein
 - Ohne Rahmen : wenn Sie keinen Rahmen wollen !
 - Depth Gadget : wollen Sie es?
 - Größensymbol : das auch?
 - Immer hinten : ob Ihr Fenster immer hinter anderen sein soll !
 - Schließsymbol : Wollen Sie das Fenster schließen können?
 - Ziehleiste : damit man das Fenster herumziehen kann

1.14 Zwischengespeicherte Objekte

In dieser Liste können Objekte abgelegt werden, die zwischen Gruppen oder Fenstern verschoben werden sollen.

1.15 Gruppe

Wird ein neues Fenster erstellt, so hat dieses schon eine voreingestelltes Kind. Dieses Kind ist die Wurzelgruppe.

Das Layout eines Fensters wird durch Anhängen von Objekten an diese Wurzelgruppe bestimmt.

Solche Kinder können wiederum auch ganze Gruppen darstellen, die wiederum eigene Kinder haben ... etc ...

Für jede Gruppe müssen diese Attribute festgelegt werden :

- Horizontal : Die Objekte dieser Gruppe werden horizontal angeordnet.
 - Page Modus : Die Gruppe zeigt nur eines ihrer Kinder auf einmal an. In der Liste müssen Sie dann die Titel der einzelnen Pages angeben.
 - Gleiche Höhe : Alle Kinder dieser Gruppe bekommen die selbe Höhe.
 - Gleiche Breite : Alle Kinder dieser Gruppe bekommen die selbe Breite.
 - Gleiche Größe : Alle Kinder dieser Gruppe bekommen die selbe Größe
 - Virtuelle Gruppe : Es handelt sich um eine virtuelle Gruppe.
 - Titel : Die Gruppe bekommt einen Titel.
 - Gewichtung : die Gewichtung einer Gruppe kann angegeben werden ↔
-

- Spalten : Ordnet die Gruppe in Spalten an.
Die Anzahl der Spalten wird in das Stringgadget eingetragen.
- Zeilen : Ordnet die Gruppe in Zeilen an.
Die Anzahl der Zeilen wird in das Stringgadget eingetragen.
- Lücke
Horiz. Lücke : Der horizontale Abstand zwischen den Objekten kann kontrolliert werden.

Vert. Lücke : Der vertikale Abstand zwischen den Objekten kann kontrolliert werden.

Der Wert des Abstandes muß im zugehörigen Stringgadget eingegeben werden

1.16 bouton

Um einen Schalter vollständig zu definieren, muß folgendes angegeben werden :

- seine Kennung
- der Text, den er enthalten soll
- das zugehörige Tastaturkürzel

Seine @{ " Gewichtung " LINK Poids } kann ebenso angegeben werden.

1.17 Liste

Um eine Liste vollständig zu definieren, muß folgendes angegeben werden :

- ihre Kennung
- ihre @{ " Gewichtung " LINK Poids }
- ob man in dieser Liste 'Doppelklicken' darf
- ob diese Liste Mehrfachauswahl unterstützt
- die Art der Liste :
 - Standard
 - eine Fließtextliste, in der Text angezeigt werden kann

- eine List von Datenträgern (und Zuweisungen)

Zusätzlich existiert noch die sogenannte @{ " Verzeichnisliste " LINK DirList ↔ } .

Diese kann durch Anwählen ihres Schalters im 'Object Choice' Fenster erstellt werden.

1.18 Verzeichnisliste

Die Verzeichnisliste zeigt die Dateien und Unterverzeichnisse im ausgewählten Verzeichnis an.

Folgendes kann eingestellt werden :

- die @{ " Gewichtung " LINK Poids }
- 'Nur Verzeichnisse' : zeigt NUR Verzeichnisse
- 'Nur Dateien' : zeigt NUR Dateien
- 'Mehrfachauswahl' : erlaubt Mehrfachselektierung der Dateien in der Liste
- '.info Dateien verbergen' : verbirgt '.info' Dateien
- 'Absteigend sortieren' : sortiert fallend
- 'Sortieren nach' : Auswahl des Sortierkriteriums (Name, Datum, Größe)

Das Stringgadget 'dir' muß den Namen des Verzeichnisses enthalten, das angezeigt werden soll.

