GFA

Richard Gordon Faika

COLLABORATORS			
	TITLE : GFA		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Richard Gordon Faika	March 3, 2023	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

GFA

1.1 rgfa-entwicklungspaket version 4.2

```
RGFA-Entwicklungspaket Version 4.2
  Inhaltsverzeichnis
  Vorwort
  Lizenzvertrag

  Umfang des Archives
  Einrichten der Komponenten

  Einzeldokumentationen
    CBO      V8.0   - Compiler/Linker-Benutzeroberflche
    Licom    V2.0   - Bibliotheksbearbeitungsprogramm
    MNDX     V1.0   - Indexgenerator
    LicomLIB V6.0   - Programmbibliothek
    LGLIB           - Licom-GEMDOS-Dateibibliothek
    GCPATCH  V0.5   - Compiler-Patchprogramm
    GL       V4.1   - Linker

  Historie
  Impressum
```

1.2 vorwort

```
Vorwort
  Die Firma GFA entwickelt bekanntlich keine Software mehr fr
  ATARI-Kompatible Systeme und hat somit vor Jahren schon die
  Pflege des GFA-Entwicklungssystems aufgegeben.

  Die letzte halbwegs vernnftige Version ist die Version 3.6TT
  des GFA-Entwicklungssystems fr Atari-Kompatible.
  Seither wurden allerlei Anpassungen und Patches, kleinerer und
  grerer Art von Privatleuten vorgenommen um Unstimmigkeiten
  und Fehler zu umgehen, da sich das Entwicklungssystem nach wie
  vor einer groen Beliebtheit erfreut und auch das schnellste
```

Entwicklungssystem ist, welches derzeit auf ATARI-Kompatiblen Systemen existiert.

Sehr gute Zusatzprogramme, wie Face Value oder Ergo!Pro wurden und werden fr das GFA-EWS entwickelt, soda das GFA-EWS auf seinem heutigen Stand den besten und professionellsten BASIC-Dialekt darstellt.

Da auch RGF Software dabei nicht unttig war, gab es vor Jahren mal eine neue GFA-Bibliothek fr den Linker und vor Kurzem auch einen neuen GFA-Linker.

Leider hat nun aber gar kein Mensch mehr den rechten berblick, wie und was er an Komponenten bentigt, um die aktuellen Grund-komponenten (GFA-Compiler, Linker, Bibliothek..) auf dem neuesten technischen Stand zur Verfgung zu haben.
Um diesem Mistand Einhalt zu gebieten und Anwendern der Produkte von RGF Software bezglich des GFA-Entwicklungssystems einen besseren berblick und ein kompaktes Paket an die Hand zu geben, wurde dieses (R)GFA-Grundpaket zusammengestellt.

Dabei wurde ein erheblicher Teil des GFA-EWS einfach ersetzt durch neu erstellte Komponenten, soda am Ende der Installation dieses Paketes (sofern Sie mchten) zur Programmerstellung aus einem GFA-Source nur noch ein gepachter Compiler von der Firma GFA-Systemtechnik brigbleibt.

Folgende Komponenten wurden von RGF Software fr das GFA-Entwicklungssystem erstellt: ein Linker, eine Linker-Bibliothek, das TTP zur Erstellung der Indexdatei der Bibliothek, ein Programm zur Bearbeitung der Bibliothek, eine grafische Benutzeroberflche fr den Compiler und den Linker (Shell) und ein Patchprogramm fr den Compiler.

Lizenzvertrag

1.3 lizenzvertrag

Lizenzvertrag
Lizenzvertrag "RGFA Entwicklungspaket"

Der Nutzer erkennt diese Bedingungen durch erstmalige Installation bzw. erste Benutzung der Software unwiderruflich an.

Ý1 RGF Software rumt Ihnen (im folgenden "Nutzer" genannt) das Nutzungsrecht an dieser Software gem den nachfolgenden Lizenzbedingungen unentgeltlich ein. Die Software darf weder abgendert, zurckentwickelt oder weiterentwickelt werden es sei denn, dies ist gesetzlich zulssig oder innerhalb dieses Vertrages ausnahmegeregelt.

Ý2 Der Nutzer ist berechtigt,

- das Programm auf einem Einzel- oder Mehrplatzrechner zu installieren und anzuwenden,
- eine Sicherungskopie zu eigenen Zwecken zu fertigen und
- die Software und das Nutzungsrecht daran an einen Dritten zu bergeben/bertragen.

