

# ShowDVI, der T<sub>E</sub>X-Previewer für den AMIGA

**Version 1.13**

**vom  
25. August 1991**

**von  
Georg Heßmann**

## **Abstract**

ShowDVI ist ein T<sub>E</sub>X-Previewer mit vielen Besonderheiten: er arbeitet schnell, läßt sich komfortabel bedienen und flexibel konfigurieren. Auch nutzt er schon viele Features des neuen Betriebssystem 2.0 wie z.B. Public-Screens, Application-Icon und die verschiedensten Screen-Modi, die seit 2.0 verfügbar sind. Aber auch als T<sub>E</sub>X-Treiber hält sich ShowDVI an die vorgegebenen Standards wie z.B. den "Level 0 DVI Driver Standard".

## Inhaltsverzeichnis:

<b>1. Copyright und ähnliches</b> .....	<b>4</b>
<b>2. Allgemeines über ShowDVI</b> .....	<b>4</b>
<b>3. Der Aufruf von ShowDVI</b> .....	<b>5</b>
3.1. Der Start von CLI .....	5
3.2. Der Start von Workbench .....	6
<b>4. Die Bedienung von ShowDVI</b> .....	<b>6</b>
4.1. Das Amiga-DOS pull-down Menü .....	7
4.1.1. Project Menü .....	7
4.1.2. Shell commands Untermenü .....	8
4.1.3. Outfit Menü .....	9
4.1.4. Move Menü .....	10
4.1.5. Resolution Menü .....	12
4.2. Das Popup-Menü .....	12
4.2.1. Das Hauptmenü des pop-up Menüs .....	12
4.2.2. Das Untermenü "outfit" des pop-up Menüs .....	13
4.2.3. Das Untermenü "load DVI" des pop-up Menüs .....	13
4.2.4. Das Untermenü "resolution" des pop-up Menüs .....	13
4.3. Gadgets, Scrollbars und Requester .....	14
4.3.1. Die Gadgets .....	14
4.3.2. Die Scrollbars .....	14
4.3.3. Die Fenster/Requester .....	15
4.4. Bedienung über die Tastatur .....	15
4.4.1. Innerhalb einer Seite scrollen .....	15
4.4.2. Einfaches Blättern .....	16
4.4.3. Blättern mittels des "PJC" .....	16
4.4.4. Verschiedenes .....	16
4.4.5. Zur physikalische Seitennummerierung ... ..	18
<b>5. ARexx Port</b> .....	<b>18</b>
5.1. Die Kommandos .....	18
5.2. Funktionstastenbelegung .....	20
5.3. Erzeugung nichtgefundener Fonts mittels Metafont .....	20
<b>6. Konfigurations-Files</b> .....	<b>20</b>
6.1. Das Startup-File .....	21
6.2. Das Fontdefinitions-File .....	24
6.3. Das Font-Caching Konfigurations-File .....	27
<b>7. "2.0" Features</b> .....	<b>29</b>
7.1. Screen Modis .....	29
7.2. Public Screen .....	29
7.3. Application Icon .....	29
<b>8. Internas</b> .....	<b>30</b>
8.1. Environment-Variablen .....	30
8.2. Verwendete Files .....	30

---

8.3. Font-Such-Pfade .....	31
8.4. Font-Caching .....	32
8.5. Font-Speicher .....	33
8.6. Seitenbitmap – Auflösungen .....	33
8.7. Ausdruck einer Seite .....	34
8.8. Taskname und Signale .....	34
8.9. Benötigter Stack .....	35
8.10. In den Hintergrund starten .....	35
8.11. Req.library .....	35
<b>9. Das Logfile .....</b>	<b>35</b>
<b>10. Benötigte Soft- Hardware .....</b>	<b>36</b>
<b>11. Bekannte Fehler .....</b>	<b>36</b>

## 1. Copyright und ähnliches

Das Programm ShowDVI ist “freely distributable copyrighted software”. Das heißt, jeder darf es sich kopieren und verwenden, das Copyright bleibt aber vollständig bei mir (Georg Heßmann). Das Programm darf auch nur unverändert weitergegeben werden.

Das Programm darf ohne meine ausdrückliche Genehmigung nicht kommerziell verkauft werden. Als PD-Software vertrieben darf der Preis 5 DM pro Diskette nicht überschreiten.

Es darf auch nur in vollständiger und unveränderter Form weitergegeben werden.

Diese Anleitung “`sdvi.tex`” ist ebenfalls Copyright Georg Heßmann und darf nicht kommerziell vervielfältigt werden. Man darf sie sich ausdrucken und auch kopieren, man darf sie aber **nicht** für Geld weitergeben.

Für weitere Erklärungen siehe das README-File.

## 2. Allgemeines über ShowDVI

ShowDVI, wozu braucht man das?

ShowDVI ist ein Programm, mit dem man die Ausgabe des  $\text{\TeX}$ -Programms auf den Bildschirm anschauen kann. Üblicherweise wird so ein Programm auch  $\text{\TeX}$ -Previewer genannt. Wozu braucht man dieses Programm?  $\text{\TeX}$  ist zwar ein exzellentes Textverarbeitungsprogramm, aber kein WYSIWYG-Programm<sup>1</sup>. Man muß die  $\text{\TeX}$ -Eingabe in einem normalen ASCII-Editor eingeben und dann mit `virtex` oder `tex` übersetzen. Das  $\text{\TeX}$ -Programm erzeugt daraus – eine fehlerfreie Eingabe vorausgesetzt – ein DVI-File<sup>2</sup>. Dieses kann mit einem Druckertreiber ausgedruckt werden. Da man aber nicht nach dem Ausbessern eines einzigen Druckfehlers wieder das ganze Dokument ausdrucken möchte, ist man dazu übergegangen, Previewer zu entwickeln, mit denen man das Dokument auf dem Bildschirm betrachten kann.

ShowDVI ist mein Beitrag zu den schon existierenden Previewern, und wie Sie in dieser Bedienungsanleitung noch sehen werden, kein schlechter.

Ein paar Besonderheiten im voraus:

- Einfache und doch sehr mächtige Bedienungsoberfläche.
- Unterstützung vieler 2.0 Besonderheiten.
- Leistungsfähiger ARexx-Port.
- Schnelle Bildschirmausgabe.
- On-Line Hilfe.
- Unterstützung von Fontlibraries (spart Speicherplatz und beschleunigt den Zugriff).
- Aufwendiger Suchalgorithmus mit Hilfe von Environment-Variablen und Definitions-Files, um Fonts zu finden.

---

<sup>1</sup> what you see is what you get

<sup>2</sup> device independent

- Hochgradig konfigurierbar. Ausgabe des jeweiligen Status in ein Konfigurations-File.

### 3. Der Aufruf von ShowDVI

ShowDVI kann sowohl von Workbench als auch von CLI aus aufgerufen werden. Da die Tooltypes, die im Info-File angegeben werden koennen, den Optionen, die man von CLI aus angeben kann, entsprechen, wird zuerst der Start von CLI aus beschrieben.

Achtung: Falls ShowDVI bereits im Hintergrund läuft, so wertet das neu gestartete ShowDVI die Parameter **nicht** aus, sondern holt nur den Screen des bereits laufenden ShowDVI Programmes nach vorne.

#### 3.1. Der Start von CLI

ShowDVI kennt eine Vielzahl von Optionen, die bei dem Aufruf angegeben werden können.

HELP	Gibt eine kurze Beschreibung aller Optionen aus. (Abkürzung: ?)
FONTDir <i>Dir</i>	Gibt ein zusätzliches Directory an, in dem nach Fontlibraries und nach pk-Fonts gesucht werden soll. (Abkürzung: -a)
FONTMem <i>Zahl</i>	Gibt an, wie groß der interne Fontspeicher sein soll, in dem die Fonts gespeichert werden. (Abkürzung: -b)
From <i>Zahl</i>	Welche Seite soll als erste angezeigt werden. (Abkürzung: -f)
HOFFset <i>ZahlXX</i>	Setzt den horizontalen Offset. <i>XX</i> ist dabei eine Einheit aus: pt Point, cm Zentimeter, in Inch, pc Pica, dd Didôt Point, cc Cicero, bp Big Point, mm Millimeter, mi Milli-Inch. (Abkürzung: -h)
VOFFset <i>ZahlXX</i>	Setzt den vertikalen Offset. <i>XX</i> ist wieder eine zwei-Buchstaben Abkürzung einer Einheit. (Abkürzung: -v)
PREload	Wenn ausgewählt, werden alle Fonts des Dokuments gleich zu Beginn geladen und nicht erst dann, wenn sie wirklich benötigt werden. (Abkürzung: -p)
RESolution <i>Zahl</i>	Setzt die Resolution, mit der das Dokument angezeigt wird. (Abkürzung: -z)
STATistic	Erzeugt ein ausführlicheres Logfile. (Abkürzung: -s)
DEBUGStat	Schließt nach jeder Zeile das Logfile. Ist nur nützlich, wenn ShowDVI einen GURU erzeugen sollte und man wissen möchte, was noch zuvor in das Logfile geschrieben wurde. Sollte normalerweise nicht benötigt werden.

	(Abkürzung: -S)
LOGName	Name Gibt den Namen/Pfad des Logfiles an.
NOLog	Unterdrückt die Erzeugung des Logfiles. (Abkürzung: -1)

Als weitere Option kann der Name eines DVI-Files angegeben werden.

Die Optionen ohne Parameter sind an/aus Schalter. Werden sie angegeben, so wird die Option als angeschaltet betrachtet. Man kann aber auch noch die Schlüsselwörter `on` oder `off` angeben. So kann man mit `statistic=on` das Flag anschalten und mit `statistic=off` wieder ausschalten.

Die Groß- Kleinschreibung ist bei den Optionen egal. Die Tatsache, daß bei der Auflistung solch eine seltsame Groß- Kleinschreibung gewählt wurde hat einen anderen Grund. Den, sich Tiparbeit zu ersparen. Bei allen Optionen reicht es, den Teil der großgeschrieben ist, anzugeben.

Damit man ständig verwendete Optionen nicht immer wieder angeben muß, wird beim Programmstart die Environment Variable `ShowDVI` ausgelesen. Diese kann man mit den Optionen belegen, die man immer verwenden möchte. Zum Beispiel:

```
setenv ShowDVI "logname=t:ShowDVI.log stat=on"
```

Wenn man dann doch einmal keine Statistik-Ausgaben haben möchte, so kann man beim Programmstart diese mit `stat=off` wieder ausschalten.

Die Default-Werte der Optionen werden bei der `HELP` Option ausgegeben.

### 3.2. Der Start von Workbench

Bei dem Start von Workbench werden die Parameter über das Info File des Programms übergeben. Dort werden die "Tool-Types" definiert. Die Optionen, die man über CLI angeben kann, können alle auch als Tool-Types definiert werden. Allerdings müssen sie dort großgeschrieben werden und dürfen auch nicht abgekürzt werden.

Man kann die Default-Parameter aber auch wie beim Start von CLI über die Environmentvariable `SHOWDVI` definieren.

Der Name des DVI-Files kann über Workbench auf zwei Arten ShowDVI übergeben werden. Zum einen, in dem man erst das ShowDVI Icon aktiviert und dann auf das DVI-File doppelklickt. Zum anderen, in dem man im Info-File des DVI-Files als "default tool" ShowDVI einträgt.

## 4. Die Bedienung von ShowDVI

Bei der Benutzeroberfläche wurde auf höchstmögliche Flexibilität und auf möglichst große Vielseitigkeit geachtet. Und das bei möglichst einfacher – intuitiver Bedienung.

Dies zeigt sich zum Beispiel darin, daß es zwei verschiedene Menüs (pop-up und pull-down) gibt, oder darin, daß praktisch alle Funktionen per einfachen Tastendruck erreichbar sind.

Als erstes soll aber das normale Amiga-DOS “pull-down” ausführlich besprochen werden, da es praktisch alle Funktionen von ShowDVI beinhaltet.

## 4.1. Das Amiga-DOS pull-down Menü

ShowDVI hat 4 Hauptmenüs. Das ist einmal “Project”, dieses ist für die Kommunikation mit der Aussenwelt zuständig. Als zweites gibt es das Menü “Outfit”. Mit diesem kann man das Aussehen von ShowDVI beeinflussen. Das dritte Menü ist das “Move” Menü. In ihm sind die Funktionen zusammengefaßt, die zum Wechseln der angezeigten Seite dienen. Mit dem vierten und letzten Menü kann man die Auflösung ändern.

Unter Kickstart 37.xx hat das Menü auch eine Besonderheit. Wenn man einen Menüpunkt anwählt (Mausknopf **nicht** loslassen) und dann die **Help** Taste betätigt, wird ein Requester mit einem kurzen Hilfetext zu dem gewählten Menüpunkt angezeigt.

### 4.1.1. Project Menü

Die Punkte des ersten Hauptmenüs “Project”:

- About** Öffnet ein Fenster, in dem eine Copyright Meldung angezeigt wird. Mit dem **help** Gadget kann man sich auch mehrere Seiten Hilfetext anzeigen lassen.  
Tastenkürzel: **HELP**
- Open again** Lädt das gerade geladene DVI-File neu ein.  
Tastenkürzel: **a**
- Open new** Öffnet eine Filerequester, mit dem man ein DVI-File auswählen kann, das angezeigt werden soll.  
Tastenkürzel: **AMIGA-0**
- Auto load again** Wenn dieser Punkt aktiviert ist, wird jedesmal wenn das ShowDVI Window aktiv wird, getestet, ob sich das DVI-File geändert hat. Ist dies der Fall, so wird es automatisch neu geladen.  
Tastenkürzel: **AMIGA-A**
- Shell commands** In diesem Untermenü sind die Aktionen zusammengefaßt, die sich auf externe Programme beziehen. Fast alle Einträge in dem Untermenü sind nur unter 2.0 auswählbar. Lediglich **newcli** funktioniert auch unter 1.3. Allerdings mit stark eingeschränkter Funktionalität. Unter 1.3 kann solange nichts mit ShowDVI machen, solange das CLI-Window offen ist. Erst nach einem **endcli** kann man wieder normal mit ShowDVI weiterarbeiten.
- Print page** Druckt die gerade angezeigte Seite über die Preference-Treiber aus. Während der Ausdruck läuft, kann die Seite nicht gewechselt werden. Abbrechen kann man den Ausdruck durch erneutes anwählen dieses Menüpunktes. Während des Ausdrucks wird ein **P** oben rechts in die Fensterkopfzeile geschrieben.  
Tastenkürzel: **p**

<b>WB to front</b>	Hohlt den Workbench-Screen nach vorne. Tastenkürzel: <b>d</b>
<b>Hide</b>	Legt den ShowDVI-Screen nach hinten. Tastenkürzel: <b>ESC</b>
<b>Save config</b>	Schreibt die aktuelle Konfiguration in das File <code>ShowDVI.config</code> . Dazu gehört zum Beispiel die gewählte Auflösung und die gewählten Farben.
<b>Quit</b>	Beendet ShowDVI. Allerdings wird erst mittels eines Requestors nach einer Bestätigung gefragt. Dies geht auch nur, wenn alle Fenster, die nicht unter der Aufsicht des ShowDVI Screens stehen, geschlossen wurden. Ohne Nachfrage kann das Programm mit CTRL-C beendet werden. Tastenkürzel: <b>AMIGA-Q</b>

#### 4.1.2. Shell commands Untermenü

Von diesem Untermenü aus, werden externe Programme gestartet. Bis auf den ersten Punkt "New CLI" sind alle anderen nur unter Betriebssystem Version 2.0 oder höher anwählbar.

<b>New CLI</b>	Öffnet ein Fenster auf dem ShowDVI-Screen und startet darin eine Shell. Tastenkürzel: <b>!</b>
<b>Execute command</b>	Arbeitet genauso wie der gleichlautende Menüpunkt der 2.0 Workbench. Es wird ein String-Requester geöffnet und nach einem Kommando (mit Argumenten) gefragt. Nach Bestätigung mittels des OK Gadgets wird das Kommando ausgeführt. Falls dieses Ausgabe erzeugt, wird auch ein Fenster auf dem ShowDVI-Screen geöffnet. Schließen kann man das Fenster entweder in dem man das CLOSE Gadget betätigt. Oder aber mittels CTRL-\. Tastenkürzel: <b>AMIGA-E</b>
<b>TeX-script</b>	Startet eine Scriptdatei, die das aktuelle DVI-File übersetzt. Das Ausgabefenster wird auf dem ShowDVI-Screen geöffnet. Welches Format-File dabei von T <sub>E</sub> X verwendet wird, bestimmt der Inhalt der Environment Variable <code>TEXFORMAT</code> . Wird dieser Menüpunkt mit der Taste <b>R</b> aufgerufen, so wird erst noch die Möglichkeit gegeben, die Environment Variable zu setzen. Die Schablone der Parameter der Script ist: <code>.key filename/a,directory/a,screenname/k</code> "Filename" ist der Name des aktuell geladenen DVI-Files, "Directory" ist das Directory, in dem das DVI-File steht. "Screenname" ist der Public-Screen Name, auf dem das Ausgabefenster der Script gestartet wird. Die Script die normalerweise aufgerufen wird ist: <code>TeX:s/ShowDVI.sh</code> . Der Name kann in dem Konfigurationsfile von ShowDVI geändert werden. Tastenkürzel: <b>r</b> oder <b>R</b>



- ARexx TeX-Shell** Startet eine ARexx-Script die die Zusammenarbeit TeX – Editor – Previewer steuert. Über mehrere weitere ARexx-Scripten kann man dabei einen Text aus dem Editor oder dem Previewer heraus übersetzen. Ist der Text fehlerfrei, wird nach dem Ende der bersetzung automatisch der ShowDVI-Screen nach vorne geholt und das DVI-File geladen. Sind in dem Text Fehler, so kann man sich diese erst ganz normal in TeX anschauen. Wenn man nun in `virtex` das `e` Kommando aufruft, wird der Editor nach vorne geholt, das Logfile geladen und die Stelle mit der Fehlerklärung angesprungen und schließlich der Cursor in der Zeile mit dem Fehler im TeX-Text plaziert. Mit einer weiteren ARexx Script `NextError` kann man sogar von einem Fehler zum nächsten springen.  
Zu der Arbeitsweise der Scripten und wie man sie installiert gibt es aber noch eine eigene Anleitung.
- Set TEXFORMAT** Öffnet einen Stringrequester. mit dem man die Environment Variable `TEXFORMAT` setzen kann. Diese Variable wird von `virtex` ausgewertet, falls kein Formatfile beim Aufruf angegeben wurde.
- SpecialHost** Startet das Programm `SpecialHost`, welches die Auswertung der `\special`-Strings vornimmt.

#### 4.1.3. Outfit Menü

Die Punkte des zweiten Hauptmenüs “Outfit”:

- Lace** Schaltet zwischen Interlace und Non-Interlace Darstellung hin und her.  
Tastenkürzel: `i`
- Scrollbar** Schaltet die Scrollbars ein und aus.  
Tastenkürzel: `o`
- Full page** Schaltet die “full-page” Darstellung ein und aus. Dabei wird die Seite so weit verkleinert, daß sie vollstaendig angezeigt werden kann. Im aktivierten “full-page” Modus wird ein `F` oben rechts in der Fensterkopfzeile angezeigt.  
Tastenkürzel: `f`
- Measure Window** Öffnet bzw. schließt ein “Measure” Fenster. In diesem werden die Seitengröße sowie die Koordinaten angezeigt, die der Mauspointer auf der imagiären Seite hat. Dazu wird noch die “magnification”, die Vergrößerung ausgegeben. Mit dieser muß man alle Werte multiplizieren, wenn man die Größen haben möchte, die die Seite nach dem Ausdrucken wirklich hat. Wenn man den linken Mausknopf drückt und dabei die Maus bewegt, so wird der Abstand zwischen der jetzigen Mausposition und der, an dem man den Knopf gedrückt hat, angezeigt. Die Einheit, in der die Werte angezeigt werden, kann frei aus den Einheiten Inch, Zentimeter und Point gewählt werden.

	Sinnvoll kann dies vor allem dann sein, wenn man Bilder exact plazieren möchte. Tastenkürzel: <b>m</b>
<b>Border line</b>	Schaltet die Anzeige von Hilfslinien an und aus. Die linke und die obere Linie zeigen an, wo horizontaler bzw. vertikaler Offset aufhört und das eigentliche Innere der Seite anfängt. Die rechte und die untere Linie bestimmen das Ende der Seite und dem Sicherheitsrand gegen "overflow boxes". Die zwei Offsets kann man über Kommandozeile Optionen ändern. Die Sicherheitsränder dagegen sind immer ein Inch groß. Tastenkürzel: <b>l</b>
<b>4 color screen</b>	Bestimmt, ob der ShowDVI-Screen zwei oder vierfarbig sein soll. Der einzige Nachteil eines vierfarbigen Screens ist der leicht erhöhte Chip-Mem Bedarf. Die Scrollgeschwindigkeit dagegen ist praktisch überhaupt nicht herabgesetzt. Der Vorteil ist das bessere Aussehen. Vor allem unter 2.0 sieht alles erheblich besser aus.
<b>Page scrollbar</b>	Damit kann man bestimmen, ob der rechte Scrollbar dazu verwendet werden soll, um innerhalb der Seite zu scrollen, oder ob man damit die anzuzeigende Seite auswählen möchte. Ist man im "Seiten-scroll" Modus, so wird immer eine Seitennummer innerhalb der Scrollbars angezeigt. Man kann nun eine neue Seite auswählen. Übernommen wird die Seite, in dem man entweder das OK Gadget drückt, oder in dem man eine der normalen Seitenwechsellasten (wie RETURN, ENTER, ...) betätigt. Diesem Modus ausschalten ohne die Seite zu wechseln erreicht man durch nochmaliges Anwählen des Menüpunktes. Tastenkürzel: <b>s</b>
<b>Unit</b>	In diesem Untermenü kann man festlegen, mit welcher Einheit das "Measure" Window seine Werte anzeigen soll. Zur Auswahl stehen Inch (in), Zentimeter (cm) und die T <sub>E</sub> X Einheit Point (pt). Alle Einheiten können auch so innerhalb T <sub>E</sub> X verwendet werden.
<b>Color</b>	Dieser Menüpunkt öffnet einen Farbrequester, mit dem man die Farben des ShowDVI-Screens ändern kann. Abgespeichert wird die Farbeinstellung mit dem "Save config" Menüpunkt. Tastenkürzel: <b>AMIGA-C</b>
<b>Clone WB color</b>	Dieser Punkt setzt die Farben des ShowDVI-Screens auf die Farben des Workbench-Screens.

#### 4.1.4. Move Menü

Die Punkte des dritten Hauptmenüs "Move":

<b>Prev page</b>	Wechselt auf die vorhergehende Seite. Tastenkürzel: <b>-</b> oder <b>BACKSPACE</b>
<b>Next page</b>	Wechselt auf die nächste Seite. Tastenkürzel: <b>+</b> oder <b>RETURN</b> oder <b>ENTER</b>

<b>First page</b>	Wechselt auf die erste Seite. Tastenkürzel: SHIFT zusammen mit - oder BACKSPACE
<b>Last page</b>	Wechselt auf die letzte Seite. Tastenkürzel: SHIFT zusammen mit + oder RETURN oder ENTER
<b>Jump to page counter</b>	Wechselt auf die Seite, die der "Page jump counter" (kurz PJC) angibt. Der Inhalt des Zählers wird oben rechts in der Fensterkopfzeile angezeigt. Gesezt wird der Zähler über die Zifferntasten. Löschen kann man ihn mittels der Taste c. Tastenkürzel: . oder g
<b>Clear page counter</b>	Löscht den "Page jump counter". Tastenkürzel: c
<b>Use order of the dvi-file</b>	ShowDVI kennt zwei verschiedene Arten, die Seiten des DVI-Files zu ordnen. Einmal die "normale" Art und Weise. Hier wird die von T <sub>E</sub> X vergebene Seitennummer als Ordnungsgrundlage verwendet. Es kann allerdings auch vorkommen, daß eine Seitennummer mehrfach im DVI-File vorkommt. Um auch damit gut umgehen zu können, gibt es auch noch eine andere Möglichkeit die Seiten zu ordnen. Diese Ordnung wird auch die "physikalische Methode" genannt, da hier einzig und alleine die Reihenfolge der Seiten innerhalb des DVI-Files als Ordnungsgrundlage verwendet. Die erste Seite des DVI-Files hat dabei immer die Seitennummer 1. Dieser Modus wird mit einem Y rechts oben in der Fensterkopfzeile angezeigt. Tastenkürzel: y
<b>Prev page</b>	Wechselt auf die, der aktuellen Seite, "physikalisch" vorangehenden Seite. Dazu wird nur für einen Seitenwechsel temporär in den "physikalischen" Modus geschaltet. Dies funktioniert mit allen Seitenwechseltasten, in dem man zusätzlich die CTRL Taste drückt. Tastenkürzel: CTRL zusammen mit - oder BACKSPACE
<b>Next page</b>	Wechselt auf die, der aktuellen Seite, "physikalisch" nachfolgenden Seite. Tastenkürzel: CTRL zusammen mit + oder RETURN ENTER
<b>First page</b>	Wechselt auf die allererste Seite des DVI-Files. Tastenkürzel: CTRL und SHIFT zusammen mit - oder BACKSPACE
<b>Last page</b>	Wechselt auf die allerletzte Seite des DVI-Files. Tastenkürzel: CTRL und SHIFT zusammen mit + RETURN oder ENTER

### 4.1.5. Resolution Menü

Dieses, letzte, Hauptmenü hat eine Besonderheit. Seine Einträge sind nicht starr vorgegeben, sondern können über das Konfigurationsfile `ShowDVI.config` frei gewählt werden. Allerdings doch nicht ganz frei. Es können nur Menüpunkte definiert werden, die die Auflösung ändern, und es dürfen auch nur maximal 10 solcher Einträge definiert werden. Sprich, man kann 10 beliebige DPI Zahlen in das Konfigurationsfile schreiben, die dann zur Auswahl angeboten werden.

Als Tastenkürzel dient AMIGA-1 für den ersten Punkt, bis AMIGA-9 für den neunten Punkt. Der zehnte, sowie der elfte (den gibt es dann, wenn man über Option eine Startauflösung gewählt hat, die sonst noch nicht in der Liste vertreten ist) haben leider keine Abkürzung über eine Tastenfolge. Diese kann man nur mit der Maus auswählen.

## 4.2. Das Popup-Menü

Je nach Konfiguration erscheint das Pop-up Menü nach Drücken der rechten Maustaste oder durch Drücken der mittleren Maustaste. Letzteres benötigt sowohl Amiga-DOS 2.0, als auch eine drei-Tasten Maus.

Das Amiga-DOS Menü ist aber auch dann erreichbar, wenn man das Popup-Menü auf die rechte Maustaste gelegt hat. Man muß dann lediglich mit der Maus in die Kopfzeile (oder noch höher) des Fensters fahren und erst dort die rechte Maustaste drücken.

Eine weitere Eigenschaft des Pop-Up Menüs läßt sich über das Konfigurationsfile einstellen. Entweder erscheint das Menü immer rechts unterhalb des Mauszeigers, oder es merkt sich, welcher Menüpunkt als letzter ausgewählt wurde und positioniert das Menü so, daß dieser Menüpunkt gleich wieder ausgewählt ist. Dieser Modus ist auch die Default-Einstellung.

Die einzelnen Einträge des Menüs sind in Zeilen untereinander angeordnet. Die Zeilen, an deren Ende ein Pfeil ist, haben noch ein Untermenü. Wenn man die Maus in die Nähe eines dieser Pfeile bringt, klappt das Untermenü auf. Ausgewählt wird ein Menüpunkt in dem über ihm die rechte Maustaste wieder losgelassen wird. Möchte man die erste Zeile einen Untermenüs auswählen, so genügt es, die dem Untermenü zugehörige Zeile im Hauptmenü auszuwählen.

### 4.2.1. Das Hauptmenü des pop-up Menüs

Das Hauptmenü beinhaltet folgende Punkte:

<code>about</code>		Zeigt einen Requester mit einem höchst interessanten Text an. Mit diesem Requester kann man sich eine kurze Hilfe zu den Tastaturkommandos anzeigen lassen.
<code>+</code>	<code>++</code>	Dies ist ein zweigeteilter Menüpunkt. Mit <code>+</code> kommt man auf die nächste und mit <code>++</code> auf die letzte Seite.
<code>-</code>	<code>--</code>	Entspricht dem Menüpunkt von oben, nur daß es hier entweder auf die vorherige oder die erste Seite geht.
<code>show full page</code>		Verkleinert die aktuelle Seite so, daß sie vollständig auf den Bildschirm paßt.

<code>outfit</code>	=>	Dies ist ein Untermenü, in dem die Kommandos zum Verändern des Aussehens des ShowDVI-Screens stehen.
<code>load DVI</code>	=>	Ein Untermenü zum Laden eines neuen DVI-Files.
<code>resolution</code>	=>	Mit diesem Untermenü kann die Auflösung der Anzeige geändert werden.
<code>print page</code>		Kommando zum Ausdruck der aktuellen Seite. Dies geschieht mittels eines Betriebssystemaufrufes. Man kann also über die Einstellung in Preferences den Druckertyp auswählen und die gewünschte Auflösung einstellen. Läuft bereits ein Druckauftrag, so heißt das Menü <code>abort print</code> . Damit kann dann der laufende Ausdruck abgebrochen werden.
<code>hide quit</code>		Noch ein zweigeteilter Menüpunkt. Mit <code>hide</code> wird der ShowDVI-Screen nach hinten gelegt und <code>quit</code> beendet, nach einer Sicherheitsabfrage, das Programm. Man kann das Programm übrigens mittels Control-C ohne Abfrage beenden.

#### 4.2.2. Das Untermenü “outfit” des pop-up Menüs

Dieses Untermenü beinhaltet folgende Punkte:

<code>unset lace</code>	Schaltet vom Interlace-Modus auf den Noninterlace-Modus. Danach lautet der Menüpunkt <code>set lace</code> .
<code>hide scrollbar</code>	Entfernt die Scrollbars. Falls die Scrollbars nicht aktiv sind, heißt der Menüpunkt <code>show scrollbar</code> .
<code>change color</code>	Ruft einen Farb-Requester auf, mit dem die Farben des ShowDVI-Bildschirms geändert werden können.
<code>save config</code>	Speichert die augenblickliche Konfiguration in ein File ab. Das File-Format wird noch in Kapitel 6 erklärt.

Wie schon einmal erwähnt, braucht man, um den Punkt `(un)set lace` aufzurufen, diesen nicht explizit auszuwählen; es genügt, wenn man den Hauptmenüpunkt `outfit` auswählt. Da `(un)set lace` der erste Punkt in diesem Untermenü ist, wird dieser aufgerufen. Ebenso funktioniert dies natürlich auch mit den ersten Zeilen der anderen Untermenüs.

#### 4.2.3. Das Untermenü “load DVI” des pop-up Menüs

Dieses Untermenü besteht aus zwei Punkten. Einmal dem `load file again`, das das aktuelle File neu einliest und wieder genau an die selbe Stelle springt, an der man vorher war. Zum anderen der Punkt `load new file`. Dieser ruft einen File-Requester auf, mit dem man ein neues DVI-File auswählen kann.

#### 4.2.4. Das Untermenü “resolution” des pop-up Menüs

Dies ist ein besonderes Menü, da es vom Benutzer vordefiniert werden kann. Es besteht aus bis zu 10 DPI-Einträgen, mit denen man die Auflösung ändern kann, in der das DVI-File angezeigt wird. Dazu muß man allerdings auch über die Fonts in den jeweiligen Auflösungen verfügen. Ohne Benutzeränderungen hat das Menü drei Einträge: `100 dpi`, `83 dpi` und `120 dpi`. Das Generieren eigener Einträge ist im Kapitel 6 nachzulesen.

### 4.3. Gadgets, Scrollbars und Requester

Das Programm hat fast 20 verschiedene Gadgets, Scrollbars und Requester. Die meisten sind zwar selbsterklärend, aber manche haben auch eine versteckte Funktion. Daher soll nun jedes einzelne Element erklärt werden.

#### 4.3.1. Die Gadgets

Der größte Teil der Gadgets ist innerhalb der Kopfzeile. Von links nach rechts:

- Das “Window-Close” Gadget. Mit ihm beendet man nach einer Sicherheitsabfrage das Programm.
- Das “--” Gadget. Durch Klicken auf dieses Gadget springt man auf die erste Seite im Dokument.
- Das “-” Gadget. Dieses Gadget blättert um eine Seite nach vorne.
- Das “+” Gadget. Dies korrespondiert mit dem “-” Gadget. Mit ihm kommt man auf die nächste Seite.
- Das “++” Gadget. Mit diesem springt man auf die letzte Seite des DVI-Files.
- Das “D” Gadget. Dies ist ein Stringgadget. Im Normalfall zeigt es das aktuelle Directory an, bzw. das Directory in welchem das aktuelle DVI-File zu finden ist. Man kann es natürlich auch ändern. Dieses Directory ist auch das aktuelle Verzeichnis des Filerequesters.
- Das “F” Gadget. In diesem Stringgadget wird das aktuelle DVI-File angezeigt. Hier kann man auch ein neues File angeben. Falls man es nur anklickt und RETURN drückt wird das aktuelle File neu eingelesen. Dies ist dann praktisch, wenn man im Hintergrund das T<sub>E</sub>X-File geändert und neu übersetzt hat.
- Das “#” Gadget. Hier wird die aktuelle Seitennummer angezeigt. Man kann damit natürlich auch eine neue Seite anwählen. Falls eine logische Seitennummer mehrfach in dem DVI-File vorkommt, so wird rechts neben diesem Gadget angezeigt, die wievielte Seite mit dieser Nummer die aktuelle Seite ist.

Ganz an dem rechten Rand kann unter bestimmten Umständen noch ein “OK” Button erscheinen. Mehr dazu bei der Besprechung der Scrollbars. Auch die anderen Gadgets werden zusammen mit den Scrollbars erklärt.

#### 4.3.2. Die Scrollbars

Am unteren und rechten Rand befinden sich die Scrollbars. Im Normalzustand verhalten sie sich nicht anders als gewohnt. Wenn man allerdings das Gadget ganz rechts unten anklickt, wird der vertikale Scrollbar in den “Seiten-Scroll-Modus” geschaltet. Dabei verändert sich das Aussehen folgendermaßen: Über dem Scrollbar erscheint ein OK-Button, der gerade angeklickte Button verwandelt sich in einen NO-Schalter und der Scrollbar selbst zeigt die aktuelle Seite an. Nun kann per Maus eine Seite ausgewählt werden. Mit dem OK Gadget springt man schließlich zur markierten Seite oder man bricht mit dem NO Knopf den Seiten-Scroll-Modus ab.

Dies kann man auch vollständig über die Tastatur steuern. Die Taste “s” schaltet den “Seiten-Scroll-Modus” ein. Mit den Cursor Tasten kann dann eine bestimmte Seite ausgewählt werden. Uebernommen wird sie mit allen Tasten, die auch sonst eine Seitenänderung bewirken (z.B. RETURN/ENTER/BACKSPACE). Nochmaliges drücken der Taste “s” entspricht dem NO Gadget, der “Seiten-Scroll-Modus”

wird ohne Änderung der Seitennummer verlassen.

Eine weitere Besonderheit halten die Auf- und Ab-Pfeile des vertikalen Scrollbars bereit. Befindet man sich am Ende einer Seite, so kann man mit einem Doppelklick auf den nach unten zeigenden Pfeil auf den Beginn der nächsten Seite springen. Ist man dagegen am Anfang einer Seite, so kommt man mit einem Doppelklick auf den nach oben zeigenden Pfeil an das Ende der vorhergehenden Seite. So kann man nur durch Klicken auf diese beiden Pfeile durch das gesamte DVI-File scrollen.

Abschalten kann man die Scrollbars natürlich auch. Entweder mit der Taste “o” oder per Menüpunkt “hide scrollbar” im outfit-Menü. Der Zustand der Scrollbars, sichtbar oder nicht, kann natürlich auch im Konfigurations-File abgespeichert werden.

### 4.3.3. Die Fenster/Requester

Folgende Fenster/Requester gibt es im Programm:

- Das Hauptfenster. Dieses füllt immer den gesamten ShowDVI-Screen. In ihm wird die T<sub>E</sub>X-Seite angezeigt.
- Das About-Hilfe-Fenster. Dieses wird per Menüpunkt `about` oder per `Help` Taste aufgerufen. Man kann dieses Fenster mit `RETURN` `q` oder dem `OK` Gadget schließen. Mit dem `help` Gadget kann man sich einen mehrseitigen Hilfetext anzeigen lassen.
- Das Measure-Fenster. Dieses wird mit dem Menüpunkt `Measure Window` oder der `m` Taste aktiviert. Mit diesem Fenster kann man die T<sub>E</sub>X-Seite abmessen. Schließen kann man das Fenster durch nochmaliges Anwählen des Menüpunktes oder wieder mit der `m` Taste.
- Der Filerequester. Aufgerufen wird dieser mit dem Menüpunkt `Open new` oder mit der Tastenkombination `AMIGA-O`. Für diesen Filerequester wird die `req.library` benötigt!
- Der Farbrequester. Aufruf: Menüpunkt `Change color` oder `AMIGA-C`. Auch für diesen Requester wird die `req.library` benötigt.
- Der Programmende Requester. Dieser wird immer dann aufgerufen, wenn man das Programm verlassen möchte (außer bei `CTRL-C`). Dabei ist wichtig zu erwähnen, daß man das Programm nur dann beenden kann, wenn man alle Fenster auf dem ShowDVI-Screen geschlossen hat, die nicht der direkten Kontrolle von ShowDVI unterstehen.

## 4.4. Bedienung über die Tastatur

Praktisch alles was man mit Hilfe der Maus an Kommandos ausführen kann, ist auch über die Tastatur erreichbar.

### 4.4.1. Innerhalb einer Seite scrollen

Innerhalb einer Seite scrollt man entweder – bei eingeschalteten Scrollbars – mit der Maus, oder aber mit den Cursor-Tasten. Mit der Cursor-Up-Taste kommt man an den Seitenanfang und mit der Cursor-Down-Taste an das untere Ende der Seite. Links/rechts geht analog. Bei gedrückter Shift-Taste ist die Scroll-Geschwindigkeit herabgesetzt. Mit Alt und einer Cursor-Taste kommt man an den jeweiligen Rand der Seite.

#### 4.4.2. Einfaches Blättern

Die aktuelle Seite verlassen und auf eine Neue springen kann man mit 7 verschiedenen Tasten. Die einfachsten davon sind:

- **RETURN**; Damit kommt man auf die nächste Seite, bleibt aber auf der selben Höhe der Seite. Zusammen mit einer Taste wie Shift oder Alt springt man auf die letzte Seite.
- **BACKSPACE** oder **-** auf der normalen Tastatur. Mit diesen zwei Tasten kommt man auf die vorhergehende Seite und bleibt ebenfalls auf dem selben Ausschnitt der Seite. Auch springt man mit Shift oder Alt auf die erste Seite.
- **ENTER** hat die selbe Funktion wie **RETURN** nur das man an den Anfang der Seite kommt. Dies ist für ein Scrollen durch das File nützlich.
- **-** auf dem Zahlenblock hat die selbe Funktion wie **BACKSPACE**, nur daß man immer ganz am unteren Ende der Seite herauskommt.

Hält man während dem Betätigen einer der obig genannten Tasten die **Ctrl** Taste gedrückt, so kommt man temporär in den “physikalischen Modus”. Eine Erklärung dazu folgt in den nächsten Abschnitten. Nur so viel vorweg: man kommt dann mit **RETURN** nicht auf die Seite mit der nächst größeren Seitennummer, sondern auf die, die im DVI-File als nächste der aktuellen Seite folgt.

Die weiteren Möglichkeiten verlangen noch eine weitere Erklärung.

#### 4.4.3. Blättern mittels des “PJC”

“PJC” oder “Page Scroll Counter”<sup>3</sup> ist eine maximal vierstellige numerische Variable, die über die Tasten 0 bis 9 gesetzt werden kann. Ihr jeweiliger Inhalt wird, wenn gesetzt, rechts oben auf der Fensterkopfzeile neben dem # Gadget angezeigt. Mit der Taste **c** kann ihr Inhalt gelöscht werden.

Wozu wird dieser Zähler benötigt?

Mit **g** oder dem **.'** von dem Zehnerblock wird auf die Seite, die der PJC angibt, gesprungen. Mit der Eingabe **c 1 5 g** springt man also auf Seite 15. Danach ist der Zähler automatisch auf Null zurückgesetzt.

Darüberhinaus geht er auch noch bei den anderen Tasten zum Seitenwechseln mit ein. Mit **3 RETURN** springt man drei Seiten vorwärts und mit **7 -** sieben Seiten zurück.

Mit Hilfe dieses Zählers reichen Zehnerblock und Cursor-Tasten vollständig aus, um sich in einem DVI-File flott zu bewegen.

Auch die Tasten **g** und **.** können mittels einer gedrückten **Ctrl** Taste temporär auf die Verwendung der physikalischen Seitennummern umgestellt werden. So kommt man mittels **1 5 Ctrl-g** auf die 15'te Seite innerhalb des DVI-Files. Diese kann ohne weiteres eine von 15 verschiedene (logische) Seitennummer haben.

#### 4.4.4. Verschiedenes

Es gibt noch eine Reihe anderer Tasten, die mit Funktionen belegt sind:

---

<sup>3</sup> Seiten-Scroll-Zähler



- f** Zeigt die aktuelle Seite im “Ganzseiten-Modus” an. Dazu wird die Seite so verkleinert, daß sie in der jeweiligen Bildschirmauflösung vollständig angezeigt werden kann. Wenn man im “full-page-modus” die Seite wechselt, so wird diese ebenfalls in diesem Modus angezeigt. In die normale Anzeigeform kommt man durch erneutes Betätigen der Taste **f** zurück. Zur Bestätigung, daß man auch wirklich im “Ganzseiten-Modus” ist, wird rechts neben dem # Gadget, in dem Status Feld, ein ‘F’ angezeigt.
- i** Schaltet zwischen Interlace- und Noninterlace-Modus hin und her.
- o** Schaltet die Scrollbars an und ab.
- m** Öffnet/schließt das “Measure” Fenster.
- l** Zeigt bzw. löscht die ‘Border” Linien. Diese zeigen die Ränder der  $\text{T}_{\text{E}}\text{X}$ -Seite an.
- c** Löscht den Page Jump Counter.
- a** Lädt das aktuelle DVI-File neu ein, bleibt aber auf der selben Seite.
- s** Schaltet den Page-Scroll-Modus an. Mit den Cursor-Up- Down-Tasten kann man dann die gewünschte Seite auswählen und mit einer der Seitenwechselfasten (**RETURN ENTER BACKSPACE - g .**) anspringen. Nochmaliges Drücken der Taste **s** wirkt wie das **NO** Gadget.
- y** Mit dieser Taste wechselt man zwischen den zwei verschiedenen Möglichkeiten eine Seitennummer zu interpretieren. Es gibt zum einen die Möglichkeit, jede Seitennummer als die zu interpretieren, die sie im  $\text{T}_{\text{E}}\text{X}$ -Text hat (logische Nummer). Setzt man am Anfang eines  $\text{T}_{\text{E}}\text{X}$ -Text zum Beispiel mittels `\pageno=7` die erste Seitennummer auf sieben, so interpretiert der Previewer im Normalfall die physikalisch erste Seite des DVI-Files als die Seite mit der Nummer sieben. Schaltet man per ‘y’ Taste allerdings in den physikalischen Seitennummer Modus um, so wird die erste Seite des DVI-Files auch mit der Seitennummer eins versehen. Dieser Modus wird mit einem ‘Y’ in der rechten Seite der Fensterkopfzeile angezeigt.
- ’ ’** (SPACE) Ähnlich dem **more** Utility. Scrollt um eine kanappe Bildschirmseite nach unten. War man ganz am unteren Rand der Seite, so wird eine Seite vorwärts geblättert.
- b** Kommt ebenfalls von dem **more** Utility. Scrollt um eine knappe Bildschirmseite nach oben. Wie bei der SPACE Taste springt man auf die vorhergehende Seite, falls man ganz am oberen Rand der Seite war.
- p** Drückt die aktuelle Seite aus. Solange der Druck noch nicht abgeschlossen ist, wird oben rechts neben dem # Gadget ein “P” angezeigt. Durch wiederholtes Drücken der **p**-Taste kann der Druck abgebrochen werden.
- d** Bringt den Workbench-Screen nach vorne und aktiviert das Fenster, das vor dem Start von ShowDVI aktiv war.
- !** Öffnet ein Shell-Fenster auf dem ShowDVI-Screen. ShowDVI wartet, bis die Shell mit `endcli` wieder geschlossen wurde.
- r** Ruft eine Script auf, die das aktuell geladene DVI-File neu übersetzt. Als Formatfile wird der Inhalt der Environmentvariable `TEXFORMAT` verwendet. (Funktioniert nur unter 2.0)
- R** Hat dieselbe Funktion wie **r**. Mit nur einem Unterschied: es wird zuerst

nach einem Formatfile gefragt. Dieses wird dann in der Environmentvariable `TEXFORMAT` abgespeichert.

**AMIGA-e** Öffnet einen Stringrequester in dem nach einem Kommando gefragt wird. Dieses wird dann ausgeführt.

**AMIGA-c** Ruft den Farb-Requester auf.

**AMIGA-o** Öffnet den File-Requester.

**AMIGA-q** Beendet das Programm.

**ESC** Diese Taste hat zwei Funktionen abhängig von der Einstellung im Konfigurations-File. Entweder beendet die Taste das Programm, oder (im Normalfall) es legt lediglich den ShowDVI-Screen nach hinten und aktiviert das Fenster, das vor dem Aufruf von ShowDVI aktiv war.

Darüberhinaus kann man die Funktionstasten mit ARexx-Scripts belegen. Genaues dazu im nächsten Kapitel.

#### 4.4.5. Zur physikalische Seitennummerierung...

Wie schon oben erwähnt, kann man mittels der 'y' Taste zwischen einer logischen Seitennummerierung (default) und einer physikalischen wählen. Was sind dabei nun aber die großen Unterschiede?

In einem normalen plain- $\text{T}_{\text{E}}\text{X}$  Text fallen sie überhaupt nicht auf. Nur wenn man die Seitennummerierung innerhalb eines  $\text{T}_{\text{E}}\text{X}$  Files ändert, ergeben sich Unterschiede.

Zum einen beim Blättern. Kommt man bei der logischen Seitennummerierung mit der `RETURN` Taste von einer Seite immer auf die mit der nächst größeren Seitennummer, so kommt man bei der physikalischen Nummerierung immer auf die Seite, die als nächstes im DVI File der aktuellen Seite folgt.

Aber auch beim Springen an das Ende des Textes ergeben sich Unterschiede. So spring man mittels `SHIFT RETURN` im logischen Modus an die Seite, die die größte logische Seitennummer hat. Im physikalischen Modus dagegen an die letzte Seite innerhalb des DVI Files.

Wie schon erwähnt wechselt man mit Hilfe der 'y' Taste zwischen den verschiedenen Seitennummerierungen. Wie aber ebenfalls schon erwähnt kann man auch nur temporär (nur für die eine auszuführende Aktion) vom logischen Modus in den physikalischen Modus umschalten. Dies geschieht mittels der Control-Taste. Zusammen mit eine der Tasten `RETURN ENTER - BACKSPACE` gedrückt, wird das folgende Wechsel der Seite im physikalischen Modus durchgeführt. Danach ist man sofort wieder im logischen Modus.

## 5. ARexx Port

Wie jedes Programm das etwas auf sich hält, hat auch das ShowDVI-Programm einen ARexx-Port. Der Name des Ports ist "showdvi". Die File-Extension für Scripts ist ".sd".

### 5.1. Die Kommandos

Folgende Kommandos beherrscht das Programm bisher:

<code>first</code>	gehe auf die erste Seite
<code>last</code>	gehe auf die letzte Seite
<code>prev [nr]</code>	gehe <i>nr</i> Seiten zurück, oder falls <i>nr</i> fehlt auf die vorherige Seite
<code>next [nr]</code>	gehe <i>nr</i> Seiten weiter, oder falls <i>nr</i> fehlt, auf die nächste Seite
<code>goto nr</code>	gehe auf Seite Nummer <i>nr</i>
<code>tofront</code>	Screen nach vorne
<code>toback</code>	Screen in den Hintergrund
<code>wbtofront</code>	Workbench-Screen nach vorne
<code>screen</code>	gib die Adresse des Screens hexadezimal aus
<code>fullpage</code>	wechsel zur Anzeige im Ganzseiten-Modus
<code>printpage</code>	drucke die aktuelle Seite aus, oder falls schon ein Ausdruck läuft, breche diesen ab.
<code>setdir dir</code>	setze das aktuelle Directory auf <i>dir</i>
<code>loadnew file</code>	lade ein neues DVI-File
<code>loadagain</code>	lade das aktuelle DVI-File neu
<code>getdir</code>	liefere das aktuelle Directory zurück
<code>getfile</code>	liefere den Filenamen des aktuellen DVI-File zurück
<code>getpage</code>	liefere die aktuelle Seitennummer zurück
<code>togglelace</code>	ändere den Anzeige-Modus ( <i>lace</i> ↔ <i>hires</i> )
<code>togglescrollbar</code>	Scrollbar an/aus
<code>setcolor nr r g b</code>	setze die Screen-Farbe <i>nr</i> auf <i>r g b</i>
<code>setresolution dpi</code>	setze die aktuelle Auflösung auf <i>dpi</i>
<code>beep</code>	setze ein DisplayBeep ab
<code>message string</code>	gib eine Meldung aus. Diese wird in die Screen-Kopfzeile eingetragen und nach sieben Sekunden wieder überschrieben.
<code>refresh</code>	Bildschirm-Refresh (Ausschalten des Fullpage-Modus)
<code>spawn prozedure</code>	interpretiere <i>prozedure.sd</i>
<code>version</code>	liefere den Version-String zurück
<code>saveconfig</code>	speichere die aktuelle Konfiguration ab
<code>activate</code>	aktiviert das ShowDVI-Fenster
<code>getnumber [def] [title]</code>	öffnet einen Requester in dem man eine Zahl eingeben kann. <i>def</i> ist ein Default-Wert und <i>title</i> ist der Titel den der Requester haben soll
<code>getstring [def] [title]</code>	Öffnet einen Requester in dem man einen String eingeben kann. <i>def</i> ist ein Default-Wert und <i>title</i> ist der Titel den der Requester haben soll
<code>okay1 text</code>	Öffnet einen Requester mit dem Text <i>text</i> und einem OK Gadget.
<code>okay2 text</code>	Öffnet einen Requester mit dem Text <i>text</i> und einem OK und einem CANCEL Gadget. Je nach Gadget wird '0' oder '1' in der RESULT Variable zurückgegeben
<code>menu men item sub</code>	Wirkt so, als ob man den entsprechenden Menüpunkt direkt per Maus angewählt hätte. Z. B. <code>menu 0 0</code> öffnet das

About-Fenster  
`exit` beende ShowDVI

Die Ausgabe der Kommandos wie `getfile` oder `getpage` werden bei gesetztem `options results` in der `RESULT` Variable übergeben.

Falls wichtige Kommandos in der bisherigen Implementierung des ARexx-Ports fehlen, so sollte man mir diese mitteilen. Falls sie sinnvoll erscheinen, werde ich sie gerne einbauen.

Falls jemand die kommerzielle AmigaTeX-Version besitzt: Die wenigen ARexx-Befehle des dortigen Previewers sind bei mir ebenfalls implementiert. Diese sind: `getcwd` (entspricht `getdir`), `filename` (= `getfile`) und `pagenum` (= `getpage`). `screen` und `exit` sind ohnehin vorhanden.

## 5.2. Funktionstastenbelegung

Im Konfigurations-File kann für jede Funktionstaste mit und ohne Shift jeweils ein ARexx-Script angegeben werden, das bei Betätigung der entsprechenden Funktionstaste ausgeführt wird. Näheres dazu im Kapitel über die Konfigurations-Files.

## 5.3. Erzeugung nichtgefundener Fonts mittels Metafont

Falls der Previewer einen zur Anzeige benötigten Font nicht finden kann und falls die Environment Variable "CALLMF" gesetzt ist, so wird der Inhalt dieser Environment Variable als Name einer ARexx Script Datei verwendet.

Diese wird zur Erzeugung des nicht gefundenen Fonts mit folgenden Parametern aufgerufen:

- dem Fontnamen des zu erzeugenden Fonts,
- der Auflösung des Fonts in DPI,
- der horizontale Grundauflösung, die zur Zeit am Previewer eingestellt ist,
- der vertikalen Grundauflösung (beim Previewer immer gleich der horizontalen),
- einer Kennung des aufrufenden Programms,
- der Name, den der Font bekommen soll und
- dem Directory, in das der Font kopiert werden soll.

Es gibt schon zwei solcher Scripten. Einmal `MakeTeXFont`. Wenn "CALLMF" mit dieser Script belegt ist, so werden alle fehlenden Fonts sofort generiert. Da dies aber manchmal recht lange dauern kann, gibt es noch die Script `MakeBatch`. Diese Erzeugt die Fonts nicht sofort, sondern generiert nur ein Scriptfile `MF:MakeTeXFont.sh`. Mit Hilfe dieser erzeugten Script kann man später einmal alle fehlenden Fonts erzeugen.

Mehr dazu ist in der Anleitung zu den "CallMF" Scripten zu finden.

# 6. Konfigurations-Files

ShowDVI arbeitet mit drei Konfigurations-Files zusammen. Dies ist zum einen das "Startup-File". Dort wird, unter anderem, das Aussehen des ShowDVI-Screens festgelegt.

In den beiden anderen wird definiert, wo Fonts und Fontlibraries zu finden sind. Eines davon ist in einem ShowDVI/DVIprint-eigenem Format.

Das andere ist im AmigaTeX<sup>4</sup>-Format. Dieses wird **nur** im Verzeichnis `TeX:pk` gesucht.

Die beiden ShowDVI/DVIprint-eigenen Files werden im Directory `TeX:config` gesucht. Sie müssen aber nicht unbedingt dort stehen, das kann mittels der Environment-Variable “`TEXCONFIG`” eingestellt werden. Nach “`setenv TEXCONFIG ram:,TeX:config`” werden diese Konfigurations-Files zum Beispiel erst im Directory `RAM:` und dann im Directory `TeX:config` gesucht. Wenn man aber ein Konfigurationsfile von ShowDVI aus abspeichert (geht nur mit dem Startup-File), so wird es grundsätzlich in das erste Verzeichnis abgespeichert. (Außer dies ist eine Schreibgeschützte Diskette)

In den beiden eigenen Definitions-Files gilt die Regel, alles was nach einem “`;`” kommt, wird mitsamt des Strichpunktes ignoriert. Außerdem sind Leerzeichen in File- oder Directory-Namen verboten. Dies gilt nicht nur für die Definitions-Files, sondern auch für die Environment-Variablen.

Das AmigaTeX-Konfiguration-File `fontvols` wird immer dann eingelesen, wenn ein Font nicht zu finden ist. Die beiden anderen werden nur einmal zum Programm Start eingelesen.

## 6.1. Das Startup-File

Der Name des ShowDVI-Konfigurations-File ist “`ShowDVI.config`”.

Das File wird vom ShowDVI-Programm Zeile für Zeile interpretiert. Zeilen die mit einem Strichpunkt beginnen, oder vor einem Strichpunkt nur Spaces oder Tabs haben, werden ignoriert. Alle anderen Zeilen bestehen aus einem Schlüsselwort und den dazugehörigen Werten. Danach kann ein Kommentar folgen.

Achtung: Wenn man vom ShowDVI-Programm aus das Konfigurationsfile abspeichert, so wird das alte einfach überschrieben. Das heißt, alle per Hand eingefügte Kommentare gehen verloren.

Ein Beispiel für ein solches Startup-File ist:

```
; This is a machine generated configuration file for ShowDVI
color_0      10,10,10      ; RGB values of background color
color_1      2,2,2        ; RGB values of foreground color
color_2      13,13,13     ; RGB values of color 2
color_3      13,1,0       ; RGB values of color 3
scrollbar    on          ; state of scrollbars
interlace    on          ; interlace
beep         off         ; beep on warnings
esc-exit     off         ; exit prog on ESC
quick-exit   on          ; exit prog immediately
```

---

<sup>4</sup> AmigaTeX ist ein kommerzielles Produkt von Radical Eye Software

popup-menu	off	; use pop-up-menu
int-menu	off	; is the pop-up-menu "intelligent"
big-menu	on	; pop-up-menu with topaz 11
middle-menu	on	; use middle-mouse-button
		; for pop-up menu
physical-numbers	off	; on == use physical numbering
auto-load-again	on	; activate window => load again
use-four-colors	on	; use 4 color screen
clone-wb-colors	off	; clone the wb colors
unit	in	; used unit
application-icon	on	; application-icon on/off
app-icon-name	TeX:config/AppIcon	; ".info" file for the app-icon
app-icon-pos	0, 0	; position 0,0 => use def. position
shell-script	TeX:s/ShowDVI.sh	; Script-file (only for OS 2.0)
arexx-start-script	rx TeX-server.rexx	; Script-file, start the TeX-server
		; (only with OS 2.0)
screen-size	0, 0	; screen size. 0,0 => clone wb-size
screen-mode	pal	; screen mode (default   pal   ntsc
		; productivity   a2024   workbench)
default_resolution	100	; default start resolution
resolution_menu	120,100,91,83,44	; entries for the resolution menu
f1	'rexx:CEDtoFront.sd'	
f2	rexx:Start_TeX.sd	
f3	rexx:Quit_TeX.sd	
F1	menu 0 6 3	
F10	menu 0 0	

Die einzelnen Einträge bedeuten dabei folgendes:

#### **color\_?**

Legt die Bildschirmfarben fest.

#### **scrollbar**

Sind die Scrollbars nach dem Programmstart angeschaltet?

#### **interlace**

Wird der Screen im Interlace-Modus geöffnet?

#### **beep**

Sollen wichtige Meldungen mit einem "DisplayBeep" unterlegt werden?

#### **esc-exit**

Soll die ESC-Taste das Programm beenden oder nur den Screen nach hinten legen?

#### **quick-exit**

Wenn dieser Punkt aktiviert "on" ist, so wird kein Abfragerequester geöffnet, wenn man das Programm verlassen möchte. Statt dessen wird das Programm sofort beendet.

#### **popup-menu**

Soll das Pop-Up-Menü anstatt des normalen Amiga-DOS Menü auf der rechten Maus Taste liegen?

#### **int-menu**

Soll sich das Pop-Up-Menü den letzten Menüpunkt merken?

**big-menu**

Soll der Font "topaz 11 für das Pop-Up-Menü verwendet werden?

**middle-menu**

Soll das Pop-Up-Menü auf der mittleren Maustaste liegen? (Funktioniert nur unter 2.0 und auch nur mit einer Maus, die (mindestens) drei Tasten hat.

**physical-numbers**

Soll das Programm sich nach dem Start gleich im "physikalischen" Seitennummerierungsmodus befinden? Siehe dazu auch die Bemerkungen zur Taste 'y'.

**auto-load-again**

Soll die Funktion "auto-load-again" defaultmäßig eingeschaltet sein?

**use-four-colors**

Soll der ShowDVI-Screen mit 4 Farben geöffnet werden?

**clone-wb-colors**

Soll die Farbwahl der `color_?` Zeilen ignoriert werden und statt dessen die Farben der Workbench übernommen werden. Diese Einstellung wird automatisch deaktiviert, wenn man mittels des Farb-Requesters die Farbeinstellung ändert und diese abspeichert.

**unit**

Welche Einheit soll im "Measure-Window" verwendet werden? Zur Auswahl stehen: `in`, `cm` oder `pt`.

**application-icon**

Soll auf der Workbench ein Application-Window geöffnet werden? Mit diesem kann man durch Doppelklick den ShowDVI-Screen nach vorne holen bzw. durch darauflegen eines Icons, dieses File laden. (Funktioniert nur unter 2.0)

**app-icon-name**

Vollständiger Pfad eines Info-Files, welches das Aussehen des App-Icons festlegt.

**app-icon-pos**

Wo auf der Workbench soll das Icon positioniert werden? Es ist allerdings empfehlenswert hier die Werte 0 zu belassen, da sich dann die Workbench selbst einen möglichst guten Platz für das Icon überlegt. Dann kann es nicht vorkommen, daß dieses icon über ein anderes gelgt wird.

**shell-script**

Pfadname einer Shell-Script, die mit dem 'r' Kommando ausgeführt wird. (Funktioniert nur unter 2.0)

**arexx-start-script**

Aufruf einer ARexx-Script, die die Zusammenarbeit Editor – virtex – Previewer steuert.

**screen-size**

Legt die Screen-Größe fest. Normalerweise sollten die Werte auf 0 stehen. Dann verwendet ShowDVI die Größe des Workbench-Screens. (Unter 2.0 wird der Text-Overscan ausgewertet) Stehen dort Werte ungleich Null und läuft das Programm unter 2.0, so wird der Screen mit dem Tag `AUTOSCROLL` geöffnet. Dies bedeutet, daß man den Screen größer als den sichtbaren Bereich

machen kann. Man scrollt dann mittels der Maus.

### **screen-mode**

In welchem Modi soll der Screen geöffnet werden? Zur Auswahl stehen: `PAL`, `NTSC`, `productivity`, `A2024`, `default` und `workbench`. Bei `A2024` legt 'Interlace' fest, ob im 10HZ oder 15Hz Modus gearbeitet werden soll. Bei `default` bestimmt die Hardware des Rechners, ob ein PAL oder NTCS Screen verwendet wird. Wenn `workbench` angegeben wurde, so wird der Screen-Mode des Workbench-Screens übernommen. Dabei wird auch die Interlace-Einstellung übernommen. D.h. falls als Screen-Mode `workbench` gesetzt wurde, so wird die `interlace` Einstellung ignoriert. (Funktioniert nur unter 2.0)

### **default\_resolution**

Mit welcher Auflösung soll normalerweise das Programm gestartet werden.

### **resolution\_menu**

Welche Einträge soll das Resolution-Menü haben? Es können maximal 10 Auflösungen angegeben werden.

#### **f1**

Belgt die Funktionstaste F1 (ohne SHIFT) mit dem nachfolgenden Text.

#### **F1**

Belgt die Funktionstaste F1 (mit SHIFT) mit dem nachfolgenden Text.

#### **f9**

Belgt die Funktionstaste F9 (ohne SHIFT) mit dem nachfolgenden Text.

Mit dem Menüpunkt `save config` oder der Tastatursequenz `AMIGA-W` wird der derzeitige Zustand abgespeichert. Dazu gehören z.B. die verwendeten Farben, die gewählte Auflösung, der Zustand der Scrollbars usw.

## **6.2. Das Fontdefinitions-File**

In diesem File können zusätzliche Angaben gemacht werden, wo einzelne Fonts oder bestimmte Fontlibraries zu finden sind. Der vollständige Suchpfad wird im nächsten Kapitel "Internas" noch genau erläutert.

Das Fontdefinitions-File wird normalerweise in `TeX:config` gesucht. Man kann dieses Directory aber auch mit der Variable `TEXCONFIG` überschreiben. In dieser Variable kann man auch einen Suchpfad definieren. Der Name des Konfigurations-Files ist `showdvi.fnt`.



Nun zum Aufbau dieses Files. Hier ein Beispiel:

flibenv		dir1 dir2		; directories with flib's
pkenv		dir1 dir2		; directories with pk-files
basepkdir_str		%d/%s.%dpk		; formatstring for pk-dir's
flib_str		PK%04d		; formatstring for flib's
pk_str		%d/%s.%dpk		; formatstring for pk-files
pkdir_str		%s.pk		; formatstring for pk-files ; which are in pkdir
basepkdir		hdpi vdpi	directory	; two args
flib		dpi	directory	; two args
font	font	dpi	directory	; three args
pkdir		hdpi vdpi	directory	; two args
basepkdir		300 300	TeX:PrtFonts	; example
flib		100	ram:fastflib	; example
font	cmr10	100	ram:fastpk	; example
pkdir		109 109	TeX:pk/110	; example
pkdir		110 110	TeX:pk/110	; example

Mit den ersten zwei Schlüsselwörtern kann man zusätzliche Directories definieren, in denen entweder nach Fontlibraries, oder nach PK-Fonts gesucht werden sollen. Von der Funktion her entsprechen diese beiden Zeilen den Environment Variablen FLIBDIR und PKDIR. Mehr dazu im Kapitel 7.1.

Die nächsten vier Zeilen definieren wie jeweils der Fontname gebildet werden soll. Als Argument wird dabei ein Format String nach C Art angegeben.

Dabei können folgende Modifier angegeben werden.

- “%s” steht für den Font Namen (z.B. cmr10),
- “%d” steht für die aktuelle dpi-Zahl,
- “%h” steht für die dpi-Auflösung in horizontaler Richtung (entspricht “%d”),
- “%v” steht für die dpi-Auflösung in vertikaler Richtung,
- “%x” ist eine ältere Einheit, entspricht der horizontalen dpi mal fünf,
- “%y” wie “%x” nur für die vertikale Auflösung,
- “%%” steht schließlich für ein “%” Zeichen.

Die “%h” “%v” “%x” “%y” Modifier sind für den Previewer nicht so sehr von Bedeutung. Sie werden vor allem für nicht quadratische Fonts (z.B. die Epson FX-80 Fonts) benötigt.

Die drei Format-Strings der obigen Tabelle sind auch gleichzeitig der jeweilige Default-Wert des Programms.

Insgesamt ist der Formatstring vollständig C-kompatibel. Beispiele hierzu:

Flib-fmt	PK-fmt	dpi	Name	Flib	Font
PK%04d	%d/%s.%dpk	100	cmr10	PK0100	100/cmr10.100pk
PK%d	%s.%d	83	cmbx8	PK83	cmbx8.100
%x.lib	%s/%d	360	lasy8	1800.lib	lasy/360

Die Zeilen, die mit “basepkdir” beginnen, definieren ein Directory, in dem ein gesamter Satz von zusammengehörenden Fonts (alle Vergrößerungsstufen) zusammengefaßt sind. Mit “`basepkdir 300 300 TeX:PrtFonts`” definiert man also, daß alle Fonts, die auf der (quadratischen) Grundauflösung 300dpi basieren (z.B. auch die Magstep 1 Fonts mit 360dpi), sich in dem Directory `TeX:PrtFonts` befinden. Hat man allerdings als eingestellte Grundauflösung 360dpi und sucht einen Magstep 0 Font (also mit 360dpi), so wird **nicht** in diesem Directory gesucht. Bei quadratischen Fonts kann man die vertikale Auflösung auch weglassen.

Der erste “basepkdir” Eintrag zu einer bestimmten Auflösung hat auch noch zwei Sonderfunktionen. Zum einen wird dieses Directory den CallMF AReXX Scripten mit übergeben, wenn diese einen fehlenden Font generieren sollen. Zum anderen wird, falls ein Font nur über das “fontvols” File (Font-Caching) gefunden wurde, in das durch das erste passende “basepkdir” kopiert. Somit ist es also möglich, die Fonts, gehörend zu den verschiedenen Druckern/Auflösungen in verschiedene Directories aufzuteilen, und diese Aufteilung wird auch bei der Generation neuer Fonts, oder bei dem Kopieren von Fonts, beibehalten.

Die Fonts auf den “fontvols” Disketten werden im übrigen mit dem Formatstring gesucht, den der erste passende “basepkdir” Eintrag hat. Falls es keinen passenden “basepkdir” Eintrag gibt, so wird der default Formatstring für pk-Files genommen.

Die Zeilen mit dem Schlüsselwort “flib” definieren einzelne Fontlibraries und in welchen Directories sie sich befinden. Nach `flib` kommen zwei Parameter: zuerst eine Zahlenangabe für die DPI der Fontlibrary, danach das Directory, in dem die Fontlibrary zu finden ist.

Ähnlich sind die Zeilen mit dem Schlüsselwort “font” aufgebaut. Diesmal sind drei Parameter anzugeben. Der erste Parameter gibt den Font-Namen an, wie er in einem  $\TeX$ -File definiert ist. Danach kommen die DPI-Angabe und das Directory, in welchem der Font zu finden ist. Auch hier wird nur ein einziges Object definiert (ein einziger Font).

Die Zeilen, die mit “pkdir” beginnen, dienen vor allem dazu, ein und den selben Fonts für verschiedene Auflösungen zu benutzen. Wird ein Font in einem dort definierten Directory gesucht, so wird der Formatstring “pkdir\_str” verwendet. In dem Beispiel von oben würde der Font `cmr10` sowohl in der Auflösung 109 dpi als auch in 110 dpi im Directory `TeX:pk/110` unter dem Namen `cmr10.pk` gesucht werden (falls er nicht schon vorher in einer Fontlibrary oder als “normaler” PK-Font gefunden wurde).

Zu beachten sind auch noch zwei Feinheiten. Zum einen wird die Reihenfolge der Aufschreibung im File *innerhalb einer* Gruppe von Einträgen (z.B. alle `flib`

Einträge) beim Suchen nach Fonts berücksichtigt. Man kann also z.B. mehrere Fontlibraries der selben Auflösung definieren und mit der Reihenfolge bestimmen, welche zuerst abgesucht werden soll.

Eine weitere Besonderheit ist, dass man die Format-Strings mehrfach definieren kann. Verwendet wird dann immer der zu letzt definierter String.

Ein Beispiel:

```

flib_str  PK%04hx%04v          ; wegen unsymmetrischen Fonts
flib      120                  TeX:fontlib ; verwende Format String von oben
flib      131                  TeX:fontlib ; ebenso
flib_str  PK%04d              ; setzt Format-String zurück
flib      100                  TeX:fontlib ; verwendet den neuen Format-String

```

Die ersten zwei definierten Fontlibraries verwenden nun den Format-String mit der horizontalen und vertikalen Auflösung. Die dritte definierte Library und *alle*, die noch bei der Suche nach Fonts während des Programmlaufs definiert werden, verwenden den zweiten definierten Format-String.

Von großem Nutzen ist dieses Definitions-File vor allem für Benutzer mit beschränktem Plattenplatz. Man kann sich zuerst heraussuchen, welche Fonts unbedingt benötigt werden. Eine große Hilfe dabei ist das Logfile, das jeden benötigten Font auflistet. Man erstellt dann für jede benötigte Auflösungsstufe eine Fontlibrary, in der die zuvor ermittelten Fonts hineingepackt werden. Diese werden auf die Festplatte kopiert. Die vollständigen Fontlibraries kann man auf Diskette halten oder, falls sie zu groß sind, auf mehrere Disketten verteilen. Wenn man alle Libraries explizit definiert, werden die Fonts in der Reihenfolge, wie sie aufgeführt sind, im File gesucht. Zuerst also auf Platte und – wenn dort nicht vorhanden – auf der Diskette, die man angegeben hat.

Einzelne Fonts zu definieren kann für Besitzer von METAFONT von großem Nutzen sein. Möchte man zum Beispiel den Font `cmr10` neu erzeugen, so braucht man ihn nicht schon während der Testphase aus allen Fontlibraries zu löschen und neu einzutragen. Statt dessen definiert man den Font im Definitions-File. Da dieser dann zuerst gefunden wird, berücksichtigt ShowDVIden gleichnamigen Font in den Fontlibraries nicht.

Möchte man sein Fontdefinitions-File überprüfen, so sollte man ShowDVI mit der Option `STAT` starten. Vor dem Programmende wird das Definitions-File mit Kommentaren (gebraucht/gefunden/nicht gefunden) in das Logfile geschrieben.

### 6.3. Das Font-Caching Konfigurations-File

Wenn ein Font nach allen bisherigen Versuchen immer noch nicht gefunden wurde, dann wird das Font-Caching<sup>5</sup> Konfigurations-File `fontvols` zu Rate gezogen.

Dieses besteht aus mehreren Zeilen mit folgendem Format:

$$\text{Volume-Name}=\text{base-dpid} . \text{dpi}_1 . \text{dpi}_2 \dots \text{dpi}_n$$

<sup>5</sup> Dieses File ist kompatibel zu dem gleichnamigen AmigaTeX Konfigurationsfile

“Volume-Name” ist der Name einer Diskette.

“base-dpi” gibt die Grundauffösung an, zu der die Fonts dieser Diskette gehören. Damit kann man bei Überschneidungen verschiedener Fontgruppen (z.B. der Satz 300dpi Fonts und der Satz 360dpi Fonts überschneiden sich) festlegen, zu welcher Grundauffösung diese Fonts gehören. (z.B. sind dies 360dpi Fonts eine Diskette 300dpi Fonts der Vergrößerung *magstep1* oder sind es 360dpi Fonts der Vergrößerung *magstep0*. Falls man diese Unterscheidung nicht treffen möchte, kann man als “base-dpi” auch “0” angeben. Dann wird immer verglichen, ob in dieser Zeile die gesuchten Einträge sind. Dies ist vor allem bei ShowDVI sehr sinnvoll, da man dort ja mit vielen verschiedenen Auflösungen arbeiten kann. Dort provoziert man auch möglichst viele Überschneidungen, um mit möglichst wenigen Fonts die Texte in möglichst vielen verschiedenen Auflösungen anzeigen zu können.

Die einzelnen “dpi”-Werte geben an, in welchen Auflösungen Fonts auf der Diskette zu finden sind.

Sowohl bei der “base-dpi” als auch bei den einzelnen “dpi”-Werten kann auch horizontale und vertikale Auflösung getrennt angegeben werden. So steht “360x180” für eine horizontale Auflösung von 360dpi und eine vertikale Auflösung von 180dpi. Nur “360” steht für die quadratische Auflösung von 360dpi sowohl für die horizontale als auch die vertikale Auflösung. “360x360” und “360” sind also identisch. (Achtung: hier wird **nur** ein kleines ‘x’ akzeptiert.)

Ein kurzes Beispiel:

Mit “FontDisk=000d.131.207” wird definiert, daß es eine Diskette mit dem Namen “FontDisk:” gibt, die zwei Directories enthält. Zum einen das Directory “FontDisk:131” mit den Fonts der Auflösung 131dpi, und zum anderen das Directory “FontDisk:207”, in dem die ausgelagerten Fonts der Größe 207dpi zu finden sind. Da “base-dpi” auf Null gesetzt ist, wird immer, egal bei welcher Grundauffösung, dieser Eintrag ausgewertet.

Eine Auflösung darf dabei mehrfach auftreten. Die Suchreihenfolge ist von oben nach unten und von links nach rechts.

Dieses Konfigurationsfile wird zuerst in “TeX:config” und dann in “TeX:pk” gesucht. Das Directory “TeX:config” kann mit der Environmentvariable `TEXCONFIG` überschrieben werden. Nach “setenv TEXCONFIG .,ram” wird das File also zuerst im aktuellen Directory, dann in `RAM:` und schließlich in `TeX:pk` gesucht.

Wird ein Font mit Hilfe des “fontvols” File gefunden, so wird es automatisch in das der Grundauffösung zugehörigen `basepkdir` oder, falls nicht definiert, nach `TeX:pk` kopiert. Dies macht dieses File äußerst interessant. Ist es doch damit möglich, alle Fonts auf mehreren Disketten zu halten und nur die Fonts, die wirklich gebraucht werden, werden in das der Grundauffösung zugeordnetem Verzeichnis kopiert. Dies erfordert die ersten paar Mal einiges an Diskettenwechsel, spart aber erheblich an Festplattenplatz!

Die Fonts auf den “fontvols” Disketten werden im übrigen mit dem Formatstring gesucht, den der erste passende “basepkdir” Eintrag hat. Falls es keinen passenden

"basepkdir" Eintrag gibt, so wird def default Formatstring für pk-Files genommen. Mit Hilfe dieses Formatstrings werden die kopierten Fonts auch angelegt.

## 7. "2.0" Features

Einige Besonderheiten des neuen AMIGA Betriebssystem konnten schon in dieser Version berücksichtigt werden. Sie sind schon an der jeweilig passenden Stelle innerhalb dieser Anleitung erwähnt worden, sollen hier aber noch einmal kurz zusammen gefaßt werden.

### 7.1. Screen Modis

Es werden alle mir bisher bekannten AMIGA Screen-Modis unter 2.0 unterstützt<sup>6</sup>.

Diese sind: *PAL*, *NTSC*, *Productivity (VGA)* und *A2024*<sup>7</sup>

Diese Modi können entweder im *interlace* oder *non-interlace* Verfahren verwendet werden. Bei der A2024 Einstellung bedeutet diese die Unterscheidung zwischen der 10 Hz und der 15 Hz Auflösung.

Ausgewählen kann man den gewünschten Monitor und Auflösung über das Konfigurationsfile. Siehe dazu Kapitel 6.1.

Unter 2.0 wird die entgültige Auflösung noch über den `Text-OverScan` ermittelt. Diese kann über die Preferences eingestellt werden.

Falls im Konfigurationsfile eine feste Größe für den Screen angegeben wurde, so wird der Screen mit dem `AUTOSCROLL` Flag geöffnet. Das heißt, daß der Screen automatisch gescrollt wird, falls er größer als der sichtbare Bereich ist und man mit der Maus an den Rand fährt.

### 7.2. Public Screen

Eine der herausstechenden Eigenschaften des neuen Betriebssystem ist sicherlich die "Public-Screen" Verwaltung. Nun wurde es möglich gemacht, daß verschiedene Applicationen sich den selben Screen teilen können. Man muß lediglich den Namen des Screen kennen. Der Name des ShowDVI-Screens ist `ShowDVI-PubScr`. Man kann nun zum Beispiel ganz einfach ein `CON: Window` auf dem ShowDVI-Screen öffnen. Es reicht nur die Angabe von:

```
"CON:0/0/300/100/Title/SCREEN ShowDVI-PubScr/CLOSE"
```

Das ShowDVI Programm verwendet diese Möglichkeit selbst immer, um ein Ausgabefenster für ein externes Programm/Shell zu öffnen.

### 7.3. Application Icon

Das Application-Icon ist eine sehr praktische Einrichtung für Leute, die gerne mit der Workbench arbeiten. Man kann nun von der Workbench per Maus durch Dop-

---

<sup>6</sup> Abgesehen von den mehrfarbigen wie HAM usw.

<sup>7</sup> Der A2024 Modus ist nicht getestet, da ich keinen solchen Monitor besitze.

pelklick auf das Icon ShowDVI nach vorne holen oder durch Ziehen eines Icons auf dieses Icon ein neues DVI-File laden.

Das Aussehen des Application-Icons wird über ein Info-File festgelegt. Der Name dieses Info-Files wird im Konfigurationsfile festgelegt. Per Default wird das File `TeX:config/AppIcon.info` verwendet.

Die Lage des Icons kann (wenn man unbedingt möchte) auch festgelegt werden. Dazu dieht der Konfigurationseintrag `app-icon-pos`. Wenn dort zweimal '0' steht, so wählt die Workbench selbst einen Platz für das Icon.

## 8. Internas

Dieses Kapitel soll nun die eine oder andere Hintergrundinformation über die Funktionsweise von ShowDVI vermitteln. Diese Informationen werden zum einfachen Gebrauch des Programms zwar nicht unbedingt benötigt, können aber recht nützlich sein.

### 8.1. Environment-Variablen

Es werden nur die von Commodore unterstützten ENV: Environment Variablen verwendet.

Folgende Variablen werden verwendet:

- SHOWDVI In dieser Variable können Kommandline-Optionen abgespeichert werden, die ständig benötigt werden.
- TEXCONFIG Liste von Pfaden, in denen nach den Konfigurationsfiles gesucht wird.
- FLIBDIR Liste von Pfaden für das Finden von Fontlibraries.
- PKDIR Liste von Pfaden für das Finden von PK-Fonts.
- CALLMF wenn diese Variable gesetzt ist, wird bei einem nicht gefundenen Font die ARexx Script aufgerufen, deren Name in der Environment Variable steht.

Eine Liste von Pfaden besteht aus mehreren Directories getrennt durch “;” oder durch “,”. Zum Beispiel:

```
ram:dir1,df1:dir2,fonts:
```

Leerzeichen innerhalb von Directories sind nicht erlaubt.

### 8.2. Verwendete Files

Folgende Files werden von ShowDVI verwendet:

- ShowDVI.config Konfigurationsfile für das allgemeine Aussehen und Verhalten von ShowDVI.
- ShowDVI.fnt Fontdefinitionsfile, das bei der Suche nach Fonts hilft.
- ShowDVI.help Hilfile für die “On-Line” Hilfe. In ihm stehen die Erklärungen zu allen Menüpunkten und die Hilfeseiten des about/help Fensters.

ShowDVI.log	Logfile in dem ShowDVI diverse Dinge mitprotokolliert. Der Name (und Pfad) des Files kann mit der Option <code>logname</code> geändert werden.
fontvols	Font-Caching File. Hier wird definiert, auf welchen Disketten welche Fontvergrößerungen abgelegt sind.

Die Namen und Pfade weiterer verwendeter Files werden in dem File "ShowDVI.config" definiert.

### 8.3. Font-Such-Pfade

Eines vorweg: das Programm bevorzugt nach Möglichkeit immer die Fontlibraries, die meiner Meinung nach viele Vorteile haben.

Das sind (kurz zusammengefaßt):

- sie sparen einiges an Speicherplatz
- auf sie kann schneller zugegriffen werden
- die Directory-Struktur ist übersichtlicher

Aber das Programm kann auch gänzlich ohne Fontlibraries arbeiten. Es ist also kein Zwang, sich solche Libraries zuzulegen. Nur in der Suchreihenfolge werden die Fontlibraries bevorzugt.

Wo werden die Fonts gesucht?

Startet man ShowDVI ohne weitere Vorbereitungen, so werden Fonts zuerst innerhalb von Fontlibraries in `TeX:FontLib` und dann als eigenständiges PK-File in `TeX:pk` gesucht. Der jeweilige Filename wird nach folgenden Format Strings ermittelt:

```
PK%04d      für die Fontlibraries, und
%d/%s.%dpk für die eigenständigen PK-Files.
```

Wie schon im Kapitel über das Fontdefinitions-File gesagt, können diese Formatstrings natürlich auch undefiniert werden.

Als erste Möglichkeit der Umkonfigurierung bieten sich zwei Environment-Variablen an. Mit `FLIBDIR` können weitere Directories definiert werden, in denen sich Fontlibraries befinden und mit `PKDIR` Directories in denen PK-Files gefunden werden können. Nach

```
setenv FLIBDIR DH0:flib1,DH1:flib2,FlibDisk:
setenv PKDIR  DH0:pk1,PKDisk:
```

wird ein Font in folgenden Directories gesucht:

- `DH0:flib1`, innerhalb einer Fontlibrary.
- `DH1:flib2`, innerhalb einer Fontlibrary.
- `FlibDisk:`, innerhalb einer Fontlibrary.
- `DH0:pk1`, als eigenständiges PK-File.
- `PKDisk:`, als eigenständiges PK-File.
- Wurde der Font nun immer noch nicht gefunden, so wird im File `TeX:pk/fontvols` nach einer Diskette gesucht, auf der der Font noch sein könnte.

Wurde mittels der Environmentvariable `FLIBDIR` oder dem Eintrag `flibenv` im Konfigurationsfile ein Suchpfad für Fontlibraries definiert, so wird der Defaulteintrag `TeX:fontlib` gelöscht. Falls also dieses Directory benötigt wird, so muß man es selbst in die Liste mit aufnehmen. Dadurch kann man zur Not auch völlig ohne das Assign `TeX:` arbeiten.

Das selbe gilt für `PKDIR` und `pkenv`.

Falls in der Kommandozeile mit der `fontdir` Option noch ein weiteres Directory angegeben wurde, so wird dieses jeweils vor dem aktuellen Directory durchsucht. In dem Directory wird also vor dem `TeX:FontLib`-Directory nach einer passenden Library gesucht und wenn bis zum Ende der Suchliste der Font immer noch nicht gefunden wurde, so wird vor dem `TeX:pk`-Directory noch in dem Directory, das nach der `fontdir` steht, Option gesucht.

Reichen diese Möglichkeiten immer noch nicht aus, so können innerhalb des Fontdefinitions-File Fontlibraries und Fonts auch noch einzeln definiert werden.

Dabei wird dann folgende Suchreihenfolge angewand:

- Als erstes wird in der Liste der einzeln definierten Fonts gesucht.
- Danach wird in der Liste der vordefinierten Fontlibraries nach Libraries der korrekten DPI-Zahl und innerhalb dieser nach dem Font gesucht.
- Falls der Font noch nicht gefunden wurde, so werden jetzt die Flib-Directories abgesucht. Definiert werden diese mit der Environment Variable `FLIBDIR` und dem Eintrag `flibenv` des Fontdefinitionsfiles.
- Nun wird in der Liste der PK-Directories nach dem passenden PK-File gesucht. Diese Liste kann geändert werden mittels der Environment Variable `PKDIR` und dem Eintrag `pkenv` des Fontdefinitionsfiles.
- Als vorletzter Versuch den Font doch noch zu finden werden als nächstes die über den `pkdir` Einträgen definierten PK-Directories abgesucht.
- Zu guter letzt wird das File `TeX:pk/fontvols` zu Rate gezogen. Falls der Font damit doch noch gefunden wurde, so wird dieser nach `TeX:pk` kopiert. Dies nennt man auch *Font-Caching*, da man so seine Fonts einfach alle auf Diskette halten kann und nur die, die wirklich benötigt werden, werden in den `TeX:pk` Bereich kopiert. Siehe dazu auch Kapitel 7.3.

Falls alle diese Versuche, einen Font zu finden, fehlschlagen und die Environment Variable “CALLMF” mit einem AREXX-Script Namen belegt ist, so wird diese Script mit den drei Parametern “Font Name”, “Font Auflösung” und “Grund Auflösung” aufgerufen. Sobald diese Script mit dem Return-Code Null beendet ist, wird der gesamte Such-Algorithmus noch einmal durchlaufen. Falls also diese Script den Font mittels Metafont erzeugt hat, braucht so ihn lediglich an einen beliebigen Ort innerhalb des Suchraumes zu kopieren, und ShowDVI verwendet ihn sofort.

## 8.4. Font-Caching

Ebenfalls von AmigaTeX übernommen wurde die Idee des “Font-Caching’s”. Wurde ein Font erst mittels des `fontvols`-File auf einer Diskette gefunden, so wird dieser in das der Grundauflösung zugeordnetem “basepkdir”-Directory oder in das `TeX:pk`



Directory kopiert. Dies erleichtert vor allem Benutzern, die ohne – oder sehr wenig – Plattenplatz auskommen müssen, die Auswahl der Fonts, die sie unbedingt schnell bereit haben müssen.

Ein  $\text{\TeX}$ -Benutzer, der lediglich zwei Diskettenlaufwerke und keine Festplatte besitzt, kann sich dann folgendermaßen helfen:

Auf eine Diskette kommt die Workbench mit den benötigten Programmen wie `virtex` und `showdvi`. Auf die zweite Diskette sollten die `tfm`-Files und die von `virtex` benötigte Format-Datei kopiert werden. Dazu kommt ein Directory mit Fontlibraries, die nur die wirklich ständig benötigten Fonts enthalten. Außerdem muß auf dieser Diskette ein Verzeichnis `TeX:pk` existieren, in der die `fontvols`-Datei steht. In dieser sind alle Disketten aufgeführt, die PK-Files enthalten.

Wird nun ein Font nicht auf der Arbeitsdiskette gefunden, so wird die nötige Zusatzdiskette angefordert und der Font in das Directory `TeX:pk` kopiert. Beim nächsten mal ist dieser Font also sofort verfügbar. Haben sich nun nach einiger Zeit mehrere Fonts in `TeX:pk` angesammelt, so können sie mit Hilfe des Fontlibrary-Programms `flib` in die Fontlibraries eingetragen werden. Dies hat den Vorteil, daß sie dann etwas weniger Platz auf der Diskette benötigen<sup>8</sup> und der Zugriff auch etwas schneller geht.

## 8.5. Font-Speicher

Um einen möglichst schnellen Seitenwechsel zu ermöglichen, werden die schon geladenen Fonts im RAM gehalten. Dies wird solange praktiziert, bis der dafür reservierte Speicherbereich voll ist. Danach wird nach der LRU<sup>9</sup>-Methode der Font, der am längsten nicht mehr benötigt wurde, aus dem Speicher entfernt.

Es kommt natürlich auf den eigenen Rechner und seinen Speicher an, um festzulegen, wieviel Speicherplatz man für die Fonts opfern kann und möchte.

Mit der Option `HELP` kann der Default-Wert abgefragt und mit `FONTMEM=bytes` beim Aufruf auf einen beliebigen Wert gesetzt werden.

Um den für den eigenen Rechner optimalen Wert zu finden, ist auch hier das Logfile eine große Hilfe. Dort wird hinter jedem Eintrag für einen geladenen Font die prozentuale Auslastung des Font-Speichers angegeben.

## 8.6. Seitenbitmap – Auflösungen

Damit das Scrollen innerhalb einer Seite möglichst flüssig ist, wird die gesamte Bitmap der aktuellen Seite im CHIP-RAM gehalten. Da das CHIP-RAM leider recht beschränkt ist, ergibt sich so auch eine Beschränkung der verwendbaren Auflösungen. Normal große Seiten kann man noch mit etwa 180 DPI anzeigen, wenn sonst nicht mehr viel CHIP-RAM benötigt wird. Bei mir sind dann noch etwa 5 kB CHIP-RAM (von 512 kB) frei.

---

<sup>8</sup> im Durchschnitt 768 Bytes pro PK-File

<sup>9</sup> last recently used

Die Größe des ShowDVI-Screens hängt von der Größe des Workbench-Screens ab. Wurde die Workbench im Overscan-Modus gestartet, so richtet sich auch ShowDVI danach. Ist der Workbench-Screen im Noninterlace-Modus und soll der ShowDVI-Screen im Interlace-Modus geöffnet werden, so wird die Y-Auflösung der Workbench verdoppelt.

## 8.7. Ausdruck einer Seite

Da intern die Seite als RastPort abgelegt ist, kann sie natürlich sehr einfach mit dem io\_Command PRD\_DUMPSPORT ausgedruckt werden. Da diese über den in den Preferences eingestellten Druckertreiber funktioniert, kann man auch mit jedem grafikfähigen Drucker diese Funktion verwendet werden. Allerdings ist die Qualität des Drucks nicht so besonders gut, da ShowDVI mit einer geringen Auflösung arbeitet. Mit 1 MB CHIP-RAM müßte man mit bis zu 300 DPI drucken können.

Einen noch nicht fertigen Ausdruck erkennt man daran, daß rechts oben in der Window-Titelzeile ein "P" steht. Während dessen kann weder die aktuelle Seite verlassen noch ein neues DVI-Files geladen werden. Abgebrochen kann ein laufender Ausdruck mit dem Menüpunkt oder der Taste, mit der er gestartet wurde.

## 8.8. Taskname und Signale

Der Taskname des Programms wird sofort nach Programmbeginn auf "ShowDVI-Task" gesetzt. Zuvor wird allerdings noch geprüft, ob es schon eine Task mit diesem Namen gibt. Wenn ja, so wird dieser ein Ctrl-E-Signal geschickt. Dieses bringt das andere ShowDVI-Programm dazu, seinen Screen nach vorne zu holen und sein Fenster aktiv zu machen. Das soeben gestartete ShowDVI-Programm beendet sich nach dem Versenden des Signals sofort<sup>10</sup>.

Wurde bei dem zweiten Aufruf von ShowDVI ein Filename angegeben (Wichtig: Filename *einzig*er Parameter), so wird nach dem Ctrl-E Signal noch das ARexx Kommando

```
address "showdvi" "loadnew %s"
```

ausgeführt, wobei für %s der Filename eingesetzt wird, der als Parameter dem Programm mit übergeben wurde. Dabei ist allerdings zu beachten, daß das File bezüglich dem Directory gesucht wird, das bei dem als erstes gestarteten ShowDVI eingestellt ist.

Noch ein weiteres Signal verwendet ShowDVI – das Ctrl-F Signal. Wenn ShowDVI dieses Signal empfängt, so wird der ShowDVI Screen nach vorne geholt und das aktuelle File neu geladen.

Man kann also auch ohne Besitzer von ARexx zu sein, im beschränkten Rahmen mit dem ShowDVI-Programm kommunizieren. Man kann es zumindestens mit einem Signal nach vorne holen, aktivieren und das bereits geladene DVI-File neu einlesen. Die Signale kann man entweder von einem eigenen Programm verschicken, mittels

---

<sup>10</sup> Daraus folgt, daß immer nur ein ShowDVI zur gleichen Zeit laufen kann.

des CLI-Kommandos `break` absetzen oder sich ganz einfach durch einen erneuten Aufruf von `ShowDVI` erzeugen lassen.

Noch ein Wort zum automatischen Aktivieren von `ShowDVI`. Wenn man mittels `Ctrl-E` oder `Ctrl-F` `ShowDVI` aktiviert, so merkt sich dieses das zuvor aktive Window. Wenn man nun mit der `ESC`-Taste den `ShowDVI`-Bildschirm wieder nach hinten legt, so wird automatisch das zuvor aktive Window wieder aktiviert.

Auf noch ein weiteres Signal achtet `ShowDVI`. Und zwar auf `Ctrl-C`. Man kann `ShowDVI` damit praktisch in jeder Lebenslage, sofort und ohne Nachfrage, beenden.

## 8.9. Benötigter Stack

Im Gegensatz zu früheren `ShowDVI` Versionen gibt sich diese nun mit 4000 Bytes Stack zufrieden.

## 8.10. In den Hintergrund starten

Wenn `ShowDVI` als Hintergrundprogramm gestartet wurde, so werden keine Ausgaben mehr nach `“stdout”` geschickt. Das bedeutet, daß im Fehlerfall auch keine Fehlermeldungen mehr sichtbar sind (falls sich der `ShowDVI`-Screen so schnell verabschiedet, daß man die Meldung dort nicht mehr lesen konnte). Es empfiehlt sich also in dem Fall das Logfile nicht auszuschalten und dann anzusehen. Dort steht die genaue Fehlerursache.

Stürzt das Programm unkontrolliert ab, so ist leider das Logfile nicht mehr lesbar (außer man hat das Programm mit der Option `DEBUGSTAT` gestartet. Man sollte es dann noch einmal im Vordergrund starten, um zu sehen, ob wenigstens auf dem Window noch eine Fehlermeldung erscheint.

Auf jeden Fall ist ein unkontrollierter Absturz etwas so Ungewöhnliches, daß man mich sofort benachrichtigen sollte. Aber bitte mit Informationen, wie ich den Fehler reproduzieren kann!

## 8.11. Req.library

Das Programm funktioniert im Prinzip auch ohne die `“req.library”`. Allerdings fallen dann einige Funktionen aus: Das sind hauptsächlich der File- und der Font-Requester. Aber auch der Stringrequester der manchmal benötigt wird, verwendet den Requester aus der `“req.library”`.

Unter den Betriebssystemversionen 1.2 und 1.3 werden auch die Bestätigungsrequester, die mittels den ARexx Kommandos `okay1` und `okay2` erscheinen, über die `“req.library”` angezeigt.

# 9. Das Logfile

Diese File wurde schon öfters im Rahmen dieser Anleitung angepriesen. Wenn man mit einer Festplatte arbeiten sollte (was sowieso zu empfehlen ist), sollte man auf die Option `NOLOG`, die das Logfile abschaltet, verzichten. Der Geschwindigkeitsverlust

durch das Anlegen des Logfiles ist so minimal, daß der etwaige Nutzen auf jeden Fall überwiegt. Man kann ja auch über die Option LOGNAME den Namen des Logfiles auf z.B. T:ShowDVI.log setzen. So ist das Logfile immer vorhanden, stört aber nicht. Normalerweise wird das Logfile im aktuellen Directory unter dem Namen ShowDVI.log angelegt.

Welche Informationen enthält dieses Logfile?

Zum einen Warnungen wie:

- nicht gefundenes Konfigurations File,
- nicht gefundene vordefinierte Font/Fontlibraries
- und fehlerhafte DVI-Files.

Informationen wie:

- welche Fonts geladen wurden,
- wie stark der Font-Speicher ausgelastet wurde,
- welches DVI-File geladen wurde
- und – mit gesetzter STAT Option – eine allgemeine Statistik über das DVI-File und im Besonderen über das Fontdefinitions-File. Dort wird dann zum Beispiel auch angegeben, wo die einzelnen Fonts gefunden wurden.

Und schließlich Fehlermeldungen, die das ShowDVI-Programm dazu veranlaßt haben, die Arbeit einzustellen.

## 10. Benötigte Soft- Hardware

An Software benötigt man zuerst einmal ein Programm, das DVI-Files erzeugt. Im allgemeinen heißt dieses Programm `virtex` oder einfach `tex`. Von sehr großem Nutzen ist auch noch die “`req.library`”. Eine große Arbeitserleichterung ist `ARexx` von William S. Hawes.

Dann wird **unbedingt** die Library “`mathieeedoubbas.library`” benötigt! Der Startup-Code von SAS-C, der diese Library öffnet bringt leider keine Fehlermeldung, falls die Library nicht geöffnet werden konnte. Das Programm läßt sich einfach nur nicht starten. :-)

An Hardware benötigt man vor allem einen Amiga. Die wichtigste Erweiterung ist dann noch eine Speichererweiterung. 1 MB an Hauptspeicher ist das absolute Minimum. Eine Festplatte ist, wenn man vernünftig arbeiten möchte, notwendig. Für das ShowDVI-Programm bracht man nicht mehr als etwa 3 MByte an Plattenplatz. Falls man allerdings noch die Zeichensätze für seinen Druckertreiber vollständig auf der Festplatte installieren möchte, so wird schon etwas mehr gebraucht.

## 11. Bekannte Fehler

Zur Zeit sind mir keine Fehler bekannt...

*(Der Rest der Seite bleibt frei für eigene Fehler Notizen)*