1.19 Stringgadget

Um ein Stringgadget vollständig zu definieren, muß folgendes angegeben werden :

- sein Titel (der verändert werden kann)
 - seine Kennung
 - der anfängliche Inhalt
 - eine Zeichenkette, die alle Zeichen enthält, die in dieses Feld eingegeben werden können
 - Eine Zeichenkette, die alle Zeichen enthält, die in dieses Feld nicht eingegeben werden können
 - die maximale Länge des Eingabefeldes
-

Auch hier kann die @{ " Gewichtung " LINK Poids } angegeben werden.

1.20 Marke

Um eine Marke vollständig zu definieren, muß folgendes angegeben werden :

- ihre Kennung
- der Text, der auf dem Schirm erscheinen soll
- ihre @{ " Gewichtung " LINK Poids }

1.21 Blättersymbol

Um ein Blättersymbol vollständig zu definieren, muß folgendes angegeben werden :

- die Liste seiner Einträge
- seine Kennung
- seine @{ " Gewichtung " LINK Poids }

1.22 Schalterleiste

Um eine Schalterleiste vollständig zu definieren, muß folgendes angegeben werden :

- die Liste der Schalter
- ihre Kennung
- ihre @{ " Gewichtung " LINK Poids }

1.23 Bild

Als erstes muß das Bild, das angezeigt werden soll, durch Anklicken ausgewählt werden.

Es gibt folgende Möglichkeiten :

- 'Vert. frei' : Das Bild kann seine vertikale Größe ändern.
 - 'Horiz. frei' : Das Bild kann seine horizontale Größe ändern.
 - 'Eingabe möglich' : Der Benutzer kann das Bild anwählen.
 - 'Feste Höhe' : setzt die Höhe eines Bildes auf den Wert
-

- des zugehörigen Eingabefeldes fest.
Damit wird 'Free Vertical' bedeutungslos.
- 'Feste Breite' : setzt die Breite eines Bildes auf den Wert
des zugehörigen Eingabefeldes fest.
Damit wird 'Free Horizontal' bedeutungslos.

Wie bei anderen Objekten, muß auch hier die Kennung angegeben werden.

1.24 Zwischenraum

Dieses Objekt stellt einen Zwischenraum zwischen zwei anderen Objekten dar und erlaubt es, die Größe des Fensters zu verändern.

1.25 Häkchen

Der Titel kann entweder links oder rechts vom Häkchen platziert werden.

Wenn ein Titel erscheinen soll, muß er im Eingabefeld 'Titel' angegeben werden.

Nicht vergessen, auch hier eine Kennung anzugeben.

1.26 Schieberegler

Für einen Schieberegler muß folgendes angegeben werden :

- ob der momentane Wert angezeigt werden soll ('Keine Nachricht')
- ob der Schieberegler einen Titel hat oder nicht
- ob der minimale Wert links liegt oder rechts ('umgekehrt')
- der maximale Wert
- der minimale Wert
- der anfängliche Wert

Es kann außerdem angegeben werden :

- der Titel
- die Kennung

1.27 Wertanzeiger

Um einen Wertanzeiger zu definieren, muß folgendes angegeben werden :

- seine Ausrichtung (horizontal oder vertikal)

- soll die Höhe auf einen festen Wert gesetzt werden, muß dieser in dem zugehörigen Eingabefeld angegeben werden. Dies wird besonders bei horizontalen Wertanzeigern genutzt.
- soll die Breite auf einen festen Wert gesetzt werden, muß dieser in dem zugehörigen Eingabefeld angegeben werden. Dies wird besonders bei vertikalen Wertanzeigern genutzt.
- ob sein Wert vor der Anzeige geteilt werden soll ('Teile durch')
- sein maximaler Wert
- seine Kennung

1.28 Skala

Eine Skala muss zusammen mit einem Wertanzeiger verwendet werden und zeigt dann den prozentualen Wert daneben an.

Es muß nur die Ausrichtung angegeben werden.

1.29 Textfeld

Um ein Textfeld vollständig zu definieren, muß folgendes angegeben werden :

- 'Maximum' : zu Anfang wird die maximale Größe des Feldes verwendet.
- 'Minimum' : zu Anfang wird die minimale Größe des Feldes verwendet.
- 'Hintergrund' : ausprobieren und staunen !
- 'Rahmen' : ausprobieren und staunen !

Der Text muß im zugehörigen Eingabefeld angegeben werden und kann jeden 'special character' , der auch in einem C Quelltext vorkommen kann, enthalten.

Und nicht die Kennung vergessen!

