

# MUIBuilder

---

Version 2.1  
Anleitung

Eric Totel

---

Copyright © 1993-1994 Eric Totel

Dies ist die zweite Version der Anleitung zu MUIBuilder und gehört zur Version 2.1 von MUI-Builder. Sie darf vervielfältigt und weitergegeben werden, sofern der Copyrightvermerk und diese Erlaubnis auf allen Kopien intakt belassen werden. Übersetzungen in andere Sprachen dürfen unter obigen Bedingungen vervielfältigt und weitergegeben werden, sofern sie die schriftliche Einverständniserklärung des Autors haben.

# 1 Einführung

Zuerst mal Danke dafür, daß Sie MUIBuilder 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** 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 oder Umweg erstellen zu lassen.

Ich hoffe, Ihnen wird dieses Programm genauso nützlich sein, wie mir.

## 1.1 Die Vorteile von MUIBuilder

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 Betatester 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. MUIBuilder ist der erste Designer, der eine einfache Anpassung an verschiedenen Sprachen (LOCALE) bietet !
6. In dieser neuen Version tauchen fantastische Funktionen auf, die in der Welt des Amiga bisher einzigartig sind :). Man kann nun die Ereignisauswertung direkt und während des Entwurfs des GUI aus MUIBuilder heraus definieren UND TESTEN.



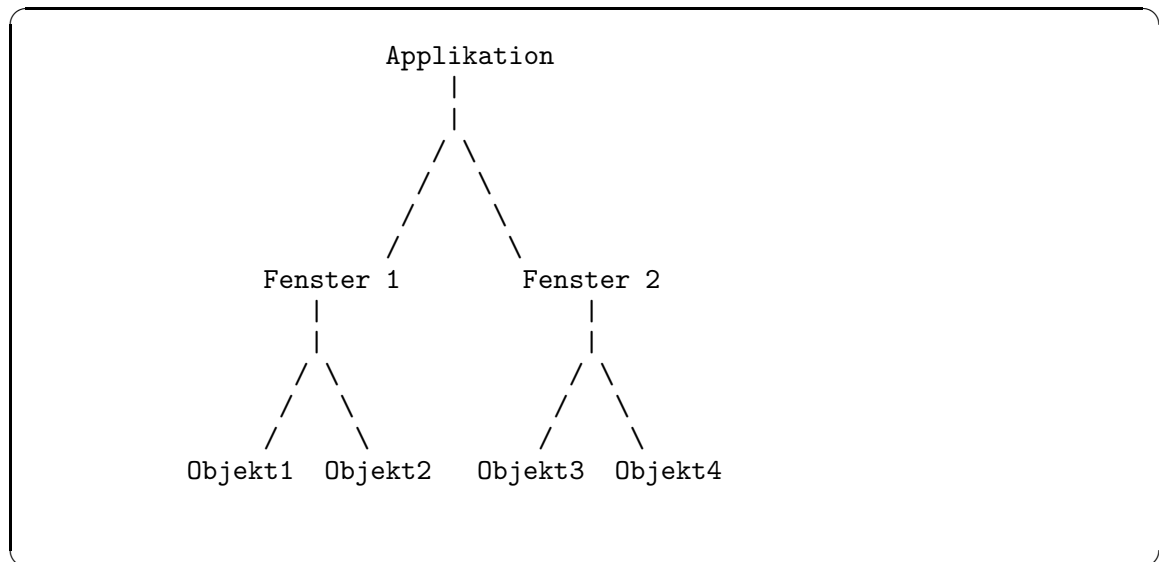
## 2 Grundlegende Prinzipien

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. Das heißt auch alle Fenster eines Programmes können mit MUIBuilder erstellt werden.

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.



## 3 Distribution

Diese Veröffentlichung besteht aus einem gepackten Archiv im LhA Format. Dieses darf unter keinen Umständen verändert werden. Dieses Programm darf auf keinen Fall in einer anderen Form ohne die schriftliche Erlaubnis des Autors verteilt werden. Desweiteren darf dieses Programm frei verteilt und weitergegeben werden, solange nicht mehr als eine realistische Kopiergebühr dafür verlangt wird (soll heißen nicht mehr als 4US-\$, 12FF, 4 DM). In jedem Fall darf mit dem Vertrieb von MUIBuilder kein Profit erzielt werden.

MUIBuilder ist **Giftware**.

Das bedeutet, daß Sie auch wenn Sie dieses Programm weiterhin benutzen, in keiner Weise verpflichtet sind, mir Geld zu schicken. Wenn Sie dieses Programm allerdings wirklich gut finden, dann können Sie mir an folgende Adresse 15\$ oder 50FF senden:

Eric Totel  
26 route de Montsuzain  
10150 Voue  
France  
E-Mail : [Eric.Totel@ramses.fdn.org](mailto:Eric.Totel@ramses.fdn.org)  
Fidonet : 2:320/104.56

Sollte Sie MUIBuilder nicht dermaßen überzeugt haben, daß Sie sich genötigt fühlen, mir Geld zu senden (oder wenn Sie z.B. ein abgebrannter Student sind :-)), dann schicken Sie mir doch bitte eine Postkarte, einen Brief oder eine EMail um mich aufzumuntern, mir Ihre Meinung zum Programm mitzuteilen, Ideen zu geben ... etc ... Einige Leute haben mir Programme geschickt, die sie selbst geschrieben haben : Diese Idee finde ich absolut klasse und möchte Sie dazu weiterhin ermuntern. Außerdem wäre es schön, wenn Sie in Ihrer Dokumentation erwähnen könnten, daß MUIBuilder bei der Erstellung der Oberfläche zum Einsatz kam.

Aber bitte senden Sie keine großen Dateien an die EMail Adresse. Schicken Sie mir einfach eine kleine Nachricht, daß Sie dies tun möchten und ich werde Ihnen mitteilen, wohin Sie diese schicken können.





## 4 Garantie

Dieses Programm wird ohne jegliche Garantiezusage zur Verfügung gestellt. Dieses Programm wird, sofern nicht anderweitig schriftlich versichert, vom Copyrightinhaber und/oder Dritten "so wie es ist" und ohne jegliche Garantie, direkt oder indirekt, zur Verfügung gestellt. Dies betrifft auch, aber nicht nur, die Garantie der Eignung für einen bestimmten Zweck. Das gesamte Risiko bezüglich Qualität und Leistung des Programms liegt bei Ihnen. Sollte es sich herausstellen, daß das Programm fehlerhaft ist, so bleiben die Kosten jeglichen nötigen Services, Reparatur oder Verbesserung bei Ihnen.

Keinesfalls, außer es wird von gültigem Recht verlangt, wird der Copyright Inhaber oder dritte, die das Programm unter obengenannten Bedingungen vertreiben, gegenüber Ihnen für Beschädigungen, inklusive alle allgemeinen, besonderen, zufälligen oder nachfolgenden Beschädigungen, die sich aus der Benutzung oder der nicht möglichen Benutzung des Programms ergeben (betreffend auch, aber nicht nur, Datenverlust, Verstümmelung von Daten oder Verluste von Ihnen oder dritten oder Versagen des Programmes im Zusammenhang mit Programmen von Drittanbietern) Verantwortlich sein, auch wenn er auf die Möglichkeit solcher Beschädigungen hingewiesen wurde.



