

# **Fluxbox Dokumentation**

**Tobias Klausmann**

Herausgegeben von  
**Engelbert Gruber**

## **Fluxbox Dokumentation**

von Tobias Klausmann

Herausgegeben von Engelbert Gruber

Versionsgeschichte

Version 0.4 2003-09-16 Geändert durch: eg

New editor in charge. Add revision history. Revisit Key Actions (crosscheck with FbCommandFactory.cc).

# Inhaltsverzeichnis

<b>1. Einleitung</b> .....	<b>1</b>
1.1. Über diese Dokumentation.....	1
1.1.1. Anmerkungen zu dieser Übersetzung.....	1
1.2. Über Fluxbox .....	2
1.2.1. Was ist Fluxbox? .....	2
1.2.2. Features.....	2
1.2.3. Fluxbox herunterladen.....	3
1.2.4. Fragen und Support .....	3
<b>2. Los gehts!</b> .....	<b>4</b>
2.1. Fluxbox installieren.....	4
2.1.1. Herunterladen des Quelltextes.....	4
2.1.2. Entpacken und kompilieren .....	4
2.1.3. Fluxbox starten .....	5
2.1.4. Dies und Jenes .....	6
<b>3. Tools</b> .....	<b>7</b>
3.1. Einleitung .....	7
3.2. fbrun.....	7
3.3. fluxbox-generate_menu.....	8
3.4. fluxspace.....	8
3.5. wmctrl .....	8
<b>4. Tabs</b> .....	<b>9</b>
4.1. Eine Einführung in Tabs.....	9
4.2. Fortgeschrittenes Tabbing .....	9
4.2.1. Sloppy Window Grouping .....	9
4.2.2. Tabbing einzelner Fenster-Arten .....	9
4.2.3. Tabs komplett abschalten.....	10
4.2.4. Tab-Platzierung.....	10
4.2.5. Automatische Gruppierung von Tabs .....	10
4.2.6. Tabs in Themes.....	11
<b>5. Tastenkombinationen</b> .....	<b>12</b>
5.1. Der Keygrabber .....	12
5.2. Tasten-Namen .....	13
5.3. Aktionen.....	13
<b>6. Desktop-Hintergründe</b> .....	<b>18</b>
<b>7. Die Slit</b> .....	<b>19</b>
<b>8. Die Toolbar</b> .....	<b>21</b>
<b>9. Menüs bearbeiten</b> .....	<b>22</b>
9.1. Den Ort der Menüdatei einstellen .....	22
9.2. Verfügbare Befehle .....	22
<b>10. Themes</b> .....	<b>25</b>
10.1. Theme basics.....	25
10.2. Advanced Theme Editing.....	26

<b>A. Einrichten von <code>.xinitrc</code>/<code>.xsession</code></b> .....	<b>29</b>
<b>B. Frequently asked questions (FAQ)</b> .....	<b>32</b>
<b>C. Theme Reference *</b> .....	<b>36</b>
C.1. Theme directives.....	36
<b>D. Artwiz Fonts für Fluxbox</b> .....	<b>41</b>
<b>E. Debugging *</b> .....	<b>44</b>

# Tabellenverzeichnis

5-1. Navigation .....	14
5-2. Fenster-Operationen .....	15
5-3. Fenstergröße .....	15
5-4. Fenster verschieben .....	16
5-5. Verschiedenes .....	16
10-1. Texture directives .....	27
C-1. Toolbar styles .....	36
C-2. Menu styles .....	37
C-3. Window styles .....	38
C-4. Handlebar styles .....	38
C-5. Resize grip styles .....	39
C-6. Window button styles .....	39
C-7. Window frame styles .....	39
C-8. Tab styles .....	39
C-9. Font on window label styles .....	40
C-10. Miscellaneous styles .....	40

# Beispiele

4-1. groups Datei .....	11
5-1. Ein Fluxbox keys-File .....	12
5-2. XMMS Tastenkombinationen .....	12
5-3. Multimedia-Tasten für XMMS .....	13
5-4. xev Ausgabe für die Cursor-Rechts-Taste .....	13
7-1. Slitlist Pfadangabe .....	20
8-1. Toolbar Tools Beispiel .....	21
9-1. Den Ort der Menüdatei vorgeben .....	22
9-2. Beispiel einer Menüdatei .....	23
10-1. Toolbar Clock style .....	26
10-2. Menu style .....	26
10-3. Font style .....	26
10-4. Texture style .....	27
10-5. Typical short style .....	28
A-1. .xinitrc .....	30
B-1. Personalisierte Tabs im Style-File .....	34
D-1. fonts.dir .....	42

# Kapitel 1. Einleitung

## 1.1. Über diese Dokumentation

Dies ist die neue Dokumentation für Fluxbox, einen X11 Windowmanager. Sie wurde aus der alten HTML-Dokumentation erzeugt, mit dem Ziel, sie leicht anpassbar an verschiedene Ausgabeformate zu machen. Die Quellen sind in DocBook 4.1 geschrieben, ein Format welches speziell für Dokumentationen entwickelt wurde.

Bisher gab es einige Personen, die an der Dokumentation mitgearbeitet haben. Was Sie jetzt lesen, basiert größtenteils auf der Arbeit von Rando Christensen <rando@babblica.net (mailto:rando@babblica.net)> oder wurde ursprünglich von ihm geschrieben, gefolgt von <klausman@users.sourceforge.net (mailto:klausman@users.sourceforge.net)> und den Beiträgen vieler anderer. Falls Danksagungen fehlen, benachrichtigen Sie bitte den <Herausgeber (mailto:grubert@users.sourceforge.net)>.

Fragen, die Fluxbox selbst betreffen, passen am besten in die entsprechenden Mailverteiler. Informationen hierzu finden sich auf der offiziellen Fluxbox Website(<http://fluxbox.sourceforge.net/download.php>).

**Einsendungen und Übersetzung:** Ich akzeptiere fast alle Einsendungen von gut geschriebener Dokumentation für alles, was in dieser Dokumentation (noch) nicht enthalten ist. Wer einen Vorschlag hat, was man hinzufügen könnte; wer wissen möchte, wie er/sie helfen kann; etc., schickt am besten einfach eine E-Mail an die oben genannte Adresse.

Sowohl die Quellen dieser Dokumentation als auch verschiedene konvertierte Formate (z.B. HTML, PostScript) stehen auf der Fluxbox Website zum Download bereit. Wenn Sie gern eine andere DTD oder ein anderes Ausgabeformat verwenden möchten, sind die Quellen Ihr Freund - obwohl Sie wahrscheinlich eine gewisse Erfahrung mit DocBook benötigen, um DTDs zu ändern. Wenn Sie die Quellen in ein anderes Format als die hier verfügbaren umwandeln möchten, könnte das Paket docbook2x hilfreich sein.

### 1.1.1. Anmerkungen zu dieser Übersetzung

Diese Übersetzung stammt von Robert Spielmann. Momentan wird daran noch gearbeitet. Mit einem "\*" gekennzeichnete Kapitel sind noch nicht oder nur teilweise übersetzt! Das Kapitel "Debugging" wird bewusst in der Originalsprache (Englisch) belassen. Ich (Robert) bin für jede Art von konstruktiver Kritik dankbar. Wenn jemand einen Fehler findet oder anderweitige Vorschläge hat, bin ich gern unter <varial@gmx.de (mailto:varial@gmx.de)> ansprechbar. Bitte verwendet ein sinnvolles Topic,

damit ich die Mail vom täglichen Spam unterscheiden kann :-) Meine Website befindet sich unter der Adresse <http://www.spielmann.info>. Danke und viel Spass mit Fluxbox!

Vielen Dank an Markus Konrad für den Hinweis auf einen Fehler in Kapitel 1.2.1!

## **1.2. Über Fluxbox**

### **1.2.1. Was ist Fluxbox?**

Fluxbox ist ein window manager für X. Fluxbox basiert auf dem Code von Blackbox 0.61.1. Fluxbox sieht aus wie Blackbox und verarbeitet Styles, Farben, Fensterpositionierung und ähnliches exakt wie Blackbox (100%ige Theme/Style Kompatibilität).

Was macht dann den Unterschied zwischen Fluxbox und Blackbox aus? Die Antwort ist einfach: eine Menge! Hier ist eine Liste von Features, die Fluxbox schon hat oder die in Bearbeitung sind:

### **1.2.2. Features**

Implementiert:

- Einstellbare window tabs
- Antialias Unterstützung (Xft)
- UTF-8 Unterstützung
- Iconbar (für minimierte/iconified Fenster)
- Fenster/Tabs auto grouping
- MouseWheel scrollen wechselt Workspace
- Einstellbare Titlebar (Platzierung der Buttons, neue Buttons etc.)
- KDE and Gnome (inkl. Gnome 2) Unterstützung
- Extended Window Manager Hints (ewmh) Unterstützung
- Eingebauter keygrabber (unterstützt emacs-ähnliche keychains)
- Maximize over slit Option
- Slit dockapp Reihenfolge einstellbar

In Planung:

- Session Management
- Windows Snapping
- Einstellbare Toolbar

- Andere Kleinigkeiten

Abgesehen von all dem sind am Code viele Verbesserungen vorgenommen worden.

### **1.2.3. Fluxbox herunterladen**

Fluxbox Quellcode kann von der Fluxbox website, <http://fluxbox.sourceforge.net/> heruntergeladen werden. Die meisten Linux-Distributionen und andere freie UNIXe enthalten auch Quellcode- und Binärpakete von Fluxbox. Im Zweifel sollten Sie die aktuellsten Pakete Ihrer Distribution verwenden.

### **1.2.4. Fragen und Support**

Obwohl Fluxbox recht einfach zu benutzen und einzustellen ist, kann es gut sein dass an der ein oder anderen Stelle Fragen oder Probleme auftauchen. Ein großer Anteil davon ist bereits häufiger aufgetaucht und deshalb möglicherweise bereits in den FAQ in Anhang B beantwortet. Bitte lesen Sie die FAQ, bevor Sie über die Mailverteiler oder im Fluxbox IRC-Channel Fragen stellen.

# Kapitel 2. Los gehts!

## 2.1. Fluxbox installieren

Dieser Abschnitt ist ein Beitrag von Jason Gillman Jr. alias "Ircaddict".

Viele Distributionen bieten fertige Pakete für Fluxbox an (oder ports/ebuilds, wenn Sie Gentoo oder FreeBSD benutzen). Diese Pakete machen die Installation recht einfach und harmlos. Es gibt aber auch Gründe, Fluxbox aus dem Quellcode zu kompilieren. Zum Beispiel ist wahrscheinlich die neueste Fluxbox-Version (noch) nicht als Paket verfügbar. Darüber hinaus könnte es von Vorteil sein, auf Ihrem System bestimmte Compiler-Flags zu verwenden. Falls Sie die Pakete Ihrer Distribution verwenden wollen, lesen Sie in der Dokumentation Ihrer Distribution nach. Für einige Distributionen gibt es auf der Fluxbox-Website sowohl Binär- als auch Quellcode-Pakete.

Dieses Dokument soll denjenigen helfen, die noch neu im Umgang mit X11 (oder Linux allgemein) sind. Es wird erklärt, wie Fluxbox kompiliert und installiert wird.