1.30 Schieber

Folgende Attribute stehen zur Auswahl:

- 'Horizontal Prop' : der Schieber soll horizontal sein
- 'Feste Breite' : setze die Breite auf den im Eingabefeld angegebenen Wert fest.
- 'Feste Höhe' : setze die Höhe auf den im Eingabefeld angegebenen Wert fest.

Dann sollte angegeben werden :

- die Anzahl der Einträge
- die Nummer des ersten Eintrages
- die Anzahl der sichtbaren Einträge

...die Kennung nicht vergessen.

1.31 'Special Characters'

In JEDEM Text der grafischen Oberfläche können folgende 'special character' angegeben werden :

- \n neue Zeile
- \r zum Anfang der Zeile
- \t Tabulator
- \e Escape
- \ das Zeichen '\'
- \" Anführungszeichen
- \xNN Das Zeichen mit dem ASCII-Wert NN (hexadezimal angegeben)
- \nnn Das Zeichen mit dem ASCII-Wert nnn (oktal angegeben)
- \c c wenn c irgend ein anderes Zeichen ist

Einige Beispiele :

- \033b für fettgedruckten Text
- \033n um wieder auf normalen Text zurückzuschalten
- \0338 für weißen Text
- \033c um den Text zu zentrieren
- \033l für linksbündigen Text
- \033r für rechtsbündigen Text
- ...

Achtung: um Anführungszeichen zu erhalten muß '\"' anstatt ''' eingegeben werden, sonst gibt es Fehler beim Übersetzen des Programms.

1.32 Quelltext

Nach Auswahl des 'Code' Schalters im Hauptfenster von MUI-Builder wird mit der Erzeugung des Quelltextes begonnen.

MUI-Builder überprüft zunächst, ob auch keine Objekt-Kennung doppelt verwendet worden ist.

Wenn alles in Ordnung ist, werden zwei Listen angezeigt:

- eine mit den Namen aller Objekte in der Applikation
- eine zweite mit allen Kennungen, die im Quelltext erzeugt werden

MUI-Builder kann in der Art, in der er die Kennungen im Quelltext erzeugt, sehr stark beeinflusst werden. Wird von dieser Möglichkeit kein Gebrauch

gemacht, wird alles automatisch erstellt.

Bevor der Quelltext erzeugt wird, müssen noch die Optionen angegeben werden.

Folgende Schalter stehen zur Verfügung :

- App Quelltext
- Object Quelltext
- Entferne Kennung
- Füge Label hinzu

Das Programm erzeugt eigentlich einen symbolischen Code, der eher eine Art Beschreibung der grafischen Oberfläche ist, als ein zum Übersetzen fertiger Quelltext.

Nachdem dieser Code erzeugt wurde (temporäre Datei in 'T:'), führt MUI-Builder eines der Module (aus dem 'modules' Verzeichnis) für die Erzeugung des Quelltextes aus, das auf die temporäre Datei zurückgreift.

Momentan werden die Sprachen C und E unterstützt.
Die Erzeugung von Assembler-Quelltext wird demnächst dazukommen.

Wenn sie glauben, daß Sie genug (!!!) wissen, um ein passendes Modul für ihre bevorzugte Programmiersprache zu schreiben : Melden Sie sich bei mir !!! Ich werde Ihnen dann erklären, wie man den Code benutzt, um daraus den Quelltext für eine Programmiersprache zu erzeugen.

Mit Hilfe des Einstellungs Fensters (Schalter 'Einstellungen') kann die gewünschte Sprache ausgewählt werden. Momentan sind das folgende :

```
@{ " C Quelltext " link Langage_C }
@{ " E Quelltext " link Langage_E }
```

1.33 Optionen

Zur Zeit stehen vier Optionen zur Verfügung :

- 'Deklarationen' : um die Deklarationen und Initialisierungen im Quelltext zu bekommen.
- 'Environment' : Schalter, um Quelltext für Includes, Abfrageschleife, Prozedurdeklaration usw. zu bekommen.
- 'Code' : Schalter, um den MUI-Quelltext zu bekommen.
- 'Lokale' : Schalter, um von MUIBuilder lokalisierten Code erzeugen zu lassen.
(siehe Lokalisation)

Jede Option kann unabhängig von den anderen gewählt werden. Man kann also genau den Teil des Quelltextes erzeugen lassen, den man benötigt.

Beispiel:

Sie haben ein Fenster erstellt und wollen einen Schalter in den Quelltext einfügen. Wählen Sie nur die 'Code' Option und der Quelltext für den Schalter wird erzeugt! Nachdem Sie diesen in Ihr Programm eingefügt haben, brauchen Sie die Deklaration des Schalters. Wählen Sie 'Declaration' ... und fügen Sie den erzeugten Text direkt dort ein, wo Sie ihn in Ihrem Quelltext haben wollen.

1.34 Applikations Quelltext

Mit diesem Schalter können Sie den Quelltext für die gesamte Applikation erzeugen lassen.

1.35 Objekt Quelltext

Mit diesem Schalter erzeugt MUI-Builder den Quelltext eines zuvor aus der 'Objects labels' Liste ausgewählten Objekts.

1.36 Entferne eine Kennung

Mit diesem Schalter kann die Erzeugung von Quelltext für das in der 'Generated Labels' Liste selektierte Objekt unterdrückt werden.

MUI-Builder erkennt automatisch, ob er die Kennung zu jedem Objekt erzeugen muß. So ist es zum Beispiel unnötig, einen Zeiger auf eine Gruppe aufrecht zu erhalten, solange man nicht während des Programmablaufs dynamisch Objekte zu ihr hinzufügen möchte.

Dieser Schalter (und der Add Label Schalter) erlauben es, MUI-Builders Standard Definitionen von Objekten zu ändern.

1.37 Füge eine Kennung hinzu

Mit diesem Schalter wird MUI-Builder mitgeteilt, daß er die Kennung des ausgewählten Objektes erzeugen soll.

Siehe auch Remove Label .

1.38 Lokalisation

Wenn Sie die Option 'Lokale' im 'Optionen' Fenster anwählen, haben Sie sich dazu entschieden, lokalisierten Quellcode zu erzeugen.

Jeder String und jedes Tastaturkürzel wird nun durch einen Aufruf einer

'GetString' Funktion ersetzt, deren Aufgabe darin besteht, einen lokalisierten String aus dem Katalog des Programmes suchen. Zahlreiche Programme können diese Funktion automatisch für Sie erstellen (ich nenne hier nur CatComp und FlexCat).

Der Name der Funktion 'GetString' kann durch Ändern des Strings, der im entsprechenden Gadget der Fenster 'Quellcode' bzw. Einstellungen steht, festgelegt werden.

Denen, die nicht so recht wissen, wie das funktionieren soll, empfehle ich, einen Blick auf eines der Beispiele zu werfen, die sich im MUIBuilder Archiv befinden.

Die Katalogbeschreibungsdatei kann durch Anwählen des Schalters @{" Katalog " ↔ link Catalogue} im Quellcode Fenster erzeugt werden.

1.39 Katalog

Der Name der '.cd' Datei muß im erscheinenden Auswahlfenster angegeben werden, bevor sie erzeugt werden kann.

Ich werde hier eine kurze Zusammenfassung dessen geben, was man über die Katalogbeschreibungsdatei (das 'xxxx.cd' File) wissen sollte.

In dieser Datei finden Sie alle Strings ihres Programmes in der voreingestellten Sprache.

Dank dieser Datei, können Sie (mithilfe von z.B. CatCom oder FlexCat) folgendes automatisch erstellen :

- die Übersetzungsdatei ('xxx.ct' File) in der sich die entsprechenden Strings einer anderen Sprache befinden
- die entsprechenden Programmdateien in Ihrer bevorzugten Sprache

(für weitere Details lesen Sie bitte die Anleitungen dieser Programme)

Die Katalogbeschreibungsdatei benutzt folgende Konvention für Tastaturkürzel:

_b Beispiel

ergibt: Beispiel

1.40 Die Sprache C

Folgende Dinge sollten beachtet werden :

- Manche Objekte benutzen Hilfsvariablen, die sich von denen unterscheiden, die sie direkt definieren. Ein Text, z.B. muß einen zugeordneten String

haben, der seinen Inhalt darstellt. Auf die selbe Art muß auch eine FloatText Liste auf eine Zeichenkette zeigen.
Diese Variablen werden automatisch erzeugt und haben, am Beispiel der Zeichenketten, den Namen 'STR_"variablen_name"'.
Beispiel : Wenn Ihr Katalog 'MUIBuilderStrings.cd' heißt, steht am Anfang des Quelltextes '#include MUIBuilderStrings.h'

- Seit lokalisierter Quellcode erzeugt wird, wird am Anfang eines Programms ein Header-File eingebunden, in dem alle Zeichenketten initialisiert werden : der Name dieses Files wird aus dem Namen des Katalogs gebildet.

