Handbuch zu Mailscan

Oliver Kopp

27. Januar 2002

Inhaltsverzeichnis

Kapitel 1

Allgemeines

Dieses Kapitel der Doku stellt ein Sammelsurium dar, das Aufgrund von Rueckmeldungen der Mailscan-User entstanden ist. Vielleicht wird es einmal zu einer FAQ umgestaltet. Falls DU dich dazu berufen fuehlst, melde dich bitte bei mir!

1.1 Support

FIDOSOFT.MAILSCAN.GER oder per Netmail: `OliverKopp@2:2471/1464`

1.2 Entwicklungsziel von Mailscan

Das Entwicklungsziel von Mailscan ist, eine eierlegende Wollmilchsau fuer Sysops zu werden. Deshalb sind die Configs auch fuer Sysops zugeschnitten. Andere User, die Mailscan 'missbrauchen', um z.B. Files zu hatchen, sollten sich deshalb nicht wundern.

1.3 Besonderheiten bei den diversen Betriebssystemen

1.3.1 Linux

Hier muessen alle Filenames lowercase sein.

1.4 Anmerkungen zum Log

1.4.1 Log-Stile

Binkley-Style Hier wird das Datum in jede Zeile am Anfang hingeschrieben

Frontdoor-Style Das Datum wird bei jedem Lauf von Mailscan am Anfang in das Log geschrieben.

1.4.2 Die Loglevel

+	Gut
-	Net so gut
!	Fehler
?	Unklare Config-Eintraege
D	Debug-Meldungen
S	Statistik
*	Wenn Verzeichnisse nicht vorhanden sind und sie erstellt werden
(Space)	Siehe #
#	Der Coder hatte wieder keine Ahnung, welchen Level er den Log-Eintrag zuordnen sollte

Kapitel 2

Mailscan und Config-Dateien

2.1 Eigene Configs

Mailscan liesst seine Configs (1. Mailscan.Cfg, dann die Configs unter Config) ein und beim Beenden schreibt Mailscan diese in sein Arbeitsverzeichnis (Work). Diese Configs sind auch schon korrigiert. Allerdings kann man von diesen Configs nur die MailRout.Cfg und die FileRout.Cfg wirklich wieder als einzu- lesende Config nehmen. Deshalb kopiert Mailscan selbststaendig diese beiden nach erfolgreichem Schreiben in sein Config-Verzeichnis.

2.1.1 Redundanz

...oder warum sehen die von Mailscan neu geschrieben Configs immer anders aus als ich es konfiguriert hatte?

In den Mailscan-eigenen Configs wird sehr auf Redundanz geachtet. Hat man z.B. als erste Adresse 2:2471/1464 in seiner Mailscan.Cfg unter [Address] eingetragen und in einer Area-Gruppe z.B. folgendes stehen:

```
*DefaultZone 2
TEST.GER Sf:\\mail\\squish\\test.ger -p2:2471/1464 2:2471/1015 1400
```

So wird daraus:

```
TEST.GER Sf:\\mail\\squish\\test.ger 2:2471/1015 1400
```

Warum?

Ganz einfach. `*DefaultZone 2` entfĎllt, da die erste Adresse der Mailscan.Cfg auch aus der Zone 2 ist. Da diese AKA die Main-AKA des Systems ist, wird angenommen, dass jede Gruppe

auch zu dieser Zone gehoert. Sprich: *DefaultZone 2. Deshalb muss dieses nicht mehr explizit da stehen.

Zu Zone 2 gehoert die AKA `2:2471/1464`, welche dann auch die primaere AKA von jedem Echo ist. Also ist die Information `-p2:2471/1464` auch ueberfluessig.

2.2 Fremde Configs

Mailscan erzeugt default-mässig Configs für andere Programme. In den meisten Fällen waerden diese komplett neu erzeugt - also die alten Dateien (falls vorhanden) überschrieben. Deshalb ist hier Vorsicht geboten.

2.3 mailscan.cfg - Internes

Dieser Abschnitt behandelt die Einstellungen, die Mailscan selbst betreffen und nicht das Erzeugen von Configs externer Programme. Dieses wird im Kapitel „mailscan.cfg - Externe Programme“ abgehandelt.

Diese Configurationsdatei ist in sogenannte Sections aufgeteilt. Format: `[<Name>]`.

Kommentare sind moeglich.

Die Reihenfolge der Sections und der Eintraege innerhalb den Sections ist egal.

Im Allgemeinen ist das Format, wenn nichts anderes in dem jeweiligen Abschnitt steht

`<Token> <Eintrag>`

<Token>1 meint bei `[Directories]` z.B. „Config“, „Data“, ...

2.3.1 [Logging]

Hiermit wird das Logfile eingestellt.

File Hier wird der Filename fuer’s Logfile angegeben.