Bei der Weitergabe der Software darf der Nutzer die Rechte an der Software nur in dem Umfang bertragen, wie er sie von RGF Software bertragen bekommen hat. Die Weitergabe darf nur erfolgen, wenn diese Lizenzbedingungen dem Empfnger der Software bekanntgegeben und von diesem in vollem Umfang anerkannt wurden.

Ý3 Mittels der Software knnen ausfhrbare Programme erstellt werden. Mit der Software dieses Paketes erstellte Programme mssen eine kurze Kennzeichnung wie "RGFA-Devkit <Version> von RGF Software" oder hnliches enthalten, die auf die Verwendung von Software dieses Paketes hinweist.

Ý4 Smtliche Leistungsdaten und sonstige Softwarebeschreibungen stellen keine Zusicherung irgendwelcher Eigenschaften dar, auch wenn sie auf DIN und/oder sonstige Normen Bezug nehmen. RGF Software bernimmt keine Gewhrleistung fr einen bestimmten Zweck oder dafr, da die Leistungs-merkmale des Programms individuellen Ansprchen entsprechen.

RGF Software haftet fr Sach- und Rechtsmngel nur im Fall arglistigen Verschweigens.

Ý5 RGF Software ist an einer bersetzung in andere Sprachen interessiert. Der Nutzer hat das Recht, die Dokumentation und oberflchenbeschreibende, nicht ausfhrbare Dateien, in eine andere Landessprache zu bersetzen, ohne da dabei inhaltlich und informelle nderungen resultieren.

Ý6 Kommt es bei der Anwendung der Software zu Datenverlusten beim Nutzer, beschrnkt sich die Haftung von RGF Software auf Vorsatz und grobe Fahrlssigkeit. RGF Software haftet nicht fr Folgeschden, insbesondere aus dem Gesichtspunkt der positiven Vertragsverletzung, sowie unvorhersehbare oder im Verantwortungsbereich des Nutzers liegende Schden. Vorstehende Haftungsregelung betrifft vertragliche wie auch auervertragliche Ansprche. Unberhrt bleibt die Haftung nach dem Produkthaftungsgesetz.

Ý7 Fr smtliche Rechtsbeziehungen der Parteien gilt das Recht der Bundesrepublik Deutschland unter Ausschlu der einheitlichen Kaufgesetze.

```
  Umfang des Archives
```

1.4 umfang des archives

```
Umfang des Archives
  Das Paket umfasst folgende Komponenten:


  /RGFA41/          Ordner
    LIZENZ.TXT      - Lizenzvertrag als Textdatei.
    RGFA.HYP        - Die Dokumentation als ST-Guide Hypertext.
    /HYPSRC/        Ordner
      RGFA.STG      - Der Sourcecode der Dokumentation.
      CBO.IMG       - Bilddatei mit Snapshot der Applikation "CBO.APP".
      LICOM.IMG     - Bilddatei mit Snapshot der Applikation "LICOM.APP".
      GCPATCH.IMG   - Bilddatei mit Snapshot der Applikation "GCPATCH.APP".

    /CBO80/         Ordner
      CBO.APP       - Applikation grafische Benutzeroberflche.
      CBO.CNF       - Konfigurationsdatei, ASCII-Format.
      CBO.RSC       - Oberflchenbeschreibende Datei (Resource-Datei).

    /LICOM20/       Ordner
      LICOM.APP     - Applikation Programm-Bibliotheksbearbeitung.
      LICOM.RSC     - Oberflchenbeschreibende Datei (Resource-Datei).
      MNDX.TTP      - Applikation Indexdateierstellung einer Programm-Bibliothek.

    /LICOMLIB60     Ordner
      LICOMLIB      - Programmbibliothek fr den Linker.
      LICOMLIB.NDX  - Indexdatei der Programmbibliothek.

    /LGLIB          Ordner
      LGLIB.LST     - Licom-GEMDOS-Dateibibliothek

    /GCPATCH        Ordner
      GCPATCH.APP   - Applikation Patchprogramm fr den GFA-Compiler.
      GCPATCH.RSC   - Oberflchenbeschreibende Datei (Resource-Datei).

    /GL41           Ordner
      GL.TTP        - Applikation Linker.


  Einrichten der Komponenten
```

1.5 einrichten der komponenten

```
Einrichten der Komponenten
  Die Installationsanleitung beschreibt eine Installation, mittels derer
  Sie direkt ausfhrbare Programme aus GFA-Sourcecode erzeugen knnen.
```

Bentigt wird hierbei alleinig ein originaler GFA-Compiler der
Version 3.6TT, wobei dieser gleich welcher Landessprache ausgefhrt
sein darf.

