

XMover

Version 1.0.4

© 2001 **b.comp** GmbH, Mainz

Das vorliegende Werk einschließlich aller Abbildungen ist urheberrechtlich geschützt. Auch die Rechte der Wiedergabe durch Vortrag Funk und Fernsehen sind vorbehalten. Jede Verwertung außerhalb der Grenzen des Urheberrechtsgesetzes ist ohne die Zustimmung der **b.comp** GmbH unzulässig und strafbar. Dies gilt insbesondere für Übersetzungen, Mikroverfilmungen und die Bearbeitung in elektronischen Systemen.

Die in diesem Handbuch erwähnten Soft- und Hardwarebezeichnungen sind in den meisten Fällen auch eingetragene Warenzeichen und unterliegen als solche den gesetzlichen Bestimmungen.

Inhaltsverzeichnis

1	Einleitung	4
2	Installation	4
2.1	Installation ohne Registrierung	4
2.2	Installation mit Registrierung	5
2.3	Mehrere XMover installieren	5
2.4	Autostart Installation	5
3	Das XMover Fenster	6
3.1	Die Menüzeile	6
3.1.1	File / Set Config File...	6
3.1.2	File / Quit	7
3.1.3	Action / Stop	7
3.1.4	Action / Start	7
3.1.5	Help / Info...	7
3.2	Das Log Fenster	7
3.3	Die Fortschrittsanzeige	8
3.4	Die Statuszeile	8
3.5	Die Start/Stopp Amatur	8
4	Die Konfigurationsdatei	8
4.1	Syntax	8
4.1.1	Abschnittsmarken	9

4.1.2	Parameterdefinitionen	9
4.1.3	Kommandos	9
4.2	Der globale Abschnitt	10
4.3	Eingangsabschnitte.	10
4.4	Parameter.	10
4.5	Kommandos	12
4.5.1	Bedingte Kommandos	12
4.5.2	Kommando Patterns	13
4.5.3	Das Kommando COPY	14
4.5.4	Das Kommando MOVE	14
4.5.5	Das Kommando DELETE	14
4.5.6	Das Kommando EXIT	15
4.5.7	Das Kommando CLEAR	15
A	Copyright Hinweise	16
A.1	regex	16

1 Einleitung

Der **XMover** ist eine MS Windows basierte Applikation zur automatisierten Übertragung von Dateien. Über eine Konfigurationsdatei gesteuert überwacht der **XMover** ein oder mehrere Eingangsordner. Dateien, die in diesen Ordnern erzeugt werden und auf eine der konfigurierten Namenskonventionen passen, können automatisch kopiert, verschoben, gelöscht oder ignoriert werden. Beim Verschieben oder Kopieren der Dateien können auch in begrenztem Umfang Änderungen am Dateinamen vorgenommen werden.

Der **XMover** ist Teil des *TruPageWebViewer* oder *TruPageViewer* Softwarepaketes und kann auf einer beliebigen Anzahl von Rechnern zur Anbindung des TruPage Servers verwendet werden.

2 Installation

Der **XMover** benötigt zur Ausführung nur die Programmdatei `XMover . EXE` und eine Konfigurationsdatei. Die Konfigurationsdatei kann über die Windows Registrierung fest eingestellt werden. Falls in der Registrierung kein passender Eintrag gefunden wird, wird die Konfigurationsdatei im Arbeitsordner des **XMover** Programms mit dem Dateinamen `XMover . cfg` gesucht.

2.1 Installation ohne Registrierung

Die einfachste Möglichkeit den **XMover** zu installieren besteht darin, die Programmdatei `XMover . EXE` und die passende Konfigurationsdatei `XMover . cfg` in einen Ordner auf dem Installationsrechner zu kopieren und die Programmdatei von diesem Ordner aus aufzurufen.

Hinweis: Wenn Sie den **XMover** auf diese Weise installieren, sollten Sie sicherstellen, dass keine Konfigurationsdatei über die Windows Registrierung konfiguriert ist. Achten Sie darauf, dass der **XMover** beim Start die folgende Meldung ausgibt:

WARNING: missing/bad configuration in registry, using defaults.

Wenn diese Meldung beim Start ausbleibt, so ist eine Konfigurationsdatei über die Windows Registrierung festgelegt. In einem solchen Fall sollten Sie den Eintrag in der Registrierung löschen oder anpassen.

