



## *Video-Tutorials mit Wink*

*Videos statt Screenshots*

## *Moblin 2 – Beta Preview*

*Die neue Distribution für Netbooks im Test*



## *Gimp tunen*

*Tipps für mehr Effekte und bequemes Arbeiten*

## Vorwort

In der letzten Ausgabe standen an dieser Stelle leicht pessimistische Zeilen – die in unserer Leserschaft eine rege Aktivität verursachten: Die Redaktion erhielt so viel Post wie schon lange nicht mehr. Größtenteils waren es aufmunternde Worte, viel Lob, aber auch Kritik. Wir haben fast jedem, der uns mit einer Mail bedachte, geantwortet – möglicherweise ist aber dennoch der eine oder andere Absender ohne Rückantwort geblieben. Auf jeden Fall haben wir uns über jede einzelne Zuschrift sehr gefreut, und natürlich setzen wir alles daran, auch weiterhin pünktlich und mit interessanten Inhalten zu erscheinen.

Natürlich haben wir darüber nachgedacht, wie wir es unseren Lesern leichter machen können, uns ihre Meinung mitzuteilen. Daher haben wir zunächst einige Gastkonten für unser Forum eingerichtet; mit den Nicknamen »Leser01« bis »Leser05« und dem Passwort »yalmleser« kann jetzt jeder einen Kommentar hinterlassen, ohne sich vorher anmelden zu müssen. Alternativ kann auch **Bugmenot** genutzt werden; dort haben wir ebenfalls einen Account hinterlegt. Eine (Nick-) Namensnennung wäre natürlich schön, um Verwechslungen zu vermeiden.

Außerdem haben wir über Twitter nachgedacht, uns aber schließlich doch für die freie Variante entschieden: Yalm ist seit kurzem bei **identi.ca** aktiv und hat dort auch eine gleichnamige **Gruppe**, in der auch einige Redaktionsmitglieder aktiv sind.

Was leider in dieser Ausgabe nicht geklappt hat, ist der Abdruck der Leserbriefe. Die Ursache hierfür ist offenbar wieder einmal der beengte Zeitraum unserer Mitarbeiter, die ja alle auch nebenbei noch ein reales Leben führen und Schule, Beruf und Familie nicht allzusehr vernachlässigen dürfen. Falls also jemand die Berufung in sich verspürt, am Projekt Yalm mitzuarbeiten: Es sind noch genügend Plätze frei. Bei dieser Gelegenheit freut sich die Redaktion, auch ganz offiziell Daniel »Fauchi95« Vigano und Maximilian »faucon« Trescher als Korrektoren (und vielleicht auch mal als Autoren) im Team willkommen zu heißen.

Genug der Vorrede – und viel Spaß mit dem Magazin, in dem Euch wieder ein kleiner, aber bunter Mix aus verschiedenen Themen erwartet. Und wer weiß, vielleicht ist auch diesmal etwas dabei, das Euch besonders gefällt. In der letzten Ausgabe war das offenbar der BitTorrent-Artikel; wir sind gespannt, welcher es diesmal ist.

Nebenbei: ein Artikel aus Yalm hat es kürzlich sogar in die PCWelt Linux (Ausgabe 04/2009) geschafft: Peter Majmeskus »**World of Goo**« aus unserer Februar-Ausgabe. Herzlichen Glückwunsch, Peter! Bei so vielen Motivationsschüben freue ich mich schon auf die nächste Ausgabe.

*Frank Brungräber*  
*calexu@yalmagazine.org*

## Inhalt

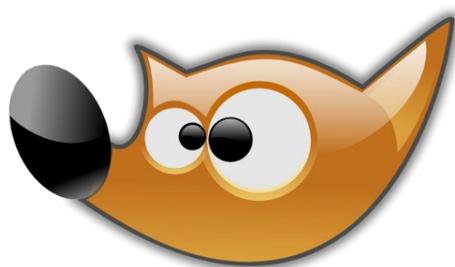
<b>Yalm - Vorwort</b> .....	<b>2</b>
Vorwort.....	2
<b>Yalm - Magazin</b> .....	<b>3</b>
GIMP in neun Schritten erweitern.....	3
Scribus (I): Ein Titelblatt erstellen.....	5
Office Shoot Out – Teil I.....	9
Websites mit dem Internet Explorer 6, 7 und 8 unter Linux testen.....	17
Moblin 2 Beta Preview.....	22
Video-Tutorials mit Wink.....	27
<b>Yalm - Schlussbemerkungen</b> .....	<b>31</b>
Schlussbemerkungen.....	31

# GIMP in neun Schritten erweitern

Jeder erfahrene Linux-Anwender kennt GIMP. Die einen nennen es ein professionelles Bildbearbeitungsprogramm, die anderen nicht. Im folgenden Teil wird gezeigt, wie man GIMP in neun Schritten gehörig erweitert. Alle im folgenden erwähnten GIMP-Ordner befinden sich versteckt im Homeordner unter `.gimp2.6`. Diesen blendet man im Nautilus-Dateimanager mit [STRG]+[H] ein.

## Schritt Nr. 1: Photoshop-PlugIns und -Pinsel mit GIMP nutzen

Das *PSPI-PlugIn* [1] erlaubt das Nutzen von Photoshop PlugIns unter GIMP. Damit kann man weiterhin Photoshop-PlugIns vom Datenträger auswählen und sie unter GIMP installieren. Leider liegt die fertig kompilierte Version nur für alte Ubuntu-Distributionen vor. Möglichkeiten sind hierzu Wine oder eine VM mit Windows. Das Tool *ABR2GBR* [2] übersetzt Photoshop-Pinsel ins GIMP-Format.



*Erweiterungen heben den Grad an Professionalität*

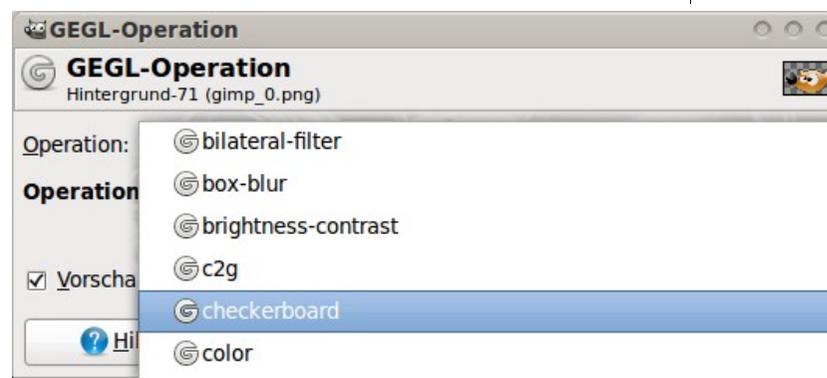
## Schritt Nr. 2: Ebenen-Styles

Möchte man Ebenen zum Beispiel mit Schatten, Farbüberlagerungen und innerem Glühen aufwerten, so hole man sich das *Layer Effects PlugIn* [3].

Es gibt davon zwei Versionen: Script-Fu und Python. Die Python-Version erlaubt eine Vorschau der Effekte. Die Installation und Auffindung in der Programmoberfläche verläuft wie vorangegangen.

## Schritt Nr. 3: CMYK Farben

Das CMYK-Farbmodell [4] steht für Cyan, Magenta, Yellow und Key (= Schwarz). Dieses ist besonders für den qualitativ hochwertigen Druck interessant. Das Fehlen dieser Farbtrennung wird von vielen bemängelt. Diese lässt sich jedoch mit dem PlugIn *Separate+* [5] nachrüsten. Unter Ubuntu werden zur Installation nur zwei Befehle benötigt. »apt-get install gimp-seperate+« und für die ICC-Profiles »apt-get install icc-profiles«. Auf Allen an-



*GEGL birgt eine Vielzahl an Features*

deren Systemen bitte immer darauf achten, dass *Seperate+* sowie die ICC-Profiles installiert sind.

## Schritt Nr. 4: Liquid Rescale

Damit [6] lassen sich Bilder in verschiedene Dimensionen wie 16:9 oder 4:3 skalieren, ohne dass diese verzerrt wirken.

## Schritt Nr. 5: Pinsel nachladen

Die Standardpinsel von GIMP sind auf Dauer langweilig. *Blendfu.com* [7] bietet mehr Pinsel-Sets als man jemals benötigen wird. Dazu kopiert man das entpackte Pinselverzeichnis nach `.gimp-2.6/brushes` im Home-Ordner. Doch Vorsicht: viele zusätzliche Pinsel verlängern den Programmstart deutlich.

## Schritt Nr. 6: Fenster verankern

Ist man es überdrüssig, die drei GIMP-Fenster immer wieder zurechtzurücken, klickt man einfach mit der rechten Maustaste auf die Fenster-Titelleiste und wählt »Immer im Vordergrund«. Danach ist die Werkzeugleiste selbst dann sichtbar, wenn man ein anderes Programm öffnet. Das Kontextmenü eröffnet noch weitere Einstellungsmöglichkeiten, die das Dreierlei der GIMP-Fenster interessant machen.

## Schritt Nr. 7: Animations-Paket

Das *GIMP Animation Package* (GAP) erlaubt die Erstellung von Animationen. Diese lassen sich gut für animierte Avatare im GIF-Format einsetzen. Bei den meisten Linux-Distributionen ist es bereits innerhalb des Paketmanagers unter der Be-

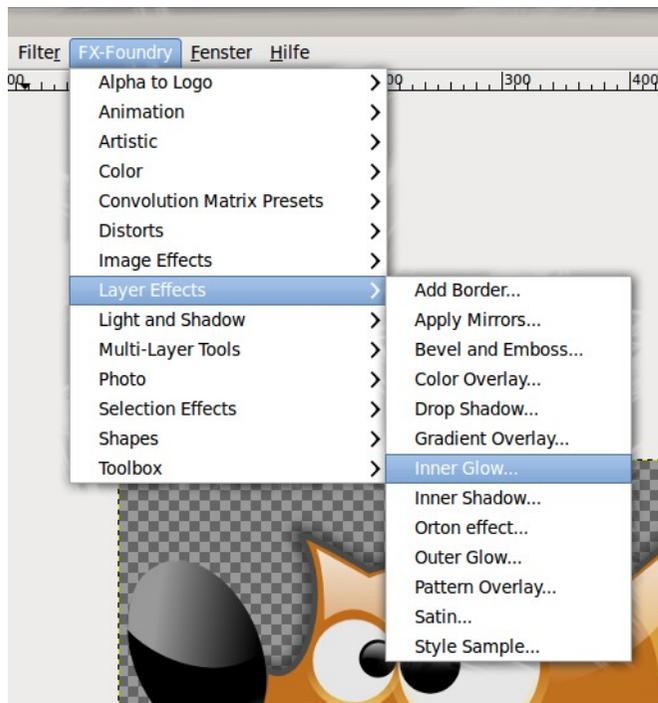
zeichnung *gimp-gap* zu finden und birgt eine Vielzahl an vorgefertigten Effekten.

### Schritt Nr. 8: GEGL

Eine der größten Neuerungen in GIMP 2.6 ist GEGL. Damit ist die Anpassungen von Ebenen-Effekten und CMYK- sowie Bit-Tiefen vereinfacht möglich. GEGL ist standardmäßig nicht eingeschaltet; man aktiviert es mittels *Farben – GEGL verwenden*. Unter *Werkzeuge* gibt es eine Auswahl an GEGL-Operationen.

### Schritt Nr. 9: 117 Effekte

Mit dem Sourceforge-Projekt *GIMP FX Foundry* [8] wird eine Vielzahl an Effekten angeboten. Zu den



Die zahlreichen Effekte laden zum Klicken ein

Skripten gehören etwa das »Roy Lichtenstein Effect Script«, das Bilder im Stil des Pop-Art-Künstlers aufbereitet, das »Planet Render Script«, das einen Planeten generiert, dessen Größe und Lichteinstrahlung der Anwender selbst bestimmt. Den Inhalt des Pakets kopiert man nach *.gimp-2.6/scripts*. Nach dem Neustart von GIMP findet man in der Menüleiste einen neuen Reiter mit dem Namen *FX-Foundry*.

### Fazit

Nachdem GIMP in den genannten neun Schritten erweitert wurde, liegt ein professionelles und sehr leistungsfähiges Bildbearbeitungsprogramm vor, mit welchem man mit wenigem Klicken und Testen ein ansprechendes Bild erstellen kann. Möchte man noch mehr Extras für GIMP haben, so sollte man die Website *Gimpstuff.org* [9] ansurfen. Weiterhin gibt es auf der offiziellen GIMP-Website [10] eine Vielzahl an Tutorials. Damit man jedoch die vollen Möglichkeiten von GIMP nutzen kann, sollte man auf Dauer damit arbeiten. »Learning-by-Doing« heißt hier die Devise. Besonders möchten wir dem Leser das Buch »GIMP 2.6 für digitale Fotografie« von Jürgen Wolf [11] ans Herz legen, das im Galileo-Verlag erschienen ist und in leicht zu lesender Weise, anhand praktischer Beispiele, die fantastischen Einsatzmöglichkeit dieser freien Software nahe bringt. Auf der Verlagsseite gibt es eine Leseprobe. Zudem liegt dem Buch eine herausnehmbare und übersichtliche Nachschlagevorlage bei, mit der man alle Tastenkürzel stets parat hat. Auch Nutzer von Mac-Rechnern, bei denen die vermeintliche Traum-Kombination »Mac plus Photo-

shop« doch nicht so gut läuft – Mike Essl zeigt in seinem Blog [12] einige Beispiele – finden mit einem wie beschrieben erweiterten Gimp möglicherweise eine gute Alternative zu proprietärer Software.

Peter Majmesku  
pe@yalmagazine.org

### Informationen

- [1] Quelle für das PSPI-Plugin: <http://www.gimp.org>
- [2] Quelle von ABR2GBR (Photoshop-Pinsel): <http://the.sunnyspot.org/gimp/tools.html>
- [3] Layer Effects Plugin Quelle: <http://www.gimp.org>
- [4] Wikipedia-Artikel über das CMYK-Farbmodell: <https://secure.wikimedia.org/wikipedia/de/wiki/CMYK>
- [5] CMYK-Plugin: <http://cue.yellowmagic.info/software/separate.html>
- [6] Liquid Rescale: <http://cue.yellowmagic.info/software/separate.html>
- [7] Pinsel von Blendfu.com: <http://www.blendfu.com>
- [8] Sourceforge-Projekt GIMP FX Foundry: <http://sourceforge.net/projects/gimpfx-foundry/>
- [9] Bei Gimpstuff.org gibt es haufenweise GIMP-Extras: <http://www.gimpstuff.org>
- [10] GIMP-Tutorials auf Gimp.org: <http://www.gimp.org/tutorials/>
- [11] Verlagsseite zum Buch GIMP 2.6 von Jürgen Wolf mit Leseprobe: <http://www.galileodesign.de/katalog/buecher/titel/gp/titelID-1777>
- [12] Mike Essls Blog über Photoshop-Abstürze unter Mac: <http://unexpectedlyquit.com/>

# Scribus (I): Ein Titelblatt erstellen

Das DTP-Programm (Desktop Publishing) Scribus eignet sich perfekt dazu, Broschüren, Flyer, Kataloge, Essays und Ähnliches zu layouten. In dieser Serie erklären wir Schritt für Schritt die Erstellung des Magazins »Die Invasion der Gummitierchen«. Der erste Teil vermittelt grundlegende Scribus-Kenntnisse und zeigt, wie eine Titelseite erstellt wird.

Davon ausgehend, dass alle Texte und benötigten Bilder bereits vorliegen, verwenden wir Scribus »nur« zum layouten. Ein DTP würde sich auch nur sehr bedingt zum Erstellen und Bearbeiten von Texten eignen.

Scribus findet sich in den Paketverwaltungen der meisten Distributionen, meistens allerdings noch in der Version 1.3.3.13. Die aktuelle Entwicklerversion 1.3.5, welche derzeit im RC2 vorliegt und also final werden wird, erwies sich als genauso stabil, etwas langsamer und viel funktionaler als ihr Vorgänger; für diese Serie verwenden wir daher Scribus 1.3.5.

Für Debian, Ubuntu und deren Derivate gibt es dafür ein eigenes Repository; die Einrichtung ist auf in der entsprechenden Rubrik [1] auf der offiziellen Homepage beschrieben. Das Paket heißt *scribus-ng*. Aber auch Nutzer anderer Distributionen werden dort fündig. [2]

## Stabilität

Vor gut einem Jahr nutzten wir Scribus, um unser Magazin zu layouten. Dies war allerdings fast unmöglich: Das Programm hatte sein eigenes Verständnis von einer »Rückgängig-Funktion«, die uns öfter zu einem Neustart zwang, Texte wurden

scheinbar willkürlich verschoben, Scribus stürzte gelegentlich ab, ohne dass man danach das Dokument in einem brauchbaren Zustand wiederherstellen konnte, usw.