Style Hiermit kannst du den Stil des Loggings festlegen. Entweder „Binkley“ oder „Frontdoor“. Default ist „Binkley“ (also „Style Binkley“)

2.3.2 [Address]

Hier kommen deine eigenen Adressen rein. Hast du in einer Zone mehrere Adressen, so ist die zuerst aufgefuehrte Adresse deine MainAKA. Diese wird z.B. bei der POINT/ZONES-Konfigurierung eines Points benutzt.

2.3.3 [Misc]

SysOpName Dein Name

SystemName Der Name deines Systems

City Die Stadt, in der sich dein System befindet

Origin Default-Origin in erzeugten Mails

Packer Der Default-Packer. z.B. fuer Downlinks.

MaxMsgSize Die maximale Groesse von Nachrichten

EchoTossLog Gibt einen Filenamen an, in dem die Echonamen reingeschrieben werden sollen, in die Mails gepostet wurden.

2.3.4 [Directories]

Die von Mailscan intern benutzten Verzeichnisse

Config Hier sind die Text-Configs von Mailscan drin

Data Hier legt Mailscan seine eigenen Daten ab. Hier sollte man nach Moeglichkeit nicht rumpfuschen.

Text Hier sind die Template-Files fuer das Announcing, die *fix-Antworten, . . . drin

Work Das Arbeitsverzeichnis von Mailscan. Dieses darf nur Mailscan zugewiesen sein, weil Mailscan ab und an auch mal das ganze Verzeichnis incl. Unterverzeichnisse löscht. `C:\` als Arbeitsverzeichnis wäre also tödlich ;)

Inbound_Secure Der Secure-Inbound. Vom Mailer werden hier die ankommenden Files von denjenigen Links abgespeichert, die ein Session-Passwort mit dir vereinbart haben.

Dieser Eintrag wird fuer's Ticken gebraucht. Denn TICs duerfen ja nur von Links mit Sessionpasswort kommen.

Inbound_UnSecure Der Insecure-Inbound. Vom Mailer werden hier die Files von den Links ohne Sessionpasswort bei dir abgespeichert.

Dieser Eintrag ist nur mal so drin. Er wird von Mailscan noch nicht gebraucht.

Outbound Der Outbound-Pfad. Dieser Configeintrag hat ein besonderes Format (s.u.)

OutBound

Format: `OUTBOUND [<FIPS|BINKLEY|FRONTDOOR>] <Pfad>`

Mailscan kann drei OutBound-Arten: BINKLEY, FIPS und FRONTDOOR. Wie diese Outbounds jeweils funktionieren, sollte bekannt sein... Default ist BINKLEY.

Den OutBoundStyle „FRONTDOOR“ kann man zwar schon in der Config angeben, allerdings wird er Mailscan-Intern noch nicht unterstützt. Die Files werden dann ins *NIRWANA* geschickt. Also bitte aufpassen.

Config-Beispiele:

- `OUTBOUND BINKLEY C:\BBS\XENIA\OUTBOUND\OUT`
  der gleiche Pfad wie der Xenia-Primaer-Outbound
- `OUTBOUND BINKLEY C:\APOINT12\OUT`
  der gleiche Pfad wie der APoint-Outbound

2.3.5 [DefaultMailAreaSettings]

2.3.6 [DefaultFileAreaSettings]

Die Default-Einstellungen fuer vorhandene Fileareas.

AnnounceAreas In welche Area die neuen Files hineinannounced werden sollen, wenn in der Gruppe nicht etwas anderes angegeben ist

2.3.7 [NewMailAreaSettings]

Path Der Basis-Pfad, unter dem die Area angelegt werden soll

Correct Wie der Filename/Verzeichnisname der Area ermittelt werden soll.

CRC32	Einfach CRC32(AreaName)
Long	Sonderzeichen aus dem AreaName raus

Announce* Falls neue Areas angelegt wurden, wie sie announced werden sollen

2.3.8 [NewFileAreaSettings]

Die Einstellungen fuer neu angelegte FileAreas. Wie bei NewMailAreaSettings

z.B. In welcher MailArea soll stehen, dass neue angelegt wurden.

2.3.9 [Linux]

Hier werden die MAPs fuer Laufwerke/Pfade eingtragen. Syntax aehnlich FTRANS von BinkD:

```
MAP f: /var/spool/fido
MAP z: /filebase
MAP e:\\temp /tmp
```

usw. - der erste passende Eintrag wird genommen.

2.3.10 [PassThroughTossing]

Da Squish seine Muskeln erst dann zeigt, wenn er alle Echomailareas passthrough durchtosst, gibt es die Moeglichkeit eine solche Config fuer Squish zu generieren. Alle Echos, die in der MailRout.Cfg *nicht* passthrough sind, werden hier (Schluesselwort EchoRoutCfg in Section [Squish]) trotzdem passthrough eingetragen, aber zusaetzlich dem Point, der bei `Point <Nr>` angegeben ist, gelinked.