## 5 Tutorium

Um Anfängern schneller auf die Sprünge zu helfen, wie dieses Programm benutzt wird, habe ich dieses Beispielprojekt mitgeliefert.

Wir werden nun Schritt für Schritt eine kleine Oberfläche bauen, deren Aufgabe sein wird, einen einfachen Text anzuzeigen, wenn ein Schalter angewählt wird.

- Zuerst müssen wir ein Fenster erstellen (**Neues Fenster** Schalter im Hauptfenster).
- Wählen Sie die einzige Gruppe in der Liste mit den Objekten an und klicken Sie den **Neues Kind** Schalter an.
- Wählen Sie Text als Objekt, definieren Sie seine Parameter (im Moment können Sie die Einstellungen übernehmen) und klicken Sie **OK** an.
- Ihr Textobjekt wird nun in Ihren GUI-Baum eingefügt.
- Fügen Sie jetzt eine Gruppe hinzu (erstellen Sie durch Ankreuzen des **Horizontal** Attributes eine horizontale Gruppe).
- Wenn alles glatt geht, sehen Sie den Gruppennamen direkt unter dem Textobjekt: wählen Sie diese Gruppe an und fügen Sie drei Schalter hinzu. Geben Sie ihnen einfach die Kennungen '1st Button', '2nd Button' und '3rd Button' sowie ihre Tastaturkürzel als '1', '2', '3'.
- Ihre Oberfläche ist jetzt fertig ! Jetzt werden wir die Notifications zu diesen Objekten definieren.
- Wählen Sie das erste Schalterobjekt und klicken Sie **Notification** an.
- Wählen Sie im sogleich erscheinenden Fenster das Ereignis **Release Button**, dann das zuvor erstellte Textobjekt und zuletzt die Aktion **Put a constant value**. Fügen Sie diese Notification einfach durch Doppelklick auf den Aktionsnamen oder durch einen Klick auf den **Erzeuge Notification** Schalter hinzu. MUIBuilder wird Sie jetzt nach einer Zeichenkette fragen. Geben Sie hier '1st Button selected' ein. Verlassen Sie, nachdem Sie diese Notification erzeugt haben, das Notification Fenster wieder.
- Wiederholen Sie die vorherige Prozedur für alle erzeugten Schalter mit entsprechenden Texten : '2nd Button selected' und '3rd Button selected'.
- Wenn Sie zum 'Fensterbaukasten' zurückgekehrt sind, klicken Sie **Testen** an und spielen Sie mit den Schaltern in der von Ihnen erzeugten Oberfläche rum.

Das ist natürlich nur eine sehr kleine Oberfläche, aber ich hoffe die Vorgehensweise wird auch bei größeren immer so einfach sein !! Mit der Hilfe von MUIBuilder sollten Sie jetzt auch sehr vertrackte Oberflächen einfach erstellen können ... hoffe ich zumindest !!! :-)



## 6 Verwendung von MUIBuilder

Die zahlreichen Möglichkeiten, die MUIBuilder für die Erstellung einer Applikation bietet, können in folgende Gruppen unterteilt werden.

### 6.1 Voreinstellungen

Das Voreinstellungsfenster bietet eine Vielzahl von Möglichkeiten an, das Programm nach Ihren Wünschen zu konfigurieren. Alle diese Voreinstellungen werden in einer ENV Variablen abgelegt, die bei der Installation von MUIBuilder erzeugt worden ist.

Es gibt folgende Möglichkeiten:

1.

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).

2. Piktogramm : wenn Sie diesen Schalter wählen, werden die Dateien, die MUIBuilder erzeugt mit einem Standardpiktogramm versehen. Dieses befindet sich normalerweise unter dem Namen `def_MUIBuilder.info` im Verzeichnis `ENV:sys/` und wird bei der Installation dorthin kopiert.

3. Aktionen bestätigen : zeigt an, ob Sie jede 'gefährliche' Aktion in einem Requester bestätigen wollen (z.B. vor dem Löschen eines Objektes).

4. Test : Wenn Sie diese Option anwählen werden alle Veränderungen an der Oberfläche direkt bei der Bearbeitung im Preview angezeigt.

5. Quellcode : zeigt an, welchen Quellcodegenerator Sie verwenden wollen.

6. Editor : Geben Sie hier Ihren Lieblingseeditor an! Vergewissern Sie sich, daß der Editor 'synchron' gestartet wird. (d.h. daß er sich, vom CLI gestartet, nicht selbst löst, sondern das Prompt erst nach Beenden des Editors zurückkehrt ! )

Folgende Editoren sollten ohne Probleme laufen:

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

7. 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 anderen als den voreingestellten angeben können). Wird dieser String nicht angegeben, so wird er aus dem Namen der Applikation gebildet ('GetApplikationsnameString').

8. Schachteltiefe : Die minimale Schachteltiefe für die Anzeige der Hierarchie im Eigenschafts Fenster.
9. Schachtelzeichen : Dieses Zeichen wird in die Anzeige eingefügt, um die Schachtelung besser lesbar zu machen

Haben Sie die passenden Einstellungen gewählt, können diese mit dem entsprechenden Schalter abspeichern ! Dabei wird auch Ihr momentan aktuelles Verzeichnis abgespeichert.

## 6.2 Speichern der Applikation

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

Der Asl Requester wird zuerst in Ihrem aktuellen Verzeichnis (Aus der ENV Variablen `MUIBuilder.env`) suchen.

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.

Außerdem werden viele von Ihnen erkannt haben, daß diese Applikationsdateien im ASCII format sind. Sie können also recht einfach mit Hilfe eines Texteditors verändert werden. Aber bitte tun Sie das nicht !! Auf diese Weise könne Sie die Datei sehr leicht unbrauchbar machen ...

## 6.3 Quellcode Erzeugung

Mit dem Quellcode Schalter im Hauptfenster starten Sie die Quellcode Erzeugung.

Zunächst prüft MUIBuilder, ob Sie auch keine Kennung doppelt vergeben haben. Sollte dies der Fall sein, werden Sie gewarnt.

Ansonsten finden Sie sich vor zwei Listen wieder. Die eine enthält alle Objektkennungen, die Sie in der Applikation definiert haben, die andere alle Kennungen, die im Quellcode generiert werden.

Damit können Sie sehr genau bestimmen, welche Kennungen MUIBuilder erzeugen soll, auf die Sie später im Programm auch zurückgreifen können, und welche nicht. Wenn Sie drauf verzichten, erledigt MUIBuilder das alles automatisch für Sie!

MUIBuilder erzeugt eigentlich einen symbolischen Code, der eher eine Art Beschreibung der grafischen Oberfläche ist, als ein 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.

Es sind Module für fast alle aktuellen Sprachen verfügbar.

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. Die muibuilder.library enthält einige Funktionen, die bei der Quellcodeerzeugung recht nützlich sind.

### 6.3.1 Quellcode Erzeugung : Optionen

Fünf Optionen werden Ihnen angeboten :

- **Deklarationen:** Um Deklarationen und Initialisierungen im Quelltext zu bekommen.
- **Environment:** Um Quelltext für Includes, Abfrageschleife, Prozedurdeklaration usw. zu bekommen.
- **Quellcode:** Um den eigentlichen MUI-Code zu erhalten.
- **Notifications:** Um die Notificationen im Quellcode zu erzeugen.
- **Locale:** Um lokalisierten Quellcode zu erhalten (siehe Abschnitt 6.4 [Locale], Seite 14).

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 Quell- text 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.

### 6.3.2 App Code

Mit diesem Schalter können Sie den Quelltext für die gesamte Applikation erzeugen lassen. Bitte beachten Sie, daß die Form des erzeugten Quellcodes natürlich von den oben eingestellten Optionen abhängt.

### 6.3.3 Object Code

Mit diesem Schalter können Sie den Quellcode für das vorher in der 'Objektkennungen' Liste ausgewählte Objekt erzeugen. Das ist speziell dann interessant, wenn Sie ein Objekt in mehreren Fenstern haben wollen oder jedes Fenster einzeln generieren, um dann, während das Programm läuft, das Objekt (und den benötigten Speicher) dynamisch allozieren.

### 6.3.4 Entferne Kennung

Mit diesem Schalter kann die Erzeugung von Quelltext für das in der 'Erzeugte Kennungen' 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.

Mit Hilfe dieses und des 'Erzeuge Kennung' (siehe Abschnitt 6.3.5 [Erzeuge Kennung], Seite 14) Schalters können Sie die Standard Einstellungen von MUIBuilder ändern.

### 6.3.5 Erzeuge Kennung

Mit Hilfe dieses Schalters, weisen Sie MUIBuilder an, das betreffende Objekt zu erzeugen (siehe Abschnitt 6.3.4 [Entferne Kennung], Seite 14).

## 6.4 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.

### 6.4.1 Katalogbeschreibung

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

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

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

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



- Die Übersetzungsdatei ( 'xxxxx.ct' file ) in der alle Strings in der gewünschten Sprache stehen.
- Den Quellcode für ihre bevorzugte Sprache

An alle Anfänger, die damit immer noch nichts anzufangen wissen: Ich empfehle Ihnen dringend, das inzwischen weit verbreitete Programm FlexCat (von Jochen Wiedmann) zu benutzen, das sich an fast alle Sprachen anpasst und eine Beschreibung beinhaltet, die das Verständnis der Lokalisierung recht einfach macht.

## 6.5 Erzeugung der Online Hilfe

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
- Der Text für ein Fenster mit dem **Fenster Node** Schalter
- Der Text für ein Object mit dem **Objekt Node** Schalter

Die Dokumentatin kann auf folgende Weise erzeugt werden:

- Für die ganze Application mit dem **Erzeuge ganzes Dokument** Schalter
- Für ein Fenster mit dem **Erzeuge Fenster Dokument** Schalter
- Für ein Object mit dem **Erzeuge Objekt Dokument** Schalter

Mit einem Doppelklick auf einen Objekt oder Fensternamen kann der Titel des zugehörigen Hilfetextes sowie der Text selbst angesehen und verändert werden. ( Wie mit den **Ändern** Schalter ).

Bei der Quelltexterstellung erzeugt MUIBuilder ein `MUIA_Help_Node` Attribut zu jedem Objekt, dessen Hilfetext nicht leer ist und das in der 'Erzeugte Kennungen' Liste auftaucht. Die erzeugte Datei ist allerdings als Online Hilfe ...und nicht als Dokumentation ... für die Applikation gedacht.

## 6.6 Verschiedene Anmerkungen

In diesem Teil der Anleitung finden Sie einen Haufen (hoffentlich) nützlicher Anmerkungen, die an anderer Stelle noch nicht erwähnt wurden.

### 6.6.1 Besondere Zeichen

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

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

Einige Beispiele :

<code>\033b</code>	für fettgedruckten Text
<code>\033n</code>	um wieder auf normalen Text zurückzuschalten
<code>\0338</code>	für weißen Text
<code>\033c</code>	um den Text zu zentrieren
<code>\033l</code>	für linksbündigen Text
<code>\033r</code>	für rechtsbündigen Text

Achtung: um Anführungszeichen zu erhalten muß '\"' anstatt '"' eingegeben werden, sonst gibt es Fehler beim Übersetzen des Programms. Außerdem ist '\e' das gleiche, wie '\033', nämlich der ESCAPE-Code. Ein häufig gemachter Fehler ist außerdem, '\33' anstatt '\033' zu tippen ...

### 6.6.2 Objektausrichtung

Um Objekte in Reih und Glied auszurichten genügt es, sie in eine Gruppe mit mehreren Zeilen (horizontal) oder Spalten (vertikal) zu packen.

Für einige Objekte (Slider, Häkchen, Stringgadget) schlägt MUIBuilder einen Titel vor. Da diese Objekte auf MUI-Ebene jedoch eigentlich keinen Titel haben, erzeugt MUIBuilder für sie

eine extra Gruppe (horizontal), die den Titel und das betreffende Objekt beinhaltet. Wenn sie das allerdings auf diese Weise verwenden, werden Sie kaum mehrere solcher Objekte in einer Gruppe ausrichten können. Erstellen Sie stattdessen eine Gruppe mit mehreren Spalten / Zeilen und verwenden Sie extra Bezeichner und Objekte ohne Titel in benachbarten Zeilen / Spalten.

### 6.6.3 Größenveränderbare Fenster

Eine der häufig gestellten Fragen ist : "Nachdem ich einige Objekte in mein Fenster eingefügt habe, kann ich seine größe nicht mehr verändern." Kurze Antwort dazu: Verwenden Sie **Zwischenraum** Objekte und fügen Sie sie in Ihre Oberfläche ein!

Hierzu ein Beispiel: Sie haben ein Häkchen und seinen Bezeichner in die Oberfläche eingefügt: Das Häkchen ist in seiner horizontalen Größe nicht veränderbar, genau wie sein Bezeichner. In diesem Fall würden Sie das Häkchen, mit zwei Zwischenräumen vorne und hinten, in eine eigene Gruppe stecken. Das resultierende 'Objekt' ist nun in seiner horizontalen Größe veränderbar, im Gegensatz zum Häkchen selbst!



## 7 Objekte

Mit MUIBuilder könne Sie alle MUI Objekte erzeugen, die Sie in ihrer Applikation haben wollen. Diese können im 'Fensterbaukasten' erzeugt oder verändert werden.

Übrigens genügt es, ein schon bestehendes Objekt doppelt anzuklicken, um es verändern zu können.

### 7.1 Area Objekt

Jedes MUI-Objekt, das Eigenschaften von dem Area Objekt erbt, hat in dem Fenster für die Einstellungen eine Register Gruppe (Ab MUI 2.0). Die zweite Seite (Area) enthält die Attribute dieser Klasse, die bei der Initialisierung des Objektes beeinflusst werden können.

- **Verstecken** : Das Objekt erscheint nicht auf der Oberfläche, es sei denn, es wäre das einzige im Fenster.
- **Disabled** : Das Objekt erscheint 'ghosted' und nimmt keine Benutzereingaben an.
- **Eingabe möglich** : Bietet die Möglichkeit, ein Objekt als Schalter zu verwenden.
- **ohne Rahmen** : Es erscheint kein Rahmen.
- **Hintergrund** : Welchen Hintergrund hätten Sie gerne?
- **Rahmen** : Welcher Rahmen darf es sein?
- **Tastaturkürzel** : Tastaturkürzel für das Objekt.
- **Titel** : Ein Titel, der oben am Rahmen des Objektes erscheint.