2.2 Installation mit Registrierung

In diesem Fall kopieren Sie die Programmdatei auf den Installationsrechner und spezifizieren den Pfad zur Konfigurationsdatei über den Windows Registrierungseditor. Der entsprechende Wert in der Registrierung wird unter folgendem Pfad erwartet:

```
HKEY_CURRENT_USER\Software\b.comp\XMover\1.0\Preferences
```

Der Wert sollte als Zeichenkette unter dem Namen ConfigFilename abgelegt sein.

2.3 Mehrere XMover installieren

Es können problemlos mehrere Instanzen des **XMover** auf einem System installiert werden und auch gleichzeitig laufen. Die empfohlene Methode hierfür ist die Installation ohne Registrierung. Für jede weitere Instanz kann nun ein Ordner mit einer Konfigurationsdatei angelegt werden. Um die einzelnen Instanzen zu realisieren, erstellen Sie jeweils eine Verknüpfung mit der **XMover** Programmdatei und tragen den Ordner mit der jeweiligen Konfigurationsdatei in das Feld "Ausführen in" ein (Sie erreichen diese Einstellung über den Eintrag "Eigenschaften" im Kontextmenü der Verknüpfung).

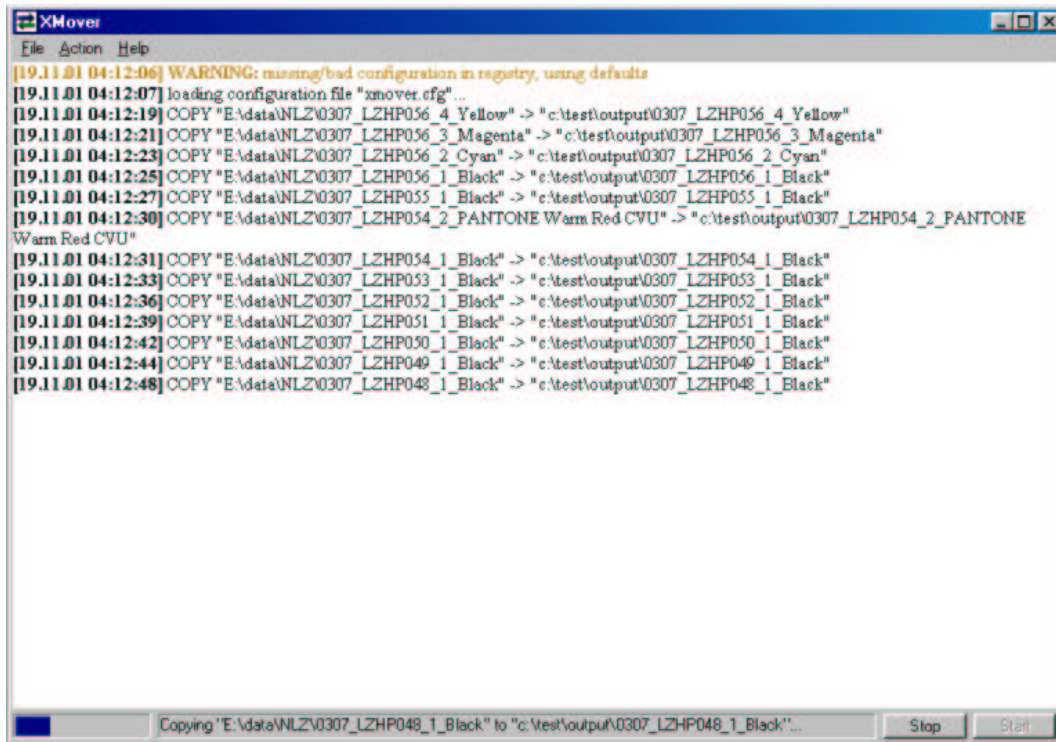
Bei einer Installation mit Registrierung ist die Verwendung mehrerer Instanzen nur dann möglich, wenn die in der Registrierung eingetragene Konfigurationsdatei über einen relativen Pfad spezifiziert ist.

2.4 Autostart Installation

In den meisten Produktionsumgebungen ist es sinnvoll, alle relevanten Instanzen des **XMover** über den "Autostart" Ordner beim Systemstart automatisch auszuführen. Der **XMover** kann problemlos über eine Verknüpfung im "Autostart" Ordner ausgeführt werden. Sollten mehrere Instanzen des **XMover** gestartet werden, so müssen die Felder "Ausführen in" in den Verknüpfungseigenschaften entsprechend angepasst werden.