Die erzeugten Mailpakete kann man z.B. einem FastEcho, IMail, CrossPoint, ... dann zu „fressen“ geben, wenn man die Mails auch wirklich lesen moechte. (Oder z.B. 1x pro Stunde, wie es hier eingerichtet ist).

Es ist geplant, dass Mailscan die Configs fuer manche Tosser, die diese Pakete bekommen auch erzeugt, so dass man seine Echos nicht doppelt anlegen muss bzw. zwei Tosser-Configs verwalten muss.

Directory In dieses Verzeichnis werden die Dateien geschrieben

Point Die Pointnummer, an die die ganzen Echos passthrough geschickt werden sollen.

Tosser hpt oder Squish

TosserOS Funktionsweise wie OS bei „externen Programmen“

NewAreaDirectory (nur bei Tosser hpt). Hier sollen die neuen Areas angelegt werden. Bei diesem Verzeichnis findet keine Überpruefung statt, da es z.B. auch fuer TosserOS lnx geschrieben wird.

2.3.11 [FILES.BBS]

Legt das Format der FILES.BBS fest. Jede darin aufgelistete Datei muss einen Punkt enthalten, da er sonst als Beschreibung der letzten Datei erkannt wird.

Bei Maximus werden mehrzeilige Beschreibungen an 31 Leerzeichen, bei Pro-Board an dem Plus als erstes Zeichen erkannt.

BBS-System Kann entweder Maximus oder Pro-Board sein.

TrueLongName Bisher ohne Funktion

2.3.12 [Kill]

Falls ein Echo geloescht wird (z.B. durch MAILSCAN KILL <Echotag>), dann werden diese auch oeffentlich aufgefuehrt.

ANNOUNCE* wie bei allen Announce-Sachen

TPL_KILLPUB Das zu verwendente Template-File. Default = killpub.txt

2.4 mailscan.cfg - externe Programme

Von Mailscan werden auch externe Programme unterstuetzt. Es werden die Configs dieser Programme geschrieben. Wenn nichts anderes angegeben ist, werden die angegebenen Configs gnadenlos überschrieben. Also aufpassen!

2.4.1 Allgemeines Format

Wenn nichts anderes in dem jeweiligen Doku-Abschnitt steht, gilt folgendes: Zuerst kommt der Directory-Eintrag, danach die Programm-Spezifischen Sachen.

Directory Hier wird das Verzeichnis angegeben, in dem sich das externe Programm befindet

OS Nur bei manchen unterstuetzt.
Hier gibt man das Betriebssystem an (w32, OS2 oder lnx). Bisher hat das nur Auswirkungen auf das Format der erzeugten Textdateien. Bei non-Unix-Betriebs- systemen wird ein CRLF geschrieben und bei Unix nur CR.

2.4.2 [Ticker]

Ticker ist hier in der Doku als externes Programm aufgefuehrt, da es programmintern ein eigenes Modul ist und von der Doku-Struktur besser so reinpasst.

Hier ist kein Directory-Eintrag moeglich, da FluppTic ein internes Modul von Mailscan ist.

BadFiles Wenn beim Einticken das TIC nicht einsortiert werden kann, dann kommt es (mit der dazugehörigen Datei (falls möglich)) in dieses Verzeichnis. Es sei denn, man aktiviert gesonderte Verzeichnisse fuer gewisse Fehler.

BadFiles_AreaNotFound Hier werden die BadTics hinverschoben, bei denen die Filearea nicht bekannt ist

BadFiles_FileNotFound Hier werden die BadTics hinverschoben, bei denen das zugehörige File nicht auffindbar ist.

BadFiles_WrongCRC Hier weden die BadTics hinverschoben, bei denen die CRC- oder MD5-Summe nicht in Ordnung ist.

Dupes In dieses Verzeichnis werden die TICs incl. File verschoben, die schon in der Zielarea vorhanden sind.

TicThru TicThru gibt das Verzeichnis an, in dem ausgehende TICs zwischengelagert werden. TICs werden immer KillFileSent verschickt, also kuemmert sich Mailscan nicht um das Loeschen gesendeter TICs, sondern der Mailer.

Ausserdem werden in diesem Verzeichnis diejenigen Files abgelegt, die passthrough durchgeroutet werden.

Falls eines der drei `BadFiles_*`-Verzeichnisse nicht konfiguriert wird, oder kein Verzeichnis angegeben wurde, dann wandern diese BadTics auch in das `BadFiles`-Verzeichnis.

2.4.3 [Squish]

AddressCfg Hier kommen alle Adressen von deinem System rein. Ersetztalso alle Address-Statements in der Squish.Cfg.

EchoRoutCfg Alle Echodefinitionen. Ersetzt also alle Echodefinitionen in der Squish.Cfg.