### 2.1.1. Herunterladen des Quelltextes

Gehen Sie als erstes auf die Download-Seite(<http://fluxbox.sourceforge.net/download.php>) und laden Sie sich den Quelltext-Tarball herunter (die Datei-Endung lautet `.tar.gz`).

Gelegentlich gibt es spezielle Developer-Versionen von Fluxbox. Diese dienen als Vorschau auf zukünftige Features und ähnliches. Da diese für Programmierer gedacht sind, fehlen ihnen möglicherweise Teile der Funktionalität und es kann vor allem Probleme mit der Stabilität geben. Wenn Sie das allerneueste Fluxbox haben wollen, probieren Sie ruhig die Developer-Versionen aus - damit nehmen Sie aber neben allen Vorteilen auch alle Risiken in Kauf. Im Newsbereich der Fluxbox-Website finden Sie Anweisungen, wo Sie die Developer-Versionen bekommen.

### 2.1.2. Entpacken und kompilieren

Jetzt, da Sie den Quelltext heruntergeladen haben, müssen Sie den Tarball entpacken. Das geht mit dem folgenden Kommando (ersetzen Sie den Dateinamen durch den des heruntergeladenen Tarballs):

```
$ tar xzvf fluxbox-0.1.12.tar.gz
```

Es erscheint eine Liste von Dateien, die entpackt werden. Nachdem der Vorgang abgeschlossen ist, wechseln Sie in das erzeugte Verzeichnis (es wird je nach Version zum Beispiel `fluxbox-0.1.12` heißen). Als nächstes muss Fluxbox konfiguriert und kompiliert werden. Während der Konfiguration können Sie einige Features von Fluxbox ein- oder ausschalten. Im Allgemeinen sollten die Standardwerte in Ordnung sein. Wenn Sie wollen, dass die Fluxbox-Slit mit KDE Panel Icons

funktioniert, müssen Sie die Option `--enable-kde` hinzufügen. Um herauszufinden welche Optionen das `configure`-Skript anbietet, benutzen Sie die Option `--help`. Wenn Sie die KDE-Unterstützung nicht wollen, sollte dies genügen:

```
$ ./configure
$ make
```

Sobald Fluxbox kompiliert ist, werden Sie `root` und führen aus:

```
# make install
```

Herzlichen Glückwunsch - Sie haben Fluxbox kompiliert und installiert!

### 2.1.3. Fluxbox starten

Schön wenn Sie Fluxbox installiert haben, aber was nützt es Ihnen wenn Sie es nicht benutzen können?

Es gibt zwei grundlegend verschiedene Wege, X11 zu starten (und damit Fluxbox). Der traditionelle Weg führt über den Befehl `startx`. Der andere Weg führt über einen grafischen Login-Manager (auch Display-Manager genannt). Der bekannteste Displaymanager ist `xdm` (dieser gehört zu XFree86). Der Displaymanager von Gnome heißt `gdm`, der von KDE `kdm`.

Wenn X11 mit der ersten Methode gestartet wird (via `startx`), heißt die für uns wichtige Datei `.xinitrc` und befindet sich in Ihrem Homeverzeichnis. Im Falle des Starts über einen Displaymanager heißt die Datei `.xsession` und befindet sich an der gleichen Stelle.

Der nächste Schritt besteht darin, die ausführbare Datei für Fluxbox zu finden. In den meisten Fällen ist das `/usr/local/bin/fluxbox`. Jetzt müssen Sie die vorhin genannte Datei (`.xinitrc`) editieren oder erzeugen. Tragen Sie einfach am Ende der Datei folgende Zeile ein:

```
exec /usr/local/bin/fluxbox
```

Ändern Sie `/usr/local/bin/fluxbox` entsprechend ab, je nachdem wo sich bei Ihrer Installation Fluxbox befindet. `/usr/local/bin/fluxbox` ist der Standard, wenn Sie Fluxbox aus dem Quellcode übersetzen. Sobald Sie die Datei bearbeitet haben, speichern und schließen Sie sie. Wenn Sie `startx` verwenden, müssen Sie dann folgendes ausführen:

```
$ chmod 700 .xinitrc
```

Im Falle von `.xsession` ist das nicht notwendig. In beiden Fällen sollten Sie das Verzeichnis erzeugen, wo Fluxbox seine Konfiguration speichert:

```
$ mkdir .fluxbox
```

Wenn Sie das Verzeichnis nicht erzeugen, werden Sie alle Einstellungen verlieren wenn Sie Fluxbox beenden und neu starten. Leider wird das Verzeichnis nicht automatisch angelegt.

#### **2.1.4. Dies und Jenes**

Wenn Sie Hilfe brauchen, kommen Sie einfach ins Fluxbox-Forum und stellen Sie dort Ihre Frage. Irgendjemand ist immer gern bereit, Ihnen zu helfen. Sie können Ihre Frage auch in der Fluxbox Mailing-Liste stellen. Ein Link zu der Liste befindet sich auf der Fluxbox Website. Als letzter Hinweis empfehle ich persönlich, die `init`, `menu`, und `titlebar` Dateien aus `/usr/local/share/fluxbox` in das Verzeichnis `.fluxbox/` in ihrem Homeverzeichnis zu kopieren.

# Kapitel 3. Tools

## 3.1. Einleitung

Fluxbox enthält eine Ansammlung von Tools, die Ihnen das Leben ein wenig erleichtern oder zusätzliche nette Features anbieten. Im Normalfall sind diese im gleichen Verzeichnis wie `fluxbox` installiert, nämlich üblicherweise in `/usr/local/bin` (es sei denn, Sie haben bei der Installation ein anderes Prefix gewählt). Die meisten Distributionen wählen auch ein anderes Verzeichnis - sollten Sie Binärpakete installiert haben, finden Sie die Dateien möglicherweise in `/usr/bin`.

## 3.2. `fbrun`

`fbrun` ist im Grunde gleichwertig zum Dialogfeld "Ausführen..." in anderen Desktop-Umgebungen. Das bedeutet, dass es eine einfache Möglichkeit zum Ausführen solcher Programme bietet, die nicht im Menü enthalten sind oder einen bestimmten Satz an Parametern für diesen einen Start benötigen.

Außerdem kann `fbrun` nützlich sein, um es mit einer vorgegebenen Kommandozeile aus dem Menü heraus zu laden. Diese Zeile kann dann editiert und ausgeführt werden. Ein Beispiel wäre der Aufbau einer SSH-Verbindung zu einem sehr langen Hostnamen und mit vielen Optionen, von denen sich nur eine häufiger ändert. In diesem Fall könnten Sie Ihrem Menü einen Eintrag für `fbrun` hinzufügen, der alle Optionen und den Hostnamen enthält. Sobald Sie den besagten Menüpunkt aufrufen, können Sie die Zeile wie benötigt editieren und sie dann ausführen.

`fbrun` hat verschiedene Optionen:

<code>-font [font name]</code>	Text Font
<code>-title [title name]</code>	Titel einstellen
<code>-text [text]</code>	Text-Eingabe
<code>-w [width]</code>	Fensterbreite in Pixeln
<code>-h [height]</code>	Fensterhöhe in Pixeln
<code>-display [display string]</code>	Displayname
<code>-pos [x] [y]</code>	Fensterposition in Pixeln
<code>-fg [color name]</code>	Vordergrund-Textfarbe
<code>-bg [color name]</code>	Hintergrundfarbe
<code>-a</code>	Antialias
<code>-hf [history file]</code>	History file (default <code>~/.fluxbox/history</code> )
<code>-help</code>	Diese Hilfe anzeigen

**fbrun Hilfetext:** Beachten Sie bitte, dass `fbrun` obige Hilfe nur in Englisch auf dem Bildschirm ausgibt. Die Texte sind nur für das bessere Verständnis übersetzt.

Die meisten dieser Optionen sollten selbsterklärend sein. Einzig die Optionen **-text** und **-hf** benötigen möglicherweise eine Erklärung. Die erste wird benutzt, um die vorgegebene (editierbare) Kommandozeile im `fbrun`-Fenster anzugeben. Wenn Sie mehrere Argumente benötigen (z.B. **ssh -X -f**), fassen Sie diese in Anführungszeichen ein:

```
fbrun -text "ssh -X -f"
```

Die **-hf** Option gibt das History-File an, in dem `fbrun` sein "Gedächtnis" ablegt, welche Kommandozeilen Sie bislang benutzt haben (genau wie die `bash` das auch tut). Normalerweise braucht man diese Option nicht, der Standardwert ist `~/fluxbox/history`. Sie kann aber nützlich sein, wenn Sie mehrere `fbrun` Menü-Einträge haben und für jeden eine getrennte History abspeichern möchten.

### 3.3. fluxbox-generate\_menu

FIXME: Diese Sektion muss noch geschrieben werden!

### 3.4. fluxspace

Zitat von: <http://fluxspace.sourceforge.net>):

Fluxspace blends Fluxbox's window management with new desktop management capabilities. It leverages existing components and the power of Python to help build a flexible desktop environment around Fluxbox and other lightweight window managers.

- Add per-workspace desktop icons and panels to Fluxbox by integrating tools like Rox Filer and Idesk.
- Decorate each workspace with different wallpaper.
- Manage your startup applications.
- Automatically start and stop applets as you enter and leave workspaces, allowing each workspace to have its own unique tools and dockapps.

Zu Deutsch: Fluxspace bietet die Möglichkeit, zum Beispiel Icons auf den Fluxbox Desktop zu legen, für jeden Workspace andere Hintergründe zu verwenden, oder auch automatisch Programme zu starten wenn Sie auf einen Workspace wechseln. Es besteht außerdem die Möglichkeit, für Fluxspace Zusatzmodule in Python selber zu schreiben.

### 3.5. wmctrl

`wmctrl` ist ein Kommandozeilen-Tool, das die Interaktion mit EWMH/NetWM kompatiblen Windowmanagern ermöglicht.

# Kapitel 4. Tabs

## 4.1. Eine Einführung in Tabs

Die Tabs von Fluxbox sind in keiner Hinsicht eine neue Idee. Die neue Implementierung ist der des PWM Windowmanagers sehr ähnlich. Und so funktioniert das ganze: Mehrere Fenster werden gruppiert und bekommen die gleiche "Geometrie" - sie haben die gleiche Größe, befinden sich an der gleichen Position auf dem Desktop, und können nur alle zusammen verschoben werden. Man könnte diese Fenster mit einem Stapel Papier vergleichen - die Tabs wären dann die kleinen Plastik-Schildchen, die man als Lesezeichen auf Seiten kleben kann um einen schnellen Zugriff zu ermöglichen.

In den Versionen bis 0.1.14 waren die Tabs am Fenster angebracht, ab 0.9.x sind sie in der Titelzeile eingebettet.

Dies ist genau die Funktionsweise von Tabs in Fluxbox: Sie wählen einfach das richtige Tab für das Fenster aus, das Sie sehen wollen, und dieses Fenster wird sichtbar.

