

SimpleCat

Guido Mersmann

COLLABORATORS

	<i>TITLE :</i> SimpleCat		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Guido Mersmann	February 12, 2023	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	SimpleCat	1
1.1	Contents	1
1.2	Distribution	2
1.3	Einführung	2
1.4	Installieren von SimpleCat	2
1.5	Featureliste	3
1.6	Die Skriptbefehle von SimpleCat	3
1.7	Generelle Informationen	3
1.8	Das Version-Kommando	5
1.9	Das Languages-Kommando	6
1.10	Das CodeSet-Kommando	6
1.11	Das AutoNum-Kommando	6
1.12	Das ASMFILe-Kommando	7
1.13	Das CFILe-Kommando	7
1.14	Der Aufruf von SimpleCat	8
1.15	Das Script-Argument	8
1.16	Das LocaleOnly-Argument	8
1.17	Das NoLenghtCheck-Argument	9
1.18	Das NoCheck-Argument	9
1.19	History	9
1.20	Was noch fehlt	10
1.21	Der Author	10

Chapter 1

SimpleCat

1.1 Contents

SimpleCat v1.1

(C) Guido Mersmann 1995

SIMPLECAT IS A GIFTWARE PRODUCT

READ THE DISTRIBUTION SECTION FOR INFORMATION ON DISTRIBUTION

Contents

- Distribution
- Featureliste
- Einführung
- Installieren
von SimpleCat
- Die Skriptkommandos
von SimpleCat
- Starten
- History
Was gabs für Versionen
- Was fehlt noch
- Der Author
Bugreports, Anregungen ...

1.2 Distribution

Distribution

Dieses Programm darf auf allen jederman zugänglichen Medien veröffentlicht werden, solange im Archiv alle Files unverändert beiliegen.

Diese Programm ist Giftware. Das bedeutet, daß jeder, der dieses Programm benutzt, ein kleines Geschenk oder eine kleine Geldsumme an den Autor schickt, um dessen Arbeit zu honorieren.

1.3 Einführung

Was ist SimpleCat ?

Alle Programmierer, die sich mit Katalogen beschäftigt haben, wissen, daß der original CatComp nicht der komfortabelste ist. Will man einen String löschen, so muß man alle IDs neu berechnen, einige Files editieren und anschließend alle Files kompilieren.

Es gibt zwar viele Katalogtools, aber alle haben ein Problem. Sie sind umständlich zu bedienen, oder erlauben das Verändern von String IDs nicht.

SimpleCat knüpft an CatComp an und ist von den Kommandos her weitgehend kompatibel. Es benötigt jedoch nur eine Datei und erzeugt daraus die Katalog- und Sourcecodefiles für alle Sprachen auf einmal. Darüber hinaus erspart er dem gestreßten Programmierer noch einiges an Arbeit.

Ich viele Localeunterstützende Programme geschrieben und bekam schon beim ersten Kontakt mit CatComp würgereize. Aus dieser Tatsache heraus wurde CatComp geschrieben und erweitert.

Es ist ein Tool für Programmierer, soll aber auch dem Anwender, der mit einem Texteditor umgehen kann, ermöglichen eigene Veränderungen zu machen.

1.4 Installieren von SimpleCat

Installieren von SimpleCat

Die Installation besteht aus dem kopieren der SimpleCat Datei in ein beliebiges Verzeichnis.

Am Besten ins C-Directory kopieren, da es so in jeder Shell zur Verfügung steht.

Fertig ist die Installation !

1.5 Featureliste

FeatureList :

`\textdegree{}` benötigt nur eine Quell-Datei.

`\textdegree{}` erzeugt alle Kataloge auf einmal. Dadurch entfallen überflüssige Überprüfungen der Quelldatei.

`\textdegree{}` kann verschiedene SourceCode-Dateien im selben Arbeitsvorgang ↔ erzeugen.

`\textdegree{}` numeriert auf Wunsch alle IDs, von einer selbst wählbaren Zahl an, ↔ durch.

`\textdegree{}` sehr schnelle Umsetzung durch spezielle interne Ablage.

`\textdegree{}` Auch Nicht-Programierer können mit diesem Tool Kataloge einfach ↔ anpassen.

1.6 Die Skriptbefehle von SimpleCat

Die Skriptbefehle von SimpleCat

Generelle Informationen

Version

Languages

CodeSet

AutoNum

ASMFile

CFile

1.7 Generelle Informationen

Generelle Informationen

Alle Zeilen, die mit einem Semikolon beginnen, werden ignoriert.

Zu Beginn möchte ich ein Beispiel Skript einwerfen, an dem die Funktion erklärt wird:

```
;----- Skriptbeginn

## languages english,deutsch
## version $VER: ViewPoint.catalog 1.00 (30.01.95)
## codeset 0
## asmfile english,Viewpoint:SRC.VP_Locale
;
MSG_ScreenTitle (1//)
Hello World !!!
Hallo Welt !!!
;
MSG_Window_Title (2/5/20)
This is an Window
Das ist ein Fenster
;
;----- Skriptende
```

Die erste Zeile, die Text enthält, bestimmt den Label. Im Beispiel sind das `MSG_ScreenTitle` und `MSG_Window_Title`. Das diese Labels später vom Assembler oder C-Compiler bearbeitet werden, darf jeder Label nur einmal im Script vorkommen.

