

# GenCodeC

---

Version 2.00  
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

**MUIBuilder** selbst produziert eigentlich keinen Quellcode, sondern eine Art Beschreibung der erstellten Oberfläche, die schließlich in einen Quellcode nahezu jeder (zumindest in solche, die MUI unterstützen) gewünschten Sprache umgewandelt werden kann (was allerdings mit etwas Aufwand verbunden ist!!).

Um diesen Oberflächenbeschreibungscodes in irgendeiner Form von einem Compiler weiterverarbeiten zu lassen, muß dieser erst mit einem externen Quellcodegenerator bearbeitet werden. GenCodeC tut dies für die Sprache C.

GenCodeC ist **FREEWARE**. Aber das Copyright bleibt bei Eric Totel. Sie können das GenCodeC verändern (Source liegt bei) und dann weiterverbreiten, wenn

1. dem Programm der Vermerk beiliegt, daß ich der Autor des Originals bin
2. mir **vor** der Weiterverbreitung eine Version zuschicken
3. diese Dokumentation (mit meinem Namen) mit dem Programm verbreitet wird (Diese darf ansonsten auch verändert werden)

## 2 GenCodeC ?

Im Archiv MUIBuilder/MB/Developer/C/Sources\_GenCodeC finden Sie den Quellcode zu GenCodeC. Haben Sie keine Hemmungen, ihn zu verändern, verbessern ... oder was Ihnen sonst einfällt!

Das Ziel dieses Programms ist es, aus den temporären Dateien in T:, die **MUIBuilder** erzeugt hat, C-Code zu machen. Dazu bedient es sich der Funktionen der muibuilder.library. Diese wiederum sind in den autodocs beschrieben.

## 3 Die Verwendung von GenCodeC

Dazu müssen Sie zwei Dinge wissen: Wie GenCodeC bedient wird, und wie der, von ihm erzeugte Quellcode zu verwenden ist. Beides wird im Folgenden erklärt:

### 3.0.1 Bedienung von GenCodeC

GenCodeC ist ein Programm, daß nur aus dem CLI gestartet werden kann. Es wird von **MUIBuilder** (mit einem **SystemTags**-Call) aufgerufen, wenn Quellcode erzeugt werden soll und Sie im "Quellcode" Feld in den Voreinstellung "C" angegeben haben.

Wenn Sie es aus einer Shell aufrufen, ohne daß **MUIBuilder** läuft, und es deshalb möglicherweise keine temporären Dateien in T: gibt, wird es wohl garnichts erzeugen. GenCodeC verlangt keine Parameter, da alles nötige in den temporären Dateien von **MUIBuilder** angegeben wird.

### 3.0.2 Der erzeugte Quellcode

GenCodeC erzeugt folgendes (für das Objekt `xy` der applikation `App`) :

- Eine `.h` Datei, in der die Definition von `xy` steht
- eine `.c` Datei (deren Name in **MUIBuilder** definiert wird),<sup>1</sup> in der die Funktion **Createxy** steht, die das Objekt `xy` (in `.h` definiert, s.o.) zurückgibt.
- eine Datei **xyExtern.h**, die alle externen Referenzen auf von Ihnen spezifizierte Variablen oder Funktionen aus Ihrem Programm enthält

Mittels zweier Konfigurations Dateien, **H-Header** und **C-Header**, können Sie den erzeugten Quellcode anpassen: GenCodeC kopiert Sie an den Anfang der erzeugten Quellcodes.

GenCodeC erzeugt sowohl Objekte, als auch Notificationen und das Öffnen der Fenster. Das Beispiel in dieser Anleitung ist also normalerweise brauchbar, um die Oberfläche einer Applikation zu testen.