### Tabbing-Grundlagen

Als erstes muss man sich merken, dass *alle* Änderungen an Tabs den dritten Mouse-Button verwenden. Zum Einstieg wählen Sie zwei Fenster aus, die Sie gern gruppieren würden. Klicken Sie mit der 3. Maustaste auf das Tab des ersten Fensters und ziehen Sie dieses auf das Tab des zweiten Fensters. Herzlichen Glückwunsch - die Fenster kleben zusammen! Ab sofort können Sie zwischen den beiden Fenstern umschalten, indem Sie die Tabs anklicken.

Um ein Tab zu entfernen, geht es genauso: klicken Sie mit der 3. Maustaste auf ein gruppiertes Tab und ziehen Sie es einfach an eine andere Stelle.

## 4.2. Fortgeschrittenes Tabbing

### 4.2.1. Sloppy Window Grouping

"Es ist mir aber lästig, das Tab auf das andere kleine Tab zu ziehen."

Da haben wir gute Nachrichten für Sie. Wählen Sie aus dem Fluxbox Konfigurations-Menü die Option 'Sloppy Window Grouping'. Diese erlaubt es Ihnen, das Tab *überall* auf dem Ziel-Fenster abzulegen um die Gruppierung zu erreichen.

### 4.2.2. Tabbing einzelner Fenster-Arten

"Das ist toll, aber ich will nur Programm X gruppieren!"

Es gibt zwei Wege, wie dies erreicht werden kann - je nachdem, wie viele Programme Tabs haben sollen. Entweder können Sie Tabs für jedes einzelne Fenster ein-/ausschalten (Rechtsklick auf die Titelzeile, 'Tab'-Option anklicken), *oder* Sie können die Tabs global abschalten (über die Option 'Configuration' -> 'Use Tabs'). Nachdem die Tabs abgeschaltet sind, können Sie diese wie beschrieben für individuelle Fenster einschalten.

### 4.2.3. Tabs komplett abschalten

"Ich glaube ich mag Tabs nicht. Kann ich sie abschalten?"

Selbstverständlich. Wählen Sie 'Use Tabs' aus dem Fluxbox Konfigurations-Menü ('Configuration'). Da dies ein "Schalter" ist, können die Tabs wieder eingeschaltet werden, indem der Menüpunkt erneut angeklickt wird. Es gibt auch eine Einstellung in der `init` Konfigurationsdatei:

```
session.tabs: true
```

Das Abschalten der Tabs kann erreicht werden, indem **true** durch **false** ersetzt wird.

### 4.2.4. Tab-Platzierung

In den Versionen bis 0.1.14 waren die Tabs am Fenster angebracht, ab 0.9.x sind sie in die Titelzeile eingebettet.

(0.1.14) Es gibt eine Option im Konfigurationsmenü namens 'Tab Placement'. Dabei handelt es sich um Stellen am Fenster, wo die Tabs platziert werden. Dies dürfte leicht verständlich sein, zu erklären sind vielleicht noch die 'Relative' Optionen. Diese Optionen sorgen dafür, dass die Gesamtlänge aller Tabs am Fenster gleich der Länge der Fensterseite ist, an der sie angebracht sind. Also: wenn das Fenster EIN Tab hat, ist das Tab so lang / breit wie das Fenster. Bei zwei Tabs haben beide eine Breite von 50% der Fensterbreite / -höhe, und so weiter. Diese Option ist sehr beliebt, da sie die Tabs unscheinbar aussehen lässt.

### 4.2.5. Automatische Gruppierung von Tabs

Es kommt vor, dass man bestimmte Anwendungen gern automatisch gruppiert haben möchte. Logischerweise nennt sich dies "Autogrouping". Hier wird erklärt, wie Autogrouping funktioniert. Zunächst benötigen Sie Fluxbox 0.1.11 oder höher. In älteren Versionen funktioniert Autogrouping nicht. Dann muss eine Datei `~/ .fluxbox/groups` angelegt werden, wenn sie noch nicht existiert. Danach editieren Sie die Datei `~/ .fluxbox/init` und fügen folgende Zeile hinzu, oder ändern die möglicherweise vorhandene Zeile entsprechend:

```
session.groupFile: ~/ .fluxbox/groups
```

Jetzt ist alles soweit. Nun muss nur noch die `groups`-Datei mit Daten gefüllt werden.

## Groupfile-Format

Jede Zeile in der Datei gibt eine Gruppe an. Es muss nur der Instanz-Name der zu gruppierenden Programme eingegeben werden. Ein Beispiel:

### Beispiel 4-1. groups Datei

```
Navigator nedit
xterm
```

Hierdurch werden zwei Gruppen erzeugt, eine mit `netscape` und `nedit`, sowie eine mit `xterm`. Das neue Fenster gruppiert sich nur mit anderen Fenstern auf dem gleichen Workspace und mit dem letzten Fenster, das den Fokus hatte. Den Namen, der im group-File einzutragen ist, erhalten Sie so:

```
xprop | awk '/WM_CLASS/{print $4}'
```

Klicken Sie dann auf das Fenster. Falls nichts angezeigt wird, versuchen Sie es statt `$4` mit `$3`.

## Autogrouping aus Tabs heraus

Autogrouping aus Tabs erlaubt es, das Root-Menü anzuzeigen indem man auf das Tab rechtsklickt, und dann eine Anwendung zu starten, deren Fenster dann zu dem Tab gruppiert wird.

**Anmerkung:** Gruppierungen dieser Art (aus Tabs heraus) können auf üble Weise mit Autogrouping interferieren.

## 4.2.6. Tabs in Themes

Es gibt einen ganzen Abschnitt zum Aussehen von Tabs in Themes (Kapitel 10). Wenn Sie daran interessiert sind, das Aussehen von Tabs in Ihrem Theme zu verändern, lesen Sie diesen Abschnitt. (Tabs in Themes nehmen normalerweise das Aussehen der Fenster-Titelzeile an, manchmal möchten Anwender dies jedoch ändern.).

# Kapitel 5. Tastenkombinationen

## 5.1. Der Keygrabber

Der Keygrabber funktioniert stark ähnlich wie `bbkeys`. `bbkeys` ist ein exzellentes Tool, das aber einige Einschränkungen hatte. Der Fluxbox Keygrabber nutzt eine völlig neue Syntax für die Konfiguration und bietet einige neue Funktionen, die Fluxbox umso mächtiger machen.

Zum einen unterstützt der neue Keygrabber "Ketten" von Tastenfolgen (so wie Emacs). Zum Beispiel könnte man einstellen: `Mod1 + M + Mod1 + F` um zum nächsten Workspace zu schalten - natürlich würde niemand wirklich genau diese Tastenfolge benutzen, sie dient nur als Beispiel.

Wenn Sie einen Teil der Sequenz schon eingegeben haben, sich dann aber entscheiden, die Sequenz abubrechen, können Sie eine andere Tastenfolge eingeben (die Sie in Ihrem keys-File festgelegt haben) um die erste Sequenz abubrechen (benutzen Sie hierzu: `AbortChain`).

Zusätzlich können Sie auch eine Tastenfolge festlegen, um zwischen gruppierten/mit Tabs gruppierten Fenstern in einer Gruppe umzuschalten (benutzen Sie hierzu: `NextTab` und `PrevTab`).

Schließlich haben wir (oder eher vlad und tarzeau) für Sie zwei Skripte zusammengestellt (beide tun das gleiche), die eine `bbkeys` Konfiguration in ein Fluxbox keys-File umwandeln. (Die Benutzung der Skripte ist ihnen selbst beschrieben). Download:  
`convertkeys`(<http://fluxbox.sourceforge.net/download/convertkeys>) oder  
`convertkeys2`(<http://fluxbox.sourceforge.net/download/convertkeys2>).

### Beispiel 5-1. Ein Fluxbox keys-File

```
Mod1 Tab :NextWindow
Mod1 F1 :Workspace 1
Mod1 F2 :Workspace 2
Mod1 F3 :Workspace 3
Mod1 F4 :Workspace 4
Control n Mod1 n :NextTab
```

Wie Sie sehen, benötigt man zunächst einen Modifier (Beispiel: Alt oder Control), und dann eine "normale" Taste. Wenn Sie eine längere Sequenz wünschen, folgt danach wieder ein Modifier und eine Taste. Abschließend benötigt man noch einen Doppelpunkt und die auszuführende Aktion.

Eine Liste gültiger Aktionen finden Sie in diesem Dokument ein Stück weiter unten!

Warum ist das alles toll für Sie? Nun, zum Beispiel können Sie nun `xmms` fernsteuern:

**Beispiel 5-2. XMMS Tastenkombinationen**

```
Mod1 P :ExecCommand xmms -p
Mod1 F :ExecCommand xmms -f
```

Wenn Sie über weitere Tasten verfügen (z.B. Multimedia-Tasten), kann xmms mit diesen Tasten ferngesteuert werden, wenn Sie sie in XFree86 richtig eingerichtet haben:

**Beispiel 5-3. Multimedia-Tasten für XMMS**

```
None XF86AudioPlay :ExecCommand xmms -u
None XF86AudioStop :ExecCommand xmms -s
```

Lesen Sie xmms --help für nähere Informationen, aber wahrscheinlich tun Sie das eh gerade...

## 5.2. Tasten-Namen

Wahrscheinlich fragen Sie sich, wie Sie die Namen von Tasten herausfinden. Starten Sie `xev`, bewegen Sie die Maus in das neue Fenster und drücken Sie eine Taste, um zu sehen wie die Ausgabe lautet. Hier ist die Ausgabe, nachdem die Cursor-Rechts-Taste gedrückt wurde:

**Beispiel 5-4. xev Ausgabe für die Cursor-Rechts-Taste**

```
KeyPress event, serial 18, synthetic NO, window 0x2c00001,
root 0x60, subw 0x0, time 3745737930, (373,380), root:(504,526),
state 0x10, keycode 102 (keysym 0xff53, Right), same_screen YES,
XLookupString gives 0 characters: ""
```

Der interessante Teil ist der Tasten-Name, der zusammen mit dem `keysym` in den Klammern steht. In diesem Beispiel steht dort (`keysym 0xff53, Right`). Also lautet der Name für die Taste `Right`.

### Spezielle Tasten

Nun folgt eine Übersicht einiger spezieller Tasten zu Ihrer Information. Beachen Sie, dass diese Tasten von `xev` sofort angezeigt werden nachdem sie gedrückt wurden.

<b>Taste</b>	<b>X11 Name</b>
Control, Strg	Control
Alt	Mod1
Super, Meta, Win* Tasten	Mod4

## 5.3. Aktionen

Es folgen die Aktionen, die Fluxbox derzeit anbietet. Sie decken den größten Teil dessen ab, was man an normalerweise auf Tasten legen würde. Beachten Sie, dass in der Konfigurationsdatei für die Tastenkombinationen das letzte Zeichen vor der Aktion ein `:` sein sollte!

Aktionen sind unabhängig von Groß-/Kleinschreibung.