3 Das XMover Fenster

Nach dem Start zeigt der **XMover** das Hauptfenster an. Das Hauptfenster besteht aus einer Menüleiste, dem Log View, der Fortschrittsanzeige, einer Statuszeile und der Start/Stopp Amatur.



3.1 Die Menüzeile

Alle Operationen des **XMover** können über die Menüzeile gesteuert werden.

3.1.1 File / Set Config File...

Über diesen Menüeintrag kann die Konfiguration des **XMover** geändert werden. Die Änderung ist nur für die laufende Instanz des **XMover** relevant und geht nach einem Neustart verloren. Dieser Menüpunkt ist geeignet, um eine neue Konfigurationsdatei zu testen, bevor die eigentliche Konfigurationsdatei geändert wird.

Hinweis: Wenn sich der **XMover** gerade in einer Kopieroperation befindet, so werden zuerst alle Dateien bearbeitet, die zum Zeitpunkt der Änderung in den Eingangsordnern lagen. Erst danach wird die neue Konfiguration geladen und aktiviert.

3.1.2 File / Quit

Dieser Menüeintrag beendet den **XMover**.

3.1.3 Action / Stop

Hält den **XMover** an. Der **XMover** kann dann über den Menüeintrag “Action / Start” wieder gestartet werden. Wird der **XMover** über diesen Menüpunkt angehalten während er gerade eine Datei kopiert, so wird die Kopie fertiggestellt bevor der **XMover** angehalten wird.

3.1.4 Action / Start

Über diesen Menüpunkt kann der **XMover** gestartet werden, falls er über den Menüeintrag “Action / Stop” oder aufgrund eines Fehlers angehalten wurde.

3.1.5 Help / Info...

Mit diesem Menüpunkt kann die Versionsnummer des **XMover** ermittelt werden.

3.2 Das Log Fenster

Alle Meldungen des **XMover** werden in das Log-Fenster ausgegeben. Das Log Fenster zeigt zu jedem Zeitpunkt die letzten 1000 Meldungen des **XMover** an. Informative Meldungen werden in Schwarz gesetzt, Warnungen in Gelb und Fehler in Rot. Die meisten Fehler führen dazu, dass der **XMover** angehalten wird. Nachdem die Fehlerursache beseitigt wurde, kann der **XMover** über den Menüpunkt “Action / Start” wieder gestartet werden.

3.3 Die Fortschrittsanzeige

Während einer Kopieroperation zeigt die Fortschrittsanzeige in der linken unteren Ecke des Hauptfensters den Fortgang der Operation an.

3.4 Die Statuszeile

Rechts neben der Fortschrittsanzeige befindet sich die Statuszeile. Die Statuszeile zeigt an, welche Operation gerade durchgeführt wird. Wenn keine Operation im Gange ist, bleibt die Statuszeile leer.

3.5 Die Start/Stopp Amatur

Alternativ zu den Menüeinträgen “Action / Start” und “Action / Stop” können auch die Knöpfe “Start” und “Stop” in der rechten unteren Ecke des Hauptfensters verwendet werden.

4 Die Konfigurationsdatei

Die Konfigurationsdatei des XMover ist in Abschnitte gegliedert. Jeder vom XMover beobachtete Eingangsordner wird in einem separaten Abschnitt konfiguriert. Optional können Konfigurationsparameter auch in einem globalen Abschnitt definiert werden. Parameter im globalen Abschnitt gelten für alle Eingangsordner, sofern der betreffende Parameter nicht dort explizit anders definiert ist.

4.1 Syntax

In der Konfigurationsdatei werden Zeilen ignoriert, welche leer sind, nur aus Leerzeichen bestehen oder mit einem Kommentarzeichen beginnen. Als Kommentarzeichen können die Zeichen “;” (Semikolon), “#” (Raute) oder “!” (Ausrufezeichen) verwendet werden. Beispiel:

```
; Dies ist ein Kommentar.
```

```
# Dies ist auch ein Kommentar.
```

```
! Dies und die voranstehende Leerzeile ebenfalls.
```

4.1.1 Abschnittsmarken

Eine Abschnittsmarke markiert den Anfang eines Abschnitts. Ein Abschnitt erstreckt sich von der Abschnittsmarke bis zur nächsten Abschnittsmarke oder bis zum Ende der Datei. Eine Abschnittsmarke markiert entweder den globalen Abschnitt oder den Abschnitt eines Eingangsordners.

Der globale Abschnitt wird mit der Marke

[global]

gekennzeichnet. Der Abschnitt eines Eingangsordners wird mit einer Marke der folgenden Form gekennzeichnet:

[<Pfad>]

<Pfad> gibt hierbei den Pfad zu dem konfigurierten Eingangsordner an.

4.1.2 Parameterdefinitionen

Parameter werden in der folgenden Form angegeben:

<Schlüssel> = <Wert>

<Schlüssel> gibt den zu definierenden Parameter an und <Wert> dessen Belegung. Der angegebene <Wert> ist dabei entweder eine Zahl, ein Wahrheitswert ("true" oder "false") oder eine Zeichenkette.

Parameterdefinitionen müssen immer innerhalb eines Abschnitts stehen. Praktisch bedeutet dies, dass eine Parameterdefinitionen nicht vor der ersten Abschnittsmarke stehen darf.

4.1.3 Kommandos

Ein Kommando kann nur in einem Konfigurationsabschnitt eines Eingangsordners stehen. Ein Kommando beginnt immer mit einem Punkt. Eine Erklärung aller gültigen Kommandos wird in §4.5 gegeben.

4.2 Der globale Abschnitt

Im globalen Abschnitt können Konfigurationsparameter angegeben werden, die für alle Eingangsordner gelten. Falls im Abschnitt für einen Eingangsordner einer der globalen Konfigurationsparameter anders definiert ist, so hat diese Definition Vorrang vor der globalen Definition.

4.3 Eingangsabschnitte

Ein Eingangsabschnitt ist ein Abschnitt der einen Eingangsordner konfiguriert. Ein solcher Abschnitt kann Konfigurationsparameter und Kommandos enthalten.

4.4 Parameter

Der **XMover** erkennt die folgenden Parameter:

`overwrite_copy` (boolean)

Gibt an, ob Dateien in Ausgabeordnern bei Kopieroperationen überschrieben werden sollten. `true`: Dateien im Ausgabeordner werden überschrieben; `false`: Dateien in Ausgabeordnern werden nicht überschrieben. (Voreinstellung: `true`)

`overwrite_move` (boolean)

Gibt an, ob Dateien in Ausgabeordnern bei Verschiebeoperationen überschrieben werden sollen. `true`: Dateien in Ausgabeordnern werden gelöscht und durch die Eingabedatei ersetzt; `false`: Dateien in Ausgabeordnern werden nicht ersetzt. (Voreinstellung: `true`)

`delay` (integer)

Gibt an, wie lange der **XMover** zwischen zwei Iterationen auf einem Eingangsordner warten soll. Die Zeit wird in Sekunden angegeben. (Voreinstellung: 10)

`allow_empty` (boolean)

Gibt an, ob leere Eingangsdateien beachtet werden sollen. `true`: Leere Eingangsdateien werden verarbeitet; `false`: leere Eingangsdateien werden ignoriert. (Voreinstellung: `false`)

`min_wait` (integer)

Gibt das Zeitintervall an, für das die Größe einer Eingangsdatei mindestens stabil bleiben muss, bevor die Datei verarbeitet wird. Die Zeit wird in Sekunden angegeben.

(Voreinstellung: 15)

`enabled` (boolean)

Gibt an, ob ein Eingangsordner berücksichtigt werden soll. `true`: Dateien in dem Eingangsordner werden verarbeitet; `false`: Dateien in dem Eingangsordner werden ignoriert.

(Voreinstellung: `true`)

`recursion` (boolean)

Gibt an, ob Dateien in Unterordnern des Eingangsordners verarbeitet werden sollen. `true`: Dateien in Unterordnern werden verarbeitet; `false`: Dateien in Unterordnern werden ignoriert. (Voreinstellung: `false`)

`order` (string)

Gibt an, in welcher Reihenfolge innerhalb einer Iteration auf einem Eingangsordner die Dateien verarbeitet werden sollen. Gültige Werte sind:

`none`

Die Dateien werden in der Reihenfolge verarbeitet, in der sie im Verzeichnis stehen (keine besondere Reihenfolge).

`lex`

Die Dateien werden in lexikographischer Reihenfolge verarbeitet (d.h. alphabetisch geordnet).

`ctime`

Die Dateien werden in der Reihenfolge ihres Erzeugungsdatums verarbeitet (die ältesten Dateien zuerst).

`mtime`

Die Dateien werden in der Reihenfolge des Datums der letzten Änderung verarbeitet (die ältesten Dateien zuerst).

(Voreinstellung: `none`)

`reverse_order` (boolean)

Gibt an, ob die durch den Parameter “order” spezifizierte Reihenfolge umgekehrt werden soll. Dieser Parameter wird nur dann ausgewertet, wenn “order” nicht auf `none` gestellt ist. `true`: Die Dateien werden in umgekehrter Reihenfolge verarbeitet; `false`: Die Dateien werden in der spezifizierten Reihenfolge verarbeitet. (Voreinstellung: `false`)

4.5 Kommandos

Für jeden Eingangsordner kann eine Reihe von Kommandos angegeben werden. In dem konfigurierten Intervall "delay" iteriert der XMover auf dem Eingangsordner. Für jede Datei, die neu im Eingangsordner erscheint und die für mindestens "min_wait" Sekunden ihre Größe nicht geändert hat, werden alle Kommandos in der angegebenen Reihenfolge ausgeführt. Die Reihenfolge, in der die Dateien verarbeitet werden, kann mit den Parametern "order" und "reverse_order" gesteuert werden.

Mit Ausnahme der Kommandos EXIT und CLEAR wird mit jedem Kommando ein regulärer Ausdruck angegeben, der mit dem Namen der Eingangsdatei verglichen wird. Falls der angegebene Ausdruck auf den Dateinamen passt, wird das Kommando ausgeführt. Ansonsten wird das Kommando übersprungen.

4.5.1 Bedingte Kommandos

Es gibt die Möglichkeit, bedingte Kommandos zu formulieren, indem die Bedingung ON SUCCESS oder ON ERROR vorangestellt wird.

`.ON SUCCESS <Kommando>`

Das angegebene <Kommando> wird nur dann ausgeführt, wenn keines der voranstehenden Kommandos einen Fehler verursacht hat. Beispiel:

```
.MOVE ".*\.tif" "z:\\ziel\\\$0"  
.ON SUCCESS EXIT  
.MOVE ".*\.tif" "f:\\fehler\\\$0"
```

Eingangsdateien mit der Endung .tif werden in den Ausgabeordner z:\ziel verschoben. Wenn bei der Operation ein Fehler auftritt, wird die Datei in den Ausgabeordner f:\fehler verschoben.

`.ON ERROR <Kommando>`

Das voranstehende Kommando wird nur dann ausgeführt, wenn eines der voranstehenden Kommandos einen Fehler verursacht hat.

```
.COPY ".*\.tif" "z:\\ziel1\\\$0"  
.ON SUCCESS MOVE ".*\.tif" "z:\\ziel2\\\$0"  
.ON ERROR DELETE ".*\.tif"
```

In diesem Beispiel werden Eingangsdateien mit der Endung `.tif` in den Ausgabeordner `z:\ziel1` kopiert. Nachdem eine der Dateien erfolgreich kopiert wurde, wird sie in den Ausgabeordner `z:\ziel2` verschoben. Wenn bei der Kopieroperation ein Fehler auftritt, so wird die Datei gelöscht.

4.5.2 Kommando Patterns

Unter dem Pattern eines Kommandos versteht man den regulären Ausdruck, der angibt, auf welche Eingangsdateien das Kommando angewendet werden soll. Der Pattern wird durch ein Leerzeichen begrenzt. Falls der Pattern selbst Leerzeichen enthält, so können diese durch einen vorangestellten Gegenschrägstrich maskiert werden. Der Pattern kann auch in doppelte Anführungszeichen gestellt werden, um die begrenzende Wirkung von Leerzeichen aufzuheben.

In einem Pattern können Sonderzeichen mit einem Gegenschrägstrich maskiert werden. Dadurch wird allerdings der Gegenschrägstrich selbst zu einem Sonderzeichen. Soll in einem Pattern ein Pfadtrenner auftauchen, so muss dieser durch zwei aufeinanderfolgende Gegenschrägstriche angegeben werden.

Sonderzeichen in einem Pattern sind alle Zeichen, die in dem regulären Ausdruck nicht für sich selbst stehen, sondern als Steuerzeichen fungieren. Dies sind die Zeichen `.` (Punkt), `*` (Stern), `+` (Plus), `()` (runde Klammern), `|` (Pipe Symbol), `[]` (eckige Klammern), `{ }` (geschweifte Klammern), `^` (Dach), `$` (Dollar).