PackCfg Falls ein User einen anderen Packer als den in Mailscan eingestellten Default-Packer benutzt, wird hier pro AKA `Pack <Packer> <AKA>` reingeschrieben. Ersetzt also alle Pack-Defintionen aus der Squish.Cfg

PktPWDCfg Falls ein User mit dir ein Paketpasswort vereinbahrt hat, wird es hier eingetragen. Ersetzt also alle Password-Statements in der Squish.Cfg

RouteCfg Bitte nur aktivieren, wenn man weiss, was man tut. Diese Config wird auch neu geschrieben.

Hier wird eingetragen, mit welchem Flavour den Links die Mail hingelegt wird. Default ist „normal“ - siehe Users.Cfg

Es werden keine anderen Route-Kommandos eingetragen. Also das sonstige Routing muss man trotzdem „von Hand“ in Squish konfigurieren.

2.4.4 [FastLst]

PasswordCfg Hier werden die Sessionpassworter der Links im Format `Password <Passwort> <AKA>` reingeschrieben. Dieses File kann man auch fuer Binkley benutzen.

2.4.5 [BinkD]

AddressCfg Alle Adressen incl. der dazugehoerigen Domain

DomainCfg Alle Domains incl. Aliasses

PasswordLst Eine Passwort-Liste im T-Mail-Format mit Domain. Kann in BinkD 0.9.4 problemlos eingebunden werden.

Fuer jeden User wird nur das Passwort fuer die erste AKA einer Zone eingetragen.

NodesCfg Eintraege im NODES-Format. Damit wird die PasswordLst ueberfluessig und die User muessen nicht extra nochmals in BinkD eingetragen werden.

2.4.6 [HPT]

Unterstuetze Eintraege: linkscfg, echoscfg, addresscfg

2.4.7 [NODELISTS]

Directory Das Verzeichnis, in dem die Nodelisten zu finden sind, wenn kein anderer Pfad angeben ist. Dieser Pfad gilt auch fuer die in der ZONES.CFG angegebenen Node-listen.

PVTNODELIST Wenn angegeben, dann erzeugt Mailscan eine Nodeliste, in der alle AKAs aller konfigurierten User gelistet sind, die die Pointnummer 0 haben.

PVTPOINTLIST Wie PVTNODELIST, nur fuer AKAs mit Pointnummer ungleich 0. Das Format ist das „Boss,“-Format.

2.4.8 [Areafix-Names]

Namen fuer Areafix-Requests

2.4.9 [Filefix-Names]

Namen fuer Filefix-Requests

2.4.10 [Mailscan-Names]

Namen fuer Mailscan-Requests (z.B. `%BBS MOVE`)

2.5 Users.Cfg

Hier werden die User konfiguriert.

Kommentare sind hinter den Config-Einträgen zwar möglich, jedoch werden sie bei der neuerstellung der USERS.CFG als extra Zeilen aufgeführt.

Die Reihenfolge der Einträge ist nicht egal. Die Schlüsselwörter sollten in der gleichen Reihenfolge benutzt werden, wie sie hier erklärt werden.

Alle Schlüsselwörter mit einem nachgestelltem Stern (*) sind Pflichteinträge.

Name* Der Name des Users. Hiermit faengt ein User-Eintrag an.

FileAreaGroups Steuert den Zugriff auf die Fileareagruppen.

MailAreaGroups Steuert den Zugriff auf die Mailareagruppen.

MailFlavour Der Flavour fuer Echomails

FileFlavour Der Flavour fuer Files

NetMailFlavour Der Flavour fuer Netmails, falls nicht angegeben, wird Mailflavour genommen.

AKA* Eine Definition von einer oder mehreren AKAs. Kann mehrmals vorkommen. Falls dieses Schlüsselwort nicht angegeben wird, muss Zones *und* Point angegeben werden.

Zones Die Zonen, auf die der User Zugriff hat. Benötigt Point-Statement.

Point Die Pointnummer, die der User hat. Benötigt Zones-Statement.

BBS-Name Der Name der BBS

BBS-Num Im Format Länderkennzahl-Nummer, so wie i.d. Nodelist üblich. Also z.B. 49-7141-290192.

BBS-FLAGS Die Flags der Line.
Zu BBS-FLAGS gehoert auch die Speed der BBS. evtl. wird das noch in einer neueren Mailscan-Version geaendert. Aber da Mailscan bisher die Flags nicht auswertet, sondern nur fuer die Nodelisten-Erzeugung braucht, genuegt eine solche Speicherung.

SystemName Hier kommt der Name des kompletten Systems rein. Ist kein SystemName angeben, so wird der Eintrag von BBS-Name genommen.

PWD* Damit wird das *fix-Passwort konfiguriert. Es gilt, falls später nicht anders konfiguriert, auch als TIC- und Sessionpasswort

PKTPWD Konfiguriert ein Paketpasswort. Wird keines angegeben, so wird das Paketpasswort aktiviert und als Passwort das *fix-Passwort genommen. Ansonsten wird kein Paketpasswort aktiviert, sprich: es ist leer.