**Tabelle 5-1. Navigation**

Aktion	Resultat
<b>Workspace</b>	Springe zu einem bestimmten Workspace. Benutzen Sie <code>:Workspace 1</code> , <code>:Workspace 2</code> , etc.
<b>NextTab</b>	Zum nächsten Tab in der aktuellen Gruppe umschalten.
<b>PrevTab</b>	Zum vorherigen Tab in der aktuellen Gruppe umschalten.
<b>NextWindow N</b>	Zum nächsten Fenster schalten. Siehe Note 1.
<b>PrevWindow N</b>	Zum vorherigen Fenster schalten. Siehe Note 1.
<b>NextWorkspace</b>	Zum nächsten Workspace springen.
<b>PrevWorkspace</b>	Zum vorherigen Workspace springen.
<b>NextGroup, PrevGroup</b>	Zur nächsten/vorherigen Fenstergruppe schalten.
<b>LeftWorkspace</b>	Das selbe wie <code>PrevWorkspace</code> .
<b>RightWorkspace</b>	Das selbe wie <code>NextWorkspace</code> .

**1. NextWindow / PrevWindow:** `NextWindow` und `PrevWindow` haben eine Zahl als Argument, die in der Tabelle nicht erklärt werden kann. So funktioniert es:

Der ganzzahlige Parameter gibt bestimmte Optionen vor. Aus der untenstehenden Tabelle werden die Optionen ausgewählt und dann per logischem ODER verknüpft:

Bitwert	Option
1	andere Tabs überspringen (Tab-Gruppen als EIN Fenster behandeln)
2	"stuck" Fenster überspringen.
4	"shaded" Fenster überspringen.

Addieren Sie die Werte der Optionen, die Sie wünschen. Das Ergebnis ist der Parameter für `NextWindow / PrevWindow`. Oder Sie wählen einfach den Wert aus dieser Tabelle:

Param	Optionen
0	Kein Überspringen.
1	Andere Tabs überspringen
2	"stuck" Fenster überspringen.

Param	Optionen
3	Andere Tabs und "stuck" Fenster überspringen.
4	"shaded" Fenster überspringen.
5	andere Tabs und "shaded" Fenster überspringen.
6	"stuck" und "shaded" Fenster überspringen.
7	andere Tabs, "stuck" Fenster und "shaded" Fenster überspringen.

Tabelle 5-2. Fenster-Operationen

Aktion	Resultat
Close	Fenster schließen.
KillWindow	Das Äquivalent dazu, <code>xkill</code> aufzurufen und dann auf das Fenster zu klicken.
Minimize	Auch als 'iconify' bekannt. Minimiert das Fenster.
ShadeWindow	Das Fenster in den 'shaded' Modus setzen, oder aus dem 'shaded' Modus herausholen.
StickWindow	Den 'sticky' Modus eines Fensters umschalten.
ToggleDecor	Einstellen ob das aktuelle Fenster einen Rahmen, Buttons und eine Titelzeile besitzt.
Raise	Das Fenster 'nach oben' bringen, es erscheint dann 'oberhalb' von Fenstern, die es überlappt.
Lower	Gegenteil von Raise.
NextTab, PrevTab	Nächstes / Vorheriges Tab aktivieren.
MoveTabLeft, MoveTabRight	Unklar, daher Original belassen: Move activate tab by the <code>n</code> tabs left/right.
DetachClient	Fenster aus der Tab-Gruppe herauslösen.

Tabelle 5-3. Fenstergröße

Aktion	Resultat
HorizontalIncrement	Macht das Fenster eine Einheit breiter (Siehe Note 2).
HorizontalDecrement	Gegenteil von HorizontalIncrement
VerticalIncrement	Macht das Fenster eine Einheit höher (Siehe Note 2).
VerticalDecrement	Gegenteil von VerticalIncrement
MaximizeHorizontal	Das Fenster horizontal maximieren.
MaximizeVertical	Das Fenster vertikal maximieren.
MaximizeWindow	Das Fenster maximieren.

Aktion	Resultat
<b>Resize</b>	Das Fenster um die angegebene Differenz in der Größe verändern, z.B. <b>resize -8 -8</b> .
<b>ResizeHorizontal</b>	Nur horizontal die Größe ändern.
<b>ResizeVertical</b>	Nur vertikal die Größe ändern.
<b>ArrangeWindows</b>	Die Fenster magisch anordnen.
<b>ShowDesktop</b>	Alle Fenster minimieren.

**2. HorizontalIncrement / HorizontalDecrement / VerticalIncrement / VerticalDecrement:** Beim Ändern der Größe kommt der Begriff "eine Einheit" vor. Dies betrifft speziell `xterm/aterm/Eterm` insofern, als bei diesen das Fenster nicht um ein Pixel in der Größe geändert wird, sondern so, dass Platz für ein zusätzliches Zeichen geschaffen wird.

Andere Programme sollten die Fenstergröße einfach um ein Pixel ändern.

Tabelle 5-4. Fenster verschieben

Aktion	Resultat
<b>SendToWorkspace</b>	Das aktuelle Fenster auf einen bestimmten Workspace verlegen. Verwenden Sie <code>:SendToWorkspace 1</code>
<b>Move</b>	um <code>delta-x delta-y</code> verschieben.
<b>MoveLeft</b>	Raten Sie mal.
<b>MoveRight</b>	Raten Sie mal.
<b>MoveUp</b>	Raten Sie mal.
<b>MoveDown</b>	Raten Sie mal.

Tabelle 5-5. Verschiedenes

Aktion	Resultat
<b>AbortKeychain</b>	In Tastenkombinationen, die aus mehreren aufeinander folgenden Tasten bestehen, die aktuelle Sequenz abbrechen.
<b>ExecCommand</b>	Einen Befehl ausführen. Beispiel <code>:ExecCommand xmms -t</code> .
<b>RootMenu</b>	Das Root-Menü aufrufen.
<b>WorkspaceMenu</b>	Das Workspace-Menü aufrufen.
<b>Restart</b>	Fluxbox neu starten.
<b>Reconfigure</b>	Fluxbox Konfiguration neu einlesen. Zum Beispiel wenn <code>keys</code> geändert wurde, aber <code>init</code> und <code>slitlist</code> vor dem Lesen möglicherweise geschrieben werden.
<b>SetStyle</b>	Das angegebene Style-File laden.

**Aktion**

`SetWorkspaceName`

`SaveRC`

`Quit`

**Resultat**

Den Workspace-Namen setzen.

Resource-Files speichern.

Fluxbox beenden.

# Kapitel 6. Desktop-Hintergründe

Fluxbox hat zwei Tools zum Einstellen des Hintergrunds: `bsetroot` und `fbsetbg`. Diese funktionieren folgendermaßen:

`bsetroot` ist ungefähr das Äquivalent von `xsetroot`. Es kann Hintergründe einstellen, so lange diese "einfach" sind - zum Beispiel eine solide Farbfüllung. `bsetroot` kann auch Farbverläufe als Hintergrund einstellen.

`fbsetbg` ist ein Wrapper der versucht ein passendes Programm zum Einstellen des Hintergrundes zu finden und überläßt diesem dann die Arbeit.

## Theme-Hintergründe übergehen

Fluxbox erlaubt es das Desktopbild unabhängig vom Theme einzustellen, indem man dies in der `~/.fluxbox/init` Datei angibt:

```
session.screen0.rootCommand: fbsetbg -f ~/backgrounds/zimdib_dark.png
```

Um das Editieren der `~/.fluxbox/init` zu vermeiden, schreibt `fbsetbg` das zuletzt verwendete Bild in die Datei `~/.fluxbox/lastwallpaper` und lädt dieses Bild wenn es mit der option `-l` aufgerufen wird. Das `rootCommand` wäre dann.

```
session.screen0.rootCommand: fbsetbg -l
```

# Kapitel 7. Die Slit

Eine der am häufigsten gestellten Fragen lautet: "Was ist die Slit?". Im Endeffekt habe ich die Frage ganz nach oben in die FAQ kopiert um dafür zu sorgen, dass danach nicht mehr 10mal täglich in #fluxbox gefragt wird, und um dem Mythos ein Ende zu machen "Slit ist nur ein anderer Name für die Toolbar".

Die Slit ist einer der vielen Teile von Fluxbox, die von Blackbox übernommen wurden. Sie ist dazu gedacht, WindowMaker Dockapps aufzunehmen, (sowie alles andere, was im sogenannten 'withdrawn' ('zurückgezogen') oder 'swallowed' Modus läuft). Solche Anwendungen haben oft eine Option **-w**, manche laufen aber auch automatisch im 'withdrawn'-Modus.

Zuerst muss sichergestellt sein, dass die Slit in Ihrer Version von Fluxbox mit kompiliert wurde. Normalerweise kann man davon ausgehen dass dies der Fall ist. Soweit mir bekannt ist, enthalten alle Pakete verschiedener Distributionen Fluxbox mit Slit. Wenn Sie denken, dass Sie ohne Slit besser zurechtkommen, kann diese vor dem Kompilieren abgeschaltet werden. Bedenken Sie jedoch, dass eine leere Slit *keinen* Platz auf dem Bildschirm benötigt, und auch nur wenig Speicher. Daher ist es im Normalfall nur notwendig, die Slit komplett abzuschalten, wenn sie mit anderer Software zu Konflikten führt.

Sie können jede 'dockfähige' Anwendung ausführen. Zum Beispiel enthält `xmms` immer das Tool `wmxmmms`. Führen Sie einfach `wmxmmms &` aus, und das Programm erscheint in der Slit. Wie oben erwähnt, benötigen manche Anwendungen (zum Beispiel `gkrellm`) die Option **-w**, damit sie in der Slit auftauchen.

## Wo bekomme ich Dockapps?

Ein guter Ausgangspunkt ist the Dockapp warehouse(<http://www.bensinclair.com/dockapp>). Dort gibt es eine sehr große Sammlung verschiedener Dockapps. Außerdem könnten Sie bei Freshmeat(<http://freshmeat.net/>) schauen, oder in den Paketen Ihrer Distribution nach Dockapps suchen.

Die `bbtools`(<http://bbtools.sourceforge.net>) Website enthält viele Blackbox/Fluxbox Tools, von denen die meisten in der Slit laufen können.

Wenn Sie KDE-Unterstützung aktiviert haben, erscheinen auch KDE Dock-Applets in der Slit.

`Dockapps.Org`(<http://www.dockapps.org/>) ist eine neue Website, die ausschließlich Dockapps gewidmet ist.

## Kann ich das Verhalten der Slit beeinflussen?

Selbstverständlich. Klicken Sie einfach mit der rechten Maustaste auf den sichtbaren Teil der Slit, und wählen Sie dort die gewünschten Optionen. Die meisten Optionen sind genau identisch mit denen der

Taskbar. Die einzige nicht identische Option ist `Direction`. Die Slit kann entweder `Horizontal` oder `Vertical` laufen.

Es gibt außerdem eine **Maximize Over Slit** Option im Konfigurationsmenü - diese erlaubt es maximierten Fenstern, die Slit zu verdecken.

## Ich will dass meine Dockapps ihre Reihenfolge beibehalten!

Ab Fluxbox 0.1.10 ist dies möglich, indem eine Datei namens `slitlist` verwendet wird. Hier einige Anweisungen zur Benutzung:

Die aktuelle Anordnung der Dockapps wird in einer Datei gespeichert, üblicherweise `~/.fluxbox/slitlist`. Wenn Dockapps in die Slit geladen werden, versucht Fluxbox die vorherige Anordnung wiederherzustellen. Dazu werden die Namen früher schon einmal ausgeführter Dockapps abgeglichen.

Es gibt eine einfache Lösung, wie Sie die Dockapps so anordnen können wie Sie möchten:

### Dockapps anordnen

1. Starten Sie Fluxbox ohne Dockapps.
2. Starten Sie Ihre Dockapps individuell in der Reihenfolge, die Sie wünschen.
3. Fügen Sie Ihre Dockapps wieder zum Autostart-Skript hinzu, zum Beispiel `.xinitrc` oder `.xsession`. Hier ist die Reihenfolge egal.

Diese Reihenfolge wird standardmäßig in `~/.fluxbox/slitlist` gespeichert und wird dann in zukünftigen Fluxbox-Sessions beibehalten.

Sie können manuell die Datei `slitlist` bearbeiten. Es handelt sich dabei um eine simple Liste von Fensternamen, pro Dockapp eine Zeile. Diese Datei sollte bearbeitet werden, während Fluxbox *nicht* läuft. Ansonsten könnten die Änderungen möglicherweise überschrieben werden.

Sie haben auch die Möglichkeit, einen anderen Ort für `slitlist` zu wählen. Folgende Zeile in `init` ändert den Pfad:

### Beispiel 7-1. Slitlist Pfadangabe

```
session.slitlistFile: /home/me/etc/slitsort
```

Beachten Sie bitte, dass es keine Option gibt um das Sortieren gänzlich abzuschalten. Wir sehen in zufälliger Sortierung keinen besonderen Vorteil.

# Kapitel 8. Die Toolbar

Die Toolbar ist ein kleiner Bereich, in dem Fluxbox Informationen anzeigt - zum Beispiel eine Uhr, oder Buttons für laufende Programme.

Sie wird durch die init-Datei und die style-Datei konfiguriert. Init-Einstellungen werden normalerweise über das Toolbar-Menü verändert.

Die Toolbar kann abgeschaltet (unsichtbar gemacht) werden, indem man in Fluxbox's init-Datei `session.screen0.toolbar.visible: false` einträgt.

Die Tools, die angezeigt werden sollen, werden normalerweise in `.fluxbox/init` eingestellt, und zwar durch den Eintrag `toolbar.tools`.

## Beispiel 8-1. Toolbar Tools Beispiel

```
session.screen0.toolbar.tools: clock, iconbar, workspacename
```

Verfügbare tools: `workspacename`, `prevworkspace`, `nextworkspace`, `iconbar`, `systemtray`, `prevwindow`, `nextwindow`, `clock`

Die Breite der Toolbar, Transparenz und Ebene können über das Toolbar-Menü eingestellt werden. Das Toolbar-Menü wird angezeigt, wenn man mit der rechten Maustaste auf die Uhr oder den Workspace-Namen in der Toolbar klickt.

Die Iconbar kann so eingestellt werden: keine Icons; Icons von allen Workspaces; Workspace Icons; alle Fenster vom aktuellen Workspace; alle Fenster von allen Workspaces.

Das Aussehen der Toolbar wird im Style-File festgelegt.

# Kapitel 9. Menüs bearbeiten

Sie haben Fluxbox installiert und sehen nun das praktische Menü, das bei einem Rechtsklick auf Ihr Desktop erscheint. Dieses Menü ist nicht sehr nützlich, wenn Sie es nicht bearbeiten können. Dieses Dokument versucht, alle diesbezüglichen Fragen zu beantworten.

Es gibt ein schönes Tool namens `fluxbox-generate_menu`, das mit Fluxbox zusammen ausgeliefert wird. Es versucht, aus der Systemumgebung die Pfade von gängigen Programmen (zum Beispiel Browser, Terminals) zu ermitteln und erzeugt eine Menü-Datei. `fluxbox-generate_menu` wird detailliert in Abschnitt 3.3 behandelt.

## 9.1. Den Ort der Menüdatei einstellen

Das Fluxbox-Menü befindet sich per Standard in `~/.fluxbox/menu`. Diese Einstellung kann in Ihrer `init`-Datei verändert werden. Hier ist ein Beispiel für die entsprechende Zeile:

### Beispiel 9-1. Den Ort der Menüdatei vorgeben

```
session.menuFile:      ~/.fluxbox/menu
```

Ändern Sie einfach `~/.fluxbox/menu`, wenn Sie eine andere Datei für Ihr Menü verwenden möchten. Die aktuelle Struktur sollte aber für die meisten User in Ordnung sein.

## 9.2. Verfügbare Befehle

Das Fluxbox-Menü ist einfach nur eine Textdatei, die es Ihnen erlaubt Unterordner zu erzeugen, Anwendungen zu starten, Workspaces zu steuern, Fluxbox einzustellen und X zu beenden. Das Menü kennt folgende Befehle:

```
[begin] (MenuTitle)
[submenu] (SubMenuName) {SubMenuTitle}
[exec] (ApplicationName) {/path/to/program}
[include] (/path/to/menufile)
[end]
[nop] (-----)
[workspaces] (SubMenuName)
[stylesdir] (/path/to/stylesdir)
[config] (FluxboxConfiguration)
[reconfigure] (Reconfigure)
[restart] (Restart)
[exit] (Exit)
```

Das meiste hiervon sollte selbsterklärend sein. Die kursiven Elemente sind die Worte, die auf dem Bildschirm erscheinen. Ich werde die schwieriger zu verstehenden Elemente weitergehend erklären. Beachten Sie, dass es für die Verschachtelungstiefe von Untermenüs keine feste Grenze gibt - höchstens eine Grenze bei der Benutzbarkeit.

**[nop]** - Erlaubt es, einfach Text oder eine Leerzeile ins Menü einzufügen. Dieser Eintrag führt nichts aus, sondern funktioniert nur als Trennelement im Menü.

**[reconfigure]** - Wenn Sie die Konfiguration von Fluxbox über das Menü verändern, gehen diese Änderungen verloren wenn Sie Fluxbox beenden. Sie müssen also in die `init`-Datei geschrieben werden um zu permanenten Änderungen zu werden. Dies erreichen Sie, indem Sie 'reconfigure' anklicken, nachdem Sie Änderungen an der Konfiguration vorgenommen haben.

**[restart]** - Nur zur Klarstellung: 'restart' startet *nur* Fluxbox neu, nicht das gesamte System.

### Beispiel 9-2. Beispiel einer Menüdatei

```
# Fluxbox menu file
[begin] (Fluxbox)
  [exec] (rxvt) {rxvt -ls}
  [exec] (netscape) {netscape -install}
  [exec] (The GIMP) {gimp}
  [exec] (XV) {xv}
  [exec] (Vim) {rxvt -geometry 132x60 -name VIM -e screen vim}
  [exec] (Mutt) {rxvt -name mutt -e mutt}
  [submenu] (mozilla)
    [exec] (browser) {mozilla -browser}
    [exec] (news) {mozilla -news}
    [exec] (mail) {mozilla -mail}
    [exec] (edit) {mozilla -edit}
    [exec] (compose) {mozilla -compose}
  [end]
  [submenu] (Startup)
    [exec] (gkrellm) {gkrellm -w}
    [exec] (xmms) {xmms -p}
    [exec] (galeon) {galeon -s}
    [exec] (kdeinit) {kdeinit}
  [end]
  [submenu] (Window Manager)
    [exec] (Edit Menus) {nedit ~/.fluxbox/menu}
    [submenu] (Style) {Which Style?}
      [stylesdir] (~/.fluxbox/styles)
      [stylesmenu] (Fluxbox Styles) {/usr/local/share/fluxbox/styles}
    [end]
  [config] (Config Options)
  [reconfig] (Reconfigure)
  [restart] (Restart)
[end]
[exit] (Log Out)
[end]
```

```
# end of menu file
```

Für weitergehende Beispiele schauen Sie sich einfach die recht komplette menu-Datei an, die bei Fluxbox dabei ist.

# Kapitel 10. Themes

## 10.1. Theme basics

This section was contributed by Justin Rebelo(<mailto:rebelo@shaw.ca>) aka "demerol".

### What is a style and how does it work?

A style is basically a theme for Fluxbox. It is a simple ASCII text file that tells Fluxbox how to generate the appearance of the different components of the window manager. They are usually located in `~/.fluxbox/styles` and in the global Fluxbox share directory, which will vary depending on the method of installation used.

### How do I make my own?

Start off by opening a style in your favorite text editor (I recommend `vim`). Look at the style, how it is structured and organized. Just looking at it will explain most of the questions you may have.

### Structure of a style

The style is made up of a few major components which then have their own sub-directives. The **toolbar**, **menu** and **window** are the major components. The **window.\*** directives control the appearance of your window frames, **window.tab.\*** controls the appearance of the window tabs. **menu.\*** controls the appearance of the popup menu that you see when you right click on the desktop. **toolbar.\*** is the bar you will see at the top or bottom of your screen. The Slit (also called a dock, wharf, etc in other WMs) is controlled by the **toolbar** settings, too if you don't set its style specifically.

### How do I change the appearance of the slit?

The slit usually uses the same options as the toolbar. Most of the time this should work reasonably well. If you want to specifically style the slit, there are three style directives you can use:

```
slit: [texture option]
slit.color: [color value]
slit.colorTo: [color value]
```

These commands work just like those for menu, window etc. when texturing the Slit.

### Can I set a background image/color?

Somewhere in the style file you might see a line with **rootCommand** at the beginning and it will be followed by a command to set the background color or image of the style. Fluxbox includes the program `bsetroot` to set color and gradient background and `fbsetbg` to display images.

## Can I add note/comments in my styles?

Sure, just start a line with a hash (#), a bang(!), or use C++ style comments (//).

## I still have more questions...

Take a look at the styles provided with Fluxbox, you should be able to find the answer there or by trying different settings. If you still can't get it, stop by #fluxbox on OPN. My nick is demerol.

## 10.2. Advanced Theme Editing

The majority of this section is taken from or heavily influenced by the man page as of Fluxbox 0.1.13. Usually, the man page is the definite authority on this matter, but this document might be more enlightening to the beginning style author.

To understand how the style mechanism works, it is nice to know a little about how X11 resources work.

X11 resources consist of a key and a value. The key is constructed of several smaller keys (sometimes referred to as children), delimited by a period (.). Keys may also contain an asterisk (\*) to serve as a wildcard, which means that one line of typed text will match several keys. This is useful for styles that are based on one or two colors.

Fluxbox allows you to configure its three main components: the toolbar, the menus and the window decorations. The slit automatically inherits its style from the toolbar but can be styled differently if need be. The little window that shows the x-y position while dragging windows, borrows its style from the window's titlebar.

Here are some quick examples to illustrate the basic syntax:

### Beispiel 10-1. Toolbar Clock style

```
toolbar.clock.color: green
```

This sets the color resource of the toolbar clock to **green**. Another example:

### Beispiel 10-2. Menu style

```
menu*color: rgb:3/4/5
```

This sets the color resource of the menu and all of its children to **rgb:3/4/5**. For a description of color names, see the X11 man pages. So this one also applies to **menu.title.color** and **menu.frame.color**. And with this:

**Beispiel 10-3. Font style**

```
*font: -b&h-lucida-medium-r-normal-***-140-*
```

you set the font resource for all keys to this font name all at once. For information about the fonts installed on your system, you can use a program like `xfontsel`, `gfontsel`, or `xlsfonts`.

Now, what makes Fluxbox just so spectacular, is its ability to render textures on the fly. Texture descriptions are specified directly to the key that they should apply to, e.g.:

**Beispiel 10-4. Texture style**

```
toolbar.clock:   Raised Gradient Diagonal Bevel1
toolbar.clock.color:  rgb:8/6/4
toolbar.clock.colorTo: rgb:4/3/2
```

Don't worry, we will explain how these directives work. A texture description consists of up to five fields, which are as follows:

**Tabelle 10-1. Texture directives**

Directive	Description
Flat / Raised / Sunken / Tiled	give the component either a flat, raised or sunken appearance. Tiled does only affect pixmaps and it will not scale them.
Gradient / Solid	draw either a solid color or a gradiented texture.
Horizontal / Vertical / Diagonal / Crossdiagonal / Pipecross / Elliptic / Rectangle / Pyramid	Select one of these texture types. They only work when Gradient is specified, too.
Interlaced	interlace the texture (darken every other line). This option is most commonly used with gradiented textures, but from Blackbox version 0.60.3 on (and thus in all versions of Fluxbox), it also works in solid textures.
Bevel1 / Bevel2	type of bevel to use. Bevel1 is the default bevel. The shading is placed on the edge of the image. Bevel2 is an alternative. The shading is placed one pixel in from the edge of the image.

Apart from the texture description, the option **ParentRelative** is also available, which makes the component appear as a part of its parent.

All gradiented textures are composed of two color values: the **color** and **colorTo** resources. When Interlaced is used in **solid** mode, the **colorTo** resource is used to find the interlacing color.

The complete list of components and which kind of value they can contain can be found in Anhang C.

Now, that seems a long list, but remember, when you create your own style, you can easily set lots of keys with a single command, e.g.

**Beispiel 10-5. Typical short style**

```
*color:          slategrey
*colorTo:        darkslategrey
*unfocus.color: darkslategrey
*unfocus.colorTo: black
*textColor:      white
*unfocus.textColor: lightgrey
*font:           lucidasans-10
```

# Anhang A. Einrichten von `.xinitrc/.xsession`

Contributed by Verin.

## *Die Bedeutung von xinit*

Ein window manager ist einfach nur eine Anwendung für X11, wie `netscape` oder `gimp` oder `xterm`. Viele Neulinge in Sachen X11 glauben, dass X11 den window manager ausführt und dass der window manager Programme ausführt. Das ist so nicht richtig. Wenn alles richtig eingestellt ist, kann man alle Anwendungen unter X11 starten, den window manager beenden und einen anderen window manager starten.

Das *wirkliche* Programm was X11 ausführt, welches dann wiederum andere Programme ausführt, ist Ihr `.xinitrc` oder `.xsession` Skript. Wenn X11 gestartet wird, wird das `.xinitrc` oder `.xsession` Skript ausgeführt und wenn das Skript beendet ist, wird auch X11 beendet. Lassen Sie mich dies wiederholen, es ist wichtig: *wenn `.xinitrc` zu Ende gelaufen ist, endet auch X11*. Dies ist NICHT der Punkt, wo der window manager die Ausführung beendet.

## *Das Skript*

Machen Sie sich zuerst folgendes klar: wenn Sie in einer Shell ein Kommando eingeben, können Sie so lange nichts anderes tun bis dieses Kommando beendet ist. Das `.xinitrc` oder `.xsession` Skript funktioniert genauso. Wenn es während seiner Ausführung auf ein Programm trifft, das lange läuft (so wie die meisten X11-Programme), wartet es genau an dieser Stelle so lange, bis das aufgerufene Programm beendet ist.

Im Idealfall sollte Ihr Skript genau *einen* solchen Punkt haben, an dem es 'wartet', und üblicherweise sollte dies am Ende des Skripts geschehen. Das bedeutet: wenn Sie andere X11-Programme ausführen wollen, bevor Sie diesen Haltepunkt erreichen, sollten diese im Hintergrund laufen. Dies erreichen Sie, indem Sie ans Ende jeder Zeile ein `&` setzen. Wenn Sie also zum Beispiel `xclock` zusätzlich zu allem anderen ausführen wollen, setzen Sie diese Zeile vor Ihren Haltepunkt:

```
xclock &
```

Das nächste was Sie sehen ist der `exec` Befehl, zu dem viele Quellen viele verschiedene Empfehlungen geben. Im Grunde ist dieser Befehl nicht notwendig, es reicht prinzipiell aus wenn Sie ihren Windowmanager in der letzten Zeile des Skripts starten - das Skript wird dann an diesem Punkt wie gewünscht 'warten'.

Warum dann überhaupt der `exec`-Befehl? Nun, angenommen Sie wollen mehrere Zeilen in Ihr Skript eintragen die einen Windowmanager starten, aber es soll nur eine davon funktionieren. Mit `exec` können Sie Ihre auserwählte Zeile ganz nach oben setzen. `exec` bedeutet nämlich folgendes:

„Ersetze 'mich selbst' mit diesem Programm, d.h. starte es und beende mich selbst, sobald das Programm beendet ist.“

Wenn Sie also zum Beispiel eine Zeile `exec wmaker` oberhalb einer Zeile `exec enlightenment` im Skript eintragen, endet das Skript wenn `wmaker` endet - es erreicht nie die nächste Zeile.

Sehen Sie, was ich mit "unnötig" meinte? Sie könnten genausogut eine Menge auskommentierter Windowmanager-Zeilen eintragen, und alles würde ganz genauso funktionieren.

### *Eine andere Methode*

Als Alternative könnten Sie auch den Windowmanager *zuerst* starten, und dessen process ID in einer Umgebungsvariablen speichern:

```
wmaker & wmpid=$!
```

Dies versetzt `wmaker` in den Hintergrund (`&`) und speichert die process ID (`$!`) in einer Variablen (`wmpid`). Dann verwenden Sie `wait`, um den Haltepunkt zu erzeugen:

```
wait $wmpid
```

oder Sie verwenden ein Programm welches Sie immer verwenden wollen, zum Beispiel `gkrellm` als Haltepunkt, indem Sie es einfach nicht in den Hintergrund versetzen. Beachten Sie jedoch in diesem Falle: sobald Sie dieses Programm beenden, endet auch Ihre X11-Session.

Ich persönlich verwende die `wait`-Methode, da ich gern vor dem Start meiner Anwendungen meinen Windowmanager auswähle. Außerdem ändere ich gern zuvor die Einstellungen meines X11-Servers, zum Beispiel Auflösung, Bildschirmschoner oder Fonts. Außerdem stelle ich gern nachher die alten Einstellungen wieder her, um sicherzugehen dass alles wieder im ursprünglichen Zustand ist - zum Beispiel haben Displaymanager manchmal Probleme, den Font-Path zurückzusetzen.

### **Beispiel A-1. `.xinitrc`**

```
# Bildschirmabschaltung aus- und energy star features einschalten
xset s off
xset dpms 600 60 60

# meine optionalen Fonts zum Font-Pfad hinzufügen
xset +fp "$X_FONTPATH"
xset fp rehash

# aktuelle Umgebungsvariablen speichern, falls Debugging nötig ist
env > ~/.xenv

# window manager
fluxbox & wmpid=$!
```

```
bbrun &  
wmCalClock &  
wmxmms &  
  
# HANG POINT - abwarten, bis der window manager beendet ist  
wait $wmpid  
  
# X Standard-Font-Pfad wiederherstellen  
xset fp default
```

# Anhang B. Frequently asked questions (FAQ)

## 1. Was ist die Slit?

Zuerst muss klar sein, dass die Slit *nicht* die Fluxbox Taskbar ist.

Die Slit ist ein Ort, wo dock-fähige Anwendungen 'andocken' können. Es gibt ein eigenes Kapitel zum Thema Slit: Kapitel 7. Bitte lesen Sie dieses Kapitel, bevor Sie Fragen zur Slit und ihrer Funktionsweise stellen.

## 2. Kann ich meine Slit Dock-Apps in eine bestimmte Reihenfolge bringen?

Ja, in Fluxbox-Versionen nach 0.1.10. In Kapitel 7 finden Sie eine Erklärung.

## 3. Wie ändere ich das Format der Uhr in der Toolbar?

Ändern Sie diese Zeile in Ihrer `init`-Datei:

```
session.screen0.strftimeFormat: %a %d %H:%M
```

Für Informationen über das Format lesen Sie die manpage von `strftime`, indem Sie **man 3 strftime** aufrufen.

## 4. Änderungen an meiner `~/fluxbox/init` werden überschrieben.

Hierbei handelt es sich um einen Bug in Fluxbox-Versionen vor 0.1.8-bugfix2. Bitte führen Sie ein Upgrade auf die neueste Version oder den neuesten Bugfix durch, bevor Sie uns einen Fehler melden.

## 5. Wie funktionieren Tabs?

Siehe Kapitel 4.

## 6. Nachdem ich Anti-Aliasing angeschaltet habe, sind meine Fonts riesig! Wie behebe ich das?

Seit der Ankunft von Xft werden Fonts anders gehandhabt. Probieren Sie als Anfänger, in Ihrem Theme (style-Datei) folgendes einzutragen:

```
*.font: Verdana:size=7
```

Natürlich können Sie eine andere Schriftart und/oder Größe verwenden, das angegebene Beispiel sollte aber die Menüs benutzbar machen. Für Informationen über die verwendbaren Fonts lesen Sie bitte die nächste Frage.

## 7. Scheinbar kann ich alte Fonts wie z.B. Snap nicht mit Anti-Aliasing verwenden! Wo liegt das Problem?

Sobald in Fluxbox AA eingeschaltet wird, verlässt Fluxbox sich beim Darstellen der Zeichen auf Freetype2. Wenn Sie AA nicht benutzen, kann X11's eingebaute Zeichendarstellung verwendet werden.

Da Freetype2 *nicht* alle Formate unterstützt, die X11 unterstützt, ist die Auswahl von Fonts für Fluxbox eingeschränkt. Momentan unterstützt Freetype2 folgende Formate:

- TrueType files (.ttf) und collections (.ttc)
- Type 1 font files sowohl in ASCII (.pfa) als auch binären (.pfb) Formaten
- Type 1 Multiple Master Fonts
- Type 1 CID-keyed Fonts
- OpenType/CFF (.otf) Fonts
- CFF/Type 2 Fonts
- Adobe CEF Fonts (.cef)
- Windows FNT/FON bitmap Fonts

Andererseits unterstützt Freetype1 nur TrueType fonts, obwohl die Unterstützung für GX und OTF Fonts durch die Verwendung einer anderen Library ermöglicht werden. Details finden Sie auf der Freetype Website.