Beispiel : Wenn Ihr Katalog 'MUIBuilderStrings.cd' heißt, steht am Anfang des Quelltextes '#include MUIBuilderStrings.h'

Außerdem stehen mit MUIBuilder zwei Module für die Erzeugung von C-Code zur Verfügung :

- @{ " GenCodeC_old " link GenCodeC_old } : hat die Vorteile und Nachteile der Version 1.0 ↵
- @{ " GenCodeC " link GenCodeC } : ein neues Modul, das es Ihnen ermöglicht, den MUI-Quelltext von Ihrem Programm zu trennen. ↵
- Der von MUI-Builder erzeugte Quelltext reagiert weder auf Eingaben, noch öffnet er Fenster etc. Wenn das Programm also schnell getestet werden soll, muß vor der Abfrageschleife zumindest ein Fenster geöffnet werden (set(WI_window, MUIA_Window_Open, TRUE)). Wird nur diese eine Zeile eingefügt, muß das Programm über das 'Exchange' Commodity beendet werden.

1.41 gencodec_old

Der erzeugte Quelltext benutzt eine Header-Datei namens 'code.h.' Sie sollte sich im selben Archiv wie MUI-Builder befinden.
Diese Datei wird von Stefan Stuntz in seinem MUI-Paket mitgeliefert und ermöglicht das Übersetzen mit den meisten C-Compilern.

1.42 gencodec

Dieses Modul ist bei weitem das interessantere der beiden !

Mit seiner Hilfe erzeugen Sie ein MUI-File, das unabhängig von Ihrem Programm ist.

Es enthält zuerst den Inhalt des Files 'C-Header', das sich im Verzeichnis 'Modules' befinden muß, das aber von Ihnen abgeändert werden kann damit MUIBuilder genau das erzeugt, das Sie wollen !!

Es werden zwei Dateien Erzeugt :

- eine 'xxxx.c' Datei, die zwei Funktionen enthält :
 - CreateObject, die einen Zeiger auf eine Struktur

- zurückgibt
- DisposeObject, die den Speicher eines erzeugten Objektes wieder freigibt
- eine 'xxxx.h' Datei, die Sie in Ihr Programm einbinden müssen und die die Definition der Struktur enthält, deren Felder die Namen der von Ihnen erzeugten Objekte Ihrer MUI-Oberfläche sind.

Siese Dateien werden folgendermaßen verwendet :

```
#include <libraries/mui.h>

#include <clib/muimaster_protos.h>
#include <clib/alib_protos.h>
#include <clib/dos_protos.h>
#include <clib/exec_protos.h>

#include <pragmas/muimaster_pragmas.h>
#include <pragmas/exec_pragmas.h>

#include "test.h"          /* ---- wird von MUIBuilder erzeugt ---- */

struct Library * MUIMasterBase;

main()
{
    struct ObjApp * app;
    BOOL    running = TRUE;
    ULONG    signal;
    extern struct Objapp * CreateSmall_Example( void );

    Init();                /* Ihre Initialisierungen          */
                          /* Libraries öffnen ( locale, MUI ... ) */

    app = CreateApp();      /* ---- wird von MUIBuilder erzeugt ---- */

                          /* Hier alle Notifikationen einfügen    */

    set( app->WI_main, MUIA_Window_Open, TRUE ); /* Fenster öffne dich... */

    while (running)        /* Hauptschleife des Programms          */
    {
        switch (DoMethod(app->app,MUIM_Application_Input,&signal))
        {
            case MUIV_Application_ReturnID_Quit:
                running = FALSE;
                break;
        }
        if (signal) Wait(signal);
    }
    DisposeApp(app);       /* ---- wird von MUIBuilder erzeugt ---- */
}
```

1.43 Die Sprache E

Bitte Sehen Sie die Dokumentation für die E Sprache.

1.44 HILFE Text

Dank MUIBuilder hat jetzt keiner mehr eine akzeptable Ausrede, sein Programm ohne das, was auf anderen Maschinen (inklusive MS-DOS, ...kein Kommentar!) weithin als Online Hilfe bekannt ist, nämlich einer HyperText Anleitung, zu erstellen. Jedem MUI-Objekt, das von MUI-Builder erstellt wird, kann ein Hilfetext zugeordnet werden.