TICPWD Setzt das Passwort, das in den TIC-Files steht (bei neu erzeugten) bzw. stehen muß (bei ankommenden).

SESSIONPWD Ändert das Session-Passwort (wird für den Mailer benötigt).

OK Definiert, was der User alles darf. Default=gar nix. Folgende Flags können dabei verwendet werden:

AV Areaverwaltung
Der User darf Mail- und Fileareas verwalten, wie z.B. durch Areafixbefehl `~DELETE.GER` eine MailArea loeschen.

FIX *fixe benutzen

FV Fileverwaltung
Der User darf Files verwalten, d.h. z.B. Files von einer Area in eine andere verschieben, Files loeschen, ...

***** alles

FAX Die Fax-Nummer des Users

VOICE Kann mehrmals vorkommen. Die Telefonnummer des Users.

CITY Die Stadt, in der der User wohnt. Wird z.B. in Nodelist-Segmenten verwendet.

PACK Definiert den Packer, womit die Mails/Files gepackt werden. Default ist der Packer, der i.d. MAILSCAN.CFG, Section [MISC] unter Packer angegeben ist. Gueltige Packer sind: LHA, ZIP und RAR. Konfiguriert werden sie über die COMPRESS.CFG.

BinkD Defniert die Zeichenkette, die 1:1 hinter die AKA in der NodesCfg geschrieben wird.

Bsp.:

```
BinkD -crc flupp.dyndns.org
```

oder

```
BinkD nomans.dyns.cx;nomans.dyndns.org
```

MaxMsgSize Die Groesse, die Nachrichten an diesen User maximal haben duerfen

END* Damit wird die Konfiguration eines Users abgeschlossen.

2.5.1 Regelung des Areazugriffs

Wenn ein User eine AKA in der Zone X hat, dann hat er automatisch auch Zugriff auf die File- und Mailgruppen, die zu dieser Zone gehoeren. Konfiguriert wird dieses in der ZONES.CFG.

Ueber FILEGROUPS bzw. MAILGROUPS kann man zusaetzliche File- und Mailgruppen angeben, auf der User Zugriff haben soll. Als Beispiel waeren z.B. XXX-Pix zu nennen, die ueber die Fido-AKA geroutet werden.

Bei *GROUPS (*=MAIL bzw. FILE) kann man entweder Gruppen, die in der *ROUT.CFG angeben sind, oder Divisions, die in *DIVS.CFG angeben sind.

2.5.2 Spezielle Flavours fuer den User

Mit NetMailFlavour, MailFlavour und FileFlavour kann man bestimmte Outbound-Flavours fuer den User global festlegen. Mögliche Flavours sind:

```
<Flavour> = normal|crash|direct|hold|invisible
```

Default ist normal.

Bei „invisible“ wird im Outbound beim Ticken ein `*.hl#`-File erzeugt. Der Mailer wertet dieses File nicht aus. Mit Tools wie z.B. „ChangeStat“ von Wilfried Brinkmann ist es moeglich, dass die User die Files per REQUEST bekommen. ChgStat benennt dann das `*.HL#` so um, dass der User die Sachen bekommt. Naeheres siehe ChgStat.Dok.

Sobald Mailscan seine Configs wesentlich schneller einlesen kann, wird auch Mailscan dieses uebernehmen koennen.

2.5.3 AKA-Definitionen

Mittels AKA

Kurzschreibweise ist erlaubt. Beispiel:

```
AKA 2:2471/1400 1464 1465
```

Hinter jede AKA koennen Flags gesetzt werden, die dann fuer jede AKA, die vor diesem Flag in dieser Zeile kommen, gelten:

***** Haengt man hinter die AKAs ein *, dann bedeutet dieses, dass der User mit diesen AKAs Echos anlegen darf.

z.B.

```
AKA 2:2471/1400
AKA 2:2471/1464 1465 *
```

Mit 2:2471/1400 darf er keine Echos anlegen, aber wohl mit 1464 und 1465.

#<AKA> Damit kann man eine andere Absender-Adresse angeben.

Beispiel:

Man hat selbst die AKAs 1400 (Hub) und 1464 (Nodenummer). MainAKA ist die 1400.

Die Fido-Downlinks bekommen die Mails von 1400, die anderen Links mit 1464. Default-maessig bekommen alle AKAs von deiner eigenen MainAKA die Mails. Also muss man es bei den Nicht-Downlinks ändern...

Nehmen wir an, 2:24/903 ist ein „externer Link“. Also gibst du i.d. Config von ihm einfach `AKA 2:24/903 #2:2471/1464` an und schon bekommt die AKA 24/903 alle Mails von der 1464 statt von der 1400.

Mittels ZONES und POINT

Ist ein User Point bei dir, so ist die Konfiguration nicht ueber AKA-Statements, sondern mittels POINT und ZONES zu machen.

