Génération Code E

COLLABORATORS			
	TITLE : Génération Code E		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		July 26, 2024	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

Génération Code E

1.1 GenCodeE v1.4

```
******************************************************************************

                              GenCodeE (v1.4)
                 Module de génération du code E pour MUIBuilder
                          Mise-à-jour du 18 avril 1994
                    © Copyright 1993, 1994, Lionel Vintenat

******************************************************************************

 Introduction
 Les fichiers générés
 Le code généré
 Les exemples
 Bugs
 Historique
 Futur
 Distribution
 L'auteur
```

1.2 Introduction

```
    Attention, cette nouvelle version de GenCodeE (1.4) ne remplace pas la
précédente (1.3) : GenCodeE v1.4 fait la même chose que GenCodeE v1.3, mais
il le fait différemment !
    En effet, GenCodeE v1.3 et GenCodeE v1.4 sont tous les 2 prévus pour
fonctionner avec MUIBuilder v1.1 et génèrent donc EXACTEMENT le même code.
Cependant, GenCodeE v1.3 modifie directement vos fichiers sources dans
le mode environnement, alors que GenCodeE v1.4 génére toujours 2 fichiers
séparés qui doivent être rassemblés avec EPP, le préprocesseur E de Barry
Wills.
    Si vous ne vous servez pas de EPP, GenCodeE v1.4 ne vous est d'aucune
utilité, dans ce cas gardez GenCodeE v1.3, mais dans le cas contraire...
    Notez que EPP est disponible dans sa dernière version (v1.4d) sur
aminet dans dev/e. Essayez-le, c'est vraiment un trés bon utilitaire pour
Amiga E v2.1b.
```

1.3 Les fichiers générés

GenCodeE v1.4 génére toujours 2 fichiers. Si dans le gadget chaîne 'Code' de la fenêtre de génération de code de MUIBuilder, vous avez écrit (par exemple) "Prog" ou "Prog.e", GenCodeE v1.4 va générer les fichiers "Prog_defs.e" et "Prog_procs.e".

Si vous n'activez pas le checkmark 'Environnement' dans cette même fenêtre, vous obtiendrez :
- dans "Prog_defs.e" une suite de chaîne du type "nom_var : LONG"
- dans "Prog_procs.e" le code "brut" de l'application ou d'un objet selon le cas

Ceci n'est intéressant que pour le cas improbable où vous devriez faire du couper/coller directement dans votre code.

Mais tout l'intérêt de GenCodeE v1.4 apparaît en mode environnement. En effet, dans ce mode, vous pouvez directement utiliser les 2 fichiers générés avec EPP en rajoutant quelque part dans votre fichier source principal la ligne "PMODULE 'Prog_defs', 'Prog_procs'". EPP se chargera de réunir intelligemment tous les morceaux.

Dans ce mode, "Prog_defs.e" contient :
- des appels à des modules E externes avec l'instruction "MODULE" : ces modules sont nécessaires pour l'utilisation de MUI
- la définition de la constante MUI_TRUE (voir le code généré)
- la définition d'un objet E de nom "obj_app" si vous avez demandé la génération du code de l'application toute entière, ou "obj_ObjetMUI" si vous avez demandé la génération du code de l'objet "ObjetMUI", au niveau de MUIBuilder; cet objet E a pour champs toutes les variables générées

Dans ce mode, "Prog_procs.e" contient :
- une procédure de nom "create_app" si vous avez demandé la génération du code de l'application toute entière, ou "create_ObjetMUI" si vous avez demandé la génération du code de l'objet "ObjetMUI", au niveau de MUIBuilder; cette procédure contient tout le code de création de l'application ou de l'objet selon le cas
- une procédure de nom "dispose_app" si vous avez demandé la génération du code de l'application toute entière, ou "dispose_ObjetMUI" si vous avez demandé la génération du code de l'objet "ObjetMUI", au niveau de MUIBuilder; cette procédure contient tout le code d'effacement de l'application ou de l'objet selon le cas
- la procédure "doMethod" pour le langage E (son source m'a été donné par Wouter van Oortmerssen, l'auteur d'Amiga E) : celle-ci est indispensable pour utiliser MUI

Aprés, l'utilisation de ces 2 fichiers est trés simple. Comme les variables ne sont pas générées seules, mais à l'intérieur d'un objet E, vous pouvez déclarez autant de variables de type l'objet "obj_ObjetMUI" que vous avez besoin d'instance de cet objet dans votre application. Ensuite, vous affectez ces variables par un appel à "create_ObjetMUI" quand vous en avez besoin. Puis, pour effacer toutes ces instances d'objet, vous utilisez à chaque fois la procédure "dispose_ObjetMUI". Et voilà, vous pouvez ainsi facilement créer de multiples fois un même objet dans votre interface graphique.

Dans le cas de l'application elle-même, bien que ce soit possible, vous n'en créerez bien-sur qu'une seule instance.

Même si cela vous semble un peu confus, faîtes des essais, tout cela est trés intuitif (et surtout trés puissant).

1.4 Le code généré

Comme MUI-Builder génére du code générique qui est ensuite interprété par des modules, le code E généré est identique dans son contenu au code C généré. Reportez vous donc au paragraphe sur le langage C pour tout ce qui concerne les variables générées, la notification et la localisation. La seule vrai différence concerne une constante MUI_TRUE qui doit être définie à 1 et qui est utilisée à la place de TRUE (qui vaut -1) pour MUI, car celui-ci ne reconnait pas toujours la valeur -1 comme vrai.