Wenn ein Kommando neben einem Pattern auch eine Zielangabe enthält, so können Zeichenketten, die auf Teilausdrücke des Patterns passen, in der Zielangabe referenziert werden. Diese Referenz erfolgt durch ein Dollar Zeichen gefolgt von einer Ziffer, die die Nummer des Teilausdrucks angibt. Die Ziffer 0 bezeichnet dabei die gesamte Zeichenkette, auf die der Ausdruck passt.

Hinweis: Pfadangaben in Abschnittsmarken sind keine Patterns, d.h. in einer Abschnittsmarke wird ein einfacher Gegenschrägstrich als Pfadtrenner verwendet.

4.5.3 Das Kommando COPY

Das Kommando COPY hat die folgende Syntax:

```
.COPY <Pattern> <Ziel>
```

Dieses Kommando kopiert die Eingabedatei an den angegebenen Zielort. Die Zielangabe muss ein vollständiger Dateiname sein, die Angabe eines Ordners genügt nicht. Beispiel:

```
.COPY "(.*)\.tif" "z:\\ziel\\$1"
```

Kopiert Dateien mit der Endung `.tif` in den Ausgabeordner `z:\ziel`, wobei die Endung `.tif` abgeschnitten wird.

4.5.4 Das Kommando MOVE

Das Kommando MOVE hat die folgende Syntax:

```
.MOVE <Pattern> <Ziel>
```

Dieses Kommando verschiebt die Eingabedatei an den angegebenen Zielort. Die Zielangabe muss ein vollständiger Dateiname sein, die Angabe eines Ordners genügt nicht.

4.5.5 Das Kommando DELETE

Das Kommando DELETE hat die folgende Syntax:

```
.DELETE <Pattern>
```

Dieses Kommando löscht die Eingabedatei aus dem Eingangsordner.

4.5.6 Das Kommando EXIT

Das Kommando EXIT wird ohne Parameter aufgerufen. Das Kommando bewirkt, dass die Abarbeitung der Kommandoliste des Eingangsordners *für die gerade bearbeitete Datei* abgebrochen wird. Das EXIT Kommando wird typischerweise bedingt formuliert, d.h. mit einem ON SUCCESS oder ON ERROR kombiniert.

4.5.7 Das Kommando CLEAR

Das Kommando CLEAR setzt für die gerade bearbeitete Datei den Fehlerindikator zurück. Typischerweise werden Kommandoblöcke, die unabhängig voneinander operieren sollen, durch ein CLEAR Kommando getrennt. Beispiel:

```
[global]
delay = 10

[e:\eingang]
enable = true
min_wait = 5
.COPY ".*\.ps" "z:\\ziel\\ps1\\\$0"
.COPY ".*\.ps" "z:\\ziel\\ps2\\\$0"
.ON SUCCESS DELETE ".*\.ps"
.ON ERROR MOVE ".*\.ps" "f:\\fehler\\ps\\\$0"
.CLEAR
.COPY ".*\.tif" "z:\\ziel\\tif\\\$0"
.ON SUCCESS DELETE ".*\.tif"
.ON ERROR MOVE ".*\.tif" "f:\\fehler\\tif\\\$0"
```


A Copyright Hinweise

A.1 regex

Die Implementierung des XMover verwendet zur Analyse von regulären Ausdrücken die regex Bibliothek von Henry Spencer. Die regex Bibliothek wurde für die Verwendung mit dem XMover modifiziert. Der Vermerk aus den Quellen:

```
This code is derived from Henry Spencer's regex library.  
Henry's original copyright notice:
```

```
Copyright 1992, 1993, 1994, 1997 Henry Spencer. All rights reserved.  
This software is not subject to any license of the American Telephone  
and Telegraph Company or of the Regents of the University of California.
```

```
Permission is granted to anyone to use this software for any purpose on  
any computer system, and to alter it and redistribute it, subject  
to the following restrictions:
```

1. The author is not responsible for the consequences of use of this software, no matter how awful, even if they arise from flaws in it.
2. The origin of this software must not be misrepresented, either by explicit claim or by omission. Since few users ever read sources, credits must appear in the documentation.
3. Altered versions must be plainly marked as such, and must not be misrepresented as being the original software. Since few users ever read sources, credits must appear in the documentation.
4. This notice may not be removed or altered.