Echos koennen von Points nicht anlegt werden, es sei denn, sie haben zusätzlich eine Node-Nummer (mittels AKA konfiguriert) und die entsprechenden Zugriffsrechte.

Verschiedene Pointnummern je Zone werden nicht unterstuetzt. Mehrere Pointnummern auch nicht. Ist IMHO auch unsinnig.

POINT und ZONES Ist ein User Point bei deinem System und moechte zig AKAs bei dir haben, dann brauchst du dich nicht immer abmuehen, deine eigenen AKAs herauszusuchen und die Pointnummer an- zuhaengen, sondern das macht Mailscan.

z.B.

```
POINT 10
ZONES 2 600
```

Der User ist Point Nr. 10 bei dir und bekommt AKAs im Fido und DreamNet.

Daraus werden die AKAs des Points generiert.

Steht in der Mailscan.Cfg folgendes:

```
[Address]
2:2471/1464
2:2471/1465
600:115/300
```

Dann bekommt der o.g. Point die AKAs 2:2471/1464.10 und 600:115/300.10. Mailscan nimmt also immer die erste konfigurierte AKA einer Zone. vgl. Erklaerung von [Address]

Konfiguration von mehreren Lines

Man ordnet dem User weitere linien zu, in dem man hinter „BBS“ noch die Liniennummer setzt. Dabei müssen nicht alle drei Einträge neu konfiguriert werden, da Mailscan bei Nichtvorhandensein die von der ersten Linie (also die ganz ohne Zahl nach „BBS“).

z.B.

```
BBS2-Name x-tReMe_BBS_Line_2
BBS2-Flags 300,CM,X75
BBS2-Num +49-7141-290840
BBS3-Name x-tReMe_BBS_Line_2
BBS3-Num +49-7141-290850
```

Die Anzahl der Lines ist nicht beschraenkt.

2.6 COMPRESS.CFG

Bitte nach Moeglichkeit Windows- bzw. OS/2-Packer konfigurieren, da DOS- Packer nicht mit langen Dateinamen zurecht kommen.

2.7 MAILROUT.CFG und FILEROUT.CFG - Gemeinsames

Hier werden die Echomailareas bzw. die Fileareas konfiguriert. Die beiden Configs sind im Prinzip gleich.

Deshalb werden hier zuerst die Gemeinsamkeiten erklaert. Dort wo es Sinn macht, wird gleich auf die Unterschiede eingegangen.

2.7.1 Die Gruppen

Eine Gruppendefinition wird mit der Zeile

`Group <Name> ["<Beschreibung>"]`

angefangen und mit

`EndGroup`

beendet. Dazwischen stehen genauere Settings (fangen alle mit * an) und danach die Echodefinitionen.

***DefaultZone** Ordnet die ganze Gruppe der angegebenen Zone zu.

***ForwardingList** Hiermit kann man Forwardlisten einbinden. Format: `<Filename> <AKA>`

Wenn ein Echo auf dem System angelegt werden soll, es aber nicht vorhanden ist, so wird in dieser Liste nachgeschaut, ob es dort drin steht und wenn ja, dann wird es bei `<AKA>` bestellt und mit der Beschreibung in `<Filename>` angelegt.

Es sind pro Gruppe mehrere Eintraege moeglich

z.B. `*ForwardingList f:\txtfiles\forward\fidoger.na 2:24/888`

2.7.2 Announceing von neuen Areas

Defaults sind die Settings aus `[NewMailAreaSettings]` bzw. `[NewFileAreaSettings]` in der Mailscan.Cfg.

2.8 Definition einer Area

Format: `<Areaname> <Pfad> <Flags> <Links> ["<Beschreibung>"]`

2.8.1 Der Areaname

Der Name der Area. Es sind keine Sonderzeichen erlaubt.

2.8.2 Der Pfad

- Bei FileAreas
  Ganz normal, ohne abschliessenden \.

- Bei MailAreas
  `<Messagebasetyp><Pfad>`
  Moegliche Messagebasetypen sind:

F	fuer *.MSG (FTS1)
J	fuer JAM-Base
S	fuer SQUISH-Base

  Der Pfad beinhaltet die messagebasespeziefische Erweiterung *nicht*.

  z.B.

  ```
  Sf:\mail\squish\announce.2471
  ```

  und nicht

  ```
  Sf:\mail\squish\announce.2471.sqd
  ```

  oder

  ```
  Ff:\mail\msg\net
  ```

  und nicht

  ```
  f:\mail\msg\net\*.msg
  ```

2.8.3 Die Flags

Die Defaults fuer die Haltezeiten werden in DefaultFileAreaSettings bzw. DefaultMailAreaSettings in der Mailscan.Cfg konfiguriert.

Allgemeine Flags

-! Forced-Echo
Das Echo darf zwar bestellt werden, aber nicht mehr abbestellt.

-o Passthrough
Die Mails/Files werden nur durchgereicht.