### 3.0.3 Beispiel

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

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

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

/* Ansi */
#include <stdlib.h>
#include <stdio.h>

/* MUIBuilder */
#include "NONE.h"

struct Library * MUIMasterBase;

/* Init function */
static void init( void )
{
    if (!(MUIMasterBase = OpenLibrary(MUIMASTER_NAME,MUIMASTER_VMIN)))
    {
        printf("Can't Open MUIMaster Library");
        exit(20);
    }
}
```

---

<sup>1</sup> Wenn Sie die `.c` Endung nicht angeben, wird sie von GenCodeC automatisch angehängt

```

/* main function */
main()
{
    struct ObjApp * App = NULL; /* Application object */
    BOOL    running = TRUE;
    ULONG    signal;

    /* Programm initialisierung ( müssen Sie selbst schreiben ) */
    init();

    /* Create Application : von MUIBuilder erzeugt */
    App = CreateApp();

    while (running)
    {
        switch (DoMethod(App->App,MUIM_Application_Input,&signal))
        {
            case MUIV_Application_ReturnID_Quit:
                running = FALSE;
                break;
        }
        if (running && signal) Wait(signal);
    }
    DisposeApp(App);
    CloseLibrary(MUIMasterBase);
    exit(0);
}

```

### 3.0.4 GUI-Objekte später hinzufügen

Ich bin sicher, daß einige Leute frustriert sein werden, weil MUIBuilder nicht alle Objekte von MUI unterstützt. Aber das ist eigentlich kein Problem.

Unter MUI kann man dynamisch (d.h. zur Laufzeit) Objekte in der Oberfläche verändern, hinzufügen, entfernen. Halten Sie in Ihrer Oberfläche einen Platz für das Objekt frei (ein Zwischenraum-Objekt genügt vollkommen) und nutzen Sie die Möglichkeiten, die MUI Ihnen bietet.

Schreiben Sie eine Funktion, die Ihr Objekt erzeugt (hier: `CreateMyObject`) und benutzen Sie sie folgendermaßen:

```

extern APTR CreateMyObject();

/* Die Oberfläche aus MUIBuilder */
App = CreateApp();

/* Fügen Sie Ihr Objekt hinzu */
DoMethod(App->EmptyGroup, OM_ADDMEMBER, CreateMyObject());

```

`EmptyGroup` ist der Name des von Ihnen vorgesehenen Platzes für das neue Objekt.

## 4 Was erzeugt GenCodeC eigentlich?

Einige Details im Bezug auf den erzeugten Quellcode sind möglicherweise von Interesse:

1. Die Struktur in der .h Datei enthält nur die Kennungen der Objekte, die Sie **MUIBuilder** als zu generierend angeben (also die, die im Fensterbaukasten mit einem 'G' vorneweg erscheinen).
2. Die Datei **xyExtern.h** enthält externe Referenzen auf Variablen oder Funktionen aus Ihrem Programm in einem Standard-Format. Nachdem eine solche Referenz zum erstenmal in dieser Datei auftaucht, verändert **GenCodeC** sie nicht mehr; Sie können sie dann also Ihren Wünschen anpassen, ohne daß sie das nächste Mal überschrieben wird.
3. Die .h Datei, die für die Locale gebraucht wird (z.B. von FlexCat erzeugt) muß **xy\_cat.h** heißen.
4. Jetzt sollten Sie die Dateien **H-Header** und **C-Header** nach Ihren Wünschen anpassen. Meiner Meinung nach sind sie wirklich wichtig, denn hier können Sie den Code, den **GenCodeC** erzeugt, nach Ihren Bedürfnissen anpassen.

## Table of Contents

<b>1</b>	<b>Einführung.....</b>	<b>1</b>
<b>2</b>	<b>GenCodeC ?.....</b>	<b>1</b>
<b>3</b>	<b>Die Verwendung von GenCodeC.....</b>	<b>1</b>
	3.0.1 Bedienung von GenCodeC.....	1
	3.0.2 Der erzeugte Quellcode.....	2
	3.0.3 Beispiel.....	2
	3.0.4 GUI-Objekte später hinzufügen.....	3
<b>4</b>	<b>Was erzeugt GenCodeC eigentlich?.....</b>	<b>4</b>