Le seul point délicat concerne les chaines entrées dans MUI-Builder. La règle générale est de les entrer comme si vous vouliez les utiliser en C à une exception près : contrairement à ce qu'il est dit avant, le caractère " peut être entré tel quel (ce n'était génant que pour le C). C'est le module de génération de code E qui s'occupera de la traduction. Plus exactement ce module reconnait :

- \r qui est remplacé par \b
- \n, \t et \e laissés tels quels
- \0oo où oo est un nombre en octal : attention le \ doit être obligatoirement suivi par un 0, suivi lui-même de 2 chiffres entre 0 et 8
- \xhh où hh est un nombre héxadécimal : attention le \ doit être obligatoirement suivi par un x en minuscule, suivi lui-même de 2 symbôles corrects (chiffre ou lettre entre a et f, ou A et F)
- le caractère ' est remplacé par \a

Attention, si la syntaxe de \0oo ou de \xhh n'est pas respectée, le module de génération du code E ne signalera pas d'erreur mais le résultat ne sera surement pas celui qui était attendu !

Si le module de génération de code E rencontre \033 ou \x1B (valeur 27 en décimal, c'est à dire le code de ESC), il le remplacera par \e. Si par contre, il rencontre un \0oo ou un \xhh de valeur décimal différente de 27, il produira une chaine E non pas sous la forme 'chaine', mais sous la forme ["c", "h", "a", "i", "n", "e", 0]:CHAR car il n'y a pas d'équivalent en E aux \0oo et aux \xhh du C. Mais les 2 formes sont totalement équivalentes.

Enfin, le code source produit contient plein de macros identiques à celles du C. Pour utiliser ces macros, vous aurez besoin d'un préprocesseur annexe car Amiga E ne le permet pas à l'origine. C'est pour cela qu'est jointe à cette archive une autre archive, Mac2E.lha (également disponible seule sur aminet). Vous trouverez dans cette archive tout ce qu'il faut pour exploiter les fichiers sources générés, avec en particulier un préprocesseur spécialisé dans le remplacement des macros et destiné à Amiga E. Reportez-vous donc à cette archive pour plus de détails.

1.5 Les exemples

Rien de spécial à dire sinon que j'ai utilisé FlexCat pour les exemples localisés. Pour chacun, j'ai fourni en plus des sources et de l'exécutable les fichiers xxxx.cd, xxxx.ct (version française), xxxx.catalog (version française) et xxxx_cat.e (le code généré par FlexCat pour ce catalogue).

1.6 Bugs

Aucun de connu.

1.7 Historique

```
Version 1.0 :  - 1ère version distribuée avec MUIBuilder v1.0
Version 1.1 :  - un petit bug corrigé (GenCodeE oubliait parfois d'indenter le
               début du code)
               - version jamais distribuée
Version 1.2 :  - ajout du support de la localisation
               - version jamais distribuée
Version 1.3 :  - ajout du mode environnement
               - 2ème version distribuée avec MUIBuilder v1.1
Version 1.4 :  - utilisation de la "compilation séparée offerte par EPP"
               - 3ème version distribuée, externe à l'archive de MUIBuilder
               - doit être utilisé avec MUIBuilder v1.1
```

1.8 Futur

Comme je pense que la compilation séparée est un aspect indispensable à tout compilateur "sérieux", toutes les versions suivantes de GenCodeE utiliseront cette caractéristique, par l'intermédiaire de EPP en attendant Amiga E v3.0, puis par Amiga E v3.0 quand celui-ci sortira.

1.9 Distribution

Cette archive peut être librement distribuée, tant que personne ne tire aucun bénéfice de cette distribution. Tout autre type de vente ne peut en aucun cas être effectué sans l'autorisation de l'auteur.
Cette archive peut être incluse dans des collections de logiciels du domaine public, tant que les conditions ci-dessus restent vérifiées.

Cependant, l'archive doit être distribuée dans son entier et tous ces fichiers (excepté les icônes) restent sous copyright de l'auteur, et ne peuvent être modifiés sans mon accord.

Enfin, je dégage toute responsabilité quant à l'utilisation de ce programme et les dommages qu'il pourrait causer : vous l'utilisez à vos risques et périls !

1.10 L'auteur

Vous pouvez me joindre par courrier :

- à mon adresse étudiante valable jusqu'en juillet 1994 inclus :

Lionel Vintenat
appartement 21
11 rue François Oulié
31500 TOULOUSE
FRANCE

- à mon adresse familiale :

Lionel Vintenat
3 impasse Boileau
Lotissement Les Termes
87270 COUZEIX
FRANCE

Ecrivez moi plutôt à mon adresse étudiante jusqu'en juillet 1994 car j'y suis beaucoup plus souvent qu'à mon domicile familiale.

Vous pouvez également me joindre sur INTERNET. Mon adresse e-mail est vintenat@irit.fr. Je préfère de trés loin que vous me contactiez par mail que par courrier. Je répondrai toujours aux questions qui me seront posées par mail, par contre n'espérez pas de réponses à un courrier (je suis trés fainéant dés qu'il s'agit de prendre un stylo...).