In der Version 1.3.5 schien sich daran zunächst nicht viel geändert zu haben: Nutzer in Scribus' IRC-Channel [3] verrieten uns aber, dass diese Fehler nicht an Scribus, sondern an dem Zusammenspiel zwischen Scribus und Ubuntu lägen. Ubuntu und Qt seien zwei verschiedene Welten, hieß es.

Für Ubuntu-Nutzer eignet sich Scribus zwar ebenfalls zum Layouten, allerdings frustrieren nervige Bugs immer wieder. Sofern man das Prinzip »save early – save often« beherzigt, kann man damit



Der Splashscreen von Scribus 1.3.5

aber noch leben. Wir testeten Scribus unter Ubuntu 8.10 und openSuse 11.1.

## Der erste Start

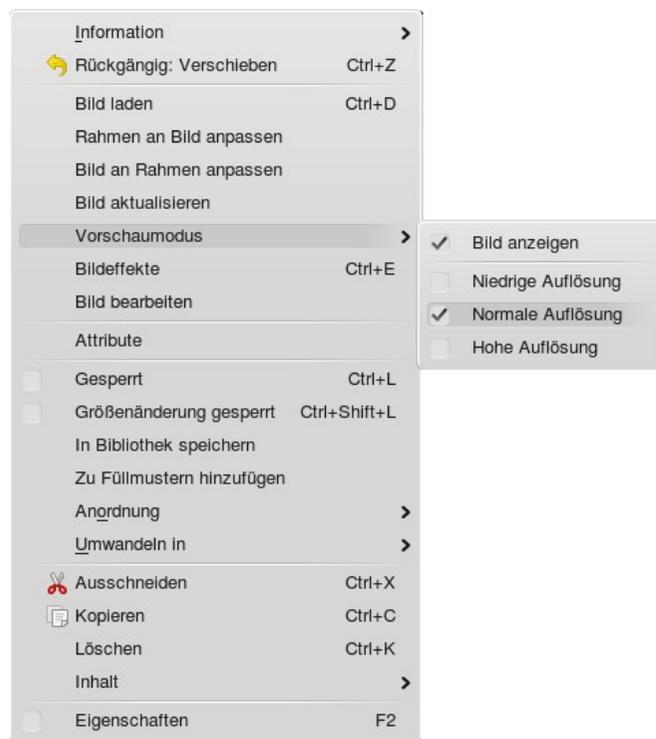
Bei jedem Start von Scribus öffnet sich der Dialog *Neues Dokument*. Hier ändern wir nur die Ausrichtung von »Hochformat« auf »Querformat«. Alle anderen Werte lassen wir bei den Voreinstellungen, auch die Seitenanzahl, da wir so später noch weitere Seiten hinzufügen können.

Das Scribus-Fenster besteht aus drei Teilen: Oben befindet sich die klassische Menüleiste, darunter folgt eine Symbolleiste, die die wichtigsten Funktionen darstellt. In der Mitte ist das eigentliche Dokument zu sehen, am Rand zwei Lineale und zwei Scrollbalken. Ganz unten gibt es noch einige Möglichkeiten, die Ansicht zu ändern.

Die erste Seite sollte schlicht sein, aber dennoch gut aussehen. Zudem sollte sie, zumindest für den Einstieg, leicht zu layouten sein. Daher entscheiden wir uns für eine simple Aufteilung in Kopfzeile, Topthemen und Fußzeile. Die Kopf- und Fußzeilen sollen rote Rechtecke mit gelber Schrift haben und sich durch einen Schatten von den Topthemen abgrenzen. Jedes der drei Topthemen wird mit einem Bild und dem dazugehörigen Lead (einer kurzen Einleitung) dargestellt.

## Kopfzeile

Soweit zur Theorie. Um das Ganze in die Praxis umzusetzen, wählt man in der Symbolleiste »Formen einfügen« (ein Quadrat) und zieht damit ein Rechteck in den oberen Ecken auf. Das Rechteck wird schwarz und ist markiert. Danach wählt



Übers Rechtsklickmenü gelangt man zu den Einstellungen

man [F2] (oder Rechtsklick – *Eigenschaften*) für den Eigenschaften-Dialog. Hier sollte zunächst ein sinnvoller Name für das Objekt vergeben werden, z. B. »Titel\_Hintergrund«.

Da sich unter dem Reiter *Farben* keine für uns geeignete findet, wird eine neue erstellt: Unter *Bearbeiten – Farben – Neu* gelangt man zum entsprechenden Menü. Wir wählen ein recht dunkles Rot und taufen es »darkred«. Danach wird wieder der Einstellungsdialog aufgerufen; unter »Farben« findet sich nun ganz unten die neue Farbe.

Nun können der Titel und die Ausgabennummer eingefügt werden: Hierzu wird ein Textrahmen aufgezogen. Das Symbol dafür findet sich natürlich wieder in der Symbolleiste. Um den Titel zentrieren zu können, empfiehlt es sich, den Textrahmen vom linken zum rechten Rand zu ziehen.

Via [Strg]+[y] (oder Rechtsklick – *Text bearbeiten*) lässt sich der »Story-Editor« öffnen. Nach dem Eintippen des Titels »Invasion der Gummitierchen« wird dieser markiert. Nun lassen sich Schrift, Schriftfarbe, Ausrichtung usw. anpassen. Leider kann man hier nicht – wie man es von Office-Programmen gewohnt ist – kursive oder fette Schrift direkt auswählen. Dazu muss man einen anderen Schriftstil der gleichen Familie auswählen. Eigentlich nicht weiter schlimm, allerdings wird das DropDown-Menü dadurch sehr unübersichtlich, was vor allem deshalb schade ist, weil man so die Schriftarten nicht suchen kann. Man kann lediglich den Anfangsbuchstaben eintippen.

Wir entscheiden uns für die Schriftart *Tribute2Ubuntu*, welche *Ubuntu-Title* um Großbuchstaben erweitert und über unsere Homepage [4] bezogen werden kann. Passend zum roten Hintergrund wählen wir eine gelbe Farbe (»Yellow«), zentrieren den Text und schließen das Menü.

Über den Eigenschaften-Dialog lässt sich unter »Text« nun die Größe einstellen. Wir wählen 36pt

und geben dem Textrahmen unter »X, Y, Z« einen passenden Namen, falls dies noch nicht geschehen ist. Jetzt ist das Einfügen einer Hilfslinie empfehlenswert. Wie von Grafik-Programmen (z. B. GIMP) gewohnt, zieht man dazu einfach eine Linie von dem oberen Lineal herunter. Ein Problem dabei ist, dass Hilfslinien standardmäßig hinter Polygonen verschwinden, also in unserem Fall hinter unserem Rechteck. Dieses Verhalten lässt sich zum Glück ändern: *Datei – Allgemeine Einstellungen – Hilfslinien – Platzierung im Dokument – Im Vordergrund anpassen*. Falls ein Dokument geöffnet ist, wird diese Einstellung auch nur für dieses Dokument gespeichert, andernfalls wird sie zur künftigen Voreinstellung. Dies gilt übrigens für alle Einstellungen unter *Allgemeine Einstellungen*.

Nun lässt sich entweder ein neuer Textrahmen anlegen und obiger Schritt wiederholen, oder man kopiert den Titel-Textrahmen, verschiebt ihn an der Hilfslinie und ändert den Text in »# 4« (die Ausgabennummer) um. Alle anderen Einstellungen können wir beibehalten.

Um das Rechteck etwas schöner wirken zu lassen, wäre ein Schatten von Vorteil. Scribus bietet leider keine Möglichkeit, ein Polygon mit Schatten zu versehen; also müssen wir uns selber einen erstellen. Das geht z. B. im GIMP mit einem Farbverlauf von Schwarz nach Weiß bzw. von Schwarz nach

## Invasion der Gummitierchen

## #4

Die Kopfzeile...

Transparenz. Die Helligkeit spielt hierbei keine Rolle, das kann in Scribus selbst angepasst werden.

Bilder lassen sich leider nicht über Copy & Paste einfügen: Man muss dazu in der Symbolleiste das Symbol »Bilderrahmen einfügen« auswählen. Wir ziehen damit ein recht schmales Rechteck unter das Polygon. Über *Rechtsklick – Bild laden* oder [Strg]+[d] lässt sich ein Bild – in unserem Fall der Schatten – laden. Im Eigenschaften-Dialog wird nun wieder ein sinnvoller Name vergeben und das Bild über *Bild – An Rahmen anpassen* an die Größe angepasst. Der Haken bei *Proportional* wird natürlich entfernt.

Gegebenenfalls müssen nun noch die Größe und die Helligkeit angepasst werden. Letzteres stellt man im Eigenschaften-Dialog unter *Bild – Bild-effekte* oder indem man mit der rechten Maustaste auf das Bild klickt und »Bildeffekte« ([Strg]+[E]) wählt. In dem geöffneten Fenster wählen wir »Helligkeit« und setzen den Wert 55. Fertig ist die Kopfzeile.

### Fußzeile

Die Fußzeile soll optisch zu der Kopfzeile passen. Daher werden das Polygon, einer der Textrahmen und der Schatten an den unteren Rand kopiert. Die drei neuen Objekten erhalten wie gewohnt sinnvolle Namen. Das Polygon und der Schatten werden dabei etwas verkleinert. Letzterer muss auch noch um 180° gedreht werden: Dazu wählt man unter *Eigenschaften – X, Y, Z* »Drehung« aus und setzt den Wert entsprechend. Scribus dreht das Bild allerdings standardmäßig nicht um dessen Mittelpunkt, sondern an der linken, oberen

Ecke. Dies lässt sich unter dem Punkt »Ursprung« anpassen, hier wird einfach die Mitte gewählt.

Der Textrahmen wird wieder an den linken und rechten Rand vergrößert. Im Story-Editor (*Rechtsklick – Text bearbeiten* oder [Strg]+[y]) erstellen wir den Text »Wir sind frei: Creative Commons Namensnennung-Weitergabe unter gleichen Bedingungen 3.0 Deutschland«, zentrieren diesen, wählen die Schriftart »FreeSerif Italic« und schließen das Fenster wieder. Schon ist die Fußzeile vollendet.

### Topthemen

Nun kümmern wir uns um das eigentliche Titelblatt, welches ja die Topthemen mit einem Bild und dem Leadtext zeigen sollte. Dafür werden zunächst drei Bilderrahmen erstellt, die Bilder geladen (*Rechtsklick – Bild laden laden* oder [Strg]+[d]) und sinnvolle Namen vergeben. Wir wählen dazu drei Gummibärchenbilder von Deviantart aus (die Quellenangaben befinden sich am Ende des Artikels).

Nach dem Einfügen der Bilder müssen diese noch bearbeitet werden: Zuerst ändern wir die Größe z. B. auf 200×135 Pixel und rufen Gimp über *Rechtsklick – Bild bearbeiten* auf. Leider kann man dies nur machen, wenn Gimp noch nicht geöffnet ist. Für das zweite und dritte Bild ist es daher sinnvoller den Öffnen-Dialog von Gimp zu benutzen. Dort wählen wir *Filter – Dekoration – Runde Ecken*, aktivieren »Schlagschatten« und erhöhen

den Radius auf 30 Pixel. Nun werden die Bilder wieder gespeichert und via Rechtsklick – *Bild aktualisieren* erneut in Scribus geladen. Danach muss man nur noch – wie beim Schatten – das Bild an den Rahmen anpassen, und die Bilder sind fertig.

Wir entscheiden uns, die drei Topthemen untereinander anzuordnen. Beim oberen und unteren Thema soll das Bild links, bei dem mittleren rechts sein. Dementsprechend setzen wir die Hilfslinien etwa 30 Pixel von den Rändern entfernt.

Nun können die Textrahmen erstellt werden. Diese sollten etwas niedriger als die Bilder und so lang sein, dass sie mit den anderen Bildern bzw. dem anderen Bild auf gleicher Höhe sind (Hilfslinien). Im Story-Editor kopieren wir nun die mehr oder weniger sinnvollen Texte und passen die Formatierung an: Für die Überschriften wählen wir die Schrift »Liberation Sans Bold Italic« und für die Leads »Liberation Sans Italic«; beide sollen die Schriftgröße 17 erhalten. Hier sollte man etwas herumprobieren: Textrahmen verschieben, Schriftgröße anpassen usw.

Nach dem Erstellen des ersten Textrahmens wird dieser kopiert, so bleiben die Größe und alle Formatierungen erhalten. Nun muss nur noch der Text eingefügt und gehofft werden, dass dieser nicht zu lang ist. Gegebenenfalls muss man weiterhin etwas hin und herschieben.

*Wir sind frei: Creative Commons Namensnennung-Weitergabe unter gleichen Bedingungen 3.0*

... und die Fußzeile sehen recht ähnlich aus

## Invasion der Gummitierchen

#4



### »3 Tage wach« – Ein Blick hinter die Kulissen

Berlins freie Gummibären-Party war auch dieses Jahr wieder ein großer Erfolg. Allerdings sollte man sich unter dieser »Party« keine Sauf-Veranstaltung vorstellen, vielmehr ist es ein Treffen der »Respect white bears«-Organisation, die sich – wie der Name vermuten lässt – für die Rechte weißer Bären einsetzt. Neben Vorträgen, Meetings und Diskussionsrunden wurde am Abend ordentlich auf die Pauke gehauen.

### Die magische Zahl 3

Bei vielen Gottheiten und Religionen gilt die Dreiheit der Götter, so glauben Christen an die Dreifaltigkeit, die Musik wäre ohne Drei-Klänge nur halb so schön, es gibt drei Grundfarben, ein Dreieck ist die einfachste geometrische Form, die Deutsche Flagge hat drei Farben und so weiter und so fort. Nun entdecken auch die Bären die Magie dieser Zahl und probieren neue Rituale aus.



### Mobbing-Probleme

Trotz Partys und Organisationen gegen Diskriminierung werden immer wieder weiße Gummibärchen ngeärgert, gehänselt, ausgelacht, bloßgestellt, bestohlen, getreten, verachtet, ausgegrenzt, misshandelt, ausgenutzt, bespuckt, falsch verdächtigt, ignoriert... Wir zeigen wie es zu dem vermeintlichen Unterschied der Bärchen kam und präsentieren daraus resultierende Probleme sowie mögliche Lösungen.

*Wir sind frei: Creative Commons Namensnennung-Weitergabe unter gleichen Bedingungen 3.0*

Das Titelblatt ist nun fertig

Das eigentliche Titelblatt ist nun fertig. Nun kann man noch hier und da Rahmen vergrößern, verkleinern oder verschieben, andere Schriftarten ausprobieren usw. Anschließend kann man das Titelbild noch über das PDF-Symbol nach PDF exportieren. Warnungen kann man hierbei meistens ignorieren.

### Ausblick

Für ein echtes Magazin fehlt natürlich noch etwas ganz Wesentliches: der Inhalt. So kümmern wir uns im nächsten Teil dieser Serie hauptsächlich um die Erstellung der Artikel und eines Inhaltsverzeichnis sowie um Links, Absatz-Stile und die Automatisierung.

Wer vom »Scribus-Fieber« gepackt wurde und nicht bis zu Teil II warten möchte, dem seien der IRC-Channel und die englischsprachige Dokumentation [5] empfohlen. Bekanntlich lernt sich das Meiste eben doch durch Ausprobieren.

Mario Fuest

[Keba@yalmagazine.org](mailto:Keba@yalmagazine.org)

### Quellen der Bilder

- Oberes Bild: **Gummibärchen** von naliha  
<http://www.piqs.de/fotos/27048.html> (Lizenz: CC-BY, Download erfordert kostenlose Registrierung)
- Mittleres Bild: **Haribo** von =bulldogg-designs:  
<http://bulldogg-designs.deviantart.com/art/Haribo-48691396> (unfrei, wir haben allerdings via Email die Berechtigung der Veröffentlichung erhalten)
- Unteres Bild: **einsam** von westpark:  
<http://www.flickr.com/photos/west-park/1408791642/> (Lizenz: CC-BY-NC-ND)

### Informationen

- [1] Download-Rubrik auf der Scribus-Homepage <http://www.scribus.net/?q=downloads>
- [2] Download des Debian-Paketes:  
<http://www.scribus.net/?q=debian>
- [3] IRC-Channel: irc.freenode.net #scribus
- [4] Direkt-Download von Tribute2Ubuntu:  
<http://yalmagazine.org/homepage/downloads/extras/tribute2ubuntu.zip>
- [5] offizielle Dokumentation: <http://docs.scribus.net/>

Download der Quelldateien: <http://yalmagazine.org/homepage/downloads/Extras/Scribus.tar.gz>

# Office Shoot Out – Teil I

Textverarbeitungen sind die Klassiker unter den Standardanwendungen. Ob für kurze Texte, Briefe, Berichte oder Diplomarbeiten – ihr Einsatzbereich ist breit und sie werden fast täglich genutzt. Unter Linux ist OpenOffice Writer der de-facto-Standard für die Bearbeitung von Texten. Als Herausforderer kommen mehrere Kandidaten in Frage. Da wären zum Beispiel AbiWord (Gnome), KWord (KDE) und LaTeX als die etwas andere Textverarbeitung. Aus rein willkürlichen Gründen treten im ersten Teil des Office Shoot Outs OpenOffice Writer und AbiWord gegeneinander an.