#### **8. Was sind die vielbesagten Artwiz Fonts?**

Hierzu gibt es eine Erklärung in Anhang D

Wenn Ihnen die Artwiz Fonts gefallen, Sie aber deren Aussehen in Terminals nicht mögen, versuchen Sie das LFP-Font-Paket von Linux Font Project(<http://dreamer.nitro.dk/linux/lfp/>). Dort gibt es zwei Font-Sets, die LFP Fixed-Width Fonts (gut für Terminals), und die LFP Variable-Width Fonts (gut für andere Zwecke). Die fixed-width Fonts sind auch für die Linux-Konsole verfügbar.

#### **9. Wie ändere ich den Hintergrund?**

Dies wird in Kapitel 6 erklärt.

#### **10. Mein Hintergrund wird hässlich, wenn ich ein anderes Theme wähle.**

Eine Lösung steht in Kapitel 6.

#### **11. Kann ich meine bestehende .blackboxrc mit Fluxbox verwenden?**

Auf jeden Fall, Sie müssen nur einige Zeilen für Titlebar und Tastenkombinationen hinzufügen. Vielleicht sollten Sie auch Ihre `blackboxrc` und `~/fluxbox/init` durch einen Symlink verbinden.

#### **12. Wie starte ich automatisch Anwendungen, wenn Fluxbox startet?**

Siehe Anhang A.

#### **13. Kann ich Blackbox Styles (Themes) mit Fluxbox verwenden?**

Ja. Die tarballs für beide Packages sollten 100%ig austauschbar sein. Auch für die Projekte Waimea und Openbox sollte dies weiterhin gelten, es gibt jedoch keine Garantie. Von beiden Projekten ist bisher kein

solches Versprechen abgegeben worden, aber ein Ziel von Fluxbox ist es, mit den Blackbox Styles kompatibel zu bleiben.

**14.** Wie konfiguriere ich `.xinitrc/.xsession`?

Siehe Anhang A.

**15.** Werden KDE-Anwendungen unterstützt?

Ja, verwenden Sie die `configure` Option `--enable-kde`. Dies stellt sicher, dass KDE Tray-Icons in der Slit erscheinen.

**16.** Werden Gnome-Anwendungen unterstützt?

Ja, verwenden Sie die `configure` Option `--enable-gnome`. Dies aktiviert die Gnome hints. In Fluxbox 0.1.12 und höher ist dies der Standard.

**17.** BBtools verwenden nicht meine aktuellen Style-Einstellungen, auch nicht nach einem Neustart.

Linken Sie einfach Ihre `~/blackboxrc` auf die `~/fluxbox/init`, zum Beispiel so:

```
$ ln -s ~/fluxbox/init ~/.blackboxrc
```

**18.** In manchen Styles sehen die Tabs hässlich aus, wie kann ich dies beheben?

*0.1.14 und frühere Versionen.*

Damit die Tabs (noch) schöner aussehen, müssen Sie einige zusätzliche Einträge im gewünschten Style (Theme) vornehmen. Beachten Sie bitte dass sie dies nicht tun MÜSSEN - Fluxbox kann die Tabs gut selbst mit einem passenden Aussehen versehen. Wenn Sie aber mehr Kontrolle über das Aussehen der Tabs haben wollen, fügen Sie einige Zeilen wie diese ein:

**Beispiel B-1. Personalisierte Tabs im Style-File**

```
! -- tab style (for Fluxbox)
window.tab.justify:                Right
window.tab.label.unfocus:        Flat Solid
window.tab.label.unfocus.color:   rgb:AC/AC/AC
window.tab.label.unfocus.textColor: black
window.tab.label.focus:           Raised Solid
window.tab.label.focus.color:      rgb:CC/CC/CC
window.tab.label.focus.textColor:  black
window.tab.borderWidth:           1
window.tab.borderColor:           rgb:10/10/10
window.tab.font:                   fixed
! --- end, tab style
```

Okay, und was bedeuten diese Zeilen? Nun - das gleiche wie alles andere in einem Theme. Ich bin sicher dass Sie all das verstehen, wenn Sie schon einmal ein Blackbox Theme selber gemacht haben (Kapitel 10 dürfte Sie ansonsten interessieren).

Beachten Sie bitte auch, dass ein Style, der diese zusätzlichen Zeilen enthält, trotzdem weiterhin perfekt in Blackbox funktioniert - Sie verlieren also nichts durch das Hinzufügen dieser Zeilen.

**19. Wie kann ich Icons auf mein Fluxbox-Desktop legen?**

Fluxbox hat für solche Zwecke sein eigenes Begleitprogramm, es nennt sich fbdesk(<http://fluxbox.sourceforge.net/fbdesk>). fbdesk ist momentan nicht im Fluxbox-Package enthalten. Es gibt auch Alternativen: das RoxFiler Projekt, idesk(<http://idesk.timmfin.net>), xdesk([http://garuda.newmail.ru/xtdesk\\_e.dhtml](http://garuda.newmail.ru/xtdesk_e.dhtml)).

**20. Fluxbox 0.9.6 ist laaangsam...**

Fügen Sie folgende Zeile zu Ihrer `.xinitrc` hinzu, bevor Sie Fluxbox ausführen:

```
export LC_ALL=C
```

Dies sollte bei neueren RedHats helfen.

# Anhang C. Theme Reference \*

## C.1. Theme directives

This is a complete list of available theme directives specifying which values they can be assigned. See Kapitel 10 for details on those directives.

**Tabelle C-1. Toolbar styles**

toolbar	Texture
toolbar.height	Number
toolbar.color	Color
toolbar.colorTo	Color
<b>Buttons</b>	
toolbar.button	Texture or ParentRelative
toolbar.button.color	Color
toolbar.button.colorTo	Color
<b>Color of unpressed button arrows</b>	
toolbar.button.picColor	Color
<b>Buttons in pressed state</b>	
toolbar.button.pressed	Texture (e.g. <b>Sunken</b> ) or <b>ParentRelative</b>
toolbar.button.pressed.color	Color
toolbar.button.pressed.colorTo	Color
<b>Color of pressed button arrows</b>	
toolbar.button.pressed.picColor	Color
<b>Workspace label</b>	
toolbar.label	Texture or <b>ParentRelative</b>
toolbar.label.color	Color
toolbar.label.colorTo	Color
toolbar.label.textColor	Color
<b>Workspace label</b>	
toolbar.workspace	Texture or <b>ParentRelative</b>
toolbar.workspace.pixmap	Pixmap
toolbar.workspace.color	Color
toolbar.workspace.colorTo	Color
toolbar.workspace.textColor	Color
toolbar.workspace.font	Font
<b>Window label</b>	
toolbar.windowLabel	Texture or <b>ParentRelative</b>
toolbar.windowLabel.color	Color

**Window label**

toolbar.windowLabel.colorTo	Color
toolbar.windowLabel.textColor	Color

**Clock**


---

toolbar.clock	Texture or <b>ParentRelative</b>
toolbar.clock.pixmap	Pixmap
toolbar.clock.color	Color
toolbar.clock.colorTo	Color
toolbar.clock.textColor	Color
toolbar.clock.font	Font

**Iconbar if empty**


---

toolbar.iconbar.empty	Texture or <b>ParentRelative</b>
toolbar.iconbar.empty.pixmap	Pixmap
toolbar.iconbar.empty.color	Color
toolbar.iconbar.empty.colorTo	Color
toolbar.iconbar.empty	Texture or <b>ParentRelative</b>

**Iconbar focused and unfocused**


---

toolbar.iconbar.focused	Texture or <b>ParentRelative</b>
toolbar.iconbar.focused.pixmap	Pixmap
toolbar.iconbar.focused.color	Color
toolbar.iconbar.focused.colorTo	Color
toolbar.iconbar.focused.textColor	Color
toolbar.iconbar.focused.font	Font

**Text**


---

toolbar.justify	<b>center, left, or right</b>
toolbar.font	Font

**Tabelle C-2. Menu styles****Title**


---

menu.title	Texture
menu.title.color	Color
menu.title.colorTo	Color
menu.title.textColor	Color
menu.title.font	Font
menu.title.justify	<b>center, left, or right</b>

**Frame**


---

menu.frame	Texture
menu.frame.color	Color
menu.frame.colorTo	Color
menu.frame.textColor	Color

**Frame**

menu.frame.disableColor	Color
menu.frame.font	Font
menu.frame.justify	<b>center, left, or right</b>

**Submenu bullets**


---

menu.bullet	<b>empty, triangle, square, or diamond</b>
menu.bullet.position	<b>right or left</b>
menu.submenu.pixmap	Pixmap

**Highlighted item**


---

menu.hilite	Texture (e.g. <b>Raised</b> )
menu.hilite.color	Color
menu.hilite.colorTo	Color
menu.hilite.textColor	Color
menu.selected.pixmap	Pixmap
menu.unselected.pixmap	Pixmap

**Tabelle C-3. Window styles****Title**


---

window.title.focus	Texture
window.title.focus.color	Color
window.title.focus.colorTo	Color
window.title.unfocus	Texture
window.title.unfocus.color	Color
window.title.unfocus.colorTo	Color
window.title.height	Number

**Label**


---

window.label.focus	Texture or <b>ParentRelative</b>
window.label.focus.color	Color
window.label.focus.colorTo	Color
window.label.focus.textColor	Color
window.label.unfocus	Texture or <b>ParentRelative</b>
window.label.unfocus.color	Color
window.label.unfocus.colorTo	Color
window.label.unfocus.textColor	Color

**Tabelle C-4. Handlebar styles**

window.handle.focus.color	Color
window.handle.focus.colorTo	Color
window.handle.unfocus	Texture

window.handle.unfocus.color	Color
window.handle.unfocus.colorTo	Color

**Tabelle C-5. Resize grip styles**

window.grip.focus	Texture
window.grip.focus.color	Color
window.grip.focus.colorTo	Color
window.grip.unfocus	Texture
window.grip.unfocus.color	Color
window.grip.unfocus.colorTo	Color

**Tabelle C-6. Window button styles**

window.button.focus	Texture or <b>ParentRelative</b>
window.button.focus.color	Color
window.button.focus.colorTo	Color
window.button.focus.picColor	Color
window.button.unfocus	Texture or <b>ParentRelative</b>
window.button.unfocus.color	Color
window.button.unfocus.colorTo	Color
window.button.unfocus.picColor	Color
window.button.pressed	Texture
window.button.pressed.color	Color
window.button.pressed.colorTo	Color

**Tabelle C-7. Window frame styles**

window.frame.focusColor	Color
window.frame.unfocusColor	Color

**Tabelle C-8. Tab styles**

window.tab.justify	<b>right, left</b> or <b>center</b>
window.tab.label.unfocus	Texture
window.tab.label.unfocus.color	Color
window.tab.label.unfocus.textColor	Color
window.tab.label.focus	Texture
window.tab.label.focus.color	Color
window.tab.label.focus.textColor	Color

window.tab.borderWidth	Number of Pixels
window.tab.borderColor	Color
window.tab.font	Font

**Tabelle C-9. Font on window label styles**

window.font	Font
window.justify	<b>center, left, or right</b>

**Tabelle C-10. Miscellaneous styles**

**Border drawn around all components**

---

borderWidth	Number of pixels
borderColor	Color
bevelWidth	Number of pixels
handleWidth	Number of pixels
frameWidth	Number of pixels

**Command to be executed whenever the style is loaded**

---

rootCommand	Shell command, e.g. <b>fbsetbg nicepiccy.jpg</b>
-------------	--

**Old BB 0.51 resources**

---

menuFont	Font
titleFont	Font

# Anhang D. Artwiz Fonts für Fluxbox

## Einleitung

Die sogenannten Artwiz Fonts wurden von jemandem erfunden, der sich (überraschenderweise) Artwiz nennt. Han hat freundlicherweise einen tarball mit den Fonts auf seine Website gestellt. Die Fonts kann man unter <http://www.xs4all.nl/~hanb/software/fluxbox/artwiz-fonts.tar.bz2> herunterladen.