-D<Haltezeit in Tagen> Wie lange die Mails/Files in diesem Echo gehalten werden sollen

-M<Maximale Mails/Files> So viele Mails/Files duerfen maximal in dem Echo sein

-P<andere Absender-AKA> Als Absender-AKA wird nicht diejenige genommen, die normalerweise zu der Zone gehoert, zu dem dieses Echo gehoert, sondern eine andere

-R read-only
Neue Links bekommen das Echo nur Read-Only.

-Z<Zone> Ordnet das Echo einer anderen Zone zu

Flags nur bei FileAras

-A- Deaktiviert das Announcement von neuen Files in dieser Area. Sinnvoll z.B. bei Nodediff-Areas

Flags nur bei MailAreas

-S<Skip> Wieviele Mails beim Aufraeumen uebersprungen werden sollen

2.8.4 Die Links

Allgemeines Format: `[<Linkflags>]<AKA>[,[Linkflags]<Datum>]`

FutureLinkState

Dabei ist der Teil `[,[Linkflags]<Datum>]` der sogenannte „FutureLinkState“. Damit kann man den Status eines Echos für einen User einem gewissen Datum automatisch geändert werden. Nützlich z.B. bei RO-Schaltungen durch MODs.

Allgemeine Linkflags

Default ist „rs“

r receive
: Der Link bekommt Zeugs aus der Area

s send
: Der Link darf auch etwas reinsenden

p passiv
: Der Link ist „passiv“ - die Link-Einstellungen bleiben zwar erhalten, aber er bekommt dennoch nichts.

U Uplink
: Der Link ist ein Uplink. Bis jetzt hat dieses Flag praktisch keine Bedeutung.

  In Zukunft soll es aber so sein: Wenn eine als Uplink gelinkte AKA ein Echo abbestellt, wird es auf dem System automatisch geloescht und bei den Down- links auch automatisch abbestellt und geloescht. (wohl auch konfigurierbar)

Linkflags nur bei FileAreas

Default ist zusaetzlich noch „FT“.

A Der Link bekommt bei neuen Files eine Announcemail.

F Der Link bekommt die Files mitgeschickt

T Der Link bekommt ein TIC-File bei neuen Files. Bekommt er kein File, so wird das TIC-File als `*.TAC` geschickt.

N NoReadWrite
Der Link bekommt weder File noch TIC und darf auch nichts reinsenden. Sinnvoll bei ÄN". Diese AKA bekommt also nur per NM die Information, dass in dieser Area etwas kam, jedoch nicht das File selber.

Linkflags nur bei MailAreas

Gibt’s keine

2.8.5 Die AKAs

Es duerfen nur AKAs aufgefuehrt sein, die auch in der USERS.CFG oder in der MAILSCAN.CFG (als eigene AKAs) konfiguriert sind.

Bei einer Aneinanderreihung kann man die AKAs auch in der Kurzschreibweise hinschreiben.

z.B. aus `2:2471/0 2:2471/1464` wird `2:2471/0 1464`.

2.8.6 Die Beschreibung

Sie muss in Anfuehrungszeichen stehen und darf keine Sonderzeichen (Umlaute hingegen schon) und vor allem keine doppelten Anfuerhungszeichen (`"`) enthalten.

2.9 Besonderheiten der filerout.cfg

2.9.1 Das Announcing

Default sind die Sachen aus `[DefaultFileAreaSettings]` aus der Mailscan.Cfg.

***NoAnnounce** hiermit wird das Announcing fuer die ganze Gruppe deaktiviert. Damit werden die ganzen *Announce...-Sachen unnoetig.

Man kann das Announcing auch fuer einzelne Areas in einer Gruppe mittels dem Flag `-A` deaktiveren. Unten steht mehr zu diesem Thema

***AnnounceEchos** Hier kann man Echos angeben, in die neue Files announced werden sollen, wenn sie nicht in die Echos gepostet werden sollen, die in der Mailscan.Cfg unter `[DefaultFileAreaSettings]`/AnnounceEchos angegeben wurden.

2.10 Besonderheiten der mailrout.cfg

2.10.1 NETAREA, BADAREA, DUPEAREA

Aus historischen Gruenden sind diese in der MailRout.Cfg aufgefuehrt und nicht in der Mailscan.Cfg. Wenn jemand damit ein Problem hat: bitte konstruktive Vorschlaege ;).

BADAREA und DUPEAREA wird von manchen externen Programmen (wie z.B. APoint) nicht unterstuetzt und sind deshalb optional.

NETAREA Definiert eine Netmail-Area. (z.Z. ist nur eine einzige moeglich)

BADAREA Hier kommen die Mails hin, die irgendwie nicht in Ordnung sind.

DUPEAREA Hier kommen die Mails hin, die als Dupe erkannt werden.

2.11 MAILDIVS.CFG und FILEDIVS.CFG

Gruppen kann man auch in sogenannte Divisions zusammenfuegen, damit man es bei der Rechte-Vergabe einfacher hat.