OpenOffice Writer wird in der Version 3.0.1 mit AbiWord in Version 2.6.6 verglichen. Apropos Vergleich – vorweg soll eines klargestellt werden: der Artikel erhebt keinerlei Anspruch auf Fairness, Ausgeglichenheit oder objektive Beurteilung der beiden Applikationen. Stattdessen versucht der Autor die beiden Kandidaten möglichst praxisnah zu testen. Abgehobene Features, die vielleicht einmal im Jahr verwendet werden, zählen wenig, wohingegen die Bedienungsfreundlichkeit einen sehr hohen Stellenwert hat. Als Testumgebung dient ein leistungsstarkes Notebook unter Ubuntu 9.04.

## Der erste Eindruck

Da OpenOffice bei Ubuntu bereits vorinstalliert ist, musste nur noch AbiWord mittels »Anwendungen – Hinzufügen/Entfernen« bereitgestellt werden. Die erste Möglichkeit für die beiden Textverarbeitungen zu punkten ergibt sich beim Start. Der Leser mag es schon ahnen; da bei OpenOffice immer die ganze Office Suite geladen wird, dauert der Start mit 10 Sekunden erheblich länger als beim Konkurrenten. AbiWord schafft den Start in schnellen 3 Sekunden und belegt 9 MB im Spei-

cher; OpenOffice genehmigt sich 20 MB. Zwischen 3 und 10 Sekunden liegen, im Empfinden des Anwenders, Welten. Deshalb geht der erste Punkt an AbiWord (OpenOffice Writer 0:1 AbiWord).

Auf den ersten zeitlichen Eindruck folgt der erste visuelle Eindruck. Writer kommt mit zwei prall ge-

füllten Werkzeugleisten daher, die keinen Platz verschwenden (kleine Icons, keine Leerräume). AbiWord ist ganz eindeutig eine Gnome-Anwendung; die Größe und Art der Icons lassen keine Zweifel aufkommen. (Alle folgenden Abbildungen zeigen zuerst OpenOffice Writer und dann AbiWord.)

In der oberen Werkzeugleiste von AbiWord ist rechts noch viel Platz vorhanden der genutzt werden könnte. Bei genauerem Betrachten fällt ein Unterschied in der Tabulatorsteuerung auf. Die Tabulatormarken im Lineal von AbiWord lassen sich unabhängiger setzen als bei Writer. Die Tabulatormarken in Writer rasten nicht an den Markierungsstrichen ein. Die untere kann nur mit gedrückter STRG-Taste unabhängig von der oberen bewegt

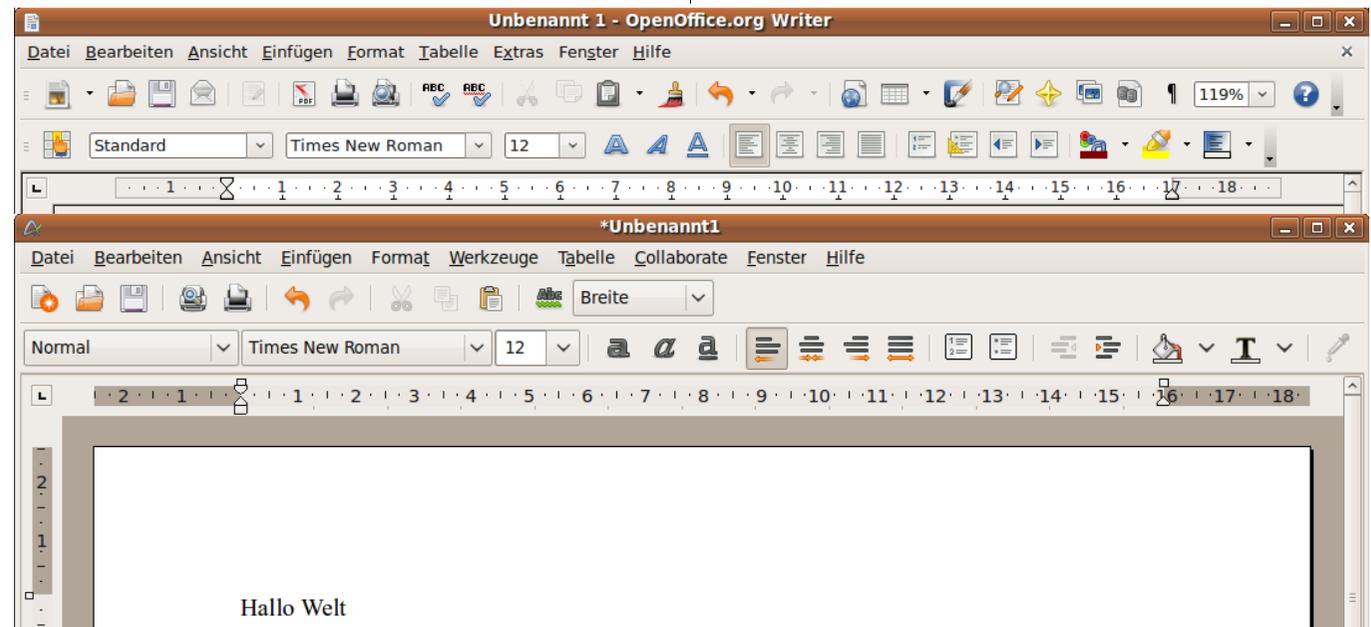


Abb. 1: Der erste Eindruck

werden. Die untere bewegt beide Tabulatormarken wohingegen die obere eigenständig bewegt werden kann. Das erscheint weniger intuitiv als in AbiWord. Dort rasten die Tabulatormarken ein und können jeweils unabhängig voneinander bewegt werden. Möchte man sie zusammen bewegen, benutzt man den rechteckigen Anfasser unterhalb der Tabulatormarken. Das ist viel einfacher; man versteht es sofort.

Die visuelle Erscheinung der beiden Programme ist sehr ähnlich. Es gibt leichte Vorteile für AbiWord; deshalb 1 Punkt für Writer und 2 Punkte für AbiWord, das macht (OpenOffice Writer 1:3 AbiWord).

### Grundeinstellung

Wie in Abbildung 1 zu erkennen ist, respektieren beide Textverarbeitungen die deutsche Sprache: die Wörter »Hallo Welt« werden als deutsche Wörter erkannt (und nicht rot unterstrichen) und die Menüs sind auch in deutsch geschrieben (außer »Collaborate« bei AbiWord).

Wenn der Autor eines nicht mag, dann ist es »Times Roman« und Schriftgröße 12. Ein neues Dokument soll als Standard »Free Sans« in Größe 11pt haben. In OpenOffice Writer lassen sich diese Einstellungen über das Menü: »Extras, Optionen, Grundschriften« vornehmen; das ist einfach und gut. In AbiWord können unter »Bearbeiten, Einstellungen« einige Angaben gemacht werden. Dort lassen sich gerade einmal 15 Einstellungen machen, die über 3 Tabs verteilt sind. Angaben zur

Standardgröße und -schriftart sucht man vergeblich.

Erst nach der Konsultation der Hilfe kommt etwas Klarheit in dieses Rätsel. Im Verzeichnis `/usr/share/abiword-2.6/templates` befindet sich eine Datei namens `normal.awt-de_DE`. Diese kopiere man in das Verzeichnis `/home/[Benutzername]/.AbiSuite/templates` und benenne sie um in `normal.awt`. Dann wird diese Datei als Vorlage für alle neuen Dokumente in AbiWord verwendet. Alternativ kann auch ein neues Dokument erstellt und unter dem Namen `normal.awt` in das oben genannten Verzeichnis gespeichert werden. Die Formatvorlagen können über das Menü: *Format, Neu / Modifizieren...* verändert werden. Dort werden die verfügbaren Stile (Formatvorlagen) aufgelistet, können umformatiert bzw. neu erstellt werden.

Als Beispiel für die Veränderung von Stilen bzw. Formatvorlagen soll der Schriftgrad der »Überschrift 1« auf 14 Punkte eingestellt werden. In Writer braucht man dazu 4 Mausklicks:

1. (Cursor befindet sich in der Überschrift) Kontextmenü öffnen
2. Absatzvorlage bearbeiten
3. Schriftgrad 14
4. OK

AbiWord benötigt für die gleiche Übung mehr als doppelt so viele Mausektionen:

1. (Cursor befindet sich irgendwo) Menü: Format, Neu / Modifizieren
2. »Heading 1« auswählen
3. Verändern

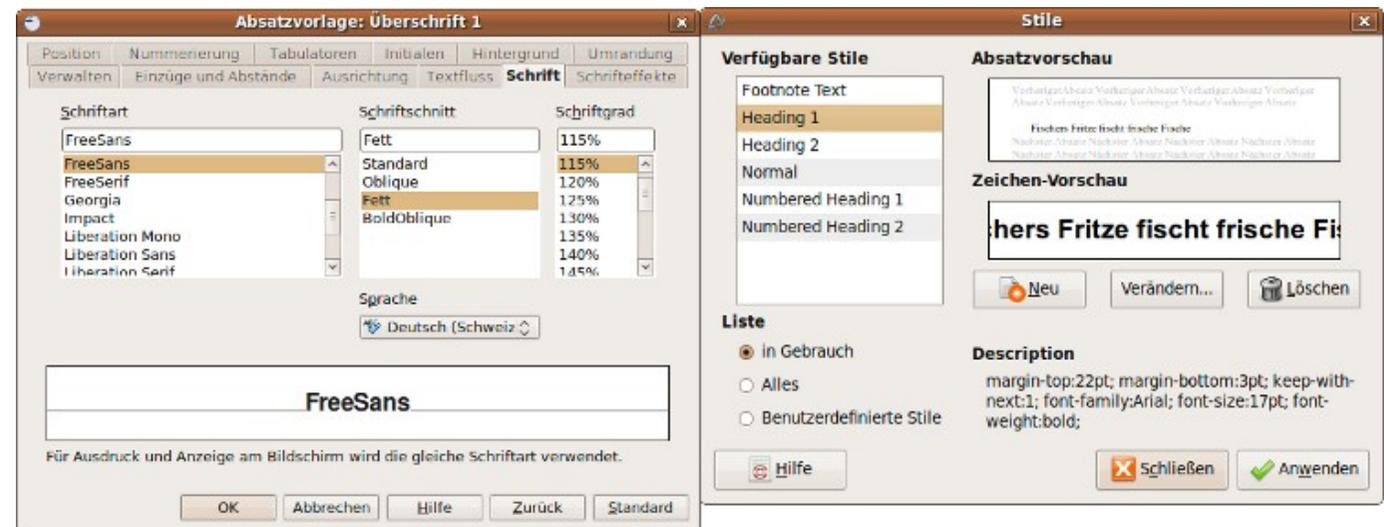


Abb. 2: Stile verändern (links: Writer, rechts: AbiWord)

4. Format
5. Schrift
6. 14
7. OK
8. OK
9. Schließen

Der Punkt für die Grundeinstellungen geht klar an Writer (OpenOffice Writer 2:3 AbiWord).

Noch ein Wort zur Hilfe. Die Unterstützung für AbiWord kann wie gewohnt mit »F1« geöffnet werden und wird als lokale Webseite im Browser angezeigt. Obwohl AbiWord auf einem System mit deutscher Spracheinstellung installiert wurde, er-

scheint die Hilfe in Englisch. Die Suchfunktion hilft auch nicht weiter da sie zwar als Link vorhanden ist jedoch nicht wirklich existiert. Da ist der Index mit ca. 100 Stichwörtern schon besser um Hilfe zu bestimmten Themen zu finden. Die Hilfe von OpenOffice Writer ist über jeden Zweifel erhaben; sie ist vorbildlich. Dadurch erkämpft sich Writer wieder den Gleichstand in der Bewertung (OpenOffice Writer 3:3 AbiWord).

#### Der Brief

Veteranen unter den Lesern können sich vielleicht noch an den »Dr.-Grauert-Brief« [1] erinnern. Er wurde in den 70er Jahren zum Testen von Druckern verwendet und fand sich später auch in Zeitungsartikeln über Textverarbeitungen als Refe-

renzbrief wieder. Da der »Grauert« völlig veraltet und zudem urheberrechtlich geschützt ist, hat der Autor keine Mühen gescheut und in Anlehnung an das Original den »Dr.-Yalm-Brief« [2] entwickelt.

Im Folgenden wird der »Yalm-Brief« Stück für Stück fortgeschrieben, um die Fähigkeiten der beiden Prüflinge weiter auszuloten. Der alte Brief wurde so umformatiert, dass die häufigsten Elemente moderner Schriftstücke vorkommen:

- Bilder
- Linien und Rahmen
- Aufzählungen
- Kopf-/Fußzeilen
- Tabellen

Los geht es mit dem Briefkopf. Darin enthalten ist ein Logo und der Absender, wobei das Logo links stehen soll, der Absender rechtsbündig daneben, abgeschlossen von einer einfachen Linie unterhalb. Das Ergebnis stellt sich so dar:

Die Abbildungen lassen die Anstrengungen beim Erstellen kaum erahnen. In OpenOffice wurde das Bild beim Einfügen auf Seitenbreite skaliert, so dass es von Hand wieder auf die Originalgröße zurück gebracht werden musste. Für die Absenderadresse konnte einfach auf rechtsbündig geschaltet und dann der Text eingegeben werden. Um die Linie unterhalb des Briefkopfes einzufügen, gibt es die Funktion »Einfügen, Horizontale Linie«. Damit

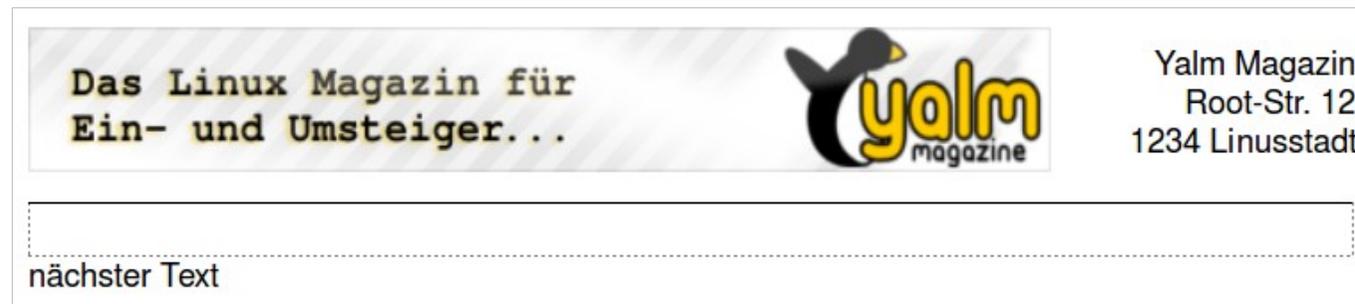


Abb. 3: Briefkopf

stehen aber nur graphische Linien zur Verfügung; einen simplen Strich gibt es nicht – oder doch? Mit einem Trick kann dennoch eine einfache Linie erzeugt werden. Dazu gibt man --- (drei Bindestriche) gefolgt von einem [RETURN] ein oder aber \_\_\_ (drei Unterstriche) plus [RETURN] um eine fette Linie zu erhalten – praktisch wenn man es weiß. Eine weitere Methode besteht darin, bei »Format, Absatz« eine passende Linienanordnung auszuwählen. Im Beispiel wurde (als weitere Alternative) eine einzellige Tabelle eingefügt und so formatiert, dass die obere Linie gezeigt wird.

AbiWord hat sich anfänglich gut gehalten. Das eingefügte Bild wird nicht auf Seitenbreite skaliert, sondern behält seine Originalgröße; das ist positiv. Doch dann wurde es immer schlimmer: der rechtsbündige Text ließ sich erst eintragen nachdem das Bild als »positioned image« formatiert wurde. Beim Schreiben der Adresse verschwand der Text ständig, so dass nur noch ab und zu etwas von ihm zu sehen war. Zum Schluss verschwand die erste Zeile der Adresse vollständig und konnte nur durch neues Skalieren des Bildes wieder hervor gebracht werden.