Bei einigen Objekten kann man allerdings nicht alle Attribute beeinflussen. Das ist kein Fehler sondern ein Feature !! Auf diese Weise will ich verhindern, daß der User irgendwelchen Unfug macht, wie z. B. einen Schalter mit einem String - Rahmen. Diese ist meine persönliche Meinung; daher waren einige Beta Tester damit nicht so ganz zufrieden, während andere mehr solcher Restriktionen forderten. Ich habe versucht, das Beste daraus zu machen : Jede Anmerkung ist natürlich willkommen !

### 7.2 Applikation

Ihre Applikation ist der Wurzeleintrag im Baum, der die komplette grafische Oberfläche darstellt. Der 'Applikation' Schalter öffnet ein Fenster mit folgenden Einstellmöglichkeiten:

- Die 'Base' der Applikation d.h. den Namen des AREXX Portes der Applikation
- Den Namen des Autors
- Den Namen der Applikation
- Die Version
- Das Copyright
- Eine kurze Beschreibung der Applikation

Außerdem befinden sich hier :

- Ein 'Menü' Schalter, der den Menübaukasten öffnet, mit dem Sie Menüs (siehe Abschnitt 7.23 [Menüs], Seite 29) basteln können. Dieses Menü kann nicht von MUIBuilder aus getestet werden, da es an eine Applikation gebunden werden muß und aus einer Applikation heraus keine neue erstellt werden kann.
- Die Möglichkeit, Notifications (siehe Kapitel 8 [Notification], Seite 31) an die Applikation zu binden.

Die Applikation kann nur eine sorte Kinder haben: das Fenster (siehe Abschnitt 7.3 [Fenster], Seite 20)

## 7.3 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 'Design', eine andere namens 'Eigenschaften' :

1. Die **Design** Page enthält 2 Listen:

In der ersten wird die graphische Oberfläche selbst aufgebaut. Diese wird in der Liste als hierarchischer Baum angezeigt. Um einen Zweig des Baumes zu falten oder zu entfalten, muß nur den kleinen Pfeil auf der linken Seite doppelt anklicken. Mit Hilfe der Schalter zwischen den beiden Listen können die Objekte der Oberfläche geändert/kopiert/verschoben werden. Außerdem gibt es zwei Menüpunkte **Falten** und **Entfalten**, mit denen der ganze Baum ge- oder entfaltet werden kann. In den zwei äußersten linken Spalten können zwei Buchstaben auftauchen :H (für Hilfe und G (für erzeugt). Mit ihnen kann direkt aus dem Fensterbaukasten bestimmt werden, ob dieses Objekt oder Onlinehilfe dazu erzeugt werden soll oder nicht. Dazu doppelklicken Sie einfach auf den entsprechenden Buchstaben (oder dort, wo er erschienen sollte).

- Die rechts Liste dient dazu, Objekte zeitweilig aufzunehmen, damit sie vervielfältigt oder verschoben werden können. (siehe Abschnitt 7.3.1 [Temporäre Liste], Seite 21).

Gruppen (siehe Abschnitt 7.4 [Gruppe], Seite 21) sind die grundlegenden Objekte, in denen alle anderen Objekte (siehe Kapitel 7 [Objekte], Seite 19) ihren Platz finden. Jedesmal wenn Sie ein Objekt erstellen, muß es als Kind einer Gruppe deklariert werden. Diese Gruppe kann durch einfaches Anwählen in der Liste spezifiziert werden.

2. In der **Eigenschaften** Page befindet sich folgendes :

- Fensterkennung und titel.
- Die eigentlichen Fensterattribute :
  - **AppWindow**: Soll's ein AppWindow sein?
  - **ohne Rand**: Mit oder ohne Rand?
  - **Depth gadget**: Mit oder ohne Depth Gadget?

- **Größensymbol:** Wollen Sie ein Größensymbol?
- **Immer Hinten:** Das Fenster hält sich schamhaft im Hintegrund
- **Schließsymbol:** Heute geschlossen.
- **Ziehleiste:** Zum rumziehen des Fensters
- **zu Beginn öffnen:** (WICHTIG) Zeigt an, ob Ihr Fenster beim späteren starten des Programms und dem Erstellen der Applikationsoberfläche geöffnet werden soll. Außerdem wird dieses Attribut überprüft, wenn die Oberfläche aus MUIBuilder heraus getestet wird, d.h. das Fenster wird gegebenenfalls geöffnet. Ist unter 'Einstellungen, Optionen' 'Testen' aktiviert, so wird das Fenster auf alle Fälle beim Erstellen zur Ansicht geöffnet.

### 7.3.1 Temporäre Liste

In dieser Liste werden Objekte plaziert, die von einer Gruppe in eine andere oder von einem Fenster in ein anderes verschoben werden sollen. Der Listeninhalt wird nicht gelöscht, wenn die Applikation gelöscht wird, oder eine neue geladen wird. Auf diese Weise können Objekte zwischen verschiedenen Applikationen ausgetauscht werden.

## 7.4 Gruppe

Wenn ein neues Fenster erstellt wird, hat es bereits eine Gruppe: die Wurzelgruppe. Die Oberfläche wird nun durch Zufügen von Kindern zu dieser Gruppe aufgebaut. Diese Kinder sind sozusagen die Knoten des Applikationsbaumes. Die Kinder einer Gruppe können natürlich selbst auch wieder Gruppen mit eigenen Kindern sein, die wiederum ...

Für jede Gruppe können folgende Attribute definiert werden :

- **Horizontal:** Die Kinder dieser Gruppe werden horizontal aufgereiht
- **Register:** Es wird jeweils nur ein Kind der Gruppe gezeigt. Dazu müssen noch in der Page 'Register' die Namen dieser Kinder (der Registerpages) definiert werden.
- **Gleiche Höhe:** Alle Kinder der Gruppen werden die gleiche Höhe haben.
- **Gleiche Breite:** Alle Kinder der Gruppen werden die gleiche Breite haben.
- **Gleiche Größe:** Alle Kinder der Gruppen werden die gleiche Größe haben.
- **Virtuelle Gruppe:** Es wird eine virtuelle Gruppe erzeugt.
- **Spalten:** Formatiert die Gruppe in Spalten, deren Anzahl rechts angegeben werden muß.
- **Zeilen:** Formatiert die Gruppe in Zeilen, deren Anzahl auch rechts angegeben werden muß.
- **Abstand**
  - Horizontal : ermöglicht die Kontrolle über den horizontalen Abstand zwischen Objekten.
  - Vertikal : ermöglicht die Kontrolle über den vertikalen Abstand zwischen Objekten.

Dieses Objekt erbt Attribute vom Area Objekt (siehe Abschnitt 7.1 [Area Objekt], Seite 19).

## 7.5 Liste

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

- Ihre Kennung
- Ob man in dieser Liste 'Doppelklicken' darf
- Ob Mehrfachauswahl unterstützt werden soll
- Ob die Eingabe von Listenelementen möglich ist
- Ob bei der Erstellung die Höhe angepasst werden soll
- Ob bei der Erstellung die Breite angepasst werden soll
- Was der Benutzer am Anfang sehen soll (Aktiviere):
  - Keinen Eintrag
  - Den ersten Eintrag
  - Den letzten Eintrag
- Die Art der Liste
  - Eine Standard Liste
  - Eine Floattext Liste, die Texte anzeigt. Dieser kann im dazugehörigen Stringgadget `Float Text`) eingegeben werden.
  - Eine Liste von Datenträgern ( `Volumes + Assigns` )
- Die Funktions Hooks (eigentlich eine einer Funktion zugeordnete Struktur, die als Verweis auf diese Funktion verwendet wird) können ebenso angegeben werden, aber diese braucht man nicht unbedingt, um eine funktionsfähige Liste zu haben. (siehe auch MUI-Autodocs!)
  - Construct Hook : wird beim Einfügen eines Objektes aufgerufen
  - Destruct Hook : wird beim Löschen eines Objektes aufgerufen
  - Compare Hook : wird beim Sortieren der Liste aufgerufen, um zwei Einträge miteinander zu vergleichen
  - Display Hook : wird aufgerufen, um ein Objekt anzuzeigen
  - Multitest Hook : wird im Zusammenhang mit Mehrfachselektion aufgerufen
- Das Format der Liste (Für Listen mit mehreren Spalten). Dieses Attribut kann nicht direkt innerhalb von MUIBuilder getestet werden.
- Den Titel der Liste.

Beachten Sie, daß es noch eine andere Art Liste gibt : die Verzeichnisliste (siehe Abschnitt 7.6 [Verzeichnisliste], Seite 23). Diese kann mittels des gleichnamigen Schalters im 'Objekte' Fenster erzeugt werden.

Dieses Objekt erbt Attribute vom Area Objekt (siehe Abschnitt 7.1 [Area Objekt], Seite 19).

## 7.6 Verzeichnisliste

Die Verzeichnisliste zeigt den die Dateien und Verzeichnisse im angegebenen Verzeichnis. Folgende Optionen können angegeben werden :



- **Nur Verzeichnisse:** zeigt NUR Verzeichnisse
- **Nur Dateien:** zeigt NUR Dateien
- **Filter Verzeichnisse:** sollen auch Verzeichnisse nach 'Akzeptiert' und 'Nicht Akzeptiert' gefiltert werden?
- **Mehrfachauswahl:** es können mehrere Dateien zugleich selektiert werden
- **Unterdrücke Icons:** zeigt keine '.info' an
- **Sort High-Low:** sortiert die Einträge absteigend
- **Sortieren:** wonach sortiert werden soll (Name, Datum, Größe)
- **Reihenfolge :** Drei Möglichkeiten :
  - Verzeichnisse erscheinen am Anfang der Liste
  - Verzeichnisse erscheinen am Ende der Liste
  - Verzeichnisse und Dateien werden gemischt (und nach **Sortieren** sortiert)
- Das anzuzeigende Verzeichnis
- Muster, wonach Dateinamen akzeptiert (angezeigt) werden
- Muster, wonach Dateinamen nicht akzeptiert (nicht angezeigt) werden
- Die Kennung

Dieses Objekt erbt Attribute vom Area Objekt (siehe Abschnitt 7.1 [Area Objekt], Seite 19).

## 7.7 Stringgadget

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

- Sein Titel (den vorgeschlagenen können Sie durch einen eigenen ersetzen)
- 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

Dieses Objekt erbt Attribute vom Area Objekt (siehe Abschnitt 7.1 [Area Objekt], Seite 19).

## 7.8 Schalter

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

- Seine Kennung
- Den Text, den er enthalten soll

Dieses Objekt erbt Attribute vom Area Objekt (siehe Abschnitt 7.1 [Area Objekt], Seite 19).

## 7.9 Bezeichner

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

- seine Kennung
- der Text, der auf dem Schirm erscheinen soll

Dieses Objekt erbt Attribute vom Area Objekt (siehe Abschnitt 7.1 [Area Objekt], Seite 19).

## 7.10 Cyclegadget

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

- Die Liste seiner Einträge
- seine Kennung

Wenn Sie auch ein Tastaturkürzel definieren, fügen Sie bitte einen Bezeichner mit dem selbem Tastaturkürzel hinzu, damit der Benutzer auch sehen kann, welches Tastaturkürzel Sie definiert haben.

Dieses Objekt erbt Attribute vom Area Objekt (siehe Abschnitt 7.1 [Area Objekt], Seite 19).

## 7.11 Radio Buttons

Um einen Radio Button vollständig zu definieren, muß folgendes angegeben werden:

- Die Einträge : Benutzen Sie die Schalter am rechten Rand der Liste, um Einträge der Liste hinzuzufügen, zu löschen oder umzubenennen.
- Seine Kennung

Dieses Objekt erbt Attribute vom Area Objekt (siehe Abschnitt 7.1 [Area Objekt], Seite 19).

Auch hier (wie bei Stringgadgets und Slidern) können Sie auswählen, ob Sie einen Titel wollen oder nicht.

Wenn ja, geben Sie ihn bitte im Feld 'Titel' an.

Außerdem müssen Sie auch hier eine Kennung angeben.

Dieses Objekt erbt Attribute vom Area Objekt (siehe Abschnitt 7.1 [Area Objekt], Seite 19).

## 7.12 Bild

Um einem Schalter ein Bild zuzuordnen, müssen Sie zuerst einmal ein solches anklicken.

Außerdem haben Sie folgende Möglichkeiten:

- **Vert. frei:** Das Bild passt automatisch seine vertikale Größe an.

- **Horiz. frei:** Das Bild passt automatisch seine horizontale Größe an.
- **Eingabe möglich:** Der Benutzer kann das Bild anwählen.
- **Feste Höhe:** Setzt die Höhe auf den im zugehörigen Stringgadget anzugebenden Wert fest; dabei wird **Horiz. frei** ignoriert.
- **Feste Breite:** Setzt die Breite auf den im zugehörigen Stringgadget anzugebenden Wert fest; dabei wird **Vert. frei** ignoriert.
- Auch hier müssen Sie eine Kennung angeben.

Dieses Objekt erbt Attribute vom Area Objekt (siehe Abschnitt 7.1 [Area Objekt], Seite 19).

### 7.13 Slider

Bei einem Slider können Sie folgendes angeben:

- Ob der momentane Zustand angezeigt werden soll oder nicht ('Keine Meldung')
- Ob der Nullpunkt unten bzw. rechts ist ('Umgekehrt')
- Ob der Slider einen Title haben soll

Folgendes muß angegeben werden:

- Der Maximale Wert
- Der Minimale Wert
- Der Anfangswert

Außerdem kann angegeben werden:

- Ein Titel, wenn vorhanden
- ... und seine Kennung !

Dieses Objekt erbt Attribute vom Area Objekt (siehe Abschnitt 7.1 [Area Objekt], Seite 19).

### 7.14 Gauge

Um eine Gauge zu definieren, muß folgendes angegeben werden:

- Ihre Ausrichtung (horizontal oder vertikal)
- Wenn die Höhe der Gauge auf einen festen Wert gestellt werden soll, so muß dieser auch im zugehörigen Stringgadget angegeben werden. Dieses bietet sich besonders bei horizontalen Gauges an.
- Wenn die Breite der Gauge auf einen festen Wert gestellt werden soll, so muß dieser auch im zugehörigen Stringgadget angegeben werden. Dieses bietet sich besonders bei vertikalen Gauges an.
- Wenn sein Wert vor der Anzeige Unterteilt werden soll, einen von NULL verschiedenen Teiler

- Ihren maximalen Wert
- Den Infotext : Dieser wird innerhalb des Balkens angezeigt. Um z.B. den momentanen Zustand anzuzeigen geben Sie hier einfach `%ld` an.
- Die Kennung

Dieses Objekt erbt Attribute vom Area Objekt (siehe Abschnitt 7.1 [Area Objekt], Seite 19).

## 7.15 Text

Um ein Textobjekt zu definieren, muß folgendes angegeben werden :

- **Text SetMax:** die maximale Größe wird als anfängliche Größe verwendet.
- **Text SetMin:** die minimale Größe wird als anfängliche Größe verwendet.
- **Label :** Seine Kennung.

Der Text muß im zugehörigen Stringgadget angegeben werden und kann alle Besonderen Zeichen (siehe Abschnitt 6.6.1 [Besondere Zeichen], Seite 16) enthalten, die in C Quellcode erlaubt sind.

Dieses Objekt erbt Attribute vom Area Objekt (siehe Abschnitt 7.1 [Area Objekt], Seite 19).

## 7.16 Skala

Eine Skala ist dazu gedacht, in Verbindung mit einer Gauge deren Inhalt zu Unterteilen und so besser lesbar zu machen.

Hier können Sie nur die Ausrichtung des Objektes angeben.

Beachten Sie, daß eine Skala, Aufgrund eines Fehlers in MUI, im Moment nur horizontal angezeigt werden kann.

Dieses Objekt erbt Attribute vom Area Objekt (siehe Abschnitt 7.1 [Area Objekt], Seite 19).

## 7.17 Prop Gadget

Geben Sie folgende Eigenschaften an:

- **Horizontal:** Das Prop Gadget soll horizontal sein.
- **Feste Breite:** Setzt die Breite auf den im zugehörigen Stringgadget angegebenen Wert.
- **Feste Höhe:** Setzt die Höhe auf den im zugehörigen Stringgadget angegebenen Wert.

Außerdem müssen Sie folgendes angeben:

- Die Anzahl der Schritte
- Die Nummer des ersten Einrages

- Die Anzahl der sichtbaren Einträge

Ach übrigens: Vergessen Sie die Kennung nicht !

Dieses Objekt erbt Attribute vom Area Objekt (siehe Abschnitt 7.1 [Area Objekt], Seite 19).

## 7.18 Zwischenraum

Mittels dieses Objektes können Sie Leerräume zwischen zwei andere Objekte einfügen, damit ein Fenster in der Größe verändert werden kann. Dies kann auf drei verschiedene Arten geschehen:

- Horizontal
- Vertikal
- Horizontal und vertikal zugleich

Zu den ersten beiden kann außerdem die Größe des Zwischenraumes angegeben werden.

## 7.19 PopAsl

Mittels dieses Objektes könne Sie standard ASL requester in Ihre Applikation einbinden. Nutzen Sie diese Möglichkeit reichlich!

Folgende Eigenschaften können angegeben werden:

- Welches Bild der Schalter enthalten soll.
- Die Art des Requester, nämlich eine der folgenden :
  - File Requester
  - Font Requester
  - Screen Mode Requester
- Außerdem können einige Hooks gesetzt werden :
  - Start Hook : Wird beim Öffnen des Requesters aufgerufen
  - Stop Hook : Wird beim Schließen des Requesters aufgerufen
- Die Kennung des Requesters

Dieses Objekt erbt Attribute vom Area Objekt (siehe Abschnitt 7.1 [Area Objekt], Seite 19).

## 7.20 PopObjekt

Mittels eines Popobjektes kann man an ein Stringgadget und einen Schalter ein Objekt binden, das erst erscheint, wenn der Schalter betätigt wird. Dieses Objekt kann ein beliebiges, mit MUIBuilder entworfenes Objekt sein (z.B. auch ein weiteres PopObjekt!) Dieses Objekt erscheint in der Baumstruktur als Kind des PopObjektes und kann wie jedes andere Objekt

auch verändert werden. Wenn es sich hierbei im Besonderen um eine Gruppe handelt, können ihr natürlich auch weitere Kinder hinzugefügt werden.

Folgendes steht zur Definition dieses Objektes zur Verfügung:

- Eine List mit den möglichen Bildern
- Einige Eigenschaften :
  - **Position relativ** : Das Pop Objekt wird relativ zum Fenster geöffnet.
  - **Ohne Rahmen** : Das Pop Object wird in einem Fenster ohne Rahmen angezeigt.
  - **Wegklickbar** : Wird der Schalter ein zweites mal betätigt, verschwindet das Pop Objekt.
- Folgende Hooks können angegeben werden :
  - **Open Hook** : Wird beim Öffnen des Pop Fensters aufgerufen
  - **Close Hook** : Wird beim Schließen des Pop Fensters aufgerufen
- Die Kennung des Objektes

Dieses Objekt erbt Attribute vom Area Objekt (siehe Abschnitt 7.1 [Area Objekt], Seite 19).

## 7.21 Rechteck

Dieses Objekt ist relativ kompliziert, da man hier eigentlich mehrere verschiedene Objekte erzeugen kann. Dies sind folgende, mittels des Cycle Gadgets auszuwählende:

- Ein rechteckiger Bereich innerhalb eines Fensters
- Ein horizontaler Balken, mit oder ohne Titel, um Objekte voneinander abzutrennen
- Ein vertikaler Balken, um Objekte voneinander abzutrennen

Ebenso kann die Höhe und Breite dieser Objekt angegeben werden.

Dieses Objekt erbt Attribute vom Area Objekt (siehe Abschnitt 7.1 [Area Objekt], Seite 19).

## 7.22 Farbfeld

Dieses einfache Objekt dient dazu, eine Farbe auf dem Bildschirm zu zeigen.

Geben Sie einfach folgendes an:

- Ob die Höhe fest sein soll
- Ob die Breite fest sein soll
- Die Farbe des Feldes

Dieses Objekt erbt Attribute vom Area Objekt (siehe Abschnitt 7.1 [Area Objekt], Seite 19).

## 7.23 Menüs

Ein Menü kann entweder an ein Fenster oder an die Applikation selbst gehängt werden.

Es besteht aus der Menüleiste mit Menütiteln, Menüeinträgen und Untermenüs. Im Menübaukasten wird eine Menüstruktur genau wie im Fensterbaukasten durch eine Baumstruktur dargestellt. Der einzige Unterschied ist, daß die Tiefe der Baumstruktur auf zwei beschränkt ist, da ein Untermenü kein weiteres Untermenü haben kann.

Moreover:

- Die Titel, Einträge und Untermenüs können mittels des Häkchens **Enable** aktiviert werden
- Für Menüeinträge können außerdem folgende Eigenschaften eingestellt werden:
  - Er kann im selektierten Status 'Abgehakt' werden
  - Ist die Option 'Häkchen' gewählt, kann der anfängliche Status vorgegeben werden
  - Ein Tastaturkürzel kann definiert werden





## 8 Notification

Die Ereignisnotification ist meiner Meinung nach die größte und wichtigste Neuerung in MUIBuilder 2.00. Außerdem ist es wohl das erstmal, daß ein Programm zur Erstellung von GUIs auf dem AMIGA diese Möglichkeit bietet.

Das Ziel ist es eine Verbindung zwischen, zum einen, eintretenden Ereignissen und, zum anderen, auszuführenden Aktionen, zu definieren. Die Ereignisse hängen mit MUI-Objekten zusammen: Sie müssen also für ein gegebenes Objekt das Ereignis definieren, das eine auszuwählende Aktion zur Folge haben soll.

Wenn solch eine Verbindung zwischen einem Ereignis und einer Aktion definiert ist, kann sie direkt aus MUIBuilder heraus getestet werden. Dazu klickt man einfach auf den **Test** Schalter im Fensterbaukasten. Jede Aktion, die nicht direkt das Applikations Objekt oder eine später im Quellcode definierte Variable beeinflusst, kann dann direkt im Fenster getestet werden.

Das Notification Fenster bietet folgende Optionen :

- Eine Liste der Ereignisse, die von dem im oberen Teil des Fensters angezeigten Objekt erzeugt werden können.
- Eine Liste mit Objekten, die durch eine Aktion beeinflusst werden können. Der Übersicht halber sind diese ihren Fenstern nach geordnet. Zuerst erscheint das Fenster in dem sich das gerade editierte Objekt befindet, dann kommt das Applikations Objekt (siehe unten) und schließlich die übrigen Fenster der Applikation.
- Sobald ein Objekt ausgewählt ist, erscheinen in der dritten Liste die Aktionen, die auf das gewählte Objekt angewendet werden können. Einige dieser Aktionen verlangen zusätzliche Parameter, deren Eingabe MUIBuilder gegebenenfalls verlangt.

Um eine Notification zu dem aktuellen Objekt hinzuzufügen, doppelklicken Sie entweder auf ein Element der Aktionsliste oder betätigen den Schalter **Erzeuge Notification**.

Das Applikations Objekt bietet alle Aktionen, die mit ihrem Programm in Verbindung stehen. Zum Beispiel Aufruf von Funktionen, ändern von Variablen

Wenn Sie eine Notification bestätigen, kann es sein, daß zusätzliche Parameter angegeben werden müssen. Dazu öffnet sich ein Fenster.

Dies können folgende Parameter sein :

Objekten kann auf folgende zwei Weisen eine Notification hinzugefügt werden :

- Für ein Fenster : Durch den **Notification** Schalter in der **Eigenschaften** page
- Für ein Objekt: Durch auswählen des Objektes in der Objekte Liste und anklicken des **Notification** Schalters darunter.

## 8.1 Funktionen

Wenn die gewählte Aktion eine **Funktion** ist, müssen Sie den Funktionsnamen angeben. In dem erscheinenden Fenster werden alle vorhandenen Funktionen angezeigt, von denen Sie entweder eine auswählen können oder einfach eine neue hinzufügen. Bei der Quellcodeerzeugung wird MUIBuilder alles erzeugen, was nötig ist, um einen Bezug zu dieser Funktion herzustellen.

## 8.2 Variablen

Einige Aktionen brauchen als Parameter eine Variable. Um diesen Variablennamen zu bekommen, öffnet MUIBuilder ein Fenster, in dem alle vorhandenen Variablen aufgelistet werden. Davon können Sie entweder eine auswählen, oder eine neue definieren.

## 8.3 Konstanten

Eine Konstante kann folgenden Typ haben:

- Ein Integer (Zahl)
- Eine Zeichenkette
- Ein einzelnes Zeichen
- Ein boolescher Wert (TRUE oder FALSE)

Je nach Art der Konstanten, kann man in eines der Gadgets den entsprechenden Wert eingeben.

## 9 Zukünftige Erweiterungen

Es bleiben noch ein Haufen Funktionen übrig, die dieses Programm zu einem der Besten seiner Art machen sollen. Einige sind bereits in dieser Version implementiert. Aber dummerweise habe ich nicht genug Zeit dazu gehabt und da ich nicht weiß, wieviel Zeit ich dazu in den nächsten Monaten haben werde, habe ich jetzt diese Version veröffentlicht.

Für nächste Versionen angestrebte Erweiterungen:

- Verbessern von einigen Objekten (z.B. PopAsl, Zwischenraum)
- Einige Objekte fehlen noch (z.B. Farbeinsteller)
- Arexx-Port mit Zugriffsmöglichkeit auf interne MUIBuilder Funktionen
- Möglicherweise Unterstützung von MUI Custom Objekten, wenn niemand vorher etwas ähnliches programmiert
- Verbesserungen bei den Notifications, um mit dem selben, momentanen GUI noch komplexere Abhängigkeiten definieren zu können.
- Verbessern der Quellcode Erzeuger : verschiedene Optionen mittels eines GUI zugänglich machen
- Lokale und globale Voreinstellungen für die Quellcode Erzeuger
- Ein Interface zwischen Texteditoren und MUIBuilder scheint eine gute Idee zu sein
- Sobald MUI Drag & Drop unterstützt, wird es bei der GUI Erstellung eingesetzt
- Die Objekt Schalter werden vielleicht Bilder haben
- Das MUIBuilder Dateiformat wird dahingehend verändert, daß man Dateien hinterher leichter editieren kann
- Verbesserung bei der Katalogerzeugung : per Hand hinzugefügte Strings werden nicht mehr gelöscht

Offensichtlich habe ich also noch einige Monate Arbeit vor mir, bevor ich mich einem neuen Projekt widmen kann. Außerdem bin ich gerne bereit, andere Vorschläge in Erwägung zu ziehen, sofern man mich nur genug bittet :-)



## Danksagungen

Ich möchte den folgenden Personen für ihre Hilfe und Unterstützung bei der Erstellung von MUIBuilder danken (ohne besondere Reihenfolge) :

- **Lionel Vintenat** für die Idee des symbolischen Codes und die Implementation des Moduls für die 'Sprache E'.
- **Albert Weinert** für die Implementation des OberonII Moduls.
- **Michal Kara** für die Übersetzung des Katalogs ins Tschechische und das Assembler Modul für MUIBuilder 1.1 (in MUI 2.2 leider nicht mehr verfügbar).
- **Bernd Noll** für seine große Hilfe beim Debuggen von MUIBuilder, und seinen riesigen Bug-Reports !!
- **Andreas Jung** für seine mehr als häufigen Bug-reports und seine unschätzbaren Vorschläge!
- **Pierre Carrette** für seine große Hilfe und seine Ideen, eine interessanter als die andere ! :-)
- **Gael Marziou, Pascal Pensa, Christian Brandel, Daniel Murrel** für ihre sorgfältigen Tests und unzähligen Vorschläge.
- **Christian Brandel** für die deutsche Anleitung und den deutschen Katalog.
- **Pascal Rabier, Mike Manzano, Daniel Murrell** für ihre Hilfe und Korrekturen bei der englischen Anleitung.
- **Tobias Ferber** für sein tolles MagicWB icon !!! ( nein, nein ... nicht toll : FANTASTISCH !!! :-) )
- **Marco Musso** für den italienischen Katalog.
- **Marcin Orlowski** (Mr Soft/W.F.M.H.) für den polnischen Katalog.
- **Stanislav Kneifl** für die tschechische Übersetzung.

Ich muß allen danken, die mir Fehlerberichte und Vorschläge geschickt haben und auf die eine oder andere Weise zum Programm beigetragen haben.

**Danke Euch allen!**



# Index

## A

Aktionen bestätigen (Voreinstellungen) .....	11
AmigaGuide .....	15
Anmerkungen .....	16
Applikation .....	19
Applikation (Quellcode) .....	13
Applikation Speichern .....	12
AppWindow .....	12
Area Objekt .....	19
Autor .....	5

## B

Besondere Zeichen .....	16
Bezeichner .....	24
Bild .....	25

## C

Catcomp .....	14
Cyclegadget .....	24

## D

Danksagungen .....	35
Deklarationen (Quellcode Optionen) .....	13
Distribution .....	5

## E

Einführung .....	1
Entferne Kennung .....	14
Environment (Quellcode Optionen) .....	13
Ereignisse .....	31
Erzeuge Kennung .....	14
Externe Module .....	13

## F

Farbfeld .....	29
Fenster .....	20
Fenster (Voreinstellungen) .....	11
Flexcat .....	14, 15
Funktionen (Notification) .....	32

## G

Garantie .....	7
Gauge .....	26

Größenveränderbare Fenster .....	17
Grundlegende Prinzipien .....	3
Gruppe .....	21

## H

Häkchen .....	24
Hilfedatei .....	15

## K

Katalog .....	14
Katalogbeschreibung .....	14
Konstanten (Notification) .....	32

## L

Liste .....	22
Locale (Quellcode Optionen) .....	13
Locale (Voreinstellungen) .....	11
Lokalisation .....	14

## M

Menüs .....	29
Module .....	13

## N

Notification .....	31
Notification Funktionen .....	32
Notification Konstanten .....	32
Notification Variablen .....	32
Notifications (Quellcode Optionen) .....	13

## O

Objekt (Quellcode) .....	13
Objekt Applikation .....	19
Objekt Area .....	19
Objekt Bezeichner .....	24
Objekt Bild .....	25
Objekt Cyclegadget .....	24
Objekt Farbfeld .....	29
Objekt Fenster .....	20
Objekt Gauge .....	26
Objekt Gruppe .....	21
Objekt Häkchen .....	24
Objekt Liste .....	22

Objekt Menüs .....	29
Objekt PopAsl.....	27
Objekt PopObjekt.....	28
Objekt Prop Gadget .....	27
Objekt Radio Buttons.....	24
Objekt Rechteck .....	28
Objekt Schalter .....	24
Objekt Skala.....	26
Objekt Slider .....	25
Objekt Stringgadget .....	23
Objekt Text .....	26
Objekt Verzeichnisliste.....	23
Objekt Zwischenraum.....	27
Objektausrichtung .....	17
Objekte.....	19
Online Hilfe.....	15

## P

Piktogramm (Voreinstellungen).....	11
PopAsl.....	27
PopObjekt.....	28
Prinzipien, grundlegende .....	3
Prop Gadget.....	27

## Q

Quellcode .....	12
Quellcode (Applikation).....	13
Quellcode (Externe Module).....	13
Quellcode (Objekt).....	13
Quellcode (Quellcode Optionen).....	13
Quellcode Erzeugung.....	12
Quellcode Optionen .....	13
Quellcode Optionen (Deklarationen).....	13
Quellcode Optionen (Environment).....	13
Quellcode Optionen (Locale) .....	13
Quellcode Optionen (Notifications).....	13
Quellcode Optionen (Quellcode).....	13
Quellcoderzeugung (Voreinstellungen) .....	11

## R

Radio Buttons .....	24
Rechteck.....	28

## S

Schalter.....	24
Skala .....	26
Slider .....	25
Speichern der Applikation.....	12
Stringgadget .....	23

## T

Temporäre Dateien.....	12
Temporäre Liste .....	21
Test Fenster .....	31
Text .....	26
Texteditor (Voreinstellungen).....	11
Tips .....	16
Tutorium .....	9

## V

Variablen (Notification).....	32
Verschachtelung (Voreinstellung) .....	12
Verwendung von MUIBuilder .....	11
Verzeichnisliste.....	23
Voreinstellungen .....	11
Voreinstellungen Aktionen bestätigen.....	11
Voreinstellungen Fenster .....	11
Voreinstellungen Locale .....	11
Voreinstellungen Piktogramm .....	11
Voreinstellungen Quellcoderzeugung.....	11
Voreinstellungen Texteditor .....	11
Voreinstellungen Verschachtelung.....	12

## W

Weiterentwicklung .....	33
-------------------------	----

## Z

Zukünftige Erweiterungen .....	33
Zwischenraum .....	27



# Table of Contents

<b>1</b>	<b>Einführung</b>	<b>1</b>
1.1	Die Vorteile von MUIBuilder	1
<b>2</b>	<b>Grundlegende Prinzipien</b>	<b>3</b>
<b>3</b>	<b>Distribution</b>	<b>5</b>
<b>4</b>	<b>Garantie</b>	<b>7</b>
<b>5</b>	<b>Tutorium</b>	<b>9</b>
<b>6</b>	<b>Verwendung von MUIBuilder</b>	<b>11</b>
6.1	Voreinstellungen	11
6.2	Speichern der Applikation	12
6.3	Quellcode Erzeugung	12
6.3.1	Quellcode Erzeugung : Optionen	13
6.3.2	App Code	13
6.3.3	Object Code	13
6.3.4	Entferne Kennung	14
6.3.5	Erzeuge Kennung	14
6.4	Lokalisation	14
6.4.1	Katalogbeschreibung	14
6.5	Erzeugung der Online Hilfe	15
6.6	Verschiedene Anmerkungen	16
6.6.1	Besondere Zeichen	16
6.6.2	Objektausrichtung	16
6.6.3	Größenveränderbare Fenster	17
<b>7</b>	<b>Objekte</b>	<b>19</b>
7.1	Area Objekt	19
7.2	Applikation	19
7.3	Fenster	20
7.3.1	Temporäre Liste	21
7.4	Gruppe	21
7.5	Liste	22
7.6	Verzeichnisliste	22
7.7	Stringgadget	23
7.8	Schalter	23
7.9	Bezeichner	24
7.10	Cyclegadget	24

7.11	Radio Buttons .....	24
7.12	Bild .....	24
7.13	Slider .....	25
7.14	Gauge .....	25
7.15	Text .....	26
7.16	Skala .....	26
7.17	Prop Gadget .....	26
7.18	Zwischenraum .....	27
7.19	PopAsl.....	27
7.20	PopObjekt.....	27
7.21	Rechteck.....	28
7.22	Farbfeld .....	28
7.23	Menüs .....	29
<b>8</b>	<b>Notification.....</b>	<b>31</b>
8.1	Funktionen.....	32
8.2	Variablen .....	32
8.3	Konstanten .....	32
<b>9</b>	<b>Zukünftige Erweiterungen.....</b>	<b>33</b>
	<b>Danksagungen .....</b>	<b>35</b>
	<b>Index .....</b>	<b>37</b>