

E code generation

COLLABORATORS

	<i>TITLE :</i> E code generation		
<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	E code generation	1
1.1	Einleitung	1
1.2	Einleitung	1
1.3	Der klassische Modus	2
1.4	Der Umgebungs-Modus	3
1.5	Bekannte Fehler	4
1.6	History	4
1.7	Der erzeugte Quelltext	4
1.8	Fehlermeldungen	5
1.9	Die Beispiele	6
1.10	Zukunftsmusik	6
1.11	Verteilung	6
1.12	Der Autor	7

Chapter 1

E code generation

1.1 Einleitung

```

                                GenCodeE (v1.3)
    E Quelltext-Generierungs-Modul für MUIBuilder
                                Update vom 10 März 1994
                                © Copyright 1993, 1994, Lionel Vinténat
  
```

Einleitung	einleitende Worte
Der klassische Modus	Arbeiten wie immer
Der Umgebungs-Modus	Neue Arbeitsmethode
Der erzeugte Quelltext	Was kommt raus ?
Die Fehlermeldungen	Fehler, Fehler, Fehler
Die Beispiele	Damit ihr was lernt...
Fehler	bekannte Fehler
History	Wie's mal war...
Zukunftsmusik	Was noch 'rein soll...
Verteilung	Wer darf was kopieren ?
Der Autor	Wer bin ich ?

1.2 Einleitung

GenCodeE arbeitet in zwei verschiedenen Arbeitsmodi: dem klassischen Modus und dem Umgebungs-Modus.

Wenn Du schon einmal mit GenCodeE v1.0 gearbeitet hast, das mit MUI-Builder v1.0 ausgeliefert wurde, wirst Du Dich schnell mit dem klassischen Modus anfreunden können. In der Tat arbeitet der klassische Modus genau so, wie GenCodeE v1.0 (daher der Name), nur, daß auch Lokalisierungs-routinen angefügt wurden.

Der Umgebungs-Modus ist die große Neuerrungenschaft von GenCodeE v1.3. In diesem Modus verändert GenCodeE direkt Deinen E-Quelltext, um verschiedenen Quelltexte-Teile anzufügen oder zu überarbeiten. Sein Ge-

brauch ist natürlich etwas trickreicher als der klassische Modus, aber dafür auch um vieles komfortabler, da viel Herumkopieren gespart wird.

Der Arbeitsmodus von GenCodeE wird durch das Setzen des "Environment"-Häkchens im Quelltext-Generierungsfenster von MUIBuilder ausgewählt.

Das "Locale" Häkchen bestimmt, ob Lokalisierungs-Routinen angefügt werden sollen oder nicht. Dieses Häkchen gilt sowohl für den klassischen als auch den Umgebungs-Modus. Die "Declarations" und "Code" Häkchen hingegen haben in beiden Modi eine andere Bedeutung. Dies wird weiter unten genauer beschrieben.

1.3 Der klassische Modus

Im klassischen Modus schreibt GenCodeE den Quelltext für die Applikation bzw. das ausgewählte Objekt in die Datei, die im "Code" String-Gadget angegeben ist. Achtung: wenn die angegebene Datei bereits existiert, wird sie ohne Warnung überschrieben.

Wenn das "Declaration" Häkchen aktiviert ist, wird der zu generierende Quelltext die Deklaration der Variablen und ihre Initialisierungswerte für das angewählte Objekt bzw. für die komplette Applikation enthalten.

Wenn das "Code" Häkchen angewählt ist, wird der zu generierende Quelltext den Quelltext zur Generierung des angewählten MUI-Objektes bzw. der gesamten Applikation enthalten.

Man kann diese beiden Gadgets auch gleichzeitig aktivieren, oder nur das "Code"-Häkchen oder nur das "Declaration"-Häkchen, ganz nach Belieben.