Die Textverarbeitung von »Gnome Office« hat auch große Mühe beim Abspeichern des Dokuments. Der Autor kämpfte minutenlang mit dem »Save« Knopf, bevor die Datei tatsächlich als gespeichert erkannt wurde. Vorher kam immer der »Save as« Dialog und der Dateiname wurde nicht in der Fensterüberschrift angezeigt. Die Linie unter

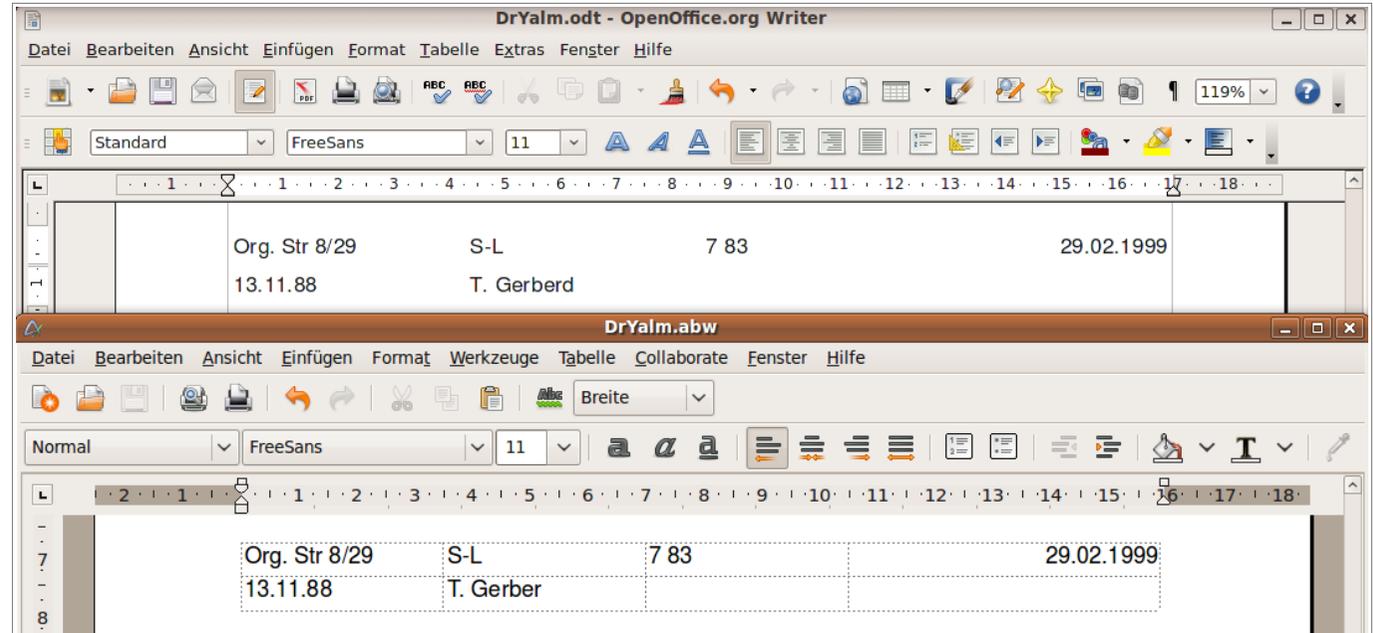


Abb. 4: Tabellen

dem Briefkopf kann nur durch eine entsprechend formatierte Tabelle erzeugt werden.

Fazit: Writer seine Aufgabe gut; AbiWord ist für diese Aufgabe kaum zu gebrauchen (OpenOffice Writer 4:3 AbiWord).

### Tabellen

Im nächsten Test wird die Adresse des Empfängers eingegeben und eine Tabelle mit Aktenzeichen. Was beim normalen Schreiben auffällt, ist die automatische Textergänzung in Writer. Kaum hat man die ersten drei Buchstaben eines bekannten Wortes geschrieben, erscheint auch schon ein Vorschlag für das gesamte Wort. Nachdem man

sich an diese Hilfestellung gewöhnt hat, geht damit das Schreiben um einiges schneller von der Hand. AbiWord hat hier nichts zu bieten.

Nun zur Tabelle: Wie im ganzen Artikel geht es nicht darum, welche Textverarbeitung mehr Funktionen zu bieten hat, sondern wie leicht einfache Aufgaben ausgeführt werden können. Im Beispiel handelt es sich um eine sehr einfache Tabelle mit 4 Spalten und 2 Zeilen. Die Abbildung 4 zeigt das Ergebnis.

Im Writer kann die Tabellengröße in der Werkzeugleiste aus einem kleinen Raster ausgewählt werden. AbiWord bietet hierfür einen ähnlichen

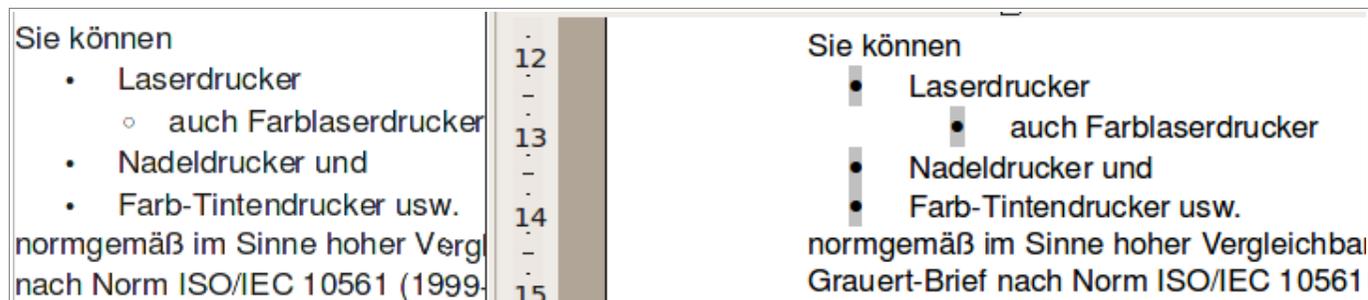


Abb. 5: Aufzählung (links Writer, rechts AbiWord)

Schalter, der jedoch erst zur Verfügung steht, nachdem man die Werkzeugleiste für Tabellen eingeschaltet hat. Im Writer erscheint automatisch eine passende Werkzeugbox, sobald sich die Schreibmarke in einer Tabelle befindet.

Im Beispielbrief soll die Tabelle keinen Rahmen haben. Dazu gibt es in Writer eine Schaltfläche mit der die Art der Umrandung ausgewählt werden kann, das ist sehr praktisch. Ein solcher Schalter fehlt bei AbiWord. Hier muss die Funktion »Tabelle formatieren« aus dem Menü benutzt werden.

Beim Testen der Kandidaten fällt auf, dass Writer nicht anzeigt, ob das aktuelle Dokument geändert wurde und deshalb evtl. gespeichert werden sollte. AbiWord zeigt dies durch einen Stern neben dem Dateinamen im Fensterkopf an.

Bei beiden Textverarbeitungen gibt es hier Licht und Schatten, deshalb bekommen beide einen Punkt für die Tabellen (OpenOffice Writer 5:4 AbiWord).

### Aufzählungen

Nummerierte Aufzählungen oder Punktlisten gehören ebenfalls zu den häufig verwendeten Formatierungen in Texten. Vorher kommt im Brief aber noch die Betreffzeile mit einem rechtsbündigen »EILT« Stempel als GIF-Bild. Das funktioniert in beiden Programmen problemlos, womit nach den Schwierigkeiten in der Kopfzeile bei AbiWord nicht zu rechnen war. In Writer ist das Einfügen von Bildern jedoch etwas einfacher und intuitiver möglich.

Kommen wir zu den Aufzählungen: Obwohl es sich dabei um eine einfache Formatierung handelt, warten die beiden Textverarbeitungen mit Unter-

schieden auf. In Abbildung 5 sind diese deutlich zu sehen.

OpenOffice Writer ändert das Symbol, wenn die Aufzählung weiter eingerückt wird; bei AbiWord bleibt das Symbol gleich und wird grau hinterlegt, obwohl die Druckansicht verwendet wurde. Dafür gibt es einen Punkt für Writer und null Punkte für AbiWord (OpenOffice Writer 6:4 AbiWord).

Beim Einrücken von Absätzen unterscheiden sich die beiden Kandidaten nicht voneinander. Für das Ein- bzw. Ausrücken gibt es Schaltflächen in der Werkzeugleiste. Alles funktioniert ohne Fehl und Tadel.

Nun gut, die grauen Felder zählen als Wiederholungsfehler und fließen nicht noch einmal in die Bewertung ein; ansonsten erfüllt AbiWord die Erwartungen. Writer hat so seine eigenen Vorstellungen davon, welche Ziffer unter 1. folgt. Während AbiWord auf der zweiten Ebene korrekt mit 1.1 fortfährt, bleibt Writer stur bei der einstelligen

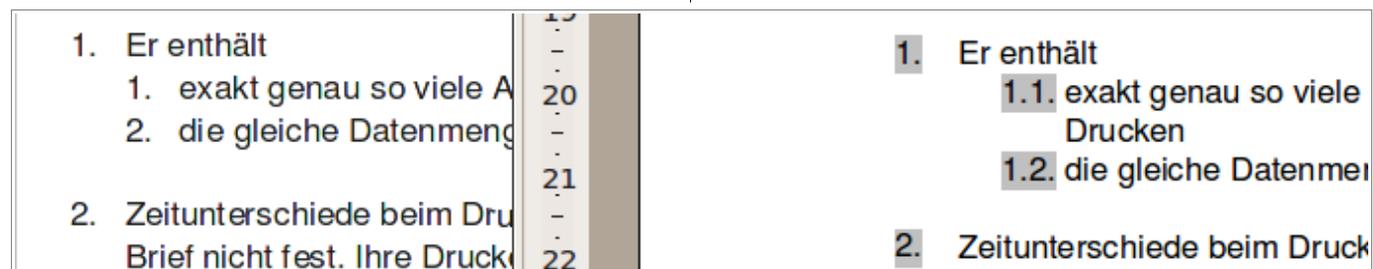


Abb. 6: Nummerierung (links Writer, rechts AbiWord)

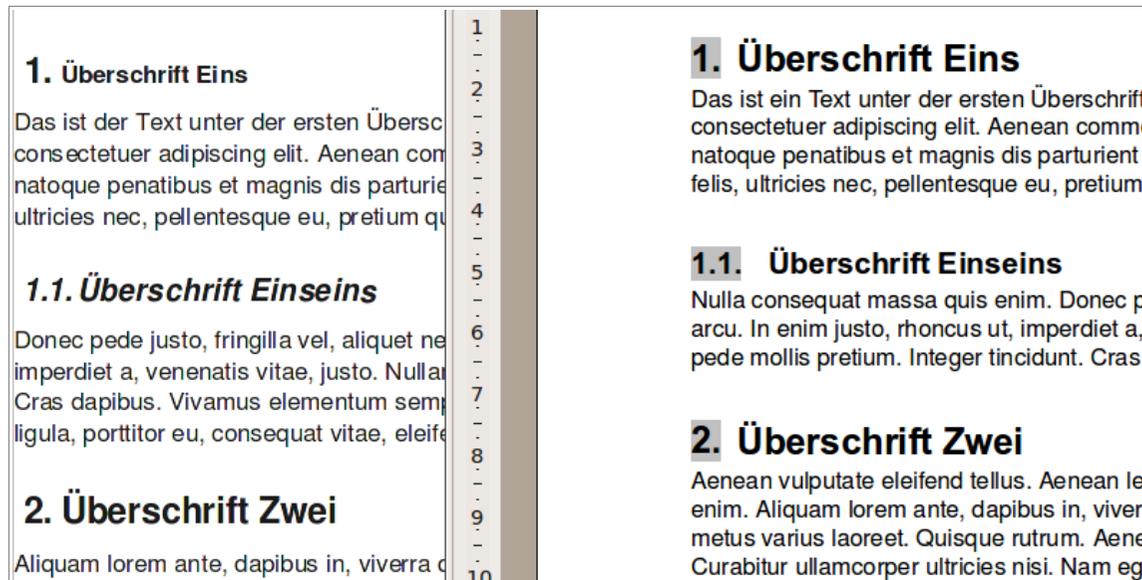


Abb. 7: Gliederung

Nummerierung. Deshalb gibt es einen Pluspunkt für AbiWord (OpenOffice Writer 6:5 AbiWord).

## Gliederung

Die Gliederung ist sozusagen die Königsdisziplin der Textverarbeitung. Dem Autor ist noch kein Programm untergekommen, das diese Aufgabe problemlos beherrscht. Ein Blick auf die Bildschirmfotos lässt Schlimmes ahnen.

Bei OpenOffice Writer hängt das Ergebnis davon ab, was man genau macht. Im Beispiel wurde der Text »Überschrift Eins« zuerst als normaler Text geschrieben und ihm dann die Formatvorlage »Überschrift 1« zugewiesen. Die fehlende Nummerierung wird über das Menü »Format, Numme-

rierung und Aufzählungszeichen« erzeugt. Im Dialog wurde im Tab »Gliederung« die erste Auswahl getroffen (siehe Abb. 8).

Bei den nächsten Überschriften kann man sich das Einschalten der Nummerierung sparen; Writer macht diese selbstständig. Es muss nur die gewünschte Formatvorlage (»Überschrift 1..2 usw.«) ausgewählt werden. Das Ergebnis sieht etwas seltsam aus: Die Größe von Nummerierung und Text stimmen nicht überein, und der kleine linke Einzug mutet auch sonderbar an.

In AbiWord wird die Gliederung ganz einfach durch Zuweisen der Formatvorlage »Numbered Heading 1« erzeugt bzw. »Numbered Heading 2« für die

nächste Gliederungsebene. Das ist deutlich einfacher als bei Writer und sieht zudem besser aus. Die Textgrößen passen zueinander und alle Überschriften in der Gliederung sind linksbündig ausgerichtet.

Das ergibt einen klaren Punkt für AbiWord (OpenOffice Writer 6:6 AbiWord).

## Speichern

OpenOffice Writer speichert seine Dokumente bekanntlich im Open Document Format, sowie in 14 anderen wenn es denn sein soll. Die ODT-Datei von Writer ist tatsächlich ein Archiv welches acht Einzeldateien beinhaltet; die meisten davon im XML Format. AbiWord kennt für seine Dateien das Format ABW, das ebenfalls XML ist, jedoch nur aus einer Datei besteht. Außer in diesem kann AbiWord seine Texte in 26 anderen Formaten speichern, auch im Open Document Format (ODT).

Wer nun meint, dass Open Document gleich Open Document ist, der hat sich getäuscht. Speichert man den AbiWord Brief im ODT-Format und öffnet

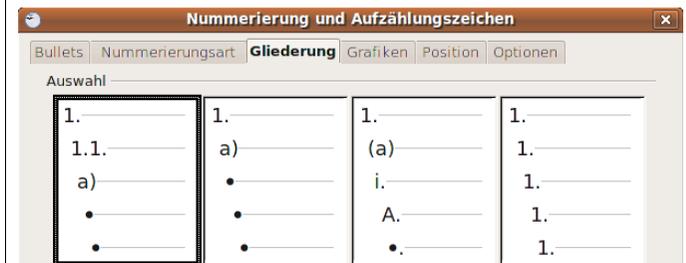


Abb. 8: Gliederungsoption in Writer

Yalm Magazin  
Root-Str. 12  
1234 Linuxstadt

Yalm-Magazin  
Root-Str. 12  
1234 Linuxstadt

---

**Das Linux Magazin für  
Ein- und Umsteiger...**



**Das Linux Magazin für  
Ein- und Umsteiger...**



---

**Eilzustellung**

Bayerische Hammerwerke GmbH  
Herrn Dr. Yalm  
Rechts der Isar 73

82367 München

Org. Str 8/29      S-L      7 83      29.02.1999  
13.11.88      T. Gerber

**Geschwindigkeitstest mit vielen Laser- und Tintendruckern  
für die Yalm-Leser zum Nachmachen**

Sehr geehrter Herr Dr. Yalm,

Sie können

- Laserdrucker
  - auch Farblasendrucker
- Nadeldrucker und
- Farb-Tintendrucker usw.

normgemäß im Sinne hoher Vergleichbarkeit testen, indem Sie im wesentlichen den Grauert-Brief nach Norm ISO/IEC 10561 (1999-05) verwenden.

Anhand dieses Dokuments lassen sich mit Ihrem Drucker ermittelte Werte mit den Herstellerangaben zum Drucktempo vergleichen.

Weil der Dr.-Grauert-Brief urheberrechtlichem Schutz unterliegt, können wir den Text nicht frei zur Verfügung stellen; die Norm kostet 41.30 Euro, als Alternative können Sie diesen Brief nutzen, der in Anlehnung an den Dr.-Grauert-Brief erstellt wurde.