MUI-Builder erzeugt dann automatisch eine Hypertext Dokumentation im Format von AmigaGuide (die nach Wahl des 'View' Schalters begutachtet werden kann).

Wie bei der Quelltexterzeugung, kann auch nur ein Teil der Dokumentation erzeugt werden, um ihn später in eine bestehende Dokumentation einzufügen. Wenn also in einer von MUIBuilder erzeugten Anleitung Änderungen vorgenommen werden, sind diese nicht verloren : Es werden einfach nur die Teile neu erzeugt, die geändert wurden, und anschließend in das Dokument eingefügt.

Verändert werden kann folgender Text:

- der Anfangstext mit dem 'App Node' Schalter
- ein Text für ein Fenster mit dem 'Window Node' Schalter
- ein Text für ein Objekt mit dem 'Object Node' Schalter

Es kann eine Dokumentation erstellt werden für :

- die Applikation mit dem 'Ganzes Doc erzeugen' Schalter
- für ein Fenster mit dem 'Fenster Doc erzeugen' Schalter
- für ein Objekt mit dem 'Objekt Doc erzeugen' Schalter

Mit einem Doppelklick auf einen Objekt oder Fensternamen kann der Titel des dazugehörigen Hilfstextes verändert und angesehen werden. (wie mit den 'Ändern' Schaltern).

1.45 Registrieren

Nichts leichter als das!!
Dieses Programm ist GiftWare!!!

Das heißt, daß Sie in KEINER Weise gezwungen werden, mir Geld zu schicken, um das Programm benutzen zu dürfen.

Aber wenn Sie das Programm wirklich mögen oder es nützlich finden, dann dürfen (ha, ha, ha!) Sie mir ca. 15 \$US oder 50FF an die folgende Adresse schicken:

Eric Totel

5 rue Riquet
31000 Toulouse
France

Wenn Sie MUI-Builder nicht dermaßen überzeugt hat, daß Sie Geld senden wollen (oder wenn Du ein abgebrannter Student bist), dann schicken Sie mir einfach eine Postkarte, einen Brief, eine EMail (totel@laas.fr), um mich zu ermutigen, mir einige Ideen, Anmerkungen usw. zukommen zu lassen.

Scheuen Sie sich in keinem Fall: Ich will unbedingt wissen, ob die Weiterentwicklung von MUI-Builder wirklich nützlich und notwendig ist.

Wenn Sie andererseits die Dokumentation in Ihre Sprache (oder eine andere, die noch nicht unterstützt wird) übersetzt haben, dann bitte nichts wie her damit !!!!

1.46 Zukünftige Erweiterungen

Zukünftige Erweiterungen werden evtl. sein :

- Notifikation (Alles was mit der Steuerung durch Ereignisse zu tun hat) direkt aus MUIBuilder heraus.
- Die Möglichkeit, via AREXX mit Editoren Text auszutauschen, damit erzeugter Quelltext direkt ins Programm eingebunden werden kann.
- Alle interessanten Ideen, die Sie mir vorschlagen (Versäumen Sie es auf keinen Fall, mir ALLE Ihre Ideen und Anmerkungen mitzuteilen, damit das Programm Ihren Anforderungen gerecht werden kann).

1.47 Grüße & Dank

Ich möchte all den Leuten danken, die mir geholfen haben und die ihren Beitrag zu MUI-Builder geliefert haben :

- Pierre Carrette für seine wirklich große Hilfe und seine Ideen, eine interessanter als die andere !!!
 - Lionel Vintenat für die Idee des symbolischen Codes und die Implementation des Moduls für die Sprache E.
 - Gael Marziou, Pascal Pensa, Christian Brandel, Daniel Murrel für ihre sorgfältigen Tests und ihre zahlreichen Vorschläge.
 - Pascal Rabier und Mike Manzano, die mir geholfen haben, die Dokumentation aus dem Französischen ins Englische zu übersetzen.
 - Christian Brandel, für seine Übersetzungen einerseits der Anleitung, andererseits des Kataloges ins Deutsche !
 - Tobias Ferber für sein wunderbares MagicWB Icon (ach quatsch ... nicht
-

wunderbar : FANTASTISCH !!! :-))

- Michal Kara, für seine übersetzung des Kataloges ins Czech, und die Implementation des Moduls für die Assembly Sprache.