In den Klammern stehen die Angaben (StringID/Minimal Size/Maximal Size)

Die StringID ist einzigartig und darf in einem Skript nur einmal vergeben werden. Sie stellt die Nummer dar, unter der man diesen String später von der `locale.library` anfordern kann.

`MinimalSize` gibt minimale Stringlänge an. Das heißt, daß sobald ein String dieser Gruppe, die angegebene Länge unterschreitet, SimpleCat mit einer Fehlermeldung abbricht. Diese Zahl kann weggelassen werden, dadurch wird die Überprüfung der Minimallänge abgeschaltet.

`MaximalSize` gibt die maximale Stringlänge an. Hier gilt das Gleiche wie für `MinimalSize`, nur gehts um das Maximum der Strings.

Der gesamte Klammersausdruck kann weggelassen werden. Dadurch werden ID und `MinimumSize` auf 0 und `MaximumSize` auf "unendlich" gesetzt.

Da aber die StringID auf 0 gesetzt wird und damit schon beim zweiten String ohne Klammersausdruck doppelt definiert würde, kann dies nur einmal geschen. Es sei denn sie Benutzen das Kommando

```
AutoNum
```

```
.
```

Beispiele:

```
-----
```

```
MSG_Demo1_Text (1/2/10)
;ID=1 MinSize=2 MaxSize=10
```

```
MSG_Demo2_Text (//)
;ID=0 MinSize=0 MaxSize=unendlich

MSG_Demo3_Text
;ID=0 MinSize=0 MaxSize=unendlich

MSG_Demo4_Text (4//)
;ID=4 MinSize=0 MaxSize=unendlich
```

Jetzt kommt der einfachere Teil. Im obigen Text wurde im Languages-Kommando english und deutsch angegeben. Dadurch ist der weitere Aufbau festgelegt. Es folgt eine Zeile "english" und eine Zeile "deutsch". Danach kommt wieder eine Label-Zeile, und so weiter.

Die landesspezifischen Zeilen können aus allen darstellbaren Zeichen aufgebaut werden. Der Backslash nimmt dabei eine besondere Stellung ein:

```
\a   fügt ASCII 7 ein. (Beep)
\b   fügt ASCII 8 ein. (Backspace)
\c   fügt ASCII 155 ein.
\e   fügt ASCII 27 ein. (Escape)
\f   fügt ASCII 12 ein. (Seitenvorschub)
\n   fügt ASCII 10 ein. (Neue Zeile)
\r   fügt ASCII 13 ein. (Zeilenanfang)
\t   fügt ASCII 9 ein. (Tabulator)
\v   fügt ASCII 11 ein. (Vertikalen Tabulator)
\xNN fügt NN ein. NN steht hier für den ASCII-Code in hexadezimal.
\NNN fügt NNN ein. NNN steht hier für den ASCII-Code in octal.
\    fügt einen Backslash ein.
```

Außerdem besteht die Möglichkeit eine landesspezifischen Zeile zu teilen, also in der nächsten Zeile weiterzumachen. Dazu muß die Zeile mit einem Backslash enden und wird daraufhin in der nächsten fortgesetzt.

Wer mit dem CatComp gearbeitet hat, wird hier keine Probleme haben.

1.8 Das Version-Kommando

Das Versionskommando

```
## VERSION $VER: name.catalog 1.0 (01.01.95)
```

Hier wird der Versionsstring nach Commodore-Style Guide angegeben. Alle Kataloge, die später von SimpleCat geschrieben werden, heißen im obigen Beispiel name.catalog. Darum ist es wichtig, das dieser Name korrekt angegeben wird.

Beispiel:

```
## VERSION $VER: SimpleCat.catalog 1.0 (01.01.95)

Erzeugt für alle Sprachen das File "SimpleCat.Catalog"
```

1.9 Das Languages-Kommando

Das Languages-Kommando

```
## LANGUAGES Sprache[,Sprache,Sprache,...]
```

Hier wird angegeben, welche Sprachen in diesem Skript vorliegen. Die Schreibweise entspricht genau der, des jeweiligen Landes.

Dieses Kommando legt außerdem fest, wieviele Strings nach einem Label kommen.

Beispiele:

```
## languages deutsch
```

Hier kommt nach einem Label genau ein String der Sprache deutsch

```
## LANGUAGES english,deutsch,português, suomi
```

Hier folgen dem Label vier Strings, die in der Reihenfolge english, deutsch, português und suomi abgelegt sein müssen.

1.10 Das CodeSet-Kommando

Das CodeSet-Kommando

```
## CodeSet Nummer
```

Der CodeSet ist eine Erfindung von Commodore und wird vom System noch nicht benutzt. Um mit späteren Versionen kompatibel zu bleiben, muß dieser Befehl entweder weggelassen werden oder die Nummer Null sein.

Beispiele :

```
## CodeSet 0
```

```
## codeSet 34
```

1.11 Das AutoNum-Kommando

Das AutoNum-Kommando

```
## AutoNum [Nummer]
```

Mit diesem Befehl wird der Autonummerierungsmodus von SimpleCat eingeschaltet. Alle Strings bekommen eine fortlaufende Nummer.

Das Argument Nummer bestimmt die erste Zahl, die automatisch an den ersten String vergeben wird.

Beispiele:

```
## AutoNum 1
```

Der erste String bekommt die Nummer 1, der Zweite 2 und so weiter.

1.12 Das ASMFIL- Kommando

Das ASMFIL-Kommando

```
## ASMFile Sprache,Filename
```

ASMFile schaltet den Erzeugungsmodus für den Assemblersourcecode ein.

Die Sprache muß eine von den im Languages-Kommando angegebenen sein. Der Filename gibt den Pfad und Namen des zu erzeugenden Sourcecodes an.

Dieser Befehl kann mehrmals benutzt werden und erzeugt demnach mehrere Sourcefiles.

Beispiele:

```
## ASMFile deutsch,SourceCodes:SimpleCat/SimpleCat_Locale.ASM
```

```
## ASMFile português,SourceCodes:SimpleCat/SimpleCat_Locale.ASM
```

1.13 Das CFILE-Kommando

Das CFILE-Kommando

```
## CFile Sprache,Filename
```

CFile schaltet den Erzeugungsmodus für den C-Sourcecode ein.

Die Sprache muß eine von den im Languages-Kommando angegebenen sein. Der Filename gibt den Pfad und Namen des zu erzeugenden Sourcecodes an.

Dieser Befehl kann mehrmals benutzt werden und erzeugt demnach mehrere Sourcefiles.

Beispiele:

```
-----  
## CFile deutsch,SourceCodes:SimpleCat/SimpleCat_Locale.c  
## CFile português,SourceCodes:SimpleCat/SimpleCat_Locale.c
```

1.14 Der Aufruf von SimpleCat

Starten von SimpleCat

In dieser Version gibt es folgendes Template :

```
SCRIPT/A, LOCALEONLY/S, NOCHECK/S, NOLENGHTCHECK/S
```

```
SCRIPT/A
```

```
LOCALEONLY/S
```

```
NOCHECK/S
```

```
NOLENGHTCHECK/S
```

1.15 Das Script-Argument

Das Script-Argument

Dieses Argument muß vorhanden sein. Es gibt den Filename des CatalogScripts an, das alle Kommandos enthält, die SimpleCat benötigt um Dateien zu erzeugen.

1.16 Das LocaleOnly-Argument

Das LocaleOnly-Argument

Mit diesem Schalter werden alle
CFile
und
ASMFile
Kommandos deaktiviert.

Dies ist z.B bei der Wartung und Berichtigung von Katalogen nützlich.

1.17 Das NoLenghtCheck-Argument

Das NoLenghtCheck-Argument

Dieser Schalter deaktiviert das Überprüfen der Minimalen und Maximalen Stringlängen. Dadurch wird der Vorgang des kompilierens ein wenig beschleunigt.

1.18 Das NoCheck-Argument

Das NoCheck-Argument

Mit diesem Argument folgende alle Überprüfungen abgeschaltet :

ID doppel definiert ?
Label doppelt definiert ?
String zu lang ?
String zu kurz ?

Diese Funktion beschleunigt das kompilieren der Cataloge und Sources erheblich. Besonders wenn man Änderungen an Gadgetbeschriftungen macht ist das sehr praktisch.

1.19 History

History

V0.90 Erste lauffähige Version
V0.91 ASMFile eingebaut.
V0.92 Kleinen Fehler behoben, der beim CS-File von BoulderDäsh das erste mal auftauchte.
V0.93 Kleine Fehler behoben.
V0.94 CFile eingebaut.
V0.95 Stringlängenprüfung eingebaut.
V0.96 Kleine Fehler in der Min/Max Routine behoben.
V0.97 Weitere Schalter in den Argumentstring eingebaut.
V1.00 Erste offizielle Version

1.20 Was noch fehlt

Was noch fehlt

\textdegree{} Erzeugung von ASM Routinen

1.21 Der Author

Der Author

Sollten Sie Bugs in SimpleCat finden, so bitte ich um Mitteilung, damit ich diese in neuen Versionen beseitigen kann.

Auch für Anregungen zur Erweiterungen von SimpleCat bin ich jederzeit dankbar.

Guido Mersmann
Glatzer Straße 12
48477 Hörstel
Deutschland

FIDONET : 2:2449/205.15
TREKNET : 37:1/501.15

Wer mir Geld überweisen möchte, kann das gerne tun:

Sparkasse Ibbenbüren

Bankleitzahl: 403 510 60
Kontonummer : 71107791