1. Er enthält

1.1. exakt genau so viele Anschläge wie der Dr.-Grauert-Brief und erzeugt beim Drucken

1.2. die gleiche Datenmenge wie das ISO/IEC-genannte Original.

2. Zeitunterschiede beim Drucken des Dokuments stellen wir im Vergleich mit dem Grauert-Brief nicht fest. Ihre Druckdauer mit dem "Yalm" sollte unseren Yalm-Labor-Resultaten daher entsprechen.

Mit freundlichen Grüßen

**EILT!**

**Eilzustellung**

Bayerische Hammerwerke GmbH  
Herrn Dr. Yalm  
Rechts der Isar 73

82367 München

Org. Str 8/29      S-L      7 83      29.02.1999  
13.11.88      T. Gerber

**Geschwindigkeitstest mit vielen Laser- und Tintendruckern  
für die Yalm-Leser zum Nachmachen**

Sehr geehrter Herr Dr. Yalm,

Sie können

- Laserdrucker
  - auch Farblasendrucker
- Nadeldrucker und
- Farb-Tintendrucker usw.

normgemäß im Sinne hoher Vergleichbarkeit testen, indem Sie im wesentlichen den Grauert-Brief nach Norm ISO/IEC 10561 (1999-05) verwenden.

Anhand dieses Dokuments lassen sich mit Ihrem Drucker ermittelte Werte mit den Herstellerangaben zum Drucktempo vergleichen.

Weil der Dr.-Grauert-Brief urheberrechtlichem Schutz unterliegt, können wir den Text nicht frei zur Verfügung stellen; die Norm kostet 41.30 Euro, als Alternative können Sie diesen Brief nutzen, der in Anlehnung an den Dr.-Grauert-Brief erstellt wurde.

1. Er enthält

2.1. exakt genau so viele Anschläge wie der Dr.-Grauert-Brief und erzeugt beim Drucken

2.2. die gleiche Datenmenge wie das ISO/IEC-genannte Original.

2. Zeitunterschiede beim Drucken des Dokuments stellen wir im Vergleich mit dem Grauert-Brief nicht fest. Ihre Druckdauer mit dem „Yalm“ sollte unseren Yalm-Labor-Resultaten daher entsprechen.

Mit freundlichen Grüßen

Abb. 9: AbiWord in Writer (links) und Writer in AbiWord (rechts)

ihn danach in Writer, erhält man als Ergebnis das, was in Abbildung 9 zu sehen ist.

Rechts daneben ist das umgekehrte Resultat zu sehen, die Writer-Datei wurde in AbiWord geöffnet. Auf den ersten Blick scheinen die beiden Briefe in Ordnung zu sein und sich sehr zu gleichen. Bei genauerem Hinsehen erkennt man im linken Brief (AbiWord als ODT gespeichert und in Writer geöffnet) einige Fehler bei der Ausrichtung des Absenders und der Aufzählungen. Der rechte Brief (Writer Brief in AbiWord geöffnet) sieht etwas besser aus: Die Linie unter der Kopfzeile wird nicht korrekt dargestellt und die Aufzählungszeichen sind ebenfalls ein wenig durcheinander geraten.

Welchem der beiden Textverarbeitungen diese kleinen Missgeschicke nun anzulasten sind, ist schwer zu sagen. Es waren ja immer beide Programme bei diesem Test beteiligt: eines zum Speichern und das andere zum Öffnen und Darstellen des Briefs. Allerdings wird eines klar: Eine Datei im Open Document Format kann nicht einfach in der Hoffnung weitergegeben werden, dass alle Formatierungen noch genauso aussehen wie im Original.

Beide Kandidaten erhalten je einen Punkt (OpenOffice Writer 7:7 AbiWord).

### Was sonst noch auffiel

Das Markieren von Text in AbiWord über mehrere Seiten hinweg funktioniert problemlos wenn man am Seitenende nicht die Geduld verliert. Ruht der

Mauszeiger bei gedrückter linker Taste einen Moment im Bereich der Statuszeile beginnt das Scrollen und Markieren auf die nächste Seite. Im Writer wird keine Geduld verlangt sondern gute Nerven. Ab der zweiten Seite erfolgt die Markierung so schnell, dass es schwierig ist den markierten Text genau zu bestimmen. Dafür erhält AbiWord einen klaren Punkt (OpenOffice Writer 7:8 AbiWord).

Ein Redaktionsmitglied berichtete über Probleme beim Scrollen unter Ubuntu 9.04 mit eingeschaltetem Compiz unter Abiword. In dieser Konstellation soll das Markieren und gleichzeitige Scrollen, über mehrere Seiten hinweg, nur bei gedrückter mittlerer Maustaste funktionieren. Der Autor konnte dieses Verhalten allerdings nicht nachvollziehen.

### Fazit

Der Vergleich zwischen Writer und AbiWord könnte noch ewig fortgeführt werden, wenn man alle Funktionen der beiden Programme berücksichtigen würde. Das Ergebnis des Vergleichs ist fast unentschieden (OpenOffice Writer 7:8 AbiWord) und – wie gesagt – absolut willkürlich.

Vielleicht hat der eine oder andere Leser einen klaren Sieg für OpenOffice Writer erwartet. Der Writer ist objektiv betrachtet und unter Berücksichtigung der schieren Funktionsvielfalt dem Produkt AbiWord haushoch überlegen. Konzentriert man sich aber auf die alltäglichen Arbeiten, die mit einer Textverarbeitung durchgeführt werden, so liegen beide Werkzeuge gleichauf. Für AbiWord

sprechen die schnelle Startzeit und die unkomplizierte Bedienung. Writer punktet mit Exaktheit (z. B. bei Grafiken) und größerem Komfort (z. B. Wortergänzung).

Somit stehen für die täglichen Textarbeiten (neben vielen anderen) mit AbiWord und Writer zwei gute und ausgereifte Anwendungen zur Verfügung. Wem diese Textverarbeitungen zu umständlich und überladen sind, dem sei der gute alte »vi« empfohlen.

*Ralf Hersel*

*[rhersel@yalmagazine.org](mailto:rhersel@yalmagazine.org)*

### Informationen

- [1] Dr.-Grauert-Brief: <http://de.wikipedia.org/wiki/Dr.-Grauert-Brief>
- [2] Download der beiden Dr. Yalm-Briefe: <http://yalmagazine.org/homepage/downloads/extras/DrYalm.zip>

# Websites mit dem Internet Explorer 6, 7 und 8 unter Linux testen

Will man seinen Webauftritt in den am weitesten verbreiteten Browsern möglichst ähnlich aussehen lassen, muss man ihn auch mit dem Internet Explorer testen. Der Wechsel zum Windows-Betriebssystem oder zu einem virtuellen Windows-PC ist zwar möglich, aber zeitraubend und umständlich. Nachhaltig bequemer geht es mit Wine und Winetricks.

Man mag sich streiten, ob die Internet Explorer-Serie gut oder schlecht ist, denn im Vergleich zum Firefox sind die Browser schwer zu erweitern. Für Webdesigner ist jedoch klar, dass die Internet Explorer wegen der Vorinstallation auf einem neuen PC oder der Verzahnung mit Windows (er lässt sich dort nur sehr aufwendig deinstallieren) eine große Anwendergemeinde haben. Damit man die Eigenheiten dieser Browser in den Griff bekommt, muss man seine Websites mit diesen Browsern (bzw. deren Rendering-Engines) testen. Dieser Artikel erläutert, wie das ohne Windows-Betriebssystem oder einer virtuellen Maschine funktioniert. Grundlage dafür ist die Windows-kompatible Laufzeitumgebung Wine [1] [2], die man über den Paketmanager oder die Hersteller-Website [3] bekommt. Für den Umgang mit dem Terminal [4] werden nur Grundkenntnisse benötigt.

## Alternative zur lokalen Installation

Wer die manuelle Installation nicht mag, kann das IE NetRenderer Plugin [5] für den Firefox 3 benutzen. Damit wird die gerade aufgerufene URL an den NetRenderer Service gesendet, wo der obere Seitenbereich mit der Engine des ausgewählten



Der IE6 neben dem IE7 unter Ubuntu 9.04.



NetRenderer zeigt eingängig die Darstellungsunterschiede bei IE6 und IE7.

## Internet Explorer 6

Dieser Browser lässt sich unter Linux am einfachsten installieren. Man muss sich hierzu nur das Programm *IES4Linux* herunterladen [7] und kann damit den IE6 mit nur wenigen Klicks installieren. Das Entpacken des Programmarchivs und das Ausführen der Datei *ies4linux* starten die Installationsroutine. Das Programm-Projekt IES4Linux wird jedoch seit 02/2006 nicht mehr weiterentwickelt.

Möchte man den Browser deinstallieren, so tut man dies über das grafische Menü von Wine, was sehr einfach ist. Mit den folgenden Internetbrowsern verhält es sich ähnlich, sodass im weiteren Verlauf nicht mehr auf die Deinstallation eingegangen wird.

*Tip am Rande:* Der größte Nachteil dieses Browsers ist die Nicht-Darstellung der Transparenz bei PNG-Bildern, egal ob unter Windows oder Linux. Mit *IEPNGFix* [8] benutzt man DirectX mittels JavaScript für die Darstellung. Es empfiehlt sich, die aktuelle Entwicklerversion von Wine zu benutzen, da dort DirectX (und sogar Teile von DirectX 10) am besten implementiert ist. Glücklicherweise nimmt die Verwendung des IE6 stetig ab und wird mit dem Erscheinen von Windows 7 zusätzlich abnehmen, sodass modernen Technologien mehr Aufmerksamkeit gewidmet werden kann.

## Internet Explorer 7

Die Datei `~/wine/user.reg` wird mit dem Editor geöffnet und um folgendes erweitert:

```
[SoftwareWineDllOverrides]
1233608859
"browseui"="native, builtin"
"comctl32"="builtin"
"crypt32"="native, builtin"
"gdipplus"="native"
"hhctrl.ocx"="native, builtin"
"hlink"="native, builtin"
"iernonce"="native, builtin"
"iexplore.exe"="native, builtin"
"itircl"="native, builtin"
"itss"="native, builtin"
"jscript"="native, builtin"
"mlang"="native, builtin"
"mshtml"="native, builtin"
"msimtf"="native, builtin"
"msxml3"="native, builtin"
"riched20"="native, builtin"
"riched32"="native, builtin"
"secur32"="native, builtin"
"shdoclc"="native, builtin"
"shdocvw"="native, builtin"
"shlwapi"="native, builtin"
"url"="native, builtin"
```

```
"urlmon"="native, builtin"
"usp10"="native, builtin"
"uxtheme"="native, builtin"
"wininet"="builtin"
"wintrust"="native, builtin"
```

Anschließend werden folgende DLL-Dateien aus dem system32-Ordner von Windows XP herauskopiert und in den gleichnamigen Ordner von Wine eingefügt:

```
msctf.dll
msimtf.dll
uxtheme.dll
xmllite.dll
```

Nun wird noch das Programm Winetricks benötigt, das gegebenenfalls noch installiert werden muss [9]. Anschließend wechselt man mit dem Terminal in den Ordner, in dem sich die winetricks-Datei befindet und führt sie mit

```
sh winetricks
```

aus. Es öffnet sich das Programmfenster von Winetricks, in dem neben folgende Einträge eine Markierung gesetzt wird:

```
comctl32
comctl32.ocx
```

```
corefonts
gdipplus
gecko
msls31
msxml3
msxml4
msxml6
riched20
riched30
tahoma
```

Nach dem Bestätigen öffnen sich nacheinander einige Installationsdialoge. Nach deren Abschluss führt man noch folgenden Befehl aus (es kann auch ohne funktionieren):

```
winecfg set comctl32.dll
```

Jetzt kann die Setup-Datei vom Internet Explorer 7 heruntergeladen [10], mit Wine installiert und benutzt werden.

### Internet Explorer 8

Es ist nicht möglich, den ganzen Internet Explorer 8 unter Wine zum Laufen zu bringen, man kann jedoch die Darstellungseengine testen. Hierbei bedient man sich der bereits erwähnten Programme IEs4Linux und Winetricks. Die Bedienelemente werden zwar später wie beim IE6 aussehen, aber das Rendering erfolgt wie beim IE8 und darum geht es schließlich beim Testen der Webseiten.

Zunächst installiert man den IE7 aus IEs4Linux heraus (er ist dort als Beta-Version angegeben). Damit es keine Konflikte mit anderen installierten Windows Programmen geben kann, wird eine eigene »Winebottle« angelegt (ein eigener Ordner mit eigenen Einstellungen).

```
mkdir .ie8
cd .ie8
cp -r ~/.ies4linux/ie7/* .
```

Dann holt man sich mit Winetricks zusätzliche Libraries.

```
WINEPREFIX=~/.ie8 winetricks mfc40
WINEPREFIX=~/.ie8 winetricks mfc42
```

Die Fehlermeldungen, die dabei auftreten, können ignoriert werden. Wichtig ist der »WINEPREFIX«, andernfalls überschreibt man die Dateien in `~/.wine/`.

Danach kann der Internet Explorer 8 von Microsoft herunter geladen werden [11]. Anschließend wird die EXE-Datei mit dem Befehl `cabextract` entpackt.

Es wird eine Anzahl DLLs benötigt, die innerhalb der EXE-Datei liegen.

```
cp wininet.dll iertutil.dll shlwapi.dll →
urlmon.dll jscript.dll vbscript.dll →
mshtml.dll mshtml.dll mshtml.dll →
```

```
advpack.dll inetctl.cpl →
~/.ie8/drive_c/windows/system32
```

Danach sollten mit `winecfg` die überflüssigen DLL-Overrides entfernt werden, da sie sonst Probleme verursachen.

```
WINEPREFIX=~/.ie8 winecfg
```

Anschließend wird ein sogenannter Wineboot durchgeführt.

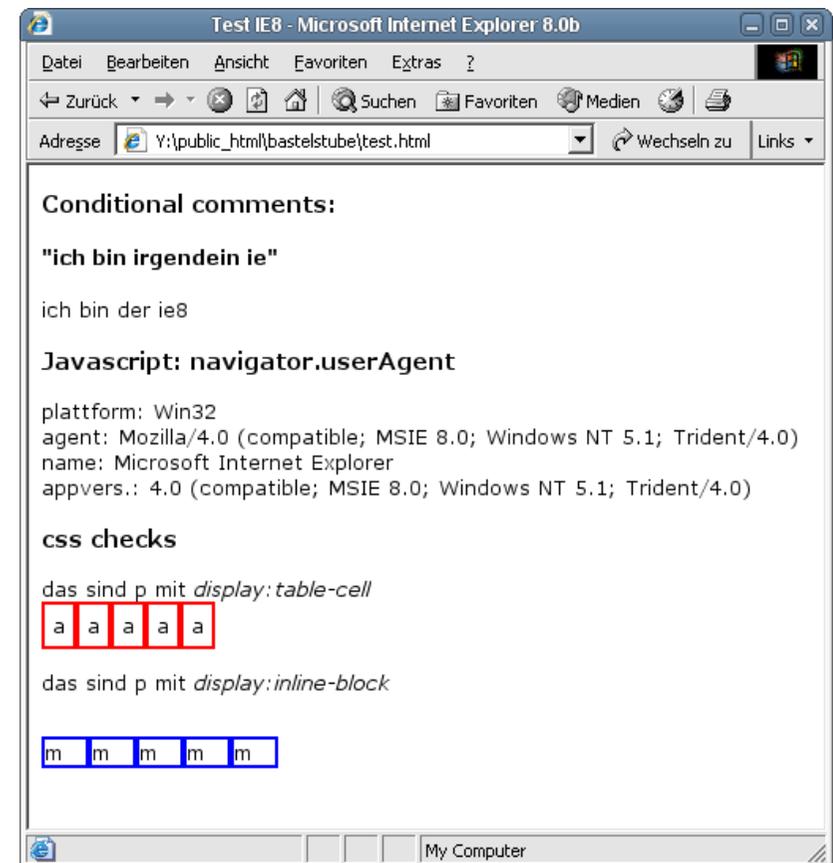
```
WINEPREFIX=~/.ie8 wineboot
```

Hier die Liste mit den erforderlichen DLL-Overrides für den IE8:

```
"iertutil"="native,builtin"
"jscript"="native,builtin"
"mshtml"="native,builtin"
"mshtml"="native,builtin"
"mshtml"="native,builtin"
"mshtml"="native,builtin"
"mshtml"="native,builtin"
"msxml3"="native,builtin"
"shdoc401"="native,builtin"
"shdoclc"="native,builtin"
"shdocvw"="native,builtin"
"shlwapi"="native,builtin"
"url"="native,builtin"
```

```
"urlmon"="native,builtin"
"vbscript"="native,builtin"
"winhttp"="native,builtin"
"wininet"="native,builtin"
```

Jetzt kommt noch ein wenig Kosmetik: der Fenstertitel wird angepasst. Da man mittels der



Die *display*-Anweisungen »*table-cell*« und »*inline-block*« sind im IE8 neu.

gegebenen Bedienelemente des IE8 nicht an die Internet Optionen heran kommt, setzt man die Startseite in der Datei *user.reg* auf »about:blank« (ohne Anführungszeichen).

```
cd ~/.ie8

sed -i -e 's/Microsoft Internet →
Explorer 7.0/Microsoft Internet →
Explorer 8.0b/' user.reg

sed -i -e 's/http:\\\\www.tatanka.com.br→
\\ies4linux\\startpage?lang=deDE&→
ieversion=ie7/about:blank/' user.reg
```

Um herauszufinden, ob tatsächlich die Engine des IE8 arbeitet, haben wir ein HTML-Dokument erstellt [12].