Sie müssen die Fonts nicht von Hand installieren, wenn Sie die Mandrake RPMs haben. Diese sind im tarball enthalten. Falls Sie die RPMs nicht haben, können Sie die Fonts wie folgt von Hand installieren. Es gibt zwei Wege, dies zu tun: systemweit oder nur für einen einzelnen User.

## System-weite Installation

Wenn Sie die Fonts für alle User auf Ihrem System installieren wollen, speichern Sie den tarball unter /tmp und gehen dann so vor:

```
# cd /usr/X11R6/lib/X11/fonts
# tar xjf /tmp/artwiz-fonts.tar.bz2
# cd fluxbox-artwiz-fonts
# mkfontdir
# chkfontpath -q -a /usr/X11R6/lib/X11/fonts/fluxbox-artwiz-fonts:unscaled
```

Danach müssen Sie den Font-Server neu starten. Bitte beachten Sie, dass Ihre Distribution möglicherweise systemweite Fonts an einer anderen Stelle ablegt, zum Beispiel /usr/share/fonts. Dieses Verzeichnis ist aber allgemein ein sinnvoller Standardwert.

## Installation für nur einen User

Wenn Sie die Fonts für nur einen User installieren wollen, geht es etwas einfacher. Speichern Sie den tarball in Ihrem Home-Verzeichnis und gehen dann so vor:

```
$ tar xjf artwiz-fonts.tar.bz2
$ mv fluxbox-artwiz-fonts .fonts
$ mkfontdir $HOME/.fonts
```

Bearbeiten Sie Ihre .xinitrc oder .xsession Datei (je nachdem, wie Sie X11 starten), so dass diese ganz zu Anfang folgende Zeile enthält:

```
xset +fp $HOME/.fonts
```

Danach müssen Sie X11 (neu)starten und mittels xlsfonts oder xfontsel überprüfen, ob die Fonts vom System erkannt wurden.

## Bugs

Die Artwiz Fonts führen manchmal zu Konflikten mit Ihren Locale-Einstellungen. Um Sie zum Laufen zu bringen, müssen Sie eventuell am Anfang Ihrer `.xinitrc` oder `.xsession` folgendes einfügen:

```
export LC=C
export LC_ALL=C
```

Diese Einstellungen betreffen Locales - sollten also später Fonts fehlen oder andere Locale-Probleme auftreten, entfernen Sie diese Zeilen wieder. `aleczapka` hat noch eine andere Lösung parat, die nun beschrieben wird.

Hier ist die Anleitung, wie man die nationalen Einstellungen unverändert benutzen und gleichzeitig die Artwiz Fonts mit Fluxbox verwenden kann:

## Locale Settings Fix

Die Lösung ist extrem einfach. Alles was Sie tun müssen ist eine Korrektur der `fonts.alias` (und/oder `fonts.dir`) Datei(en).

Dieser Schritt wird auch Probleme mit anderen Anwendungen (z.B. Evolution und UTF-8) beheben. Zuerst müssen Sie eine korrekte `fonts.dir` Datei erzeugen. Diese sollte sich in dem Verzeichnis befinden, wo Sie die Artwiz Fonts installiert haben. Wenn nicht, wechseln Sie in dieses Verzeichnis und führen `mkfontdir` aus.

Die Syntax dieser Datei ist einfach. Die erste Zeile enthält nur die Anzahl der Fonts in dem Verzeichnis. Alle folgenden Zeilen haben die Form

```
font_filename fontname
```

Hier ist ein Beispiel für eine `fonts.dir` Datei:

### Beispiel D-1. `fonts.dir`

```
14
glisp.pcf.gz -artwiz-glisp-medium-r-normal--11-110-75-75-p-90-iso646.1991-irv
gelly.pcf.gz -artwiz-gelly-medium-r-normal--10-100-75-75-p-90-iso646.1991-irv
edges.pcf.gz -artwiz-edges-medium-r-normal--10-100-75-75-m-50-iso646.1991-irv
nu.pcf.gz nu
drift.pcf.gz drift
cure.pcf.gz cure
aqui.pcf.gz aqui
lime.pcf.gz -artwiz-lime-medium-r-normal--10-100-75-75-m-50-iso646.1991-irv
snap.pcf.gz -artwiz-snap-medium-r-normal--10-100-75-75-p-90-iso646.1991-irv
```

Was uns interessiert, sind Einträge die nur aus einer kurzen fontspec bestehen - in diesem Fall die Fonts Nu, Drift, Cure und Aqui. Das Problem ist, dass diesen Fonts vollständige X11 Font-Namen fehlen.

Ändern Sie die Datei zu folgendem:

```
14
glisp.pcf.gz -artwiz-glisp-medium-r-normal--11-110-75-75-p-90-iso646.1991-irv
gelly.pcf.gz -artwiz-gelly-medium-r-normal--10-100-75-75-p-90-iso646.1991-irv
edges.pcf.gz -artwiz-edges-medium-r-normal--10-100-75-75-m-50-iso646.1991-irv
nu.pcf.gz -artwiz-nu-medium-r-normal--11-110-75-75-p-90-iso646.1991-irv
drift.pcf.gz -artwiz-drift-medium-r-normal--11-110-75-75-p-90-iso646.1991-irv
cure.pcf.gz -artwiz-cure-medium-r-normal--11-110-75-75-p-90-iso646.1991-irv
aqui.pcf.gz -artwiz-aqui-medium-r-normal--11-110-75-75-p-90-iso646.1991-irv
lime.pcf.gz -artwiz-lime-medium-r-normal--10-100-75-75-m-50-iso646.1991-irv
snap.pcf.gz -artwiz-snap-medium-r-normal--10-100-75-75-p-90-iso646.1991-irv
```

Als letztes müssen Sie Ihre `fonts.alias` Datei korrigieren (um die Fonts mit anderen Encodings als `iso646` zu benutzen).

Die Syntax ist **font\_alias font\_name**. Um also zum Beispiel die Artwiz Fonts mit ISO-8859-2 zu benutzen, tragen Sie wie folgt einen Alias ein (alles in *einer* Zeile).

```
-artwiz-anorexia-medium-r-normal--11-110-75-75-p-90-iso8859-2
-artwiz-anorexia-medium-r-normal--11-110-75-75-p-90-iso646.1991-irv
```

Sie können auch einen Blick auf meine eigenen

`fonts.dir`(<http://fluxbox.sourceforge.net/download/fonts.dir>) und

`fonts.alias`(<http://fluxbox.sourceforge.net/download/fonts.alias>) Files werfen, diese unterstützen

ISO-8859-1, ISO-8859-2, und ISO10646-1 (UTF-8). Wenn Sie alles wie oben angegeben einstellen, sind keine Änderungen an den `LC_*` Flags mehr notwendig. Darüber hinaus werden ALLE Anwendungen (nicht nur Fluxbox) korrekt funktionieren, ohne Fehlermeldungen wie zum Beispiel "can't convert character set" oder ähnliches.

# Anhang E. Debugging \*

## General information

Debugging is a skill which you can learn. There are quite a few good documents that describe how to make a good bug reports. Please read them before you get into action, your bug won't run away and it won't kill you. Rather we want to kill the bug and we want your help :-). So lets go to battle well prepared. Two nice documents are the Bugzilla Bug Reporting HOWTO(<http://www.mozilla.org/quality/bug-writing-guidelines.html>) and Simon Tatham's How to Report Bugs Effectively(<http://www.chiark.greenend.org.uk/~sgtatham/bugs.html>).

Due to the nature of some unofficial patches, we can't support them all. This means (for you) that you should check if Fluxbox crashes *without* the patches you applied, too. If the patches of your Fluxbox were applied by some distribution maintainer, contact him about the issue.

If you require more help with debugging than this guide provides, you may also see if one of people on #fluxbox at irc.freenode.net is of help. As the main developers and quite a bunch of qualified people are usually there, chances are that we can sort things out.

## Specific Fluxbox things

### Output from Fluxbox

Like any other application Fluxbox sends all messages to the console it was started from. The problem is that under normal circumstances all output goes to the *text* console. Change your Fluxbox start line to **exec xterm** or whatever your favorite terminal emulator may be. Start X11 and start Fluxbox in that xterm, and lo and behold, you can easily see all messages from Fluxbox.

### The info we want

We want to know a lot of things. Make sure you know all about them.

- Your operating system / distribution and its version.
- Fluxbox version, if CVS (or development tarball) then which day?
- When did it happen? Did you do something? Can you reproduce it?
- Your settings. They can be found in `~/.fluxbox/init`

### What to do with core dumps.

So if you ever get a core dump with Fluxbox do the following: Go to the channel #fluxbox on OPN and tell fluxgen you got a core dump. Give him all the info he asks for. He will probably also ask you to do the following. It's quite a lot of work but anybody with a bit of common sense and some Unix experience can do it. Oh, and you need the GNU debugger, called `gdb`.

## Rebuild Fluxbox

Yes you read it right. To become a real debugger you have to rebuild Fluxbox with debugging symbols.

Build Fluxbox as usual and add the following option when you do the make:

```
$ CFLAGS=-Wall -g3 CXXFLAGS=-Wall -g3 make
```

If you use the source RPMs from the Fluxbox site or your distribution you can do something like this:

```
$ su
# rpm -ivh fluxbox-0.1.11.lmdk.src.rpm
# cd /usr/src/RPM/SPECS
# env DEBUG=true rpm -ba fluxbox.spec
# rpm -Uvh --force /usr/src/RPM/RPMS/i686/fluxbox*
# exit
$ mkdir -p ~/src/fluxbox
$ cp -R /usr/src/RPM/BUILD/fluxbox* ~/src/fluxbox
```

Go to the Fluxbox directory (So we get the core dump in the right place). The shell has a nifty feature that disables core-files so make sure you really get a core file with:

```
$ ulimit -c unlimited
```

Start X11 and let's debug. Do whatever it takes to make Fluxbox dump core. And now we start debugging:

```
$ gdb fluxbox core
```

And issue this command in `gdb` (the first part is `gdb`s prompt, don't type it :)):

```
(gdb) where
```

Now you get a lot of output. Fluxgen wants to know all the output that starts with a #.

Now paste that into an e-mail to fluxgen and also attach the four settings files in `~/ .fluxbox/`.

Please do not remove the core and the source code directory yet. Fluxgen may want to ask you a few extra question and then you will need them. Please do not send the core file to fluxgen if he does not ask specifically - since it is specific to your system, it most probably is of no use to him and core files tend to be quite big.