Hinweis: Im klassischen Modus generiert GenCodeE nur Quelltext, um das GUI aufzubauen. Im Gegensatz zu GenCodeE v1.0 wird kein Quelltext produziert, um die "muimaster.library" zu öffnen oder die Standard-schleife für MUI-Programme. Aus diesem Grunde sind dem Archiv zwei Programm-Skelette einer MUI-Applikation beigefügt:

- Application.e als eine Beispiel für eine Applikation ohne Lokalisierung
- Application+Locale.e als Beispiel für eine Applikation mit Lokalisierung

Zum Aufbau des Teils von Application+Locale.e, der die Lokalisierungs-routinen enthält, inspirierte mich der durch FlexCat erzeugte Quelltext. Die Anleitung dieses Programms enthält mehr Erklärungen. Darüber hinaus ist es ein exzellentes Programm, das man sich wirklich mal ansehen sollte (der Autor, Jochen Wiedmann, kann im InterNet unter der Adresse wiedmann@mailserv.zdv.uni-tuebingen.de erreicht werden).

Diese Quelltexte sind nur Beispiele, die ruhigen Gewissens ohne Erlaubnis verändert und denn eigenen Bedürfnissen angepaßt werden können. Die doMethod() Funktion wurde mir direkt von Wouter van Oortmerssen (Danke Wouter !), dem Autoren von Amiga E, zugeschickt.

1.4 Der Umgebungs-Modus

In diesem Modus versucht GenCodeE die Datei, die im "Code" String-Gadget angegeben ist, so zu verändern, daß alle Änderungen, die an einem Objekt bzw. der gesamten Applikation vorgenommen wurden, übernommen werden.

Um dies zu erreichen müssen in der Quelltext-Datei einige Regeln beachtet werden, da GenCodeE leider nicht über künstliche Intelligenz verfügt.

Genauer gesagt sucht GenCodeE nach zwei verschiedenen Stellen im Quelltext:

- die Stelle, wo die globalen Variablen deklariert werden
- die Prozedur, in der alle Variablen initialisiert werden und der Quelltext zur Erzeugung der Objekte eingefügt werden muß

Tatsächlich generiert GenCodeE für jedes Objekt eine eigene Prozedur, in der das Objekt, bzw. die Applikation generiert wird. Deshalb müssen alle Variablen, die für das GUI benutzt werden, global deklariert werden, damit jede Prozedur auf sie zugreifen kann.

Damit GenCodeE diese Variablen im Quelltext finden kann, müssen sie in einer Zeile stehen, die mit /*MUIB*/ beginnt. GenCodeE löscht bei jeder Generierung die Zeilen, die mit /*MUIB*/ beginnen und schreibt die Definitionen neu hinein. Die Deklarationen müssen unbedingt aufeinander folgen, ohne Leerzeile, da GenCodeE abbricht, wenn eine Zeile nicht mehr mit /*MUIB*/ beginnt.

Das einzige, was demnach bei der ersten Generierung des Sources zu tun ist, ist eine Zeile, die nur /*MUIB*/ enthält, an der Stelle einzufügen, an die die Deklarationen eingefügt werden. Den Rest erledigt GenCodeE.

Im Umgebungs-Modus legt GenCodeE die Variable-Initialisierungen und den Quelltext zur Generierung der einzelnen Objekte für jedes Objekt in eine eigene Prozedur. Die Prozedur zur Generierung der Applikation heißt `create_app()`, die Prozedur für ein Objekt, das `ObjLabel` heißt, wird `create_ObjLabel()` benannt.

Um den Quelltext zu generieren, sucht GenCodeE nach einer Zeile, die mit `"PROC create_app()"` oder `"PROC create_ObjLabel()"` und ersetzt den kompletten Quelltext zwischen PROC und ENDPROC durch den neuen Quelltext. Das einzige, was zu tun ist, bevor GenCodeE den Quelltext das erste mal bearbeitet, ist für jedes Objekt eine Zeile `"PROC create_ObjLabel()"` bzw. für die Applikation `"PROC create_app()"` einzufügen, gefolgt von einer Zeile, die `"ENDPROC"` beinhaltet. Den Rest erledigt GenCodeE.

Da die Auswirkungen der "Declaration"- und "Code"-Häkchen auf den erzeugten Quelltext ein wenig komplexer Natur sind, hier noch eine Tabelle, in der alle Möglichkeiten verzeichnet sind:

+-----+	+-----+	+-----+
	"Declarations" ein	"Declarations" aus
+-----+	+-----+	+-----+

"Code" ein	Deklarationen + Prozeduren	nur Prozeduren	
"Code" aus	nur Deklarationen	nix !	
+-----+-----+-----+			

Ein kleines Beispiel zum Verständnis:

Du möchtest ein GUI generieren, daß mit einem Aufruf generiert wird, aber auch Zugriff auf ein bestimmtes Objekt, sagen wir, das List-Objekt `my_list`, haben. Also fügst Du Deinem Source folgende Zeilen hinzu:

- eine Zeile, die `/*MUIB*/` enthält
- die leere Prozedur `create_app()`
- die leere Prozedur `create_my_list()`

Danach, zur Generierung oder dem Update der Variablen, klickst Du auf "Appli Code" mit dem "Declarations"-Häkchen aktiviert und dem "Code"-Häkchen deaktiviert. Um den Quelltext und die Initialisierung auf den aktuellen Stand zu bringen, klickst Du müssen beide Häkchen aktiviert sein. Anstelle von "Appli Code" kann natürlich auch der "Objekt Code" Button verwendet werden, wenn nur das selektierte Objekt geändert werden soll.

Achtung: wenn nur das "Declarations"-Häkchen aktiviert ist und auf "Object Code" geklickt wird, werden die globalen Variablen komplett durch die des Objekts ersetzt. Dies ist meistens nicht das, was gewünscht wird.

Ein letzter Tip: experimentiere ein wenig mit dem Programm herum, bevor Du irgendwelche wichtigen Dateien modifizierst, da die Dateien ja automatisch verändert werden.

1.5 Bekannte Fehler

keine Fehler bisher bekannt

1.6 History

Version 1.0 : - erste, mit MUIBuilder v1.0 ausgelieferte Version

Version 1.1 : - kleiner Bugfix (manchmal wurden die Einrückungen vergessen)

- niemals ausgeliefert

Version 1.2 : - Lokalisierungsroutinen angefügt

- niemals ausgeliefert

Version 1.3 : - Umgebungs-Modus hinzugefügt

- mit MUIBuilder v1.1 ausgelieferte Version

1.7 Der erzeugte Quelltext

Da MUIBuilder allgemeinen Quelltext generiert, der hinterher durch die Module in die entsprechende Sprache gewandelt wird, enthält der E-Quelltext prinzipiell dieselben Informationen, wie der C-Quelltext. Deshalb kann das, was zum C-Quelltext gesagt wurde, auch auf den E-Quelltext angewandt werden. Einzige Ausnahme: die MUI_TRUE Konstante wurde mit dem Wert 1 belegt, da MUI die E-Konstante TRUE, die mit -1 belegt ist, nicht als TRUE anerkennt, sondern von 1 als TRUE ausgeht (C TRUE = 1).

Der einzige diffizile Punkt ist die Behandlung von Strings, die im MUIBuilder eingegeben wurden. Generell kann jeder String wie ein C-String eingegeben werden mit der Ausnahme, daß ''' auch als ''' eingegeben werden kann, ohne ein "\" voranzustellen. Das E-Quelltext-Modul wird es richtig übersetzen. Etwas genauer:

- \r wird durch \b ersetzt
- \n, \t und \e bleiben unverändert
- \0oo mit oo eine oktale Zahl : \ muß von "0" gefolgt werden, wiederum gefolgt von zwei Ziffern zwischen "0" und "8"
- \xhh mit hh eine hexadezimale Zahl : \ muß von "x" gefolgt werden, wiederum gefolgt von zwei Ziffern zwischen "0" und "9", "a" und "f" oder "A" und "F"
- ' wird durch \a ersetzt

Achtung: wenn der Syntax von \hxx oder \0oo verletzt wird, beschwert sich GenCodeE nicht sondern erzeugt Quelltext, der nicht ganz dem Erwarteten entspricht.

Wenn GenCodeE auf \033 oder \x1B (dezimal 27, Escape) trifft, wird es durch \e ersetzt. Andererseits, wenn es auf \0oo oder \xhh trifft, das einen anderen Wert als 27 hat, wird kein String vom Typ 'string' generiert, sondern vom Typ ["s","t","r","i","n","g",0]:CHAR, weil es in E kein Äquivalent zu \0oo oder \xhh in C gibt. Aber diese beiden Formen sind vollkommen äquivalent.

Der erzeugte Quelltext ist voll mit Makros identisch denen in C. Um dieses Makros zu benutzen, braucht man einen Annex-Preprozessor, da Amiga E Makro-Expandierung nicht unterstützt. Deshalb ist MUIBuilder ein weiteres archiv beigelegt, das alles enthält, um die Quelltexte zu behandeln, im besonderen ein Preprozessor. Näheres in der Anleitung im Archiv enthalten.

1.8 Fehlermeldungen

Sollte ein Fehler auftreten, gibt GenCodeE diesen in einem Requester aus:

- "Out of memory !" GenCodeE braucht mehr Speicher
- "I/O Problem !" GenCodeE kann eine Datei nicht
finden, schreiben oder lesen
- "Incorrect source format !" GenCodeE kann in der Quelltext-

- datei die Schlüsselworte
"/*MUIB*/", "create_app()",
"create_ObjLabel" oder "ENDPROC"
nicht finden
- "You must generate a label Du hast Quelltext für ein aus-
for the object !" gewähltes Objekt angefordert, ohne
ein Label dafür zu generieren.

1.9 Die Beispiele

Alle Beispiele, die mitgeliefert werden, sind im Umgebungs-Modus erstellt. Zum nicht-lokaliserten Beispiel ist nichts besonderes zu sagen. Ich bin einfach der Standardprozedur gefolgt, die im Kapitel über den Umgebungs-Modus erklärt ist und habe ein paar Zeilen zum Öffnen und Schließen des Fensters angefügt.

Im lokalisierten Beispiel habe ich das Wort "application" durch den Namen der Applikation ersetzt (case sensitiv !) und ein bißchen Quelltext, der durch FlexCat (nochmal: wenn Du dieses Programm nicht kennst, sofort ausprobieren !) erzeugt wurde, eingefügt. Nur der Quelltext von Click mußte ein bisschen verändert werden, da im Quelltext von FlexCat zwei "\033" in "\e" geändert werden mußten.

Zu guter Letzt habe ich zu jedem lokalisierten Beispiel, zusätzlich zum Quelltext und dem Executable, noch die Dateien xxxx.cd, xxxx.ct (französische Version), xxxx.catalog (französische Version) und xxxx_cat.e (Quelltext generiert durch Flexcat für diesen Catalog) beigelegt.

1.10 Zukunftsmusik

Das Ideal eines solchen Quelltext-Generators ist es, sich auf getrennte Kompilation abzustützen. Das heißt, daß Modifikationen am GUI nur eine einzige Datei betreffen, die vom kompletten Modul referenziert wird. Dies wäre die einfachste und komfortabelste Lösung für alle.

Leider ist Amiga E nicht in der Lage, dieses Konzept zu unterstützen. Deshalb wird GenCodeE sich in dem Maße weiterentwickeln, in dem sich MUIBuilder und Amiga E weiterentwickelt.

1.11 Verteilung

GenCodeE und alle anderen Dateien im E Verzeichnis dürfen nicht ohne MUIBuilder kopiert werden, außer, ich gebe mein Einverständnis dafür. Desweiteren unterliegen alle Dateien dem Copyright des Autoren und keine darf ohne meine explizite Einwilligung verändert werden.

Ich kann nicht verantwortlich gemacht werden für irgendwelche Schäden, die durch den Gebrauch von GenCodeE hervorgerufen werden:

Gebrauch und Mißbrauch geschieht auf eigene Gefahr !

1.12 Der Autor

Man kann mich per Post erreichen unter:

- Studentenadresse, bis einschließlich Juli 1994:

Lionel Vinténat
appartement 21
11 rue François Oulié

31500 TOULOUSE
Frankreich

- Heimatadresse:

Lionel Vinténat
3 impasse Boileau
Lotissement Les Termes

87270 COUZEIX
Frankreich

Schreibt mir bitte bis Juli 1994 an meine Studentenadresse, da ich dort öfter und länger bin, als zu Hause.

Außerdem kann man mich auch über das InterNet erreichen. Meine e-mail-Adresse hier ist vinténat@irit.fr. Ich würde es vorziehen, wenn ihr mir per e-mail schreibt. Ich werde versuchen, alle Anfragen, die als e-mail eintreffen, zu beantworten, aber erwartet nicht von mir, daß ich einen Brief schreibe (Ich bin sehr faul, wenn es darum geht, einen Stift in die Hand zu nehmen...).