Wenn Sie eigene Komponenten mit diesem Paket verwenden beachten Sie bitte,
da innerhalb dieser Dokumentation auf die folgende Installation Bezug
genommen wird.

Installation:

1. - Kopieren Sie das entpackte Archiv auf ein Laufwerk und/oder in einen
     Ordner ihrer Wahl.

2.   Kopieren Sie Ihren bisher verwendeten GFA-Compiler "GFA_BCOM.PRG"
     der Version 3.6TT in das Verzeichnis GCPATCH.

3. - Starten Sie die Applikation GCPATCH.APP im Ordner GCPATCH.
  3.1 - Beachten Sie die Ausgaben im Ausgabebereich des Fensterdialoges
        (weies Feld).
  3.2 - Bei Vorhandensein des "GFA_BCOM.PRG" im Ordner "GCPATCH" wird der
        Knopf "Apply" anwhlbar geschaltet, den Sie nun anwhlen.
  3.3 - Ist der Patch erfolgreich, finden Sie im Ordner "GCPATCH" eine neue
        Datei namens "GC.TTP" - dies ist der aktualisierte GFA-Compiler.

4. - Starten Sie die Applikation "CBO.APP" im Ordner "CBO80".
  4.1 - Im rechten, unteren Teil des erscheinenden Fensterdialoges finden
        Sie mehrere, anklickbare Dateipfadangaben.
  4.2 - Klicken Sie die Dateipfadangabe fr den Compiler an und whlen Sie
        die Datei "GC.TTP" im Ordner "GCPATCH" aus.
  4.3 - Klicken Sie die Dateipfadangabe fr den Linker an und whlen Sie
        die Datei "GL.TTP" im Ordner "GL41" aus.
  4.4 - Klicken Sie die Dateipfadangabe fr die Programmbibliothek an
        und whlen Sie die Datei "LICOMLIB" im Ordner "LICOMLIB60" aus.
  4.5 - Bettigen Sie den Knopf "save" im Fensterdialog um die Konfiguration
        zu speichern.

Ihr RGFA-Paket ist nun einsatzbereit.
Zur Erstellung einer ausfhrbaren Programmdatei, studieren Sie bitte das
Kapitel "CBO".

Einzeldokumentationen

1.6 cbo

CBO
Dies ist die Compiler-Benutzeroberflche CBO.

Im Linken Teil finden Sie die Optionen zum Compiler.
Im rechten oberen Teil die Optionen zum Linker und darunter die
Einstellungen zu den Pfadangaben zu Compiler, Linker, Programmbibliothek
und Sourcedatei.

Die Compiler- und Linkeroptionen knnen auch innerhalb des Sourcecodes gendert werden, wobei immer jene Einstellung aktiv ist, ab der die Option vom Compiler gefunden wurde.

Beispiel:

im CBO wurde die Optione "C+ Regsave vor C:()" deaktiviert, dann wird die Einstellung wie folgt vom Compiler behandelt:

```
[...]
~C:adresse()      - Register werden nicht gesichert
$C+               - ab jetzt werden die Register gesichert
~C:adresse()      - Register werden gesichert beim Aufruf
$C-               - ab jetzt werden Register nicht mehr gesichert
~C:adresse()      - Register werden nicht gesichert
[...]
```

Die Compileroptionen (von oben nach unten):

Einstellung Registersicherung
- Wenn aktiviert, werden die Register des Prozessors gesichert, wenn ein Maschinenprogrammaufruf mittels "C:adresse%(..)" erfolgt. Ntig, wenn Maschinenprogramme Verwendung finden, die die Register A3-A6 verndern.

Einstellung Bombenabfang
- Wenn aktiviert, werden Buserror-Interrupts abgefangen. (mit der LicomLIB nicht verwendbar!).

Einstellung Funktionsrckgabe
- Wenn aktiviert, werden Funktionsergebnisse, die Integer sind, also keine Stringrckgabe, oder Realwertrckgabe haben, als reiner Integerwert zurckgeliefert (Byte, Word, Long). Die Option ist generell empfehlenswert.

Einstellung der Gre des Variablenspeichers
- Wenn aktiviert, wird hiermit die Gre des GFA-internen Variablenspeichers bestimmt. Der Variablenspeicher ist der Speicherbereich, in dem temporre Daten gehalten werden. Dazu zhlen erzeugte Strings und Felder. Emfehlenswert ist das Minimum von 16kB (16384 Bytes). Ein zu geringer Variablenspeicher kann zu Programmabbruch fhren, die mit der LICOMLIB compilierten Programme melden dann einen Fehlernummer #08 und die Gre des zum Zeitpunkt des Abbruchs noch vorhandenen Variablenspeichers.

Einstellung der Wertbreite von SELECT/CASE/ENDSELECT-Konstrukten.
- Einstellung der Wertbreite zwischen 16 und 32 Bit (Word/Long). Empfehlenswert ist hier die Einstellung WORD. Wenn Sie innerhalb des Programmes ein SELECT/CASE/ENDSELECT-Konstrukt haben, in dem Sie 32 Bit breite Werte auswerten, klammern Sie das Konstrukt wie folgt:

```
$S%
SELECT
[..]
ENDSELECT
$S&
```

Einstellung der Optimierung von SELECT/CASE/ENDSELECT-Konstrukten.
 - Einstellung zur Optimierung der Konstrukte vom Compiler.
   Empfehlenswert ist hier die Angabe "SELECT auf Speed opt.", da die
   Lngenoptimierung nur sehr gering ausfllt, jedoch einen erheblichen
   Geschwindigkeitsverlust mit sich bringt.

Einstellung der Form der Fehlerrckgabe.
 - Einstellung der Form der Fehlerrckgabe als Text oder als Nummer.
   Empfehlenswert ist die Rckgabe als Nummer, da die Fehlertexte
   das Programm unntig aufblhen.

Einstellung von PROCEDURE/RETURN-Konstrukten.
 - Hiermit kann man die Art der PROCEDURE/RETURN-Unterprogramme
   einstellen, wobei der Compiler 68k- oder GFA-Unterprogramme erzeugt.
   68k-Unterprogramme, also Unterprogramme die allein mit "rts"
   abgeschlossen sind, sind etwas krzer aber nur dann schneller als
   die GFA-Unterprogramme, wenn KEINE Parameter bergeben werden.
   Die Option "68k-Procedure" ist also nur dann empfehlenswert, wenn
   man viele parameterlose Unterprogramme im Programm hat.

Einstellung von ENDFUNC.
 - Hiermit kann man einstellen, ob der Compiler den Befehl "ENDFUNC"
   erzeugt (ist wie ein RETURN von PROCEDURE), soda bei Funktionen, die
   nicht mit "RETURN <wert>" verlassen werdenkein Programmabbruch erfolgt.
   Die Option sollte auf "ENDFUNC egal" stehen, da das Programm sonst
   unntig aufgeblht wird und Fehler bezglich falscher oder fehlender
   Parameterrckgabe sonst sehr schwer zu finden sind.

Einstellung von Multiplikationsanweisungen
 - Hier ist die Breite von Multiplikationen einstellbar. Bei der Einstellung
   "MUL 16/32 Bit", werden Multiplikationen maximal mit 16 Bit als Eingabe
   und Ergebnis ausgefhrt. Bei "MUL 64 Bit" werden Multiplikationen
   generell mit Eingabe von max. 32 Bit integer oder 32 Bit real und
   Ausgabe von 64 Bit integer oder 32 Bit real ausgefhrt.
   Die Einstellung sollte auf 16/32 Bit stehen, wenn sie Werte multi-
   plizieren, die als Eingabe max. 16 Bit haben (Ausgabe max. 32 Bit).
   Verwenden Sie dann den Befehl "MUL(a,b)", wenn sie Realwerte zu
   Multiplizieren haben, nehmen Sie den Ausdruck "a*b". Mit dem
   Befehl "MUL()" ausgedrckte Multiplikationen werden im Kompilat bei
   Wortbreiten Integerwerten zum schnellen Maschinenbefehl "muls ...",
   wobei der Ausdruck "a*b" z.Bsp. die etwas langsamere, eigene
   Multiplikationsroutine aufruft.

Einstellung von Divisionsanweisungen
 - quivalent zur Multiplikation

Einstellung von RC_INTERSECT
 - Die Berechnung von sich berschneidenden Rechtecken mittels "RC_INTERSECT"
   kann im WORD-Format oder im LONG-Format erfolgen, wo bei die Version
   mit 16 Bit breiten Datenwerten wesentlich flinker ist.
   Hierbei ist die Einstellung "RC_INTERSECT in WORD" empfehlenswert, wenn
   Sie die Funktion z.Bsp. nur verwenden um Bildschirmausgaben zu berechnen.

Die Linkeroptionen:

Einstellung der Programmflags
- F+ = Fastload
  M+ = Fastram darf verwendet werden
  PrgM+ = Programm darf ins Fastram geladen werden
  Shared = bei mehrmaliger Ausfhrung wird das gleiche Textsegment verwendet (nur mit LICOMLIB und gepachtem Compiler).

  Mit der Listenauswahl darunter lsst sich der Speicherschutz einstellen, der fr normale Programme auf "global" oder "nur lesen" stehen sollte.

Einstellung Symboltabelle
- Wenn eine Symboltabelle erstellt werden soll, die in das ausfhrbare Programm geschrieben wird, ist das Hkchen "-s Symboltabelle" zu setzen.

Einstellung "Objektdatei lschen"
- Soll die vom Compiler erzeugte Objektdatei (*.O) nach Erstellung der Applikation gelscht werden, ist dieses Hkchen zu setzen.

Einstellung von "Objectfile"
- Wenn Sie externe Programmfunktionen einbinden mchten, knnen Sie hier eine Objektdatei im DRI-Objektformat angeben, die der Linker in das Programm mit einbindet und die aus dem GFA-Programm mittels der Compilerfunktion $X<name> eingebunden werden knnen.

Erstellen eines lauffhigen Programmes:

1. Klicken Sie die Dateipfadangabe fr "Sourcefile" an und whlen Sie Ihren GFA-Sourcecode aus.

2. setzen Sie die entsprechenden Compiler- und Linkeroptionen oder verwenden Sie die Vorgabeinstellung.

3. Klicken Sie auf den Knopf "make", es erscheint die Maus als Biene und die Knpfe "make", "compile" und "link" drcken sich selbststndig ein und zeigen somit den Fortschritt an.
   Nach Abschlu ist Ihr Programm erzeugt, der Verlauf ist in der gleichnamigen Mitschriftdatei mit der Dateiendung "*.log" (Vorgabe) gespeichert. Treten Fehler bei der Kompilierung oder beim Linken auf, erscheint eine Alertbox mit der Fehlernummer des Compilers und/oder des Linkers. Schauen Sie dabei nach Abschlu in die Mitschriftdatei, die im Verzeichnis angelegt wird wo die GFA-Sourcedatei liegt.

- Der Knopf "make" kompiliert und linkt die GFA-Sourcedatei.
- Der Knopf "compile" kompiliert nur die GFA-Sourcedatei und erstellt daraus die Objektdatei.
- Der Knopf "link" Linkt die erstellte Objektdatei zu einem lauffhigen Programm.
- Der Knopf "save" sichert die aktuellen zum GFA-Sourcecode gehrenden Compiler und Linkereinstellungen in einer dem GFA-Sourcecode gleichnamigen Datei mit der Endung "*.cbo", wobei die Pfadangaben zum Compiler, Linker und zur Programmbibliothek nicht darin mit abgespeichert werden.

Laden Sie nun eine andere GFA-Sourcedatei, wobei schon eine *.cbo-Datei

existiert, werden automatisch die dafr zugehrigen Einstellungen gesetzt.
Achtung! Das Format der *.cbo und *.cnf-Dateien des CBO hat sich
gendert (jetzt im ASCII-Format), soda diese nicht zu lteren CBO-
Versionen als 8.0 kompatibel sind.

Licom

1.7 licom

Licom
Dies ist das Programmbibliothek-Bearbeitungsprogramm Licom.

Die Programmbibliothek besteht aus mehreren, zusammengefassten Objektdateien,
welche die eigentlichen Funktionen enthalten die viele GFA-Befehle bentigen.

Mit der Applikation "Licom" knnen Sie nun eine Programm-Bibliothek komfortabel
mit einer grafischen Benutzeroberflche bearbeiten, sofern dies notwendig ist.
Nachdem Sie eine Bibliothek bearbeitet haben, ist es notwendig eine neue ↩
Indexdatei
zu erstellen. Hierzu bettigen Sie einfach den Knopf "make NDX", wobei sich dafr
im gleichen Verzeichnis das Indexerstellungsprogramm MNDX.TTP befinden sollte.

Nachdem Sie Ihre Bibliothek geladen haben, erscheinen links in der Liste
alle in der Bibliothek vorhandenen Objektdateien, die Sie nun importieren,
exportieren, ersetzen oder einfgen und lschen knnen.

Eine Scriptabarbeitung ist ebenfalls integriert, wobei folgende Scriptbefehle
in einem Script zugelassen sind, welches, wenn sie in eine ASCII-Datei mit der
Dateiendung "*.lsc" geschrieben wurden, von Licom ausgefhrt wird:

```
libload <dateipfad>             - Bibliothek einladen
replace <OBJECT>,<dateipfad>    - Objekt in der Bibliothek durch eine externe
                                  Objektdatei ersetzen.
paste <dateipfad>               - Eine externe Objektdatei einfgen.
lsave                           - Die Bibliothek unter dem gleichen Namen  ↩
   speichern.
lsave_as <dateipfad>            - Die Bibliothek unter einem neuen Namen  ↩
   speichern.
make_ndx <dateipfad>            - Indexdatei der Bibliothek <dateipfad>  ↩
   erstellen.
end                             - Ende der Scriptausfhrung.
```

Das Programm ist sonst im Groen und Ganzen selbsterklrend und bedarf somit
keiner weiteren, tiefschrfenden Ausfhrung.

MNDX

1.8 mndx

MNDX
MNDX ist ein Kommandozeilenprogramm, welches aus einer
Programmbibliothek eine Indexdatei erstellt, die der Linker
zustzlich zur Programmbibliothek bentigt.

Aufruf: mndx <dateipfad>

LicomLIB

1.9 licomlib

LicomLIB
Die LicomLIB ist eine Programmbibliothek, welche Grundlegende nderungen
beinhaltet, die die Ausfhrung von mit ihr erstellter Applikation sicherer,
sauberer und schneller macht.

Die LicomLIB untersttzt auch eine automatische FPU-untersttzung, soda
arithmetische Befehle bei Rechnern mit FPU bis zu 10 mal schneller ausgefhrt
werden.
Zuknftig sind Ausfhrungen der LicomLIB direkt fr FPU, 030er und 030+FPU
geplant.

Wenn man Programme mit der LicomLIB erstellt, sind einige grundlegende
Dinge zu beachten, ohne die das erstellen einer Applikation mit der
LicomLIB sinnlos ist. Machen Sie sich von der Vorstellung frei, da
Sie alte "Schweineprogramme" mit der LicomLIB kompilieren und danach
eine saubere Applikation haben. Die LicomLIB wurde mit dem Ziel entwickelt,
mglichst saubere und schnelle Compilate zu erzeugen. Wenn Sie GEM-Programme
erstellen und wirklich mchten, da Ihre Software auch unter Speicherschutz-
Systemen, auf allen mglichen Emulatoren, ATARI-Rechnern und Clones
grundstzlich lauffhig ist, dann sollten Sie sich ein Herz fassen und
auf die LicomLIB umstellen.

Bei der Erstellung Ihrer Applikation ist dann zu beachten, da Sie gewisse
Befehlsgruppen nicht mehr oder nur eingeschrnkt verwenden drfen.

Was Sie nicht verwenden drfen:

Alle GFA-Dateibefehle
- In der LicomLIB sind sie erneuert und Fehlerbehandelt, aber nach wie vor
unsauber und machen eine Applikation instabil durch die ntige interne

```
      Fehlerbehandlung, die ja normalerweise durch den Programmierer gesteuert
      werden sollte.
      Wenn Sie wirklich mchten, da Ihre Applikation stabil luft, sollten
      Sie auf eine GEMDOS-Bibliothek (z.Bsp. die Licom-GEMDOS-Bibliothek (LGLIB) ↩
          )
      zurckgreifen!

  Alle ehemaligen LINE-A-Befehle
    - 95% aller ehemaligen Line-A-Funktionen sind nur noch Dummyfunktionen
      und stehen somit nicht mehr zur Verfgung, soda ein Programmierer
      gezwungen wird sauberere VDI-Befehle zu verwenden. Die restlichen 5%
      sind die GFA-eigenen Fensterbefehle OPENW CLOSEW TOPW usw. (nicht
      ~WIND_OPEN() etc.!)


Was Sie verwenden drfen:

    - alle sonstigen Befehle, mit Einschrnkung bei RESERVE, welches im
      kompilierten Zustand nicht ausgefhrt werden sollte
      (empfohlen).

      Beispiel wie es im kompilierten Zustand nicht ausgefhrt wird:

      '
      compiled!=BYTE{BASEPAGE+256}<>96
      '
      IF NOT compiled!
        RESERVE 32768
      ENDIF
      '
      [...]
      '
      IF NOT compiled!
        RESERVE
      ENDIF
      EDIT
      '



LGLIB
```

1.10 lglib

```
Licom GEMDOS-Dateibibliothek
  Die LGLIB ist eine kleine Bibliothek mit den bentigten
  Dateifunktionen inclusive VFAT-Erkennung und Umlenkung
  von Bildschirm-Textausgaben in eine Datei.

  Sie finden den Sourcecode als LST-Datei im Ordner LGLIB.
```

GCPATCH

1.11 gcpatch

GCPATCH
Dies ist das Compiler-Patchprogramm GCPATCH.

GCPATCH ist ein Patchprogramm fr den originalen GFA-Compiler 3.6TT, wobei dieser gleich welcher Landessprache ausgefhrt sein darf.

Beim Programmstart sucht GCPATCH nach der Datei "GFA_BCOM.PRG" im gleichen Verzeichnis und gibt eine entsprechende Meldung im Textfeld aus.

Nun brauchen Sie nur noch den Knopf "Apply" anklicken. GCPATCH ldt den alten Compiler ein, tested ihn auf Behandelbarkeit und nimmt bei Erfolg verschiedene Patches vor um danach den "neuen" Compiler unter dem Dateinamen "GC.TTP" im gleichen Verzeichnis abzuspeichern.

Treten Inkompatibilitten auf, wird kein Patch durchgefhrt und entsprechende Meldungen im Textfeld ausgegeben.

Sollte wider Erwarten der Patch bei Ihnen nicht funktionieren, obwohl Sie einen originalen Compiler verwendet haben, schreiben Sie mir bitte umgehend eine Email: richie@rgfsoft.com

GL

1.12 gl

GL
Version 4.1
GL.TTP ist ein neugeschriebener Linker fr das GFA-Entwicklungssystem.

folgende Vorteile gegenber dem originalen Linker erwarten Sie:

- kann mit langen Dateinamen umgehen
- luft mit Speicherschutzsystemen
- kann anderslautende Dateinamen verarbeiten (z.Bsp. "licomlib.lib" statt "licomlib")
- luft auch unter Magic-PC mit langen Dateinamen
- ist flink

Impressum

1.13 historie

```
Historie
  Historie ab 1.7.2001

   ú 01.07.2001 (Version 4.1)

          - Erstverffentlichung des kompletten RGFA-Paketes.


   ú 02.07.2001 (Version 4.2)

          FIX: Kleinen Bug im Patch zum Compiler behoben (fehlerhaftes
          anhngen von Buchstaben an das GFA-File auf Partitionen mit
          langen Dateinamen).
```

1.14 impressum

```
Impressum
  Grafik, Layout & Hypertext-Sourcecode Copyright (c)2001 RGF Software,
  Richard Gordon Faika
  Nachdruck und Verffentlichung dieser Seiten oder Teile daraus
  (auch auszugsweise) nur mit schriftlicher Genemigung.
  Erstellt wurde die Dokumentation auf einem Atari-TT030 unter
  MagiC mittels dem High-Tech-Editor Luna.
```


```
  Richard Gordon Faika
  RGF Software
  Richard Sorge Strae 24
  10249 Berlin

  eMail: richie@rgfsoft.com

  www.rgfsoft.com
  www.rgfsoft.org
  www.myluna.de
  IRC: #atari.de
```