Zum Schluss ist ein Skript zum Starten und Unterdrücken von Fehlermeldungen nützlich. Man lege dieses in den Ausführfad unter dem Namen »ie8« und ersetze »jesper« durch den eigenen Benutzernamen.

```
#!/usr/bin/env bash

# IEs 4 Linux script to run ie7 - →
http://tatanka.com.br/ies4linux

# changed by xela

debugPipe() {
    while read line; do [ "$DEBUG" →
= "true" ] && echo $line; done
```

```
}
cd
export WINEPREFIX="/home/jesper/.ie8"

wine "/home/jesper/.ie8/drive_c/Program →
Files/Internet Explorer/IEXPLORE.EXE" →
"$@" 2>&1 | debugPipe
```

Der Autor hätte gerne den Acid2-Test mit dem IE8 aufgerufen; angeblich erhält er dort, im Vergleich zu seinen Vorgängern, eine gute Punktzahl. Allerdings wird der Browser dabei jedoch instabil. Damit eignet sich der so installierte Internet Explorer 8 zum Testen, ist aber nicht als Standardbrowser zu empfehlen. Sollte die Installation nicht auf Anhieb funktionieren, kann man verschiedene Konstellationen der DLL-Overrides ausprobieren. Wem eine Installation des IE8 ohne die Bedienelemente des IE6 von IEs4Linux gelingt, der benötigt zum Testen keinen separaten IE7. Denn dann kann man im Menü die Rendering-Engine des IE7 einstellen.

*Peter Majmesku*  
pe@yalmagazine.org

## Informationen

- [1] Wine-Homepage: <http://www.winehq.org/>
- [2] Wikipedia-Artikel zum Thema Wine: <https://secure.wikimedia.org/wikipedia/de/wiki/Wine>
- [3] Homepage von Wine: <http://www.winehq.org/>
- [4] Artikel über das Terminal im Ubuntuusers-Wiki: <http://wiki.ubuntuusers.de/Terminal?highlight=terminal>
- [5] Quelle für das IE NetRenderer Plugin: <https://addons.mozilla.org/de/firefox/addon/6455>
- [6] Browsershots-Webservice: <http://browsershots.org/>
- [7] Homepage von IEs4Linux: <http://www.tatanka.com.br/ies4linux/page/De/Hauptseite>
- [8] Transparenz beim IE6 mit IEPNGFix: <http://www.twinhelix.com/>
- [9] Wiki-Seite von Winetricks: [http://wiki.winehq.org/winetricks\\_de](http://wiki.winehq.org/winetricks_de)
- [10] Downloadseite vom IE7: <http://www.microsoft.com/Downloads/details.aspx?familyid=9AE91EBE-3385-447C-8A30-081805B2F90B&displaylang=en>
- [11] Downloadseite des IE8: <http://www.microsoft.com/germany/windows/internet-explorer/default.aspx>
- [12] HTML-Dokument zum Test der IE8-Engine: <http://www.netzor.de/ie8test.htm>

# Moblin 2 Beta Preview

Das einst von Intel entwickelte und nun von der Linux Foundation betreute Projekt erreichte kürzlich den Betastatus. Nach mehreren faszinierenden Videos konnten wir bald nicht mehr anders als das vielversprechende Betriebssystem zu testen.

Einer der Gründe, warum XP auf Netbooks mittlerweile Linux abgehängt hat ist die Tatsache, dass die Hersteller unterschiedliche Distributionen auf den Netbooks vorinstallieren und diese praktisch unkonfigurierbar machen. Software nachinstallieren ist mit dem Standardbetriebssystem kaum möglich, Softwareupdates der installierten Software werden minimal bis gar nicht angeboten. Das führte unter anderem dazu, dass viele angepasste, von der Community gepflegte Distributionen wie EeeUbuntu entstanden, die speziell auf die Bedürfnisse der Netbookbesitzer abgestimmt wurden.

Hier versucht Intel mit Moblin eine sehr vielversprechende Distribution für Netbooks zu etablie-

<b>Eingesetzter Kernel</b>	2.6.29.3
<b>Standarddateisystem</b>	ext3
<b>Benötigter Speicherplatz</b>	rund 2.5 GB
<b>Paketmanagement</b>	packagekit & yum
<b>Browser</b>	Firefox 3.0.10
<b>Messenger</b>	Empathy 2.26.1
<b>Mediaplayer</b>	Eigenentwicklung

*Einige Details*

ren. Die Motivation dürfte wohl die in Netbooks verbaute Hardware (fast immer kommen Intel CPUs und GPUs zum Einsatz) und die allgemeine Enttäuschung über die Hardwareanforderungen aktueller Windowsversionen sein [1].

Mitte April dieses Jahres wurde Moblin von der Linux Foundation übernommen. Obwohl Moblin nicht mehr, wie ursprünglich, auf Ubuntu basiert, will Canonical künftig auf angepasste Moblin-Versionen setzen [2].

## Unterstützte Plattformen

Folgende Netbooks werden offiziell von Moblin unterstützt:

- Acer Aspire One (ZG5)
- Asus EeePC 901, 1000, 1000H
- Dell Inspiron 910
- MSI Wind
- Lenovo S10
- Samsung NC10
- HP mini 1010 and 1120NR (kein WLAN)

Nicht unterstützt werden explizit Netbooks mit einem Intel GMA-500 Grafikchip, worunter einige der teuren Geräte von MSI und Sony fallen.

## Installation

Unser Testnotebook war ein Acer Aspire One 110L Netbook.

Zuerst laden wir uns das Image und das in Python geschriebene Installationskript herunter. Es ist auch möglich, die img-Datei direkt mit `dd` auf den USB-Stick zu schreiben, wir erhielten aber nach mehreren Versuchen immer einen schwarzen Bildschirm beim Booten.

```
wget http://mirrors.kernel.org/moblin/ →
releases/test/betaimages/moblin-netbook-
-beta-20090524-001.1.img
wget http://git.moblin.org/cgit.cgi/ →
moblin-image-creator/plain/image-writer
chmod +x image-writer
sudo ./image-writer moblin-netbook-beta-
-20090524-001.1.img
```

Nach dem Ausführen des Skriptes muss man den gewünschten USB-Stick angeben, mit dem man später Moblin installieren will. *Achtung:* Hier besonders aufpassen, da alle Daten auf dem USB-Stick überschrieben werden; andere externe Datenspeicher eventuell vorher zur Sicherheit ausstecken!

Ist der USB-Stick mit dem Moblin-Image beschrieben, stecken wir ihn in einen USB Slot und booten davon Moblin (Hinweis: Dazu sind eventuell Anpassungen der Bootreihenfolge im BIOS notwendig).

Die Installation selbst ähnelt der von Ubuntu stark; positiv aufgefallen ist uns ein einfacher Assistent für die Festplattenverschlüsselung.

Genauere Informationen und Troubleshooting bezüglich der Installation findet man auf der Homepage [3].

### Oberfläche

Nachdem man das System in rund 15 Sekunden gebootet hat, begrüßt einen eine schlichte und schicke Oberfläche. Als Windowmanager wird Mutter (ausgesprochen: Matter) verwendet, ein Mix von Metacity und Clutter, welcher schicke und flüssige Effekte ermöglicht.

Die Benutzeroberfläche und Steuerung sind sehr intuitiv und können sich durchaus mit Apple-Produkten messen. Sämtliche Programme und Einstellungen sind über die elf Kategorien erreichbar, welche schick animiert über das Dock angesteuert werden können.

### Kategorien

Insgesamt elf Kategorien bündeln Applikationen und ermöglichen eine intuitive Bedienung der Oberfläche.



Die »M\_Zone« bietet einen schnellen Überblick über Aktivitäten

### 1. M\_Zone

Die M\_Zone bietet eine grobe Übersicht über interessante Informationen: Die nächsten Termine werden am linken Rand angezeigt, im Browser häufig besuchte Seiten und kürzlich abgespielte Medien kann man direkt über einen Klick erreichen. Am rechten Rand werden die Albencover der zuletzt gehörten Lieder angezeigt; die Informationen werden von Last.fm bezogen.

### 2. Status

Hier können Nachrichten für Microblogs wie Twitter eingegeben werden. Später sollen auch andere Dienste wie identi.ca, auf welchem auch Yalm seit neuestem zwitschert, ermöglicht werden.

### 3. People

Hier werden die Instant Messenger-Kontakte angezeigt. Das derzeit installierte Empathy unterstützt erst drei Protokolle: Jabber, Salut und GoogleTalk. Da die Einrichtung des Jabber-Kontos nicht funktioniert hat, können wir über die genaue Funktionsweise und weiteren Funktionen nur mutmaßen; das muss wohl an der Beta liegen.

### 4. Browser

Es sind standardmäßig zwei Browser verfügbar: Firefox und eine Eigenentwicklung, welche vermutlich auf XULRunner basiert. Firefox findet man komischerweise unter »Applications – Sonstige – Web Browser«.

Flash ist standardmäßig nicht dabei, kann aber relativ einfach über den Paketmanager installiert werden:

```
sudo yum install flash-plugin
```

Der Standardbrowser macht einen guten Eindruck. Er beherrscht eine Art »Speed Dial«, die Google Chrome sehr ähnlich ist: Häufige Seiten werden als Thumbnails beim Öffnen neuer Tabs angezeigt und erlauben einen schnellen Zugriff auf die meistgenutzten Seiten. Des Weiteren können Tabs mit einem Klick als grafische Vorschau angezeigt werden.

Der größte Nachteil des Standardbrowsers ist jedoch die fehlende Erweiterbarkeit. Wir haben keine Möglichkeit gefunden, Addons nachzuinstallieren.

## 5. Media

Eine der vielversprechendsten Kategorien, obwohl sie nur zur Hälfte implementiert ist. Hier werden kürzlich angeschaute Medien aufgelistet und allem Anschein nach lokale Medien und Netzwerkfreigaben aufgelistet.

Vor allem der Mediaplayer ist sowohl optisch wie auch von der Bedienung sensationell gelungen.

## 6. Pasteboard

Dieser Bereich bietet eine Übersicht über kopierte Texte und Dateien, ist aber im Grunde genommen nicht sonderlich nützlich.

## 7. Applications

Diesen Bereich wird man wohl am meisten aufrufen. Hier sitzen die Einstellungen, die Systemverwaltung und installierte Programme, die übersichtlich durch Kategorien getrennt sind. Eine Schnellsuchleiste erlaubt wie in den meisten Kategorien einen schnellen Zugriff auf Anwendungen.

## 8. Zones

Zones bietet eine Übersicht über geöffnete Programme und Arbeitsflächen und ist der Exposé-Funktion von Compiz bzw. Mac OS X nachempfunden. Auch Fenster können wir einfach auf andere Arbeitsflächen verschieben.

## 9 & 10. Power, Brightness & Volume

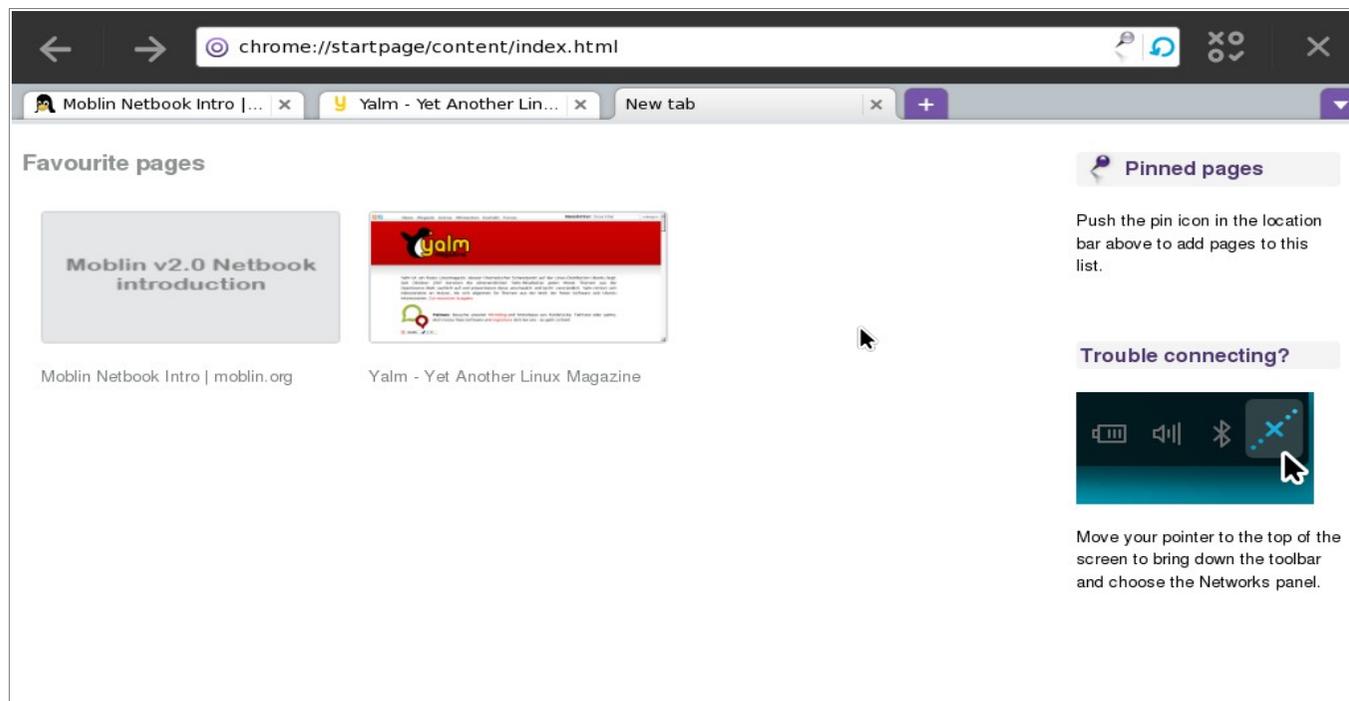
Hier befinden sich selbsterklärende Einstellungen für die Regelung von Bildschirmhelligkeit und Lautstärke. Das Ändern der Bildschirmhelligkeit wird anscheinend für unseren Laptop nicht unterstützt, geht aber problemlos über die Sondertasten vonstatten.

## 11. Networks

Unter diesem Reiter verbirgt sich die Netzwerkkonfiguration, welche äußerst übersichtlich und einfach gehalten ist. Blitzschnell sind wir mit Drahtlosnetzwerken verbunden; bei geschützten Netzwerken müssen wir lediglich das Passwort eingeben und sind dann verbunden.

## Softwareangebot

Das Softwareangebot ist im Moment noch extrem dünn gesät und beinhaltet oft fehlerhafte Pakete.



*Der angepasste Browser beherrscht Speed Dial und Tabs*

Ein IRC-Client ist ebenso wenig vorhanden wie OpenOffice; dafür gibt es Gnome Office, welches aber von Darstellungsfehlern geplagt wird und nicht standardmäßig installiert ist. Als Büroprogramme werden nur eine Kalender-, Notiz- und Adressverwaltung angeboten.

Soweit wir sehen konnten, war auch ein Mailclient nicht vorhanden.

Spiele sind standardmäßig nur drei installiert, zwei weitere lassen sich nachinstallieren.

Nach weiterer erfolgloser Suche ist klar: Ohne zusätzliche Repositories macht Moblin keinen Spaß. Gott sei Dank können wir diese aber selber nachrüsten.

### Neue Paketquellen hinzufügen

Da der Softwarefundus sehr begrenzt ist und einige installierte Programme wie Empathy im jetzigen Zustand nicht zufrieden stellen, muss man sich nach Paketquellen umsehen. Geeignete, wenn auch nicht zu 100% kompatible Quellen stellen die Repositories von Fedora Core 10 dar.

Um sie hinzuzufügen, legen wir die Datei

```
/etc/yum.repos.d/fedora10.repo
```

mit folgendem Inhalt an:

```
[fedora10-i386]
name=Fedora 10
baseurl=http://download.fedora.redhat.com/pub/fedora/linux/releases/10/Everything/i386/os
enabled=1
gpgcheck=1
```

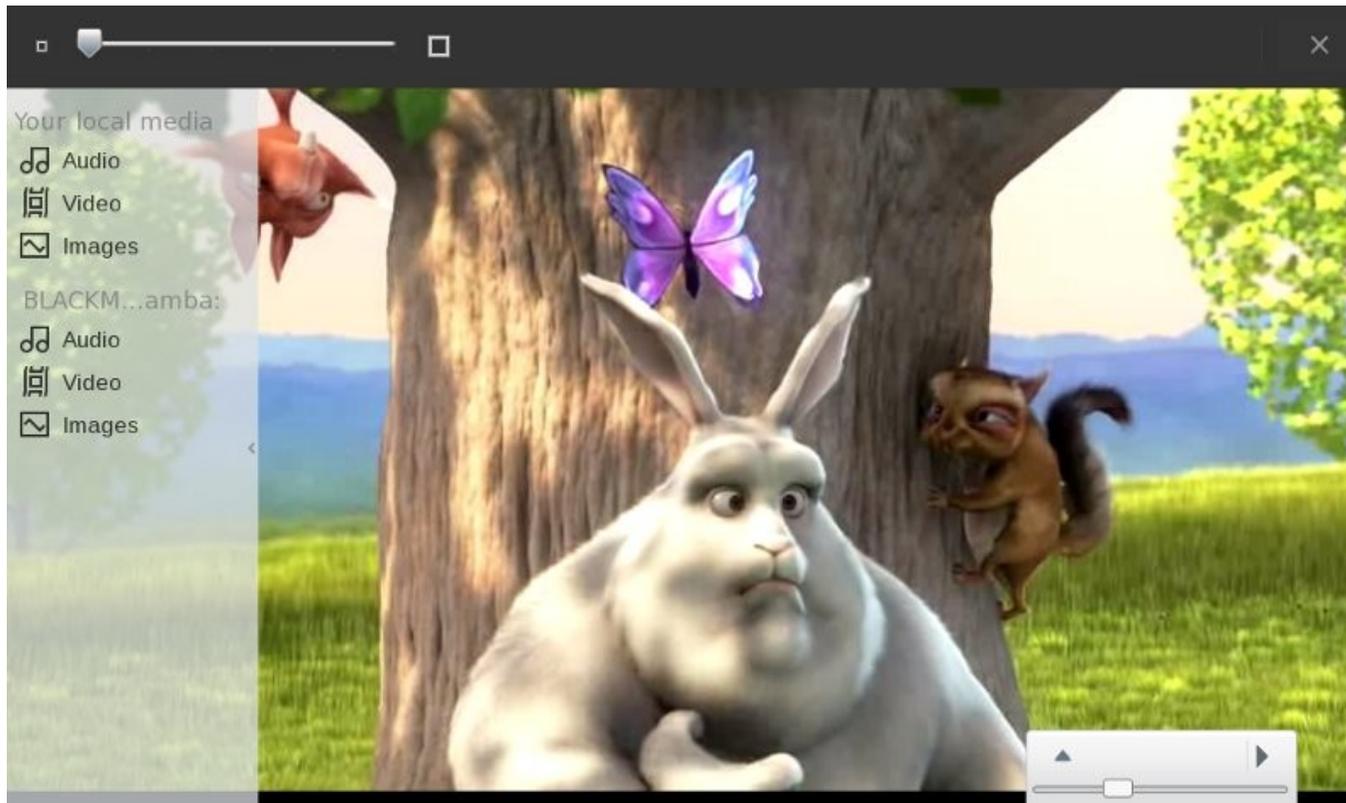
Als Editor verwenden wir vi oder gedit, welche über das Terminal aufgerufen werden können. Dieses befindet sich in »Applications – Zubehör«.

```
sudo gedit /etc/yum.repos.d/fedora10.repo
```

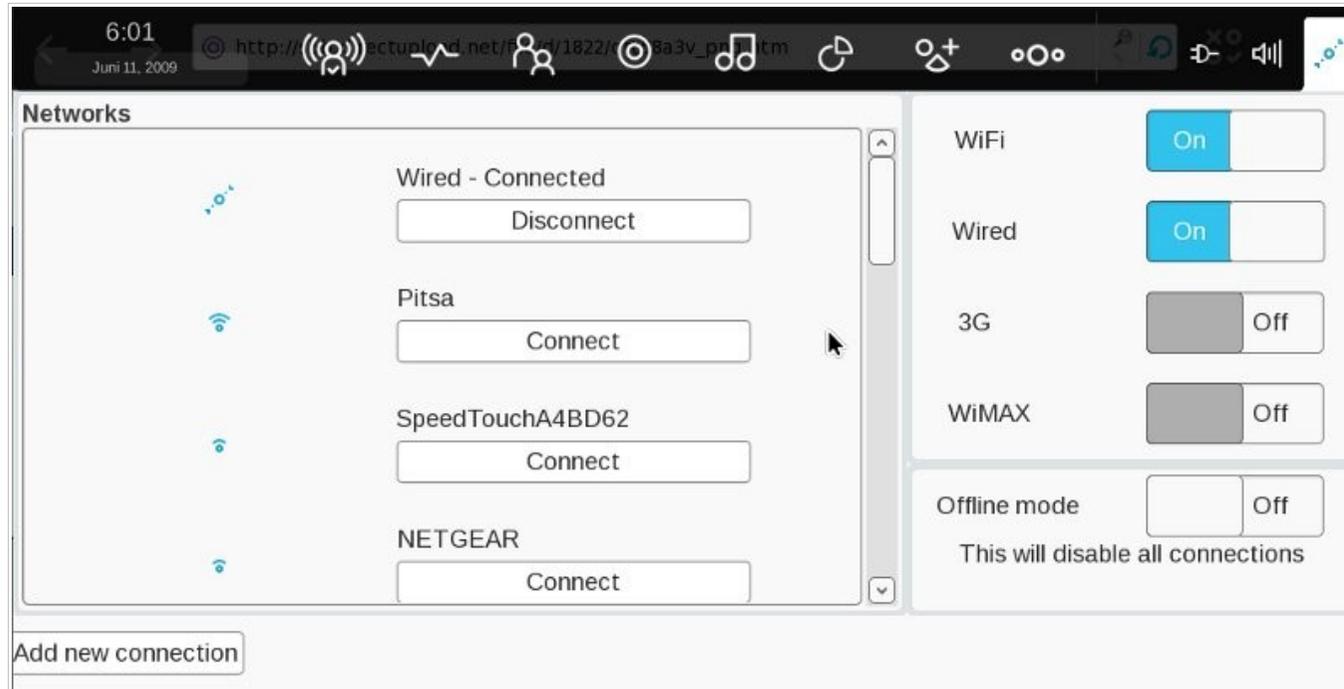
Nach einem

```
sudo yum makecache
```

ist der neue Softwarefundus verfügbar. Doch Vorsicht: Einige Pakete aus den Fedora Repositories kollidieren mit bereits installierter Software. So setzt Moblin beispielsweise auf Upstart, während Fedora noch das ältere Sysvinit nutzt. Auch einige Abhängigkeiten fehlen. Daher sollte mit aktivierten Fedora Repositories kein Upgrade durchgeführt werden.



Der Mediaplayer gewährt Zugriff auf lokale Medien und Samba Freigaben



Netzwerke werden übersichtlich angezeigt und die Verbindung ist ein Kinderspiel

Wir haben die unbrauchbare Empathy-Version durch Pidgin ersetzt:

```
sudo yum install pidgin
```

Dieses kann fortan über »Applications – Internet« gestartet werden.

Sind alle externen Applikationen installiert, empfiehlt es sich, die Fedora-Quellen wieder zu deaktivieren, um Moblin zu ermöglichen, sich sauber zu aktualisieren. Die Fedora-Repositories lassen sich grafisch unter »Applications – Einstellungen – Softwarequellen« deaktivieren.

### Fazit

Moblin bietet eine atemberaubend schöne, gut bedienbare und schnelle Oberfläche, welche das Beste aus dem Netbook herausholt. Der größte Haken jedoch ist das Softwareangebot: Die Standardquellen sind äußerst dürftig. Erst mit dem (unsauberen) Hinzufügen der Fedora-Repositories war es möglich, wichtige Software nachzuinstallieren.

Da es sich aber um eine Beta handelt, ist ein finales Fazit nicht möglich und wir hoffen, dass unsere Kritikpunkte bis zum finalen Release besei-

tigt werden. Sollten alle Programme wie vorgesehen funktionieren und die Repositories massiv erweitert werden, hat Moblin das Zeug zum Referenzsystem für Netbooks.

Bernhard Posselt  
ray@yalmagazine.org

### Informationen

- [1] Intel setzt auf Linux für Netbooks:  
<http://derstandard.at/fs/1244117133360/Finanzkrise-Intel-sieht-in-Windows-7-kein-Mittel-gegen-Krise>
- [2] Ubuntu setzt auf Moblin:  
<http://derstandard.at/fs/1242317178668/Ubuntu-setzt-kuenftig-auf-Moblin>
- [3] Hilfe um Moblin installieren:  
<http://moblin.org/documentation/test-drive-moblin/using-moblin-live-image>

## Video-Tutorials mit Wink

Wie man mit Linux Screenshots erstellt ist weithin bekannt. Dieser Artikel beschäftigt sich mit dem nächsten Schritt – dem Erstellen von umfangreichen Video-Tutorials.

### Screenshots

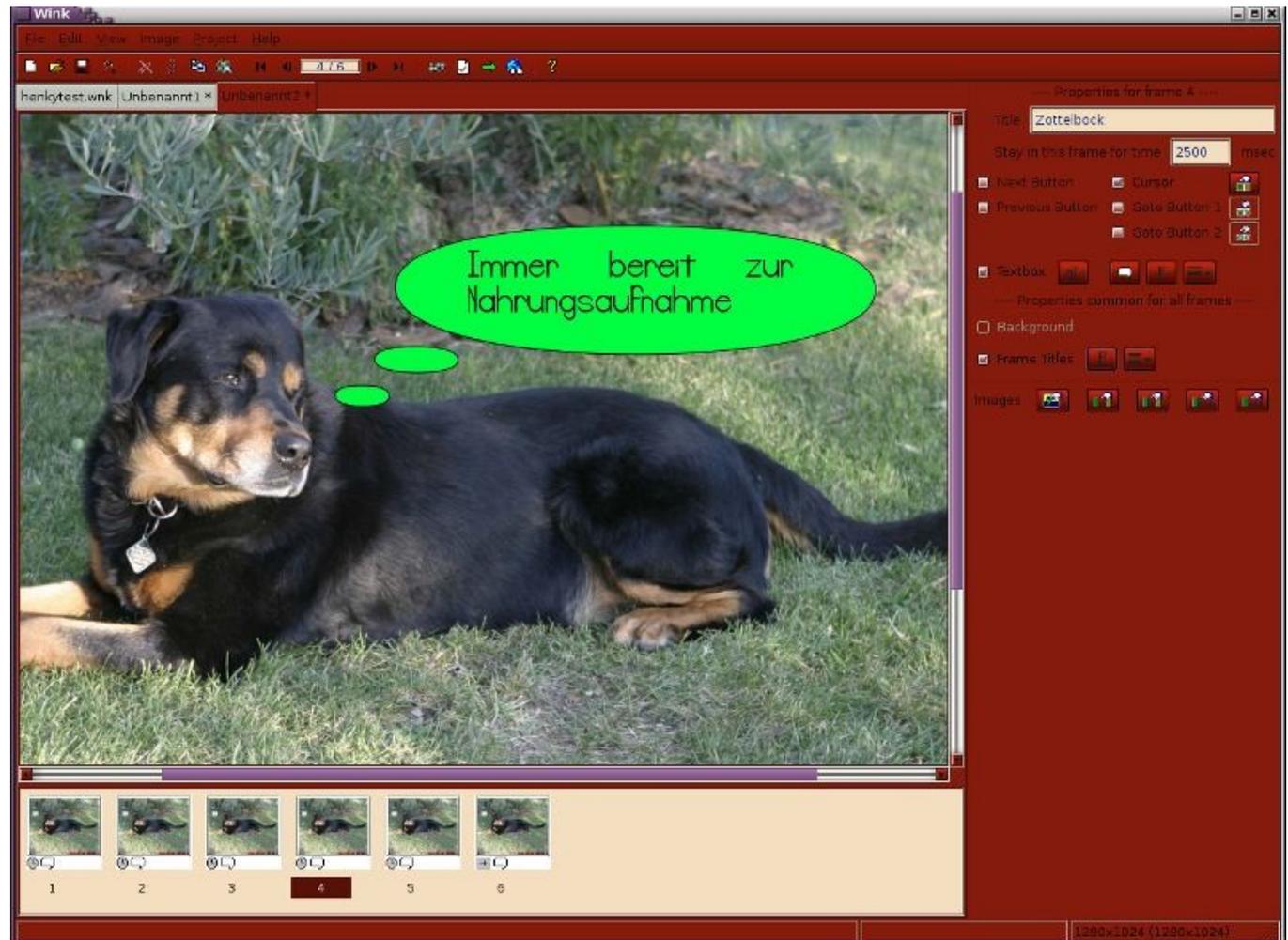
Der Vollständigkeit halber soll dieses Thema kurz angerissen werden. Linux-Distributionen mit KDE- oder GNOME-Desktop haben von Haus aus Programme an Bord, um Bildschirmfotos zu erstellen. Bei Ubuntu genügt hierfür die [Druck]-Taste. Nach deren Betätigung öffnet sich ein Fenster, in dem der fotografierte Bereich links im Bild dargestellt wird. Man kann noch einen Namen vergeben und einen Speicherort wählen und das war es schon.

Komfortabler ist das Tool *KSnapshot*, das mit der Tastenkombination [Alt]+[F2] und der Eingabe von *ksnapshot* gestartet wird. Neben der Möglichkeit, das Bild zu speichern, kann man es auch direkt ausdrucken lassen. Falls man noch etwas verändern möchte bevor das Bild ausgelöst wird, kann man das Bild mit einer Zeitverzögerung von bis zu 99 Sekunden aufnehmen lassen. Möchte man nicht den kompletten Bildschirm ablichten, so kann man im *capture mode* noch verschiedene Bereiche auswählen.

Auch Wink bietet die Funktion eines einzelnen Snapshots an. Ausgelöst wird er mit Hilfe der Taste [Pause].

### Das Programm

Bei Wink [1] handelt es sich um Software, die kostenlos zur privaten Nutzung angeboten wird; sie unterliegt jedoch keiner Lizenzierung, die sie



Ein Tutorial als PDF

HTML-Seite. Das Ganze kann mit Erläuterungen (Callouts) versehen werden. Eine Audio-Aufnahme, die einen Film während des Abspielens begleitet, ist leider nicht möglich, lediglich verbale Erläuterungen zu einzelnen Bildern können eingespielt werden.

Um an das Programm zu gelangen gibt es mehrere Wege. Die aktuelle Version für Linux, 1.5 build 1060, findet man sowohl in den Ubuntu-Repositories als auch zum Download von der Website des Projekts. In der Wink-Hilfe finden sich zwei Tutorial-Projekte, die die Funktionsweise der Software erläutern.

Um sich mit anderen Wink-Nutzern auszutauschen, existiert ein Forum [3], Tipps und Tricks finden sich auf dieser Website [4].

### Die ersten Schritte

Wer mag, kann nach dem ersten Start von Wink unter *File – Choose Language* Deutsch als Sprache einstellen. Da das beim Autor leider nicht funktioniert hat, gehen wir im Folgenden von der englischen Spracheinstellung aus.

Mit der Taste [F4] ruft man einen Dialog zur Feineinstellung von Wink auf, dem man aber anfänglich keine Beachtung schenken muss, denn die Voreinstellungen sind für unsere Zwecke in der Regel geeignet.

Bevor man mit der Aufzeichnung eines Tutorials beginnt, sollte man sich allerdings erst einmal

Gedanken darüber machen, einige Notizen schaden nicht.

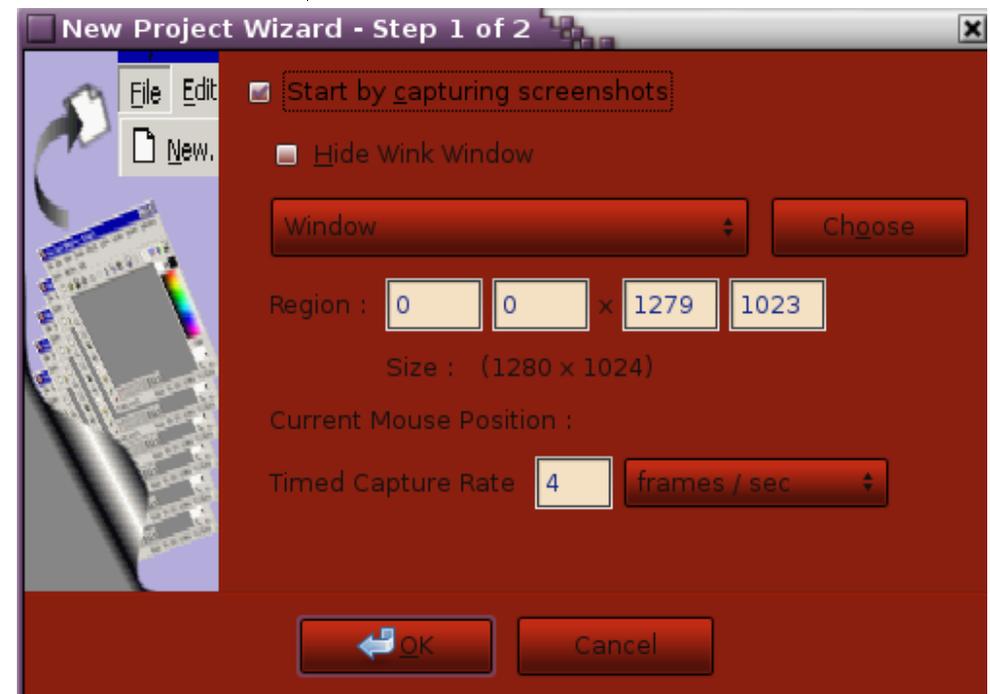
Zunächst sollte man die Bildschirmauflösung derjenigen bedenken, die sich das Werk später anschauen sollen. Verfügen sie über eine geringere Auflösung als die, in der das Tutorial erstellt wurde, wird ihnen lediglich ein Teilbereich des Videos gezeigt werden. Dann sollte man sich darüber im Klaren sein, ob nur ein einzelnes Fenster oder der gesamte Bildschirm angezeigt werden soll. Die Anzeige des kompletten Bildschirms ermöglicht es, auch während der Aufnahme Fenster zu öffnen und zu schließen, die größer als das Original-Fenster sind und bei der Aufzeichnung eines einzelnen Fensters abgeschnitten würden. Je genauer man vorher weiß was man darstellen möchte und wie die einzelnen Abläufe sind, desto eher werden falsche Mausbewegungen oder gar das Stocken des gesamten Ablaufs vermieden.

Vor Beginn einer Session unter Linux ist es wichtig, den Ziffernblock der Tastatur abzuschalten – Wink arbeitet ansonsten nicht.

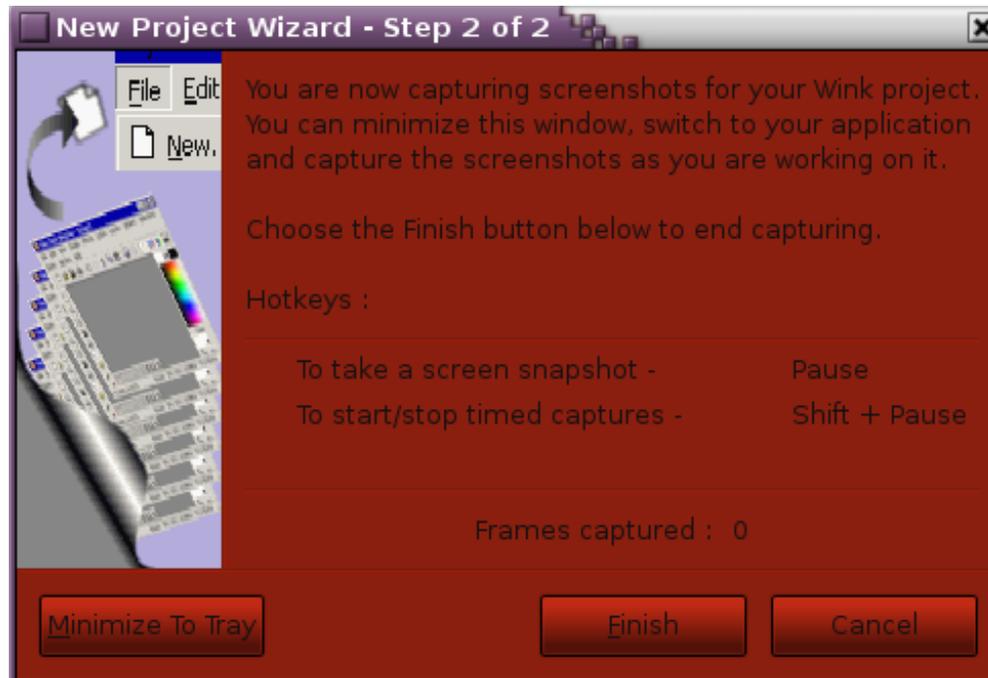
Um ein neues Projekt zu starten, klickt man in der

Menüleiste *File – New* an und öffnet damit ein Fenster 1, in dem einige Einstellungsmöglichkeiten angeboten werden. Das Häkchen im Feld *Start by capturing screenshots* belassen wir. Wink beginnt dann ohne Zeitversatz mit der Aufnahme, sobald diese gestartet wird.

Um den kompletten Bildschirm-Inhalt aufzuzeichnen, wählt man *Screen* als Objekt. Ein Eintrag in *Region* bleibt damit erspart. In den Feldern rechts wird die Bildschirmauflösung dargestellt. Falls nur ein einzelnes Fenster für die Aufnahme vorgesehen ist, wählt man *Window* und klickt auf den *Choose*-Button. Danach wird der Mauszeiger auf



Die Details für die Aufnahme werden festgelegt



Wer mit der Maus arbeiten möchte, entscheidet sich für »Minimize To Tray«

die obere Fensterleiste des aufzuzeichnenden Fensters geführt und darauf geklickt. Im Bereich *Region* sieht man, dass Wink die Koordinaten des Fensters gespeichert hat.

Die Anzahl der Aufnahmen pro Zeiteinheit, die *Timed Capture Rate*, kann in Aufnahmen pro Sekunde, Minute oder Stunde eingeteilt werden, wobei wir zur Erzeugung eines Flash-Videos die Einstellung bei frames / sec belassen können. Zu Testzwecken kann man zunächst die Einstellung von 4 Bildern pro Sekunde übernehmen. Wichtig zu wissen ist hierbei, dass sich die Datenmenge

analog zur Bildrate pro Zeiteinheit verhält. Erhöht man beispielsweise die Anzahl der frames/sec von 4 auf 8, so verdoppelt sich damit auch die Datenmenge.

Soll ein PDF gestaltet werden, so wird frames / minute die bessere Wahl dafür sein. Da man das PDF in einem übersichtlichen Rahmen halten möchte, soll es nicht zu umfangreich werden, außerdem vermeidet man Hektik während der Aufzeichnung.

Ist man mit seinen Einstellungen zufrieden, wird zuletzt ein Häkchen in das Feld *Hide Wink Window* gesetzt, um dafür zu sorgen, dass das Wink-Fenster während der Aufnahme unsichtbar ist. Danach klickt man auf OK.

Es öffnet sich Fenster 2, das mit der Steuerung der Aufnahme per Hotkeys vertraut macht. Um lediglich einen Screenshot zu erzeugen, genügt ein Druck auf die Taste [Pause]. Um eine Bildfolge zu starten und zu beenden bedient man sich der Tastenkombination [Umschalt]+[Pause]. Nach

einem Klick auf *Finish* ist das Programm aufnahmebereit.

Wem die Arbeit mit der Maus besser gefällt, klickt statt auf *Finish* auf *Minimize To Tray* und sieht anschließend das Wink-Symbol in der Menüleiste. Mit Hilfe der rechten Maustaste öffnet sich ein Fenster, das die Steuerung von Wink ermöglicht. Mit *Start Timed Capture* beginnt, mit *Stop Timed Capture* beendet man die Aufnahme.

Um sich mit dem Programm vertraut zu machen, kann man ein erstes Video erstellen, indem man einfach einige Kreise mit der Maus auf seinem Desktop zieht, einige Fenster öffnet, verkleinert, Dinge markiert und tut, was einem sonst noch so einfällt.

Ist man damit fertig, wird die Aufzeichnung wie oben erläutert beendet.

### Arbeiten mit Wink

Nachdem die Aufnahme erstellt wurde, klickt man auf *Restore Capture Window*, und in einem Fenster wird angezeigt, wie viele Bilder erzeugt wurden. Mit einem Klick auf *Finish* wird Wink dazu veranlasst, den Aufzeichnungsvorgang zu beenden. Je nach Anzahl der erzeugten Einzelbilder kann das einen Moment dauern.

Im Hauptfenster zeigt Wink das erste Bild der Sequenz an. Am unteren Rand sieht man die Reihenfolge aller erstellten Einzelbilder seines Videos.

Rechts im Fenster befinden sich die Manipulationswerkzeuge, um Kommentare einzufügen oder Zeitabläufe zu steuern.

Wer lediglich ein unkommentiertes Video haben möchte, speichert es unter *File – Save As* und rendert es anschließend mit Tastendruck auf [F7]. Zur Darstellung wird der Standard-Webbrowser verwendet.

Meistens wird es aber nötig sein, noch etwas an dem Werk zu feilen – und Wink bietet dafür eine umfangreiche Werkzeugkiste an.

Beginnen wir mit einer Einzelbilddarstellung, wie sie beispielsweise auch zur Erstellung eines PDF genutzt werden kann. Zunächst muss man entscheiden, ob die Produktion als Film automatisch ablaufen, ein Einzelbild jeweils für eine ihm zugemessene Zeitspanne gezeigt werden, oder ob der Betrachter per Pfeiltasten durch die Darbietung navigieren soll. In der Grundeinstellung geht Wink davon aus, dass die Pfeiltasten genutzt werden sollen. Diese Darstellungsform ist bei komplizierteren Abläufen recht nützlich, bei denen der Betrachter die Möglichkeit haben soll, in der Präsentation auch wieder zurück zu gehen, ohne sich das Ganze noch einmal von Anfang an betrachten zu müssen. Wählt man also keine Steuerungsoption aus, werden automatisch Pfeile eingearbeitet.

Möchte man ein Einzelbild für eine bestimmte Zeit zeigen, so trage man die gewünschte Dauer in das

Feld *Stay in this frame for time – msec* in Millisekunden ein. Der Wert 1000 steht hier also für eine Sekunde. So kann man jedem Bild je nach Informationswert eine individuelle Zeitspanne zuordnen.

Setzt man ein Häkchen in das Kästchen vor *Textbox*, erscheint ein »Callout« genanntes, mit einem Pfeil versehenes Feld, in das ein erklärender Text geschrieben werden kann. Um es zu beschreiben genügt ein Doppelklick in das Feld oder auf das Kästchen rechts neben *Textbox*. Neben vielen verschiedenen Schriftarten kann man die Schriftgröße, die Schriftfarbe, Fett, Kursiv und Unterstrichen wählen.

Höhe und Breite des Callouts bestimmt man durch Ziehen mit der Maus, ebenso den Ort, an dem das Feld platziert werden soll. Auch Form und Farbe, die Richtung der Pfeile oder die Formatierung des Textes können je nach Geschmack verändert werden. Einfach ausprobieren – alle Änderungen können vor dem Speichern wieder rückgängig gemacht werden.

Ist man mit seiner Arbeit zufrieden, so kann sie nach dem Speichern nun unter *File* als PDF, HTML oder Postscript zur direkten Druckausgabe exportiert werden. Wer die Ausgabe als Flash-Video bevorzugt, rendert den Film (Taste F7) und schaut sich das fertige Produkt in seinem Webbrowser an.

## Fazit

Ein feines Programm, um Vorgänge auf dem Bildschirm darzustellen und sie zu erläutern zu können. Optimal wäre es, wenn man während des Aufnahmevorgangs noch verbal erklären könnte, was bei den Darstellungen besonders wichtig ist, aber man kann ja den Film auch noch nachträglich mit einer Tonspur versehen.

Jürgen Weidner

[joschi@yalmagazine.org](mailto:joschi@yalmagazine.org)

## Informationen

- [1] <http://www.debugmode.com/wink/>
- [2] <http://www.gnu.org/>
- [3] <http://www.debugmode.com/newforums/>
- [4] <http://groups.google.com/group/wink-discuss/web/how-tos-tips-tricks>

# Schlussbemerkungen

Yalm ist ein privates, nichtkommerzielles Projekt. Die Zeitschrift erscheint am dritten Sonntag eines Monats.

Rückmeldungen zu unserem Magazin – seien es Artikelwünsche, Verbesserungsvorschläge, Lob oder auch Kritik – sind herzlich willkommen. Schreibt einfach an [redaktion@yalmagazine.org](mailto:redaktion@yalmagazine.org) oder postet in unserem Forum auf <http://www.yalmagazine.org/forum> (Login: »Leser01« bis »Leser05«, Passwort: yalmleser). Ein **Bugmenot-Account** ist ebenfalls verfügbar.

Wir suchen engagierte und zuverlässige Helfer, die bei unserem Magazin mitarbeiten wollen. Nicht nur Layouter mit guten OpenOffice-Kenntnissen und natürlich Autoren sind gerne gesehen, sondern auch Programmierer und Entwickler sind herzlich eingeladen, bei Yalm mitzumachen. Schreibt uns bei Interesse bitte eine E-Mail an [redaktion@yalmagazine.org](mailto:redaktion@yalmagazine.org) oder seht euch für weitere Details die Rubrik »Mitmachen« auf unserer Homepage an.

## Layout

Die PDF-Ausgabe von Yalm wird mit OpenOffice 3.0.1 erstellt; als Redaktionssystem und für die HTML-Ausgabe verwenden wir **Dokuwiki**. Die jeweils gültige Dokumentvorlage kann von der Yalm-Homepage (Rubrik »Extras«) **heruntergeladen** werden.

## Listings und weiterführende Informationen

Layoutbedingte Zeilenumbrüche werden mit einem Pfeil → dargestellt. Eventuell notwendige Leerzeichen stehen vor diesem Pfeil.

Weiterführende Informationen, Listings und Dateien zu einzelnen Artikeln werden bei Bedarf in der Rubrik »Extras« der Yalm-Homepage zum Download angeboten.

## An dieser Ausgabe haben mitgewirkt:

Bernhard Posselt (Admin, Autor)  
 Daniel Vigano (Autor, Korrektur)  
 Frank Brungräber (Layout, Korrektur, Autor)  
 Heiko Andresen (Korrektur)  
 Jürgen Weidner (Autor, Korrektur)  
 Mario Fuest (Autor, Korrektur)  
 Maximilian Trescher (Korrektur)  
 Peter Majmesku (Autor, Korrektur)  
 Ralf Hersel (Autor)  
 Stefan Zaun (Korrektur)

## Lizenz

Yalm wird unter der **Creative Commons Namensnennung - Weitergabe unter gleichen Bedingungen 3.0 Deutschland Lizenz** veröffentlicht.



Kurz: Yalm-Ausgaben oder einzelne Artikel dürfen kopiert, verbreitet und öffentlich zugänglich gemacht werden; die Inhalte dürfen abgewandelt und bearbeitet werden. Voraussetzung hierfür ist, dass sowohl der Autor als auch Yalm genannt werden und die Weitergabe unter den gleichen Lizenzbedingungen erfolgt.

## Redaktion und Homepage

Kontakt: [redaktion@yalmagazine.org](mailto:redaktion@yalmagazine.org)  
 Yalm-Homepage: <http://www.yalmagazine.org>  
 V.i.S.d.P.: Tobias Kündig  
 Sagenblickweg 6  
 CH-6030 Ebikon  
[tobias@yalmagazine.org](mailto:tobias@yalmagazine.org)

## Bildquellen

Die Inhaber der Bildrechte werden in den Bildunterschriften oder in den Artikelinformationen genannt. Für den Fall, dass die Verwendung eines Bildes nicht zulässig oder gewünscht ist, bitten wir um eine kurze Information; wir werden es dann umgehend entfernen. Quellen der auf der Titelseite verwendeten Grafiken folgen:

Das **GIMP-Logo** wurde der **Open Clip Art Library** entnommen und ist gemeinfrei.

Das Wink-Logo wurde vom Urheber unter der GPL und der Creative Commons-Lizenz (CC BY SA) veröffentlicht; es stammt aus **Wikimedia Commons**.

Die Verwendung des Moblin-Logos erfolgt mit freundlicher Genehmigung von Mike Shaver, Webmaster von **moblin.org**.

*Yalm 07/2009 erscheint  
am 19. Juli 2009*

Yalmagazine.org wird von **NETzor.de** gehostet.