Allgemeines Format:

```
<Divisionname> <Gruppe> [<Gruppe>] [<Gruppe>] [...]
```

Angenommen, man hat die FidoNet-Echos in nationale und internationale Echos auf- geteilt, dann müsste man jedem User, der Zugriff auf das FidoNet hat die Gruppen „FidoNet-Int“ und „FidoNet-Nat“ geben. Wenn er mit seiner Zone-2-AKA Zugriff auf noch mehr Gruppen bekommen soll, kann dieses ganz schnell ausarten. Deshalb kann man die Gruppen unter einer „Oberüberschrift“ (dem Division-Namen) zusammen- fallsen. Dies geht mit

```
FidoNet FidoNet-Int FidoNet-Nat
```

2.12 Zones.Cfg

Die Zones.Cfg dient dazu, zu einzelnen Zonen genauere Informationen zu liefern.

Damit ein User mit einer Zone-2-AKA automatisch Zugriff auf alle FidoNet-Areas (nehmen wir mal an, die Gruppe heisst „FIDONET“) hat, muss man hier in der Zones.Cfg bei Zone 2 als Mail-Gruppe „FidoNet“ angeben. Natürlich kann man auch Divisions angeben.

Beispiel:

```
[Zone 2]
Name FidoNet
Desc Das Netz der Netze
Domain fidonet
DomainAlias fidonet.org
DomainAlias fido
DomainAlias fido.net
DomainAlias fnet
MailGroup FidoNet
FileGroup Misc
```

Bei MailGroup und FileGroup darf auch eine Division stehen.

Kapitel 3

Kommandozeilenparameter

3.1 Hatch

Dient zum Hatchen einer oder mehreren Datei(en).

Wenn sich das File nicht in dem Pfad der Zielarea befindet, dann wird es dorthin verschoben.

Wird keine Beschreibung angegeben, wird sie nach Moeglichkeit aus der `file_id.diz` im Archiv genommen.

3.1.1 nur ein File

```
%HATCH <Filename> <Area> [*<Replaces>] [<shortdesc>] [@<file_id.diz>]
```

z.B.

```
HATCH c:\testfile.zip TESTAREA "ein ganz kleiner Test" @c:\temp\file_id.d
```

oder (ohne eine kurze Beschreibung)

```
HATCH c:\testfile.zip TESTAREA @c:\temp\desc.fle
```

Die Beschreibung wird hier aus der Datei desc.fle genommen.

oder (ganz ohne Beschreibungen)

```
HATCH c:\testfile.zip TESTAREA
```

`*<Replaces>` gibt einen Filenamen an, der das zu ersetzende File angibt. Nuetzlich beim Hatchen von gebugfixten Versionen aelterer Software. Das File wird in der lokalen Filebase erst dann ersetzt, wenn das File auch wirklich gehatched wird.

3.1.2 mehrere Files

Aehnlich wie einem File - Statt <Filename> gibt man eben WildCards an

z.B. `HATCH C:\TOUPLOAD\* UNCHECK`

Gibt man zusaetzlich noch eine ShortDesc und/oder eine `file_id.diz` an, wird diese Beschreibung nur dann genommen, wenn sich aus dem Archiv oder der FILES.BBS (falls das File schon in der Ziel-Area ist und ein FILES.BBS-Eintrag existiert) keine Beschreibung rausziehen laesst, genommen.

3.1.3 NoDescNoHatch

Gibt man als shortdesc NODESCNOHATCH an, wird das File nicht gehatched, wenn keine Beschreibung fuer das File gefunden werden konnte.

d.h. - keine `file_id.diz` und (wenn das File schon im Zielareapfad liegt) kein FILES.BBS-Eintrag.

3.2 TIC

Hiermit wird Mailscan zum TICken veranlasst. Die neuen Files werden nicht gleich announced, sondern erst mit dem Paramter `ANNOUNCE`. So ist es z.B. moeglich, waehrend einer Session die TICs gleich zu verarbeiten, aber erst spaeter, nach der kompletten Mailsession, (der Uebersicht wegen) gesammelt zu announcen.

`*.TIC`-Files werden natuerlich nur im Secure-Inbound gesucht.

3.2.1 Die Reihenfolge der Fehlerbestimmung

Natuerlich koennen in den TICs auch Fehler sein (Datei nicht gefunden, ...). Hier die Reihenfolge, in der die Fehler ueberprueft werden:

1. Gibt es das File?
2. ZielAKA ueberhaupt eine von deinem System? (Entfällt, wenn keine im TIC [1])
3. AbsenderAKA ueberhaupt eine von irgendeinem User?
4. Gibt es die TIC-Area?
5. Stimmt der TimeStamp? (Nur wenn TimeStamp-Check aktiviert)

[1] Tritt z.B. bei TICs von Requestprozessoren auf

6. Stimmt die Dateigroesse? (Nur wenn Grossen-Check aