

Seminar Betriebssysteme: Linux
Programmportabilität auf andere
Betriebssysteme

Markus Nöbauer

28. Mai 2004

Copyright ©2004 Markus Nöbauer

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or and later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled GNU Free Documentation License.

Zusammenfassung

Diese Arbeit ist im Rahmen der Lehrveranstaltung Seminar:Betriebssysteme an der Johannes Kepler Universität im Sommersemester 2004 entstanden. Die Seminararbeit gibt Einblick in die Problematik der Portierung nativen Codes von Linux nach Windows. Der Schwerpunkt liegt auf der Portierung von graphischen Benutzeroberflächen.

Mein Ziel war es die zwei Ansätze für die Entwicklung portable Programme vorzustellen. Der Old School Ansatz, Code in einer portablen Programmiersprache wie C oder C++ zu schreiben und auf den Zielplattformen jeweils neu zu kompilieren, ist der verbreitetste. Dazu gibt es eine ganze Reihe von Bibliotheken. Kapitel 2 GTK+ und Kapitel 3 Qt stellen die zwei bekanntesten Toolkits vor.

Spätestens seit Java ist Plattformunabhängigkeit salonreif. Den zweite Ansatz, Compile once Run anywhere, behandelt Kapitel 4. Dieses letzte Kapitel stellt das Mono Projekt vor, eine freie Implementierung des .NET Frameworks von Microsoft.

Inhaltsverzeichnis

1	Allgemeines zu Widget Toolkits	3
1.1	Begriffsbestimmung	3
1.2	Warum Portabilität ?	4
1.3	Wie erreicht man Portabilität ?	5
2	GTK+	6
2.1	GLib	6
2.1.1	Typedefs	6
2.1.2	Speicherverwaltung	7
2.1.3	Strings	8
2.1.4	Timer	9
2.1.5	einfach und doppelt verkettete Listen	10
2.1.6	Mehr	11
2.2	GTK+ Widgets	11
2.2.1	Ein einfaches Fenster	11
2.2.2	Container und Frames	13
2.2.3	Buttons und Combobox	17
2.2.4	Textelemente	18
2.3	Beispielanwendung	19
2.3.1	Tools für die Entwicklung unter Linux	19
2.3.2	Tools für die Entwicklung unter Windows	21
2.3.3	Beispielprogramm	21
2.3.4	Portable Anwendungen	25
3	Qt	27
3.1	Über Qt	27
3.1.1	Installation	27
3.2	Grafische Benutzeroberflächen mit Qt	29
3.2.1	Allgemeines	29
4	.NET und Mono	30
4.1	Ein paar Worte zu .NET	30
4.1.1	Die Common Language Specification	31
4.1.2	Das .NET Framework	31
4.1.3	ASP.NET und WindowsForms	32
4.2	Mono	32
4.2.1	Mono Komponenten	33
4.2.2	Installation von Mono auf Linux und Windows	33

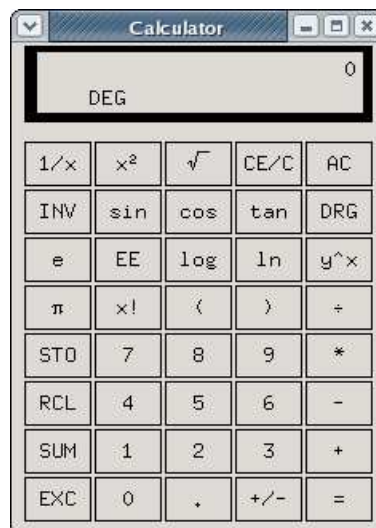
4.3	Portabilität	34
4.3.1	Von Mono nach Mono	34
4.3.2	Von Mono nach .NET	34
4.3.3	Von .NET nach Mono	35
4.4	Mono in der Praxis	36
5	Fazit	38
5.1	Entscheidung für eine Strategie	38
5.1.1	Old School	38
5.1.2	Platformunabhängigkeit	38
A	GNU Free Documentation License	40
A.1	Preamble	40
A.2	Applicability and Definitions	40
A.3	Verbatim copying	42
A.4	Copying in quantity	42
A.5	Modifications	43
A.6	Combining documents	44
A.7	Collections of documents	45
A.8	Aggregation with independent works	45
A.9	Translation	45
A.10	Termination	45
A.11	Future revision of this license	46
A.12	Addendum: How to use this License for your documents	46
B	GNU General Public License	47
B.1	Preamble	47
B.2	GNU General Public License - Terms and conditions for copying, distribution and modification	48
B.3	How to Apply These Terms to Your New Programs	52
C	GNU Lesser General Public License	54
C.1	Preamble	54
C.2	GNU Lesser General Public License - Terms and conditions for copying, distribution and modification	56
C.3	How to Apply These Terms to Your New Libraries	61

Kapitel 1

Allgemeines zu Widget Toolkits

1.1 Begriffsbestimmung

Toolkits für die Erstellung grafischer Benutzeroberflächen sind in der Unix und Linux Welt weit verbreitet. Im Umfeld von Microsoft Windows sind GUI Toolkits hingegen kaum bekannt. Das Windows API bietet eine Hi-Level Schnittstelle für die Erstellung von GUI Elementen wie Fenster, Scrollbalken, Buttons usw. Das X Window System im Linux Umfeld hat diese Funktionalität nicht und unterstützt nur das Zeichnen einfacher Formen und das Setzen von Farben. Was nicht bedeutet dass es unmöglich ist Anwendungen mit einer grafischen Benutzeroberfläche zu erstellen. Ein gutes Beispiel ist `xcalc`.



Neben `xterm`, `xclock` und `xload` gehört `xcalc` zu den gängigen X Applikationen die auf nahezu allen Systemen vorhanden sind. Start:

```
[developer@localhost]$ xcalc
```

Der Begriff Widget ist ein Kunstwort und setzt sich aus Window und Gadget zusammen. Ein Widget ist also ein Komponente einer GUI, muss aber nicht zwangsweise sichtbar sein. Container Widgets sind nicht sichtbare GUI Elemente die andere Widgets in sich aufnehmen und strukturieren können.

Es existieren viele verschiedenen Toolkits für das Erstellen von grafischen Benutzeroberflächen. Einige der bekanntesten sind Motif, Lesstif¹, GTK+, Qt von der Firma Trolltech und WxWidgets. Toolkits wie GTK+ können längst mehr als nur Fenster und Buttons zeichnen. Zusätzliche Bibliotheken wie GLib sind auf Portabilität getrimmt und haben einen großen Funktionsumfang. Widget Toolkits erlauben erst die Portierung von nativen Programmen mit grafischer Benutzeroberfläche auf andere Betriebssysteme. Sie bieten dem Programmierer eine einheitliche Schnittstelle und sind nach unten auf die jeweilige Plattform zugeschnitten. In der Funktionsweise lassen sich zwei Prinzipien unterscheiden. Eine Klasse von Toolkits kapselt Aufrufe von vorhandener Hi-Level APIs, die andere Klasse zeichnet die Komponenten selbst und emuliert ein bestimmtes Look And Feel.

1.2 Warum Portabilität ?

Warum ist Portabilität ein anstrebenswertes Ziel, und lohnt sich der Mehraufwand an Zeit und Kosten ? Um portable Programme zu schreiben ist Wissen über die angestrebten Plattformen notwendig. Eine Tatsache die sich in Personal-, Hardware-, Lizenz- und Schulungskosten niederschlägt. Dennoch übersteigt der Nutzen oft die Kosten.

Mit portablen Programmen kann man einen größeren Markt erschließen. Windows ist ein sehr weit verbreitetes Betriebssystem und nur wenige Firmen können es sich leisten diesen Markt auszuschließen. Umgekehrt ist Linux eine aufstrebende Alternative und macht Windows auf vielen Feldern Konkurrenz. Es ist nicht ratsam diesen wachsenden Markt zu ignorieren.

Sind Programme von Anfang an auf Portabilität ausgelegt können mehr Kunden zu geringeren Kosten angesprochen werden. Die Entwicklung einer portablen Anwendung ist in der Regel billiger als das Portieren einer vorhandene Anwendung auf eine bestimmte Plattform.

Durch Fokus auf Portierbarkeit können sogar die Entwicklungskosten gesenkt werden. Für 10 Windows Entwickler muß man 10 Betriebssystem Lizenzen und 10 Compiler Lizenzen bezahlen. Läßt man stattdessen 8 Entwickler auf Linux mit einer freien Entwicklungsumgebung arbeiten, sinken die Lizenzkosten. [Dalheimer]

Portable Programme neigen zu weniger Fehlern. Viele Bugs kommen nur in ganz bestimmten Situationen zur Geltung. Diese böartigen Fehler überleben meistens auch die Qualitätssicherung und treten erst nach einiger Zeit im laufenden Betrieb beim Kunden auf. Wenn man für verschiedene Plattformen entwickelt ist die Wahrscheinlichkeit genau diese Bugs zu finden höher weil die

¹Ein Motif Klon aber unter Library General Public License <http://www.lesstif.org>

Programme auf unterschiedlichen Betriebssystemen mit unterschiedlicher Speicher-verwaltung, Timesharing, usw. arbeiten.

1.3 Wie erreicht man Portabilität ?

Will man portable Programme schreiben, ist einiges zu beachten. Leider kann man nicht immer mustergültige portable Software erstellen und muss an manchen Stellen Kompromisse zu Gunsten der Entwicklungszeit und Kosten machen.

Grundvoraussetzung für Portabilität ist Aufrufe des Betriebssystem API zu vermeiden. Funktionen des Win32 API sind so gut wie nie auf Linux oder Unix verfügbar. Für die Interaktion mit dem Betriebssystem gibt es Bibliotheken die eine einheitliche Schnittstelle für den User² haben und das Arbeiten mit Dateien und das Ansprechen von angeschlossenen Geräten kapseln.

Manchmal decken die verwendeten Bibliotheken nicht alle notwendigen Funktionen ab. Wenn das der Fall ist, kann der POSIX³ Standard helfen. Darin ist der Zugriff auf Systemspezifische Dateien und Prozesse festgeschrieben. Unix deckt fast den ganzen Posix Standard ab, und Windows NT hat einen eigenen Posix Layer. Der zusätzliche POSIX Layer für Windows kostet allerdings ein wenig Performance.[Dalheimer]

Im schlechtesten Fall deckt weder eine vorhandene Bibliothek noch der POSIX Standard die gewünschte Funktionalität ab. Dann ist Handarbeit gefragt. Am besten kapselt man die Funktionsaufrufe statt sie überall im Quellcode umzuschreiben.

²Ein User kann auch eine andere Bibliothek oder ein Builder Tool sein, nicht notwendigerweise ein Mensch.

³IEEE Spezifikation für Portable Operating System Interface for Computing Environments

Kapitel 2

GTK+

GTK+ ist ein sehr weit verbreitetes Toolkit. Das Plus im Namen steht für die Weiterentwicklung und Verbesserungen im Toolkit. GTK+ wurde ursprünglich für das Grafikprogramm GIMP¹ entwickelt. Die Buchstaben GTK stehen für GIMP Toolkit. Die bekanntesten Anwendungen von GTK+ sind GIMP und der GNOME² Desktop. GTK+ selbst unterliegt der GNU Lesser GPL³ und basiert auf 3 wichtigen Komponenten: [IBM1]

- Pango ist eine Layout- und Renderbibliothek für Text mit Schwerpunkt auf Internationalisierung. Sie ist verantwortlich für Text und Schrifthandhabung in GTK+
- Die ATK Bibliothek beinhaltet Schnittstellen für alternative Ausgabegeräte.
- Die GLib ist eine Low Level C Bibliothek und bildet das Herzstück von GTK+. Die GLib umfaßt Typedefinitionen, Makros, Datenstrukturen, Fehlerbehandlung uvm.

2.1 GLib

Die GLib Bibliothek ist eine Erweiterung der Standard C Bibliotheken. Sie bietet viel Funktionalität und nimmt Rücksicht auf Portierbarkeit. Obwohl in C geschrieben hat GLib einen starken Objektorientierten Ansatz.[GLib] Dieser Teil der Seminararbeit ist keine Auflistung des GLib API, und stellt nur einige nützliche Teile vor. Die GLib und GTK+ Beispiele lassen sich in der Shell wie folgt kompilieren:

```
gcc dateiname.c `gtk-config --cflags --libs` -o ausgabedatei
```

2.1.1 Typedefs

Die GLib bietet eine ganze Reihe an Datentypen welche die Standarddatatype aus C erweitern. Einerseits sparen die GLib Typen Tipparbeit, andererseits

¹GNU Image Manipulation Program

²GNU Network Object Model

³siehe Anhang C

erhöhen sie die Protabilität und Lesbarkeit des Codes. Datentypen in C haben die unangenehme Eigenschaft stark von der verwendeten Plattform abhängig zu sein. So ist zB. ein Pointer auf Alpha 8 Bytes, auf einer Intel 80x86 Architektur aber 4 Bytes groß Folgende Datentypen kommen mit der GLib hinzu[GLib]

char	gchar
short	gshort
long	glong
int	gint
char	gboolean
float	gfloat
double	gdouble
long double	gldouble
unsigned char	guchar
unsigned short	gushort
unsigned long	gulong
unsigned int	guint
void* gpointer	
gint8	8 Bit Integer
guint8	8 Bit unsigned Integer
gint16	16 Bit Integer
guint16	16 Bit unsigned Integer
gint32	32 Bit Integer
guint32	32 Bit unsigned Integer

Quelle:[GLib]

2.1.2 Speicherverwaltung

Die Speicherverwaltung unter C ist eine nie versiegen wollende Quelle für allerlei gemeiner Fehler. Die C Funktion `malloc()` garantiert **nicht** dass auch tatsächlich Speicher allokiert wurde, beendet aber ohne Fehler. Die Funktion `free()` hingegen crasht wenn NULL übergeben wird. Auch hier hilft die GLib aus.[IBM1]

Die Funktion `gpointer g_malloc(gulong size)` garantiert dass auch tatsächlich Speicher allokiert wurde. Wenn die Allokation fehlschlägt bricht das Programm ab. Das klingt hart, ist aber besser als eine Speicherschutzverletzung zu provozieren.

Um Speicher ab einer gewissen Adresse mit einer bestimmten Größe zu allokieren gibt es die Funktion `g_realloc(gpointer mem,gulong size)`. Das ist nützlich um frei gegebene Daten zu retten, aber potentiell gefährlich weil der Speicher vielleicht überschrieben wurde.

Das Pendant zu `free()` in der GLib ist `void g_free(gpointer mem)`. Anders als `free()` stürzt die GLib Funktion nicht bei einem NULL Pointer ab sondern macht nichts.

2.1.3 Strings

Strings sind in C Character Arrays und ihre Verwendung sehr Fehleranfällig. Zu kleine Puffer führen gerne zu Buffer Overflows. Die GLib bringt einen neuen Datentyp `GString` mit der dynamisch wächst um Überläufe zu verhindern und wie folgt definiert ist:

```
struct GString
{
    gchar *str; /* Pointer to the String's current \0-terminated value */
    gint len; /* Current length */
}
```

Quelle: [GLib]

Die Funktion `GString *g_string_new(gchar *init)` erzeugt einen neuen String. Der Initialwert `gchar *init` kann auch NULL sein, sonst wird der Wert von `*init` kopiert. Folgende Funktionen sind eine kleine Auswahl für die Manipulation eines GStrings. Ein komplette Auflistung aller Funktionen findet sich in der API Dokumentation <http://www.gtk.org/api>.

Vorne anhängen

```
GString* g_string_append(GString *string, gchar *val)
GString* g_string_append_c(GString *string, gchar c)
```

Hinten anhängen

```
GString* g_string_prepend(GString *string, gchar *val)
GString* g_string_prepend_c(GString *string, gchar c)
```

An Position einfügen

```
GString* g_string_insert(GString *string, gint pos, gchar *val);
GString* g_string_insert_c(GString *string, gint pos, gchar c);
```

Ausgabe

```
void g_string_sprintf(GString string, gchar *fmt, ...)
void g_string_sprintf_a(GString string, gchar *fmt, ...)
```

Quelle: [GLib]

Dazu ein kleines Beispiel. Folgendes Programm erzeugt einen neuen `GString`, hängt `Hello GLib` vorne und `Hello World` hinten an; Ergänzt den String hinten um ein Rufzeichen und stellt einen Zeilenumbruch an den Anfang. Abschließend wird ein String bestehend aus einem Beistrich und einem Leerzeichen an Position 12 eingefügt und der Puffer des GStrings ausgegeben.

```

#include <stdio.h>
#include <glib.h>

GString *s;

int main()
{
    s=g_string_new(NULL);
    g_string_append(s,"Hello GLib");
    g_string_prepend(s,"Hello World");
    g_string_append_c(s,'!');
    g_string_prepend_c(s,'\n');

    g_string_insert(s,12," ");

    /* Zugriff auf GString Buffer */
    printf("%s \nLength: %d \n",s->str,s->len);
}

```

2.1.4 Timer

Ein sehr nützlicher Teil der GLib sind die Timer Funktionen. Mit folgenden Funktionen kann ein neuer Timer angelegt, gestartet, gestoppt, gelöscht, zurückgesetzt und die verstrichene Zeit ausgelesen werden.

```

GTimer* g_timer_new(void)
void g_timer_start(GTimer *timer);
void g_timer_stop(GTimer *timer);
gdouble g_timer_elapsed(GTimer *timer,gulong *microseconds);
void g_timer_reset(GTimer *timer);
void g_timer_destroy(GTimer *timer);

```

Quelle: [GLib]

Folgendes Beispiel Programm illustriert die Funktionsweise von Timern.

```

#include <stdio.h>
#include <glib.h>

int x;
gdouble secs;
GTimer *timer;

int main()
{
    timer=g_timer_new();
    g_timer_start(timer);

    scanf("%d",&x);

    g_timer_stop(timer);
}

```

```

secs=g_timer_elapsed(timer,NULL);
g_timer_destroy(timer);

printf("%f",secs);
return 0;
}

```

2.1.5 einfach und doppelt verkettete Listen

Die GLib stellt neben neuen Datentypen auch Strukturen bereit. Die beliebtesten sind einfach und doppelt verkettete Listen. Eine Liste mit GLib kann entweder NULL oder vom Typ `GList` bzw. `GSLList` für eine einfach verkettete Liste sein; Wobei jedes Element der Liste vom Typ `GList` bzw. `GSLList` ist. `GList` besteht aus drei Pointern; Einen `next` Pointer, einen `prev` Pointer der bei einfach verketteten Listen fehlt, und einen `data` Pointer. Weil der Speicherbedarf für einen `prev` Pointer verschwindend gering ist, sollte man immer eine doppelt verkettete Liste verwenden.

Es gibt keine spezielle Funktion um eine neue Liste anzulegen. Die Funktionen zur Manipulation einer Liste behandeln einen NULL Pointer als neue Liste. Folgende Funktionen stehen für Manipulation und Zugriff zur Verfügung:

```

struct      GList;

GList*      g_list_append(GList *list, gpointer data);
GList*      g_list_prepend(GList *list, gpointer data);
GList*      g_list_insert(GList *list, gpointer data, gint position);
GList*      g_list_insert_sorted(GList *list, gpointer data, GCompareFunc func);
GList*      g_list_remove(GList *list, gpointer data);
GList*      g_list_remove_link(GList *list, GList *llink);

void        g_list_free(GList *list);
GList*      g_list_alloc(void);
void        g_list_free_1(GList *list);

guint       g_list_length(GList *list);
GList*      g_list_copy(GList *list);
GList*      g_list_reverse(GList *list);
GList*      g_list_sort(GList *list, GCompareFunc compare_func);
GList*      g_list_concat(GList *list1, GList *list2);
void        g_list_foreach(GList *list, GFunc func, gpointer user_data);
void        (*GFunc)(gpointer data, gpointer user_data);

GList*      g_list_first(GList *list);
GList*      g_list_last(GList *list);
#define      g_list_previous(list)
#define      g_list_next(list)
GList*      g_list_nth(GList *list, guint n);
gpointer    g_list_nth_data(GList *list, guint n);
GList*      g_list_find(GList *list, gpointer data);

```

```

GList*      g_list_find_custom(GList *list, gpointer data, GCompareFunc func);
gint       g_list_position(GList *list, GList *llink);
gint       g_list_index(GList *list, gpointer data);

void       g_list_push_allocator(GAllocator *allocator);
void       g_list_pop_allocator(void);

```

Quelle:[GLib]

Die Funktion `g_list_free(GList *list)` gibt den Speicher für die Liste wieder frei, und nur den Speicher für die Liste. Die Elemente auf welche die `data` Pointer zeigen werden nicht mit frei gegeben. Darum muß sich der Programmierer selber kümmern.

2.1.6 Mehr ...

Das war natürlich nicht alles, aber eine erschöpfende Behandlung würde den Umfang dieses Dokuments sprengen. Die GLib hat noch viele nützliche Funktionen und Datenstrukturen zu bieten, wie Hashtables, Balanzierte Bäume usw. Der interessierte Leser möge sich die vorhandene Literatur zu Gemüte führen.

2.2 GTK+ Widgets

Die GTK+ Widget Bibliothek ist zu groß um sie erschöpfend in dieser Arbeit zu behandeln. Dieser Abschnitt stellt die wichtigsten Widgets für die Erstellung einer GTK+ Anwendung vor. Eine vollständige Auflistung aller Widgets und Funktionen ist im GTK+ API auf <http://www.gtk.org/api> zu finden.

2.2.1 Ein einfaches Fenster

Folgender Code startet ein GTK+ Programm und erstellt ein einfaches Fenster.

```

#include <gtk/gtk.h>

int main(int argc, char *argv[])
{
    GtkWidget *fenster, *button;

    gtk_init(&argc, &argv);

    fenster=gtk_window_new(GTK_WINDOW_TOPLEVEL);
    gtk_window_set_title(GTK_WINDOW(fenster), "Beispiel 1");
    gtk_window_set_default_size(GTK_WINDOW(fenster), 320, 220);
    gtk_signal_connect(GTK_OBJECT(fenster),
                       "delete_event",
                       GTK_SIGNAL_FUNC(gtk_main_quit),
                       NULL);

    button=gtk_button_new_with_label("Klick mich");

    gtk_container_add(GTK_CONTAINER(fenster), button);

```

```

gtk_widget_show(button);
gtk_widget_show(fenster);

gtk_main();
}

```

Um die GTK+ Widget Bibliothek verwenden zu können, ist das Einbinden von `gtk.h` notwendig. Das einzige Widget in diesem Beispiel ist ein Fenster und vom Typ `GtkWidget`. Nach der Deklaration erfolgt die Initialisierung.

Die GTK+ Funktionen sind alle gleich aufgebaut, sie folgen dem einfachen Muster `gtk_Widgettyp_Funktionalität`. Die erste Funktion erstellt ein neues Toplevel Fenster. Mit der zweiten Funktion wird der Titel gesetzt, der dann in der Fensterleiste erscheint. Die dritte Funktion legt die Anfangsgröße des Fensters mit 320 x 220 Pixel fest.

```

gtk_signal_connect(GTK_OBJECT(fenster),
                  "delete_event",
                  GTK_SIGNAL_FUNC(gtk_main_quit),
                  NULL);

```

...

```

gtk_main();

```

GTK+ ist eine ereignisbasierte GUI Bibliothek. Die Anweisung `gtk_main()` startet eine Endlosschleife die auf Ereignisse wartet. Alles was mit der GUI zu tun hat ist ein Ereignis, wie das Anklicken oder eine Mausbewegung. Wenn ein Ereignis eintritt sendet das entsprechende Widget ein Signal aus. Diese Signale können vom Programmierer mit Funktionen verknüpft werden, sogenannten **Callbacks**.

In diesem Beispiel wird das `delete_event` mit der Funktion `gtk_main_quit` verknüpft. Der Button in diesem Beispiel hat keine Funktionalität. Ohne die Verknüpfung würde zwar das Fenster geschlossen, aber das Programm weiterlaufen. Die `gtk_main_quit` Funktion ist Teil von GTK+.[Fischer]

```

gtk_container_add(GTK_CONTAINER(fenster),button);
gtk_widget_show(button);
gtk_widget_show(fenster);

```

Die erste Funktion packt den Button in das Containerwidget Fenster. Damit ein Widget zu sehen ist, muß es mit `gtk_widget_show` sichtbar gemacht werden.



2.2.2 Container und Frames

Mit Ausnahme des Hauptfensters müssen alle `GtkWidgets` in Container gepackt werden. Das wichtigste Containerwidget ist das Hauptfenster selbst. Das Hauptfenster kann aber nicht mit beliebig vielen `gtk_container_add()` Aufrufen vollgepackt werden. Um Struktur in die Oberfläche zu bringen gibt es drei wichtige Containerwidgets.

Boxen

Mit Boxen läßt sich ein Containerwidget entweder horizontal oder vertikal einteilen. Die entsprechenden Widgets heißen `GtkHBox` und `GtkVBox`. Die Boxen selbst sind nicht sichtbar können aber durch Abstände zwischen den Widgets Einfluß auf die Darstellung nehmen. Die Funktionen für das Erstellen einer horizontalen bzw. vertikalen Box sind bis auf das `h` oder `v` im Namen gleich.

```
GtkWidget *gtk_hbox_new(gint homogeneous, gint spacing)
```

Mit dem Parameter `homogeneous` läßt sich einstellen ob alle Widgets gleich viel Platz in der Box bekommen. Möglich sind die Werte `TRUE` oder `FALSE`. Mit `TRUE` werden alle Widgets auf die Größe des größten Widgets in der Box ausgedehnt. Der Parameter `spacing` bestimmt den Zwischenraum. [Fischer]

Um die Box mit Elementen zu füllen verwendet man die Packfunktionen. Der Unterschied zwischen den beiden Funktionen ist ob mit dem Füllen von vorne (links bzw. oben) oder von hinten (rechts bzw. unten) begonnen wird. Mit `expand`, `fill` und `padding` läßt sich die Darstellung und der Innenabstand bestimmen.

```
void gtk_box_pack_start(GtkBox *box,  
                        GtkWidget widget,  
                        gint expand,  
                        gint fill,  
                        gint padding);
```

```
void gtk_box_pack_end(GtkBox *box,
```



```

GtkWidget widget,
gint expand,
gint fill,
gint padding);

```

[Fischer]

Mit diesem Code werden 5 Buttons in einer HBox verteilt. Die Elemente eines Containers sollten immer vor dem Container mit `gtk_widget_show` sichtbar gemacht werden. Anderfalls können Flimmereffekte entstehen.

```

GtkWidget *box;
GtkWidget *buttons[5];

gtk_init(&argc,&argv);

/* Fenster und Callback mit gtk_main_quit erstellen */

box=gtk_box_new(TRUE,10);
for(i=0;i<5;i++)
{
    buttons[i]=gtk_button_new_with_label("Button");
    gtk_box_pack_start(GTK_BOX(box),buttons[i],TRUE,TRUE,10);
    gtk_widget_show(buttons[i]);
}

gtk_container_add(GTK_CONTAINER(fenster),box);
gtk_widget_show(box);
gtk_widget_show(fenster);

gtk_main();

```



Tabellen

Tabellen sind mächtiger als Boxen und teilen ein Containerwidget in Zeilen und Spalten. Die Gitternetzlinien der Tabelle werden horizontal und vertikal mit 0 beginnend durchnummeriert. Einem Widget wird ein Platz in der Tabelle zugewiesen indem man den linken und rechten sowie den oberen und unteren Rand bestimmt. So kann ein Widget auch mehr als nur eine Zelle belegen. Die Funktion zum Anlegen einer Tabelle verlangt die Anzahl der Zeilen, die Anzahl der Spalten und einen `homogeneous` Wert. Letzterer bestimmt wieder ob alle Felder der Tabelle so groß wie das größte Widget in der Tabelle sein sollen.

```

GtkWidget *gtk_table_new(gint rows, gint cols, gint homogeneous)

```

Um ein Widget in einer Tabelle zu platzieren gibt es eine umfangreicher Funktion, und eine Funktion für Default Werte. Mit der umfangreichen Funktion lassen sich auch Werte für Padding und Darstellung einstellen.

```

void gtk_table_attach(GtkTable *table,
                    GtkWidget *child,
                    gint left_attach, gint right_attach,
                    gint top_attach, gint bottom_attach,
                    gint xoptions, gint yoptions,
                    gint xpadding, gint ypadding)

void gtk_table_attach_defaults(GtkTable *table,
                              GtkWidget *child,
                              gint left_attach, gint right_attach,
                              gint top_attach, gint bottom_attach)

```

[Fischer]

Nachstehendes Beispiel erstelle eine 3x3 Tabelle und füllt sie mit Buttons.

```

#include <gtk/gtk.h>

int main(int argc, char *argv[])
{
    GtkWidget *fenster, *table, *button1, *button2, *button3, *button4;

    gtk_init(&argc, &argv);

    fenster=gtk_window_new(GTK_WINDOW_TOPLEVEL);
    gtk_signal_connect(GTK_OBJECT(fenster), "delete_event",
                      GTK_SIGNAL_FUNC(gtk_main_quit), NULL);
    gtk_window_set_title(GTK_WINDOW(fenster), "Beispiel3");
    gtk_window_set_default_size(GTK_WINDOW(fenster), 320, 220);

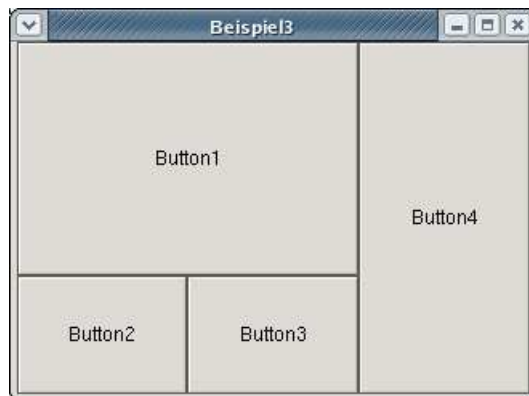
    table=gtk_table_new(3, 3, TRUE);
    button1=gtk_button_new_with_label("Button1");
    button2=gtk_button_new_with_label("Button2");
    button3=gtk_button_new_with_label("Button3");
    button4=gtk_button_new_with_label("Button4");

    gtk_table_attach_defaults(GTK_TABLE(table), button1, 0, 2, 0, 2);
    gtk_table_attach_defaults(GTK_TABLE(table), button2, 0, 1, 2, 3);
    gtk_table_attach_defaults(GTK_TABLE(table), button3, 1, 2, 2, 3);
    gtk_table_attach_defaults(GTK_TABLE(table), button4, 2, 3, 0, 3);

    gtk_container_add(GTK_CONTAINER(fenster), table);
    gtk_widget_show(button1);
    gtk_widget_show(button2);
    gtk_widget_show(button3);
    gtk_widget_show(button4);

    gtk_widget_show(table);
    gtk_widget_show(fenster);
    gtk_main();
}

```



Frames

Rahmen sind ein gutes Mittel um Elemente einer graphischen Benutzeroberfläche zu gruppieren. Frames können als Containerwidget verwendet werden, aber nur ein Element aufnehmen. Ein neuer Rahmen mit Beschriftung wird nach bekanntem Muster erstellt.

```
GtkWidget *gtk_frame_new(const gchar *label)
```

Rahmen lassen sich den eigenen Wünschen anpassen. Mit folgenden drei Funktionen lassen sich die Beschriftung und deren Ausrichtung ändern, sowie das Aussehen der Umrandung.

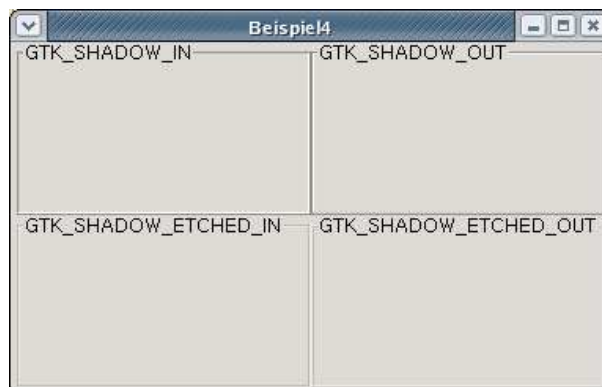
```
void gtk_frame_set_label(GtkFrame *frame, const char* label)
```

```
void gtk_frame_set_label_align(GtkFrame *frame,
                               gfloat xalign, gfloat yalign)
```

```
void gtk_frame_set_shadow_type(GtkFrame *frame, GtkShadowType type)
```

Für die Umrandung kommen 5 Typen in Frage:

- GTK_SHADOW_NONE
- GTK_SHADOW_IN
- GTK_SHADOW_OUT
- GTK_SHADOW_ETCHED_IN
- GTK_SHADOW_ETCHED_OUT



2.2.3 Buttons und Combobox

Buttons gehören zur Grundausstattung jeder GUI Bibliothek. In GTK+ sind sie in verschiedenen Varianten vertreten. Die einfachste Form eines Buttons wurde schon im Beispiel 1 verwendet.

Toggle- und Checkbuttons

Etwas interessanter ist der ToggleButton. Im Prinzip ist ein ToggleButton ein normaler Button der entweder gedrückt oder nicht gedrückt ist.

Mit `GTK_TOGGLE_BUTTON(tglbutton_widget)->status` kann der aktuelle Zustand ermittelt werden. Unabhängig vom aktuellen Status kann ein Button mit `void gtk_toggle_button_toggled(GtkToggleButton *tglbutton)` umgeschaltet werden. Checkbuttons sind ähnlich trivial wie normale Buttons. Genau genommen sind Checkboxes besser ToggleButtons. Für den den Zugriff und die Manipulation sind die gleichen Funktionen wie für ToggleButtons vorgesehen.

Radiobuttons

Radiobuttons sind trickreicher als Toggle- und Checkbuttons. Es kann immer nur ein Radiobutton eine Gruppe aktiv sein. Die Gruppenzugehörigkeit wird in GTK+ auf eine GSList abgebildet. Die Funktionen für das Erstellen eines Buttons und Verknüpfen mit einer Gruppe sind:

```
GtkWidget *gtk_radio_button_new_with_label(GSList *gruppe,
                                           gchar *label);
GSList *gtk_radio_button_group(GtkRadioButton *button);
```

[Fischer]

In diesem Beispiel wird die Verwendung der vorgestellten Funktionen demonstriert.

```
#include <glib.h>
#include <gtk/gtk.h>

int main(int argc, char *argv[])
{
    GSList *liste;
    GtkWidget *fenster, *box, *radio1, *radio2;

    gtk_init(&argc, &argv);

    fenster = gtk_window_new(GTK_WINDOW_TOPLEVEL);
    gtk_signal_connect(GTK_OBJECT(fenster), "delete_event",
                      GTK_SIGNAL_FUNC(gtk_main_quit), NULL);
    gtk_window_set_title(GTK_WINDOW(fenster), "Beispiel5");
    gtk_window_set_default_size(GTK_WINDOW(fenster), 320, 120);

    box = gtk_vbox_new(TRUE, 5);
```

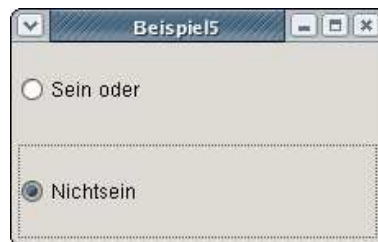
```

radio1=gtk_radio_button_new_with_label(NULL,"Sein oder");
gtk_box_pack_start(GTK_BOX(box),radio1,TRUE,TRUE,0);
liste = gtk_radio_button_group(GTK_RADIO_BUTTON(radio1));

radio2=gtk_radio_button_new_with_label(liste,"Nichtsein");
gtk_box_pack_start(GTK_BOX(box),radio2,TRUE,TRUE,0);
liste = gtk_radio_button_group(GTK_RADIO_BUTTON(radio2));

gtk_container_add(GTK_CONTAINER(fenster),box);
gtk_widget_show(radio1);
gtk_widget_show(radio2);
gtk_widget_show(box);
gtk_widget_show(fenster);
gtk_main();
}

```



2.2.4 Textelemente

Zur Grundausstattung jeder Widget Bibliothek gehören Textelemente. Dazu zählen einerseits Labels, die ausser ihrer nackten Existenz keine weitere Funktionalität haben, und Eingabefelder aller Art.

Label

Labels sind trivial und kaum eine Erwähnung Wert. Sie finden auch Verwendung in anderen Widgets die aus mehreren Komponenten zusammengesetzt sind. Die Beschriftung eines Frames beispielsweise wird mit einem Label realisiert.

Eingabefelder

Einzeilige und mehrzeilige Eingabefelder finden sich in vielen Benutzeroberflächen, und dürfen auch in GTK+ nicht fehlen. Funktionen bilden das Pendant zu den Get- und Setmethoden einer Objektorientierten Programmiersprache. Hier exemplarisch für ein einzeiliges Textfeld `GtkEntry`.

```

GtkWidget *gtk_entry_new();
GtkWidget *gtk_entry_new_with_max_length(guint16 max);

void gtk_entry_set_text(GtkEntry *entry, const gchar *text);
void gtk_entry_append_text(GtkEntry *entry, const gchar *text);
void gtk_entry_prepend_text(GtkEntry *entry, const gchar *text);
void gtk_entry_set_position(GtkEntry *entry, gint position);

gchar* gtk_entry_get_text(GtkEntry *entry);

[GLib]

```

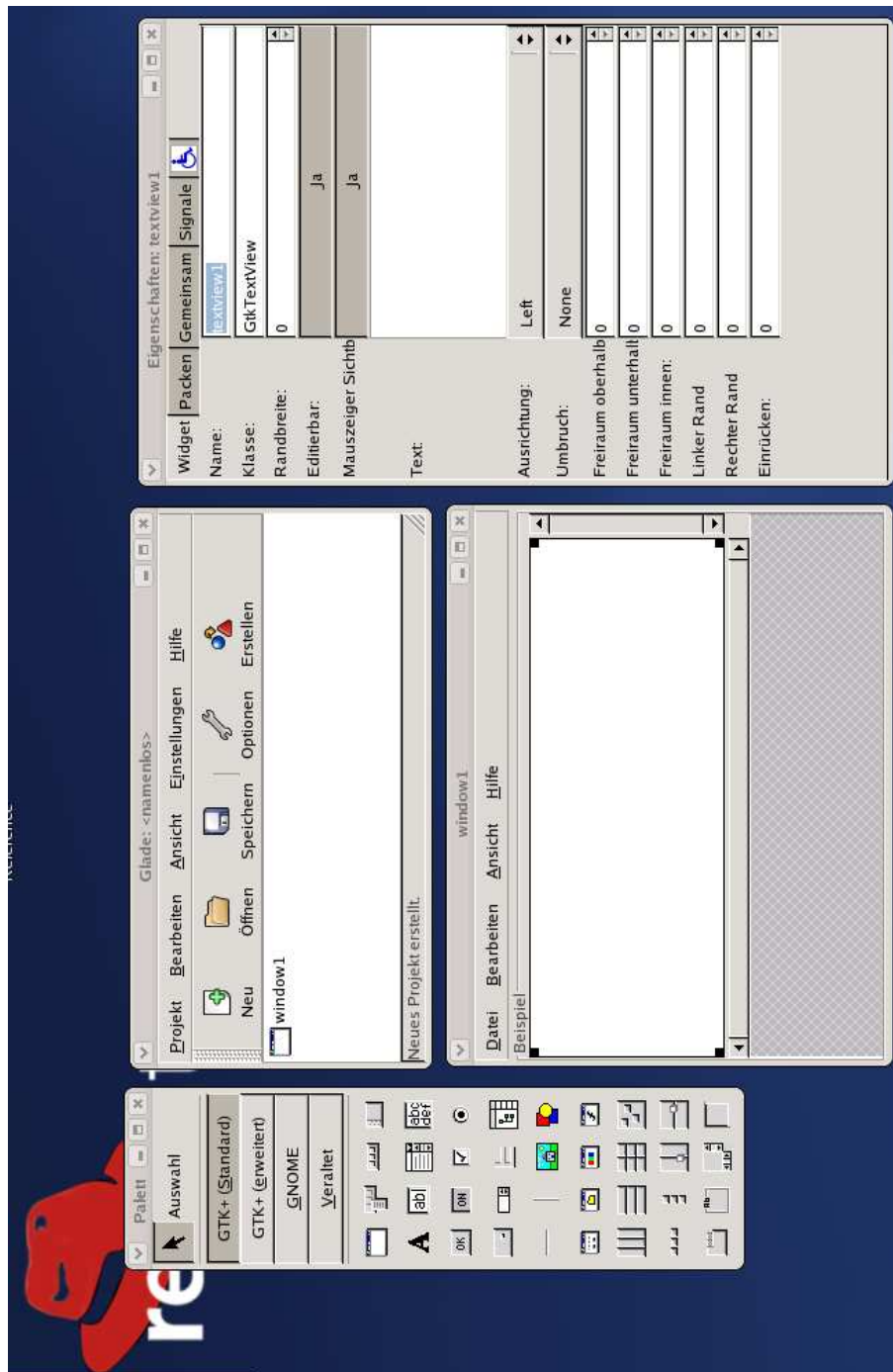


2.3 Beispielanwendung

Nach dem langen Grundlagenteil geht es jetzt in die Praxis. Eine Beispielanwendung mit GTK+ Benutzeroberfläche soll von Linux nach Windows portiert werden. Zur Demonstration dient eine kleine Anwendung, die nicht viel leistet. Das Beispiel liesse sich nach belieben ausbauen, was aber die Programmlogik im Verhältnis zur Benutzeroberfläche unnötig aufblähen würden. Der Fokus liegt jedoch auf der Benutzeroberfläche.

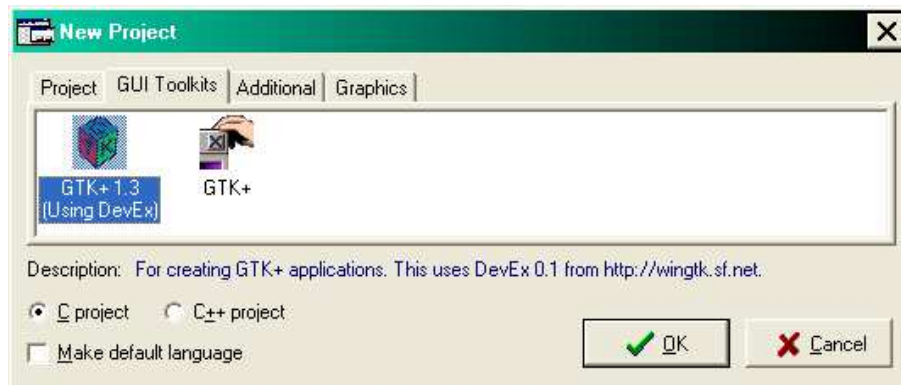
2.3.1 Tools für die Entwicklung unter Linux

Für die Entwicklung mit GTK+ genügt im Prinzip ein Texteditor, die GTK+ Bibliotheken und ein C Compiler. Komfortabler ist hingegen ein GUI Builder wie Glade.



2.3.2 Tools für die Entwicklung unter Windows

Unter Windows bietet sich die freie IDE Bloodshed Dev-C++ an. Die aktuelle und ältere Version sind von der Bloodshed Homepage <http://www.bloodshed.net> zu haben. Für die GTK+ Entwicklung mit Dev-C++ gibt es ein eigenes Package auf der Website des GIMP Projekts <http://www.gimp.org/>.



Ein neues GTK+ Projekt mit Dev-C++. Ein kleiner Bug plagt das GTK+ Paket für Dev-C++. Vor dem erstmaligen Schliessen des Projekts müssen die Projekteinstellungen gesichert werden. Einfaches **Datei / Speichern** sichert nur das Sourcefile. Wird das Projekt wieder geöffnet beklagt sich der Linker. Die Lösung ist **Datei / Alles Speichern**.

2.3.3 Beispielprogramm

Folgender Code ist ein Beispielprogramm in C mit GTK+ Oberfläche. Wie erwähnt ist die Funktionalität bescheiden, es rechnet nur Dezimalzahlen in Binärzahlen um und umgekehrt. Die GUI besteht aus den typischen Komponenten wie Fenster, Menü, horizontale und vertikale Boxen, Textfelder, Buttons, Rahmen und Callbacks.

```
#include <math.h>
#include <glib.h>
#include <gtk/gtk.h>

int BinToInt(char *bin);
char *IntToBin(int value);
char *IntToStr(int value);
gint toBin(GtkWidget widget,GdkEvent event,gpointer data);
gint toDec(GtkWidget widget,GdkEvent event,gpointer data);

GtkWidget *dec_text,*xBase_text;

/* Menu */
static GtkItemFactoryEntry menu[]={
    {"/_Datei",NULL,NULL,0,"<Branch>"},
    {"/Datei/Beenden","<control>Q",gtk_main_quit,0,NULL}
};
```



```

int main(int argc, char *argv[])
{

    GtkWidget *fenster;
    GtkWidget *main_box, *dec_box, *xBASE_box, *frame_base10, *frame_baseX, *frame_box;
    GtkWidget *dec_button, *xBASE_button;

    GtkWidget *items;
    GtkAccelGroup *accl;
    GtkWidget *menu_bar;
    gint n_menu=2;

    gtk_init(&argc, &argv);

    /* create Window */
    fenster=gtk_window_new(GTK_WINDOW_TOPLEVEL);
    gtk_window_set_title(GTK_WINDOW(fenster), "xBASE");
    gtk_window_set_default_size(GTK_WINDOW(fenster), 320, 100);
    gtk_signal_connect(GTK_OBJECT(fenster), "delete_event",
                      GTK_SIGNAL_FUNC(gtk_main_quit), NULL);

    /* create Menu */
    accl=gtk_accel_group_new();
    items=gtk_item_factory_new(GTK_TYPE_MENU_BAR, "<main>", accl);
    gtk_item_factory_create_items(items, n_menu, menu, NULL);
    gtk_accel_group_attach(accl, GTK_OBJECT(fenster));
    menu_bar=gtk_item_factory_get_widget(items, "<main>");

    /* decimal input */
    dec_text=gtk_entry_new();
    dec_button=gtk_button_new_with_label("Umrechnen");
    gtk_signal_connect(GTK_OBJECT(dec_button), "clicked",
                      GTK_SIGNAL_FUNC(toBin), NULL);

    /* xBase input */
    xBASE_text=gtk_entry_new();
    xBASE_button=gtk_button_new_with_label("Dezimal");
    gtk_signal_connect(GTK_OBJECT(xBASE_button), "clicked",
                      GTK_SIGNAL_FUNC(toDec), NULL);

    /* Box for decimal side */
    dec_box=gtk_vbox_new(TRUE, 0);
    gtk_box_pack_start(GTK_BOX(dec_box), dec_text, FALSE, FALSE, 5);
    gtk_box_pack_start(GTK_BOX(dec_box), dec_button, FALSE, FALSE, 5);

    /* Box for xBase side */
    xBASE_box=gtk_vbox_new(TRUE, 0);
    gtk_box_pack_start(GTK_BOX(xBASE_box), xBASE_text, FALSE, FALSE, 5);
    gtk_box_pack_start(GTK_BOX(xBASE_box), xBASE_button, FALSE, FALSE, 5);

```

```

/* Frame for decimal side */
frame_base10=gtk_frame_new("Dezimal");
gtk_frame_set_shadow_type(GTK_FRAME(frame_base10),GTK_SHADOW_ETCHED_IN);
gtk_container_add(GTK_CONTAINER(frame_base10),dec_box);

/* Frame for xBase */
frame_baseX =gtk_frame_new("Binaer");
gtk_frame_set_shadow_type(GTK_FRAME(frame_baseX),GTK_SHADOW_ETCHED_IN);
gtk_container_add(GTK_CONTAINER(frame_baseX),xBase_box);

/* Box for Frames */
frame_box=gtk_hbox_new(TRUE,0);
gtk_box_pack_start(GTK_BOX(frame_box),frame_base10,TRUE,TRUE,5);
gtk_box_pack_start(GTK_BOX(frame_box),frame_baseX,TRUE,TRUE,5);

/* Box for frames */
main_box=gtk_vbox_new(FALSE,0);
gtk_box_pack_start(GTK_BOX(main_box),menu_bar,FALSE,TRUE,0);
gtk_box_pack_start(GTK_BOX(main_box),frame_box,FALSE,TRUE,0);

gtk_container_add(GTK_CONTAINER(fenster),main_box);
gtk_widget_show_all(fenster);

gtk_main();
return 0;
}

int BinToInt(char *bin)
{
    double value=0.0;
    int i,length=strlen(bin);

    for(i=length-1;i>=0;i--)
    {
        if(bin[i]=='1') value+=pow(2.0,length-1-i);
    }

    return (int)value;
}

char *IntToBin(int value)
{
    int i,length;
    char *bin=NULL;

    /* log(0) doesn't work */
    if(value==0) return "0";

    length=1+(int)floor(log(value)/log(2));

```

```

bin=(char *)g_malloc(sizeof(char)*length+1);

for(i=0;i<length;i++)
{
    if( value-pow(2,length-1-i)>=0 )
    {
        value=value-pow(2,length-1-i);
        bin[i]='1';
    }
    else {
        bin[i]='0';
    }
}
bin[length]='\0';
return bin;
}

char *IntToStr(int value)
{
    int i,tmp,length;
    char *str;

    length=(int)log10(value)+1;
    str=(char *)g_malloc(sizeof(char)*length+1);

    printf("\n");
    tmp=0;
    for(i=length-1;i>=0;i--)
    {
        tmp=(int)value/pow(10,i);
        value-=tmp*pow(10,i);
        printf("%d ",tmp);

        if(tmp==0) str[length-1-i]='0';
        if(tmp==1) str[length-1-i]='1';
        if(tmp==2) str[length-1-i]='2';
        if(tmp==3) str[length-1-i]='3';
        if(tmp==4) str[length-1-i]='4';
        if(tmp==5) str[length-1-i]='5';
        if(tmp==6) str[length-1-i]='6';
        if(tmp==7) str[length-1-i]='7';
        if(tmp==8) str[length-1-i]='8';
        if(tmp==9) str[length-1-i]='9';

    }
    str[length]='\0';

    return str;
}

```

```

gint toBin(GtkWidget widget,GdkEvent event,gpointer data)
{
    char *value=IntToBin( atoi(GTK_ENTRY(dec_text))) );
    gtk_entry_set_text(GTK_ENTRY(xBase_text),value);
    return 0;
}

gint toDec(GtkWidget widget,GdkEvent event,gpointer data)
{
    int intval=BinToInt( GTK_ENTRY(xBase_text)) );
    gtk_entry_set_text(GTK_ENTRY(dec_text),IntToStr(intval));
    return 0;
}

```

Mit diesem Code lässt sich unter Linux mit dem GNU C Compiler und GTK+ ein funktionierendes Programm erstellen.

```

[developer@localhost GTK]# gcc bsp.c `gtk-config --cflags --libs` -o bsp
[developer@localhost GTK]# ./bsp

```



Der Lohn für die Verwendung von Widget Toolkits ist Portabilität. Der obige Code kann 1:1 übernommen, und per Copy&Paste in ein neues GTK+ Projekt in Dev-C++ unter Windows eingefügt werden. Ohne Änderung des Sourcecodes kann man damit eine Win32 Anwendung mit graphischer Benutzeroberfläche erzeugen.



2.3.4 Portable Anwendungen

Das kleine Beispiel sollte die Verwendung von GTK+ für die Entwicklung portabler Programme demonstrieren. Dass portable Programme mit graphischer Benutzeroberfläche nicht nur eine akademische Spielerei sind demonstriert GIMP

GIMP

Das GNU Image Manipulation Program zählt zu den Vorzeigeprojekten der Opensource Community. GIMP ist ein hochwertiges Grafiktool, das sich nicht vor kommerziellen Konkurrenten wie Photoshop verstecken muss. Seit kurzem ist GIMP 2.0 verfügbar. Die Sourcefiles sind auf <http://www.gimp.org/downloads> verfügbar.

Sodipodi

Ein relativ junger Stern am Opensource Himmel ist Sodipodi; Ein Werkzeug für das Erstellen von SVG Grafiken. Die aktuelle Version ist 0.34 und wurde unter der GNU General Public License veröffentlicht. Sodipodi wurde in C geschrieben und verwendet GTK+ für die graphische Benutzeroberfläche. Die Sodipodi Homepage ist unter <http://www.sodipodi.com> erreichbar.



Sodipodi 0.34 unter Linux

Kapitel 3

Qt

Dieses Kapitel über Qt ist extrem kurz. Die Gründe dafür sind mangelnde C++ Kenntnisse und ausreichend Literatur für den interessierten Leser.

3.1 Über Qt

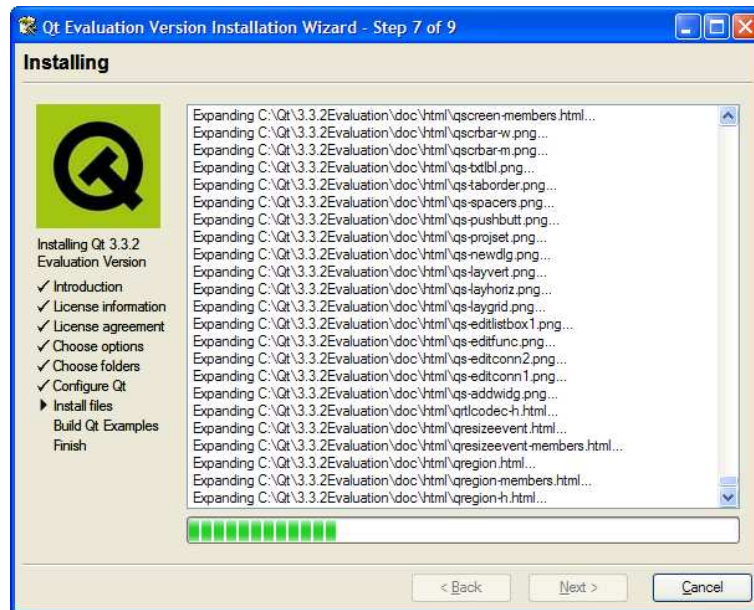
Qt ist eines der bekanntesten, wenn nicht das bekannteste Widget Toolkit unter Linux. Diese Popularität hat es dem Aushängeschild KDE zu verdanken. Qt ist ein kommerzielles Produkt der Firma Trolltech¹, und der Einsatz von Qt für KDE war daher nicht unumstritten. Der Gedanke ein kommerzielles Widget Toolkit für das Desktop Environment eines freien Betriebssystems zu verwenden schmeckte viele Entwicklern nicht. Inzwischen ist Qt für die Entwicklung freier Software unter der GPL und Q Public License verfügbar. Für die Entwicklung kommerzieller Software mit Qt muß man aber stolze Preise von einigen hundert bis zu tausenden Euros bezahlen. Näheres ist auf der Trolltech Homepage <http://www.trolltech.com/products/qt/pricing.html> zu erfahren.

3.1.1 Installation

Windows

Die Installation von Qt unter Windows erfolgt mittels eines Installers. Das Qt Paket arbeitet mit den Entwicklungsumgebungen Microsoft Visual C++ und Microsoft Visual Studio .NET zusammen. Das Qt Paket bringt auch gleich den Oberflächendesigner Qt Designer mit.

¹<http://www.trolltech.com>

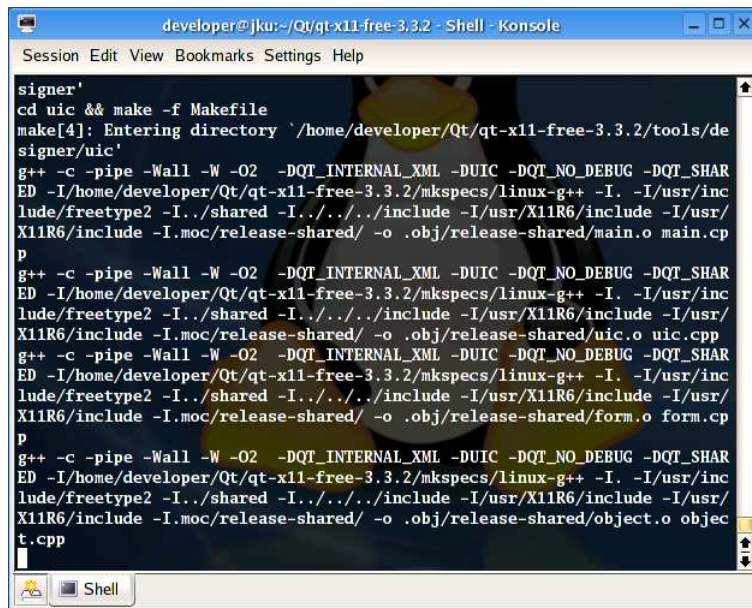


Qt Installation unter Windows XP

Linux

Sollten die Pakete für die Qt Entwicklung nicht installiert sein, kann man die aktuelle Qt Version für Linux/X11 von der Trolltech Homepage beziehen. Das tar Archiv für die aktuelle Version 3.3.2 ist 16954 kB groß und enthält die Quellen sowie die Dokumentation. Die Installation läuft nach bekanntem Muster und dauert eine ganze Weile.

```
[developer@localhost Qt]$ tar -xvzf qt-x11-free-3.3.2.tar.gz
[developer@localhost Qt]$ cd qt-x11-free-3.3.2
[developer@localhost Qt]$ ./configure
[developer@localhost Qt]$ make
```



```
developer@jku:~/Qt/qt-x11-free-3.3.2 - Shell - Konsole
Session Edit View Bookmarks Settings Help

signer'
cd uic && make -f Makefile
make[4]: Entering directory `/home/developer/Qt/qt-x11-free-3.3.2/tools/de
signer/uic'
g++ -c -pipe -Wall -W -O2 -DQT_INTERNAL_XML -DUIC -DQT_NO_DEBUG -DQT_SHAR
ED -I/home/developer/Qt/qt-x11-free-3.3.2/mkspecs/linux-g++ -I. -I/usr/inc
lude/freetype2 -I../shared -I../../../../include -I/usr/X11R6/include -I/usr/
X11R6/include -I.moc/release-shared/ -o .obj/release-shared/main.o main.cp
p
g++ -c -pipe -Wall -W -O2 -DQT_INTERNAL_XML -DUIC -DQT_NO_DEBUG -DQT_SHAR
ED -I/home/developer/Qt/qt-x11-free-3.3.2/mkspecs/linux-g++ -I. -I/usr/inc
lude/freetype2 -I../shared -I../../../../include -I/usr/X11R6/include -I/usr/
X11R6/include -I.moc/release-shared/ -o .obj/release-shared/uic.o uic.cpp
g++ -c -pipe -Wall -W -O2 -DQT_INTERNAL_XML -DUIC -DQT_NO_DEBUG -DQT_SHAR
ED -I/home/developer/Qt/qt-x11-free-3.3.2/mkspecs/linux-g++ -I. -I/usr/inc
lude/freetype2 -I../shared -I../../../../include -I/usr/X11R6/include -I/usr/
X11R6/include -I.moc/release-shared/ -o .obj/release-shared/form.o form.cp
p
g++ -c -pipe -Wall -W -O2 -DQT_INTERNAL_XML -DUIC -DQT_NO_DEBUG -DQT_SHAR
ED -I/home/developer/Qt/qt-x11-free-3.3.2/mkspecs/linux-g++ -I. -I/usr/inc
lude/freetype2 -I../shared -I../../../../include -I/usr/X11R6/include -I/usr/
X11R6/include -I.moc/release-shared/ -o .obj/release-shared/object.o objec
t.cpp
```

Die Installation von Qt dauert eine halbe Ewigkeit. Es ist daher ratsam die mitgelieferten Bibliotheken der eigenen Distribution zu verwenden. Sollte man dennoch spezielle Features einer neueren Version brauchen muss man Zeit für ein Update investieren.

Als Entwicklungsumgebung unter Linux biete sich das beliebte KDevelop an. Sofern noch nicht installiert ist die momentan aktuelle stabile Version 3.0.3 auf der KDevelop Homepage <http://www.kdevelop.org> zu haben.

3.2 Grafische Benutzeroberflächen mit Qt

3.2.1 Allgemeines

Wie schon erwähnt ist Qt eine C++ Klassenbibliothek für die Entwicklung graphischer Benutzeroberflächen. Der Umfang von Qt ist im Laufe der Zeit gewachsen. Im Rahmen des Mono Projekts findet Qt dank C# Language Bindings unter dem Namen Qt# Verwendung. Ähnlich wie GTK+ beschränkt sich Qt nicht auf die GUI Entwicklung, sondern bringt auch Hilfsklassen wie Stacks und Hashtables mit. [LinuxMagQt].

Qt Oberflächen lassen sich mit dem Builder Tool Qt Designer einfach zusammenklicken. Assistenten führen auf Wunsch durch verschiedene Dialoge und schreiben im Hintergrund den Quellcode. Der Qt Designer ist im moment noch eine Standalone Anwendung. Ab Version 4 von Qt soll sich der Qt Designer in IDEs wie Visual Studio .NET einbinden. [ct1004]

Kapitel 4

.NET und Mono

Was macht ein Kapitel über .NET in einer Seminararbeit zu Linux ? Microsoft vermarktet eine Reihe von Technologien und Strategien unter dem Sammelbegriff .NET . Eine wichtige Eigenschaft ist Portabilität. Assemblies¹ die von einem .NET Compiler erstellt wurden können auf jeder Plattform mit .NET verwendet werden. Selbstverständlich ist die .NET Laufzeitumgebung nur für Windows verfügbar. Weil aber .NET eine ganze Reihe offener Standards wie XML und SOAP² verwendet und sowohl C#³ als auch die Common Language Infrastructure⁴ ECMA⁵ Standards sind, ist eine Implementierung von .NET auch für andere Betriebssysteme möglich. Die bekannteste Implementierung für Linux, Unix und Windows ist Mono⁶.

4.1 Ein paar Worte zu .NET

.NET umfasst alle Produkte und Technologien, die Managed Code nutzen und Web Services zur Konnektivität nutzen. Diese Produkte tragen unser .NET Connected Logo. ... Eine Ausnahme haben wir definiert für solche Produkte, die wir als Plattform-Produkte für andere .NET Produkte verstehen.

John Montgomery, Director of Product Management for the Microsoft Developer Platform

[iX0504]

Um mit Mono arbeiten zu können ist es wichtig einen Überblick über .NET zu haben. Der Weg den Microsoft mit .NET eingeschlagen hat ist komplementär zu Java. Während Java, dank der Laufzeitumgebung, eine Sprache für viele Plattformen ist, führt .NET viele Sprachen in ein System zusammen.

¹in dt. gerne als Baustein übersetzt

²Simple Object Access Protocol

³www.ecma-international.org/publications/standards/Ecma-334.HTML

⁴www.ecma-international.org/publications/standards/Ecma-335.HTML

⁵European Computer Manufacturers Association

⁶span. für Affe

Die Sprache C# wurde speziell für .NET entwickelt und die gängigen Sprachen für Windowsprogrammierung wie C++ und Visual Basic auf .NET getrimmt⁷.

4.1.1 Die Common Language Specification

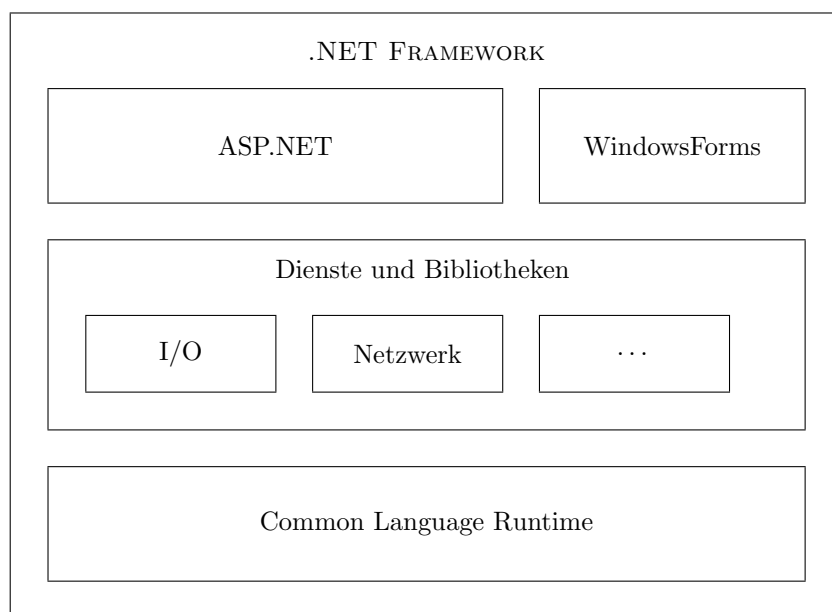
Um viele Sprachen in .NET zusammenzuführen muss eine gemeinsame Basis für die verwendeten Datentypen gefunden werden. Diese Basis ist in .NET das Common Type System. Daraus lassen sich alle weiteren Datentype ableiten.

Wie erwähnt verwenden mehrere Sprachen das .NET Framework, und nicht jede dieser Sprachen unterstützt alle möglichen Datentypen. Der kleinste gemeinsame Nenner, den alle .NET Sprachen beherrschen müssen wird die Common Language Specification genannt. Folgendes Beispiel illustriert die Situation:

In Visual Basic gibt es keine vorzeichenlose Datentypen. Einen `unit` gibt es in VB nicht, in C# und C++ aber schon. Daher müssen alle Klassen die der CLS genügen wollen, nach außen vorzeichenbehaftete Datentypen verwenden.[Hanisch]

4.1.2 Das .NET Framework

Das Kernstück von .NET bildet das .NET Framework. Das Framework besteht aus mehreren Teilen die im folgenden besprochen werden.



Quelle:[DevGuide]

Die Common Language Runtime

Die Laufzeitumgebung von .NET selbst besteht auch wieder aus verschiedenen Teilen. Soll ein .NET Programm ausgeführt werden, muß im ersten Schritt die

⁷siehe CLS

Laufzeitumgebung gestartet werden. Das Programm liegt in einer Intermediate Language vor und muß vom Class Loader aufgelöst werden. Der Class Loader löst Verweise auf und lädt die notwendige Komponenten.

Im zweiten Schritt wird der Code vom Class Loader mit einem Just In Time (JIT) Compiler in ausführbaren Code übersetzt. Der JIT Compiler ist Teil der CLR. Die Aufteilung in zwei Kompilervorgänge, von Quellcode in IL und von IL in ausführbaren Code erlaubt eine Optimierung vor Ort für das jeweilige System.

Ein .NET Programm läuft unter der Kontrolle der CLR, daher auch der Begriff **managed Code**. Die CLR übernimmt das Laden von Komponenten die erst zur Laufzeit benötigt werden sowie die Speicherverwaltung. Weiter regelt die CLR die Ausführung mehrerer unabhängiger .NET Anwendungen damit nicht ein .NET Programm auf den Speicherbereich eines Anderen zugreifen kann.

Das Service Framework

Das .NET Framework ist sehr umfangreich, es hat mehr als 7000 Klassen und 150.000 Methoden. [Hanisch] Im Service Framework sind die Klassen und Methoden die von einer .NET Anwendung benutzt werden können. Die Klassen sind in Namespaces strukturiert, die jeweils einen bestimmten Bereich von .NET abdecken. Das Service Framework ist mehr als nur eine große Klassenbibliothek, es erlaubt auch den Zugriff auf Dienste des Betriebssystems. [DevGuide]

4.1.3 ASP.NET und WindowsForms

Die Active Server Pages und WindowsForms dienen der Erstellung von Benutzeroberflächen. Wie der Name schon verrät deckt ASP.NET die Webservices ab. Mit WebForms trennt man das Aussehen von Webservices von der Business Logik. Die ASP.NET Seiten werden beim ersten Zugriff in IL kompiliert und für jeden weiteren Zugriff der kompilierte IL Code verwendet. Die WindowsForms erlauben die Erstellung traditioneller Windows Oberflächen.[DevGuide]

4.2 Mono

Nach der kurzen .NET Einführung stellen sich die Fragen was genau Mono ist, und welche .NET Konzepte von Mono abgedeckt werden. Das Mono Projekt wurde Mitte 2001 gestartet mit dem Ziel eine Entwicklungsplattform für Unix Systeme zu erstellen. Führend bei der Unterstützung für Mono ist die Firma Ximian⁸, welche inzwischen zu Novell gehört. Mono setzt die ECMA Standards um welche von Microsoft vorgeschlagen wurden. Das Framework besteht aus vier Komponenten. [Mono]

- Eine Laufzeitumgebung
- Die Standard Klassenbibliothek
- Neue Klassenbibliotheken
- Compiler

⁸<http://developer.ximian.com/projects/mono/>

4.2.1 Mono Komponenten

Die Laufzeitumgebung von Mono entspricht der CLR des .NET Frameworks. Die Laufzeitumgebung übernimmt die JIT Kompilierung von IL Code in ausführbaren Code. Ebenso Garbage Collection, Threadverwaltung, plattformübergreifende Ein-/Ausgabe und mehrfache Virtual-Machine Domains.[Mono]

Die Klassenbibliothek bildet wie im .NET Framework das Herzstück. Mono hält sich an die ECMA Standard und lehnt sich stark an die Implementierung von .NET an. Neben den Klassenbibliotheken die der ECMA Standard vorschreibt, faßt Mono eine Reihe weiterer neuer Bibliotheken. Dazu gehören Datenbank Anbindungen, die Verwendung von Unix spezifischen Funktionen, und speziellen Elementen für die GUI Entwicklung mit GNOME und KDE.

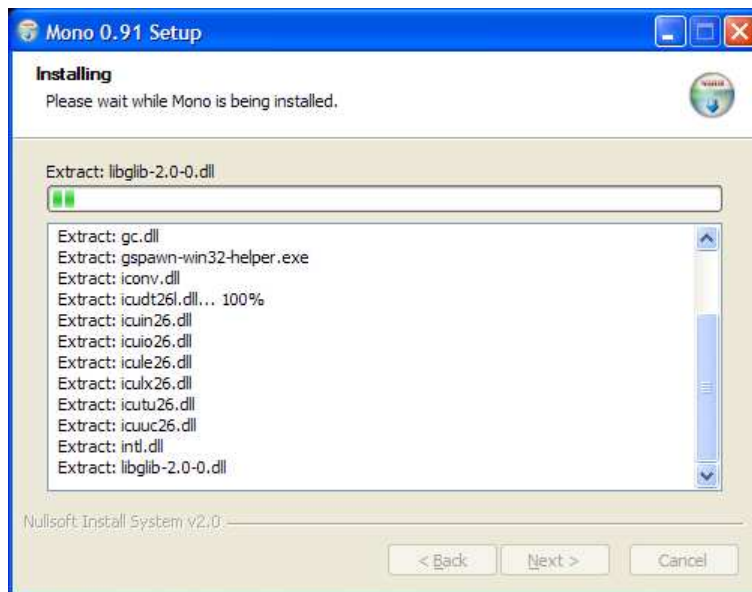
Für das Mono Framework existieren eine Reihe von Compilern. Diese sind mit den Compilern aus dem Hause Microsoft voll kompatibel. Zwar lassen sich für Mono auch .NET Sprachen Compiler verwenden, aber für den Einsatz auf anderen Betriebssystemen mussten eigenen Compiler geschrieben werden.

4.2.2 Installation von Mono auf Linux und Windows

Vorweg die schlechte Nachricht; Es gibt **noch** keine Final Version von Mono, das letzte große Release war die Beta 1. Wenn man der Roadmap Glauben schenkt kommt Mono 1.0 im zweiten Quartal 2004. Wer das aktuelle Beta 1 Release verwendet muß auch noch auf die Windows Forms verzichten. Die Alternative für GUI Entwicklung mit Mono sind C# Bindings für GTK+, genannt GTK#. Der aktuelle Stand der Implementierung ist auf der Mono Homepage unter Class Status zu finden <http://www.go-mono.org/class-status.html>

Windows

Die Installation unter Windows erfolgt mit einem Installer. Nach erfolgreicher Installation kann das Mono Framework in gewohnter Weise verwendet werden. Für Mac OS X und Linux sind die entsprechenden Packages oder Sourcefiles auf der Downloadseite <http://www.go-mono.org/download.html> verfügbar. Die notwendigen Pakete für die GUI Programmierung mit GTK# sind auf der Website des GTK# Projekts <http://gtk-sharp.sourceforge.net> zu finden. Die momentan aktuelle Version 0.91.1 ist im Sourcecode als `.tar` 1315 kB groß.



Linux

Unter Linux läuft die Installation wie gewohnt ab. Die notwendigen Pakete sind von der Mono Homepage zu haben, entweder als RPMs oder als Sourcen. Nach erfolgreichem Download lassen sich die Pakete mit `rpm -Uvh *rpm` installieren. Die Minimalinstallation umfasst die Archive für den C# Compiler und die Laufzeitumgebung.

- Mono Runtime 0.91
- Mono Class Libraries and C# Compiler 0.91

Für die entsprechenden Abhängigkeiten müssen noch weitere Pakete installiert werden. Für die aktuelle Beta1 sind fertige RPMs für Redhat, Fedora und SuSE verfügbar.

```
[root@localhost Beta1]# rpm -Uvh *rpm
```

4.3 Portabilität

4.3.1 Von Mono nach Mono

Die Portierung von Mono nach Mono macht die wenigsten Probleme. Ärgerlich sind nur Konflikte die durch unterschiedliche Versionen des installierten Mono Frameworks resultieren.

4.3.2 Von Mono nach .NET

Das Mono Projekt hat sich Kompatibilität mit dem Microsoft Vorbild auf die Fahnen geschrieben. Die Portierung von Mono nach .NET hat daher hohe Erfolgschancen. Der Portierbarkeit abträglich ist die Verwendung von Mono spezifischen Features, etwa Klassen für die GUI Entwicklung mit GNOME oder

KDE. Wie erwähnt fehlen momentan noch die WindowsForms und GTK# ist der brauchbarste Kompromiss.

4.3.3 Von .NET nach Mono

Das Fernziel von Mono ist vollständige Kompatibilität. Zum jetzigen Zeitpunkt ist Mono noch ein gutes Stück davon entfernt. Ein .NET Anwendung die man unter Windows mit Visual Studio, Sharpdevelop o.ä. entwickelt hat, läuft mit hoher Wahrscheinlichkeit nicht mit Mono auf einem Linux Rechner.

Beispiel

Dieses Beispiel illustriert das Problem. Folgender Code erzeugt ein einfaches Fenster mit .NET und läßt sich problemlos unter Windows mit dem C# Compiler von Microsoft compilieren.

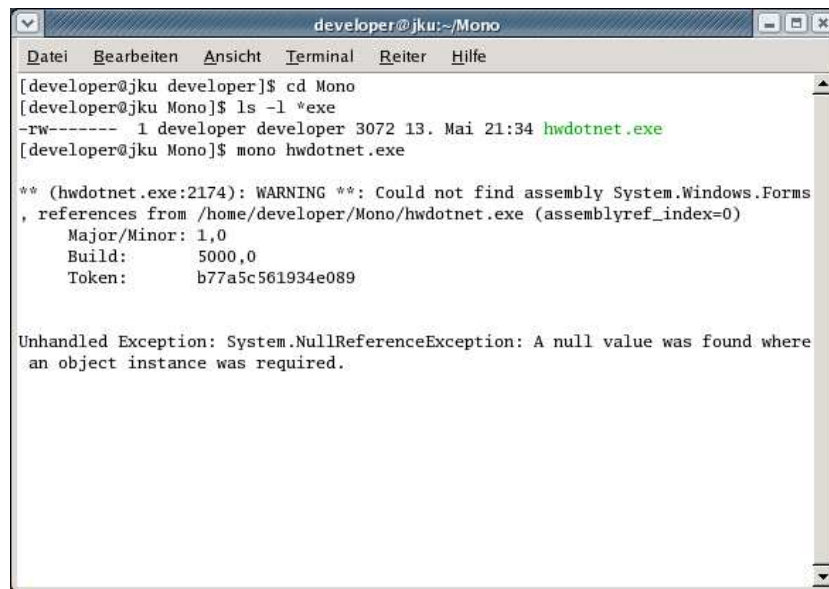
```
using System;
using System.Windows.Forms;
using System.Drawing;

namespace hwdotnet
{
    public class SimpleWin : Form
    {
        public SimpleWin()
        {
            this.Text = "Hello World from .NET";
        }

        static void Main(string arg[])
        {
            Application.Run( new SimpleWin() );
        }
    }
}
```



Der Versuch die .NET Anwendung unter Linux mit dem aktuellen Mono Framework zu starten scheitert jedoch kläglich. Die verwendeten WindowsForms sind noch nicht verfügbar.



```
developer@jku:~/Mono
Datei Bearbeiten Ansicht Terminal Reiter Hilfe
[developer@jku developer]$ cd Mono
[developer@jku Mono]$ ls -l *exe
-rw----- 1 developer developer 3072 13. Mai 21:34 hwdotnet.exe
[developer@jku Mono]$ mono hwdotnet.exe

** (hwdotnet.exe:2174): WARNING **: Could not find assembly System.Windows.Forms
, references from /home/developer/Mono/hwdotnet.exe (assemblyref_index=0)
  Major/Minor: 1,0
  Build:      5000,0
  Token:     b77a5c561934e089

Unhandled Exception: System.NullReferenceException: A null value was found where
an object instance was required.
```

4.4 Mono in der Praxis

Jetzt bleibt nur noch die Frage offen, wie man Mono verwendet. Nach erfolgreicher Installation kann man den Mono C# Compiler verwenden. Der Compiler hält sich an die ECMA-334 C# Language Specification, der Entwicklung mit C# steht also nichts im Wege. Der Compiler wird mit `mcs` aufgerufen und produziert eine `.exe` Datei.

Die Mono Laufzeitumgebung, das Pendant zur .NET CLR, ruft man mit `mono` auf. Damit läßt sich eine Mono Anwendung oder eine .NET Anwendung in IL ausführen. Im Idealfall läßt sich die Mono Anwendung per `xcopy deployment` auch auf einem Windows Rechner mit .NET Laufzeitumgebung ausführen.

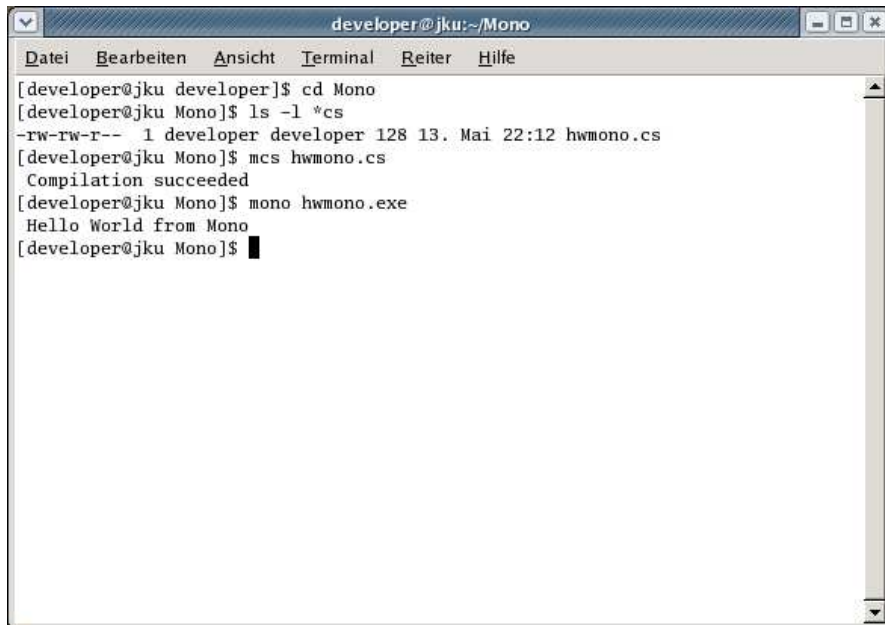
Beispiel

Als Beispiel soll das obligatorische Hello World Programm dienen.

```
using System;

class hwmono
{
    public static void Main(string[] arg)
    {
        Console.WriteLine("Hello World from Mono");
    }
}
```

Das Programm wird als `hwmono.cs` Datei gespeichert und mit dem Mono C# Compiler compiliert. Das Ergebnis ist die Datei `hwmono.exe` die in IL vorliegt. Gestartet wird das Hello World Programm mit `mono hwmono.exe`.



```
developer@jku:~/Mono
Datei Bearbeiten Ansicht Terminal Reiter Hilfe
[developer@jku developer]$ cd Mono
[developer@jku Mono]$ ls -l *.cs
-rw-rw-r-- 1 developer developer 128 13. Mai 22:12 hwmono.cs
[developer@jku Mono]$ mcs hwmono.cs
Compilation succeeded
[developer@jku Mono]$ mono hwmono.exe
Hello World from Mono
[developer@jku Mono]$
```


Kapitel 5

Fazit

Portabilität ist eine feine Sache. Hauptgrund für die Entwicklung portabler Programme ist Wunsch einen größeren Markt zu erschliessen. Die Entwicklung und das Test auf verschiedenen Systemen macht portable Programme dazu weniger fehleranfällig. Doch wo Licht ist, ist auch Schatten. Die Entwicklung für verschiedene Plattformen erfordert ein Mehr an Wissen über die Zielplattformen. Man muss mehr Zeit, Geld oder Personal einsetzen als wenn man nur für eine Plattform entwickelt. Eine Komponente die auf einer Plattform funktioniert muss noch lange nicht auf einer anderen funktionieren.

5.1 Entscheidung für eine Strategie

Ist die Entscheidung erst einmal gefallen ein Programm portabel zu entwickeln stellt sich die Frage nach einer Strategie. Grundsätzlich stehen zwei Wege offen. Zum einen nativen Code zu erzeugen, oder auf eine Laufzeitumgebung zu setzen.

5.1.1 Old School

Der alt hergebrachte Weg portable Programme mit graphischer Benutzeroberfläche zu entwickeln sind Widget Toolkist. Welches Toolkit und welche Sprache man verwendet muss von Fall zu Fall entschieden werden. In den gängigen Toolkits wie GTK+ und Qt stecken Jahre an Erfahrung. Es existiert daher ein solider Grundstock an Wissen und Personal. Die meisten Widget Toolkits sind C oder C++ Bibliotheken. Beide Sprachen sind weit verbreitet, es mangelt auch hier nicht an Erfahrung und qualifizierten Personal.

5.1.2 Plattformunabhängigkeit

Für plattformunabhängige Programme gibt es neben Java kaum Alternativen. Mono ist zwar ein ehrgeiziges Projekt, das von Monat zu Monat wächst, aber noch nicht ausgereift ist. Bis heute gibt es keine final Version von Mono. Man sollte sich den Namen auf jeden Fall merken.

Literaturverzeichnis

- [Fischer] Thorsten Fischer, GUI Programmierung mit GTK+, Handbuch und Referenz, SuSE Press 2000
- [GLib] GLib Reference Manual, <http://developer.gnome.org/doc/API/glib/index.html>
- [IBM1] The wonders of GLib,
<http://www-106.ibm.com/developerworks/linux/library/l-glib.html>
- [IBM2] The wonders of GLib Part 2,
<http://www-106.ibm.com/developerworks/linux/library/l-glib2.html>
- [ct1004] c't, Magazin für Computer Technik, Ausgabe 10 2004, Qt4 in diesem Jahr, Seite 42
- [Dalheimer] Matthias Kalle Dalheimer, Programming with Qt, Writing Portable GUI Applications on Unix & Win32, O'Reilly 1999
- [LinuxMagQt] Linux Magazin, Ausgabe 05 1996, Aus dem hohen Norden, Graphische Benutzeroberflächen mit Qt, <http://www.linux-magazin.de/Artikel/ausgabe/1996/05/Qt/qt.html>
- [DevGuide] Andreas Maslo, Jörg M. Freiburger, .NET Framework Developer's Guide, Markt und Technik Verlag 2002
- [Hanisch] Armin Hanisch, GoTo C#, Addison-Wesley 2002
- [Mono] Ewald Geschwinde, Hans-Jürgen Schönig, Mono, .NET-kompatible Anwendungen mit dem Open Source-Framework, Markt und Technik Verlag 2003
- [iX0504] iX, Magazin für Professionelle Informationstechnik, Ausgabe 5 2004, Die Zukunft von .NET, Seite 86f

Anhang A

GNU Free Documentation License

GNU Free Documentation License Version 1.2, November 2002

Copyright (C) 2000,2001,2002 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

A.1 Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of 'copyleft', which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

A.2 Applicability and Definitions

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license,

unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance

of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

A.3 Verbatim copying

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

A.4 Copying in quantity

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must

take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

A.5 Modifications

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission. B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement. C. State on the Title page the name of the publisher of the Modified Version, as the publisher. D. Preserve all the copyright notices of the Document. E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices. F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below. G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice. H. Include an unaltered copy of this License. I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence. J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission. K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein. L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not consid-

red part of the section titles. M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version. N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section. O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

A.6 Combining documents

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

A.7 Collections of documents

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

A.8 Aggregation with independent works

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

A.9 Translation

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

A.10 Termination

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify,

sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

A.11 Future revision of this license

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

A.12 Addendum: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright (c) YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Anhang B

GNU General Public License

GNU GENERAL PUBLIC LICENSE Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

B.1 Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software—to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation’s software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or

modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

B.2 GNU General Public License - Terms and conditions for copying, distribution and modification

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for non-commercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary

form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through

that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE

THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

B.3 How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it does.>  
Copyright (C) <year> <name of author>
```

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author  
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type 'show w'.  
This is free software, and you are welcome to redistribute it under certain  
conditions; type 'show c' for details.
```

The hypothetical commands 'show w' and 'show c' should show the appro-

priate parts of the General Public License. Of course, the commands you use may be called something other than ‘show w’ and ‘show c’; they could even be mouse-clicks or menu items—whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a “copyright disclaimer” for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program ‘Gnomovision’ (which makes passes at compilers) written by James Hacker.

< signature of Ty Coon >, 1 April 1989 Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

Anhang C

GNU Lesser General Public License

GNU LESSER GENERAL PUBLIC LICENSE Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

C.1 Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software—to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages—typically libraries—of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified

version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

C.2 GNU Lesser General Public License - Terms and conditions for copying, distribution and modification

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) The modified work must itself be a software library.
- b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
- c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
- d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A

suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library

or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published

by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

C.3 How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.> Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library 'Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990 Ty Coon, President of Vice

That's all there is to it!