

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	GenCodeE v1.4	1
1.2	Einleitung	1
1.3	Die erstellte dateien	2
1.4	Bekannte Fehler	3
1.5	History	3
1.6	Der erzeugte Quelltext	3
1.7	Die Beispiele	4
1.8	Zukunftsmusik	4
1.9	Verteilung	4
1.10	Der Autor	5

Chapter 1

E code generation

1.1 GenCodeE v1.4

GenCodeE (v1.4)
E Quelltext-Generierungs-Modul für MUIBuilder
Update of 18 april 1994
© Copyright 1993, 1994, Lionel Vintenat

Deutsche Überstzung von Schröer

Einleitung	einleitende Worte
Die erstellte dateien	Was ist von GenCodeE erstellt ?
Der erzeugte Quelltext	Was kommt raus ?
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

Warnung ! Diese neue Version von GenCodeE (1.4) ersetzt nicht die vorhergehende. GenCodeE v1.4 tut genau dasselbe, aber es tut es anders !

GenCodeE v1.3 und GenCodeE v1.4 sind alle für den Gebrauch mit MUIBuilder v1.1 geschrieben und generieren genau denselben Quelltext. GenCodeE v1.3 jedoch modifiziert um Umgebungs-Modus die Quelltext-Dateien direkt, wogegen GenCodeE v1.4 zwei Dateien erzeugt, die mit Hilfe des E-Preprozessors EPP von Barry Wills verbunden werden müssen.

Wenn Sie kein EPP-Benutzer sind, ist GenCodeE v1.4 vollkommen nutzlos für Sie, aber andernfalls...

EPP ist im AmiNet im Verzeichnis dev/e erhältlich und ein wirklich gutes Utility für Amiga E v2.1b.

1.3 Die erstellte dateien

GenCodeE v1.4 generiert immer zwei Dateien. Sollten Sie im 'Code'-String-Gadget im Code-Generierungs-Fenster von MUIBuilder beispielsweise "Prog" oder "Prog.e" angegeben haben, generiert GenCodeE v1.4 die Dateien "Prog_defs.e" und "Prog_procs.e".

Wenn Das 'Environment' Häkchen nicht aktiviert war, wird folgendes erzeugt:

- in "Prog_defs.e" einige Strings "var_name : LONG"
- in "Prog_procs.e" der "rohe" Quelltext der Applikation oder der Objekte, je nach Einstellung

Dies ist jedoch nur interessant, wenn Sie ungewöhnlicherweise einige Quelltext-Teile mit Cut/Copy/Paste in Ihren eigenen Quelltext einfügen wollen.

Alles wichtige jedoch geschieht bei GenCodeE v1.4 im Umgebungs-Modus. In diesem Modus können dir beiden Dateien, die GenCodeE erzeugt direkt mit EPP eingebunden werden, wenn irgendwo in der Hauptdatei die Zeile "PMODULE 'Prog_defs', 'Prog_procs'" auftaucht. EPP wird intelligent alle Teile zusammenfügen.

In diesem Modus enthält "Prog_defs.e":

- einige Aufrufe von externen E Modulen durch die "MODULE"-Anweisung (diese Aufrufe sind nötig, um MUI zu benutzen)
- die MUI_TRUE Konstanten-Definition (siehe Der erzeugte Quelltext)
- die Definition eines E Objektes, dessen Name "obj_app" ist, wenn die Generierung der gesamten Applikation gewünscht wurde, sonst "obj_ObjectMUI", wenn die Generierung des Objektes "ObjectMUI" verlangt wurde. Die Felder dieses Objektes sind alle generierten Variablen.

"Prog_procs.e" enthält in diesem Modus:

- eine Prozedur mit dem Namen "create_app", wenn die Generierung der gesamten Applikation verlangt wurde, sonst "create_ObjectMUI", wenn nur die Generierung des Objektes "ObjectMUI" verlangt wurde. Diese Prozedur enthält den gesamten Generierungs-Quelltext der Applikation oder des Objektes
 - eine Prozedur "dispose_app", wenn die gesamte Applikation generiert werden sollte bzw. "dispose_ObjectMUI", wenn nur das Objekt "ObjectMUI" generiert wurde
 - die "doMethod"-Prozduer für E (ihr Source wurde mir von Wouter van Oortmerssen, dem Amiga E Autoren, geschickt): diese ist essentiell
-

wichtig für die Programmierung von MUI.

Die Benutzung dieser beiden Dateien ist sehr einfach: da die Variablen nicht global, sondern innerhalb eines E Objektes generiert wurden, können soviele Variablen vom Typ "ObjectMUI" deklariert werden, wie Kopien dieses Objektes gebraucht werden. Dann werden diesen Variablen Werte zugewiesen, indem "create_ObjectMUI" aufgerufen wird, wenn Sie es brauchen. Schließlich werden alle Objekte am Ende durch "dispose_ObjectMUI" freigegeben. Das ist alles ! Dadurch kann beliebig oft dasselbe Objekt in einem GUI verwendet werden. Dies geht natürlich auch für eine Applikation, aber dafür ist es recht sinnlos.

Auch wenn das alles ein bisschen kompliziert klingt, machen Sie ein paar Versuch damit und sehen Sie, daß es sehr intuitiv und mächtig ist.

1.4 Bekannte Fehler

keine Fehler bisher bekannt

1.5 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

Version 1.4 : - dritte ausgelieferte Version von GenCodeE, ohne MUIBuilder

- benötigt MUIBuilder v1.1

1.6 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.

1.7 Die Beispiele

Zu den Beispielen gibt es nichts besonderes zu sagen, außer, daß für die Lokalisierung FlexCat benutzt wurde. Zu jedem habe ich, zusätzlich zum Quelltext und dem ausführbaren Programm, auch die Dateien xxxx.cd, xxxx.ct (französische Version), xxxx.catalog (französische Version) und xxxx_cat.e (der Quelltext, der von FlexCat erzeugt wurde) beigelegt.

1.8 Zukunftsmusik

Da ich denke, daß separate Kompilation ein essentieller Bestandteil eines jeden "seriösen" Compilers ist, werden alle nachfolgenden Versionen von GenCodeE dieses Feature mit Hilfe von EPP benutzen, während wir auf AmigaE v3.0 warten.

1.9 Verteilung

Dieses Archiv darf solange weiterkopiert werden, solange niemand einen Profit daraus schlägt. Ohne Einwilligung des Autors darf keine andere Form der Weiterverbreitung gewählt werden.

Dieses Archiv darf in PD-Sammlungen aufgenommen werden, solange die obigen Konditionen nicht verletzt werden.

However the archive must be distributed in it's entirety and all its files (except the icons) are copyrighted by the author. None of them can be modified without my permission.

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

Gebrauch und Mißbrauch geschieht auf eigene Gefahr !

1.10 Der Autor

Man kann mich per Post erreichen unter:

- Studentenadresse, bis einschließlich Juli 1994:

Lionel Vintenat
appartement 21
11 rue François Oulié

31500 TOULOUSE
Frankreich

- Heimatadresse:

Lionel Vintenat
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 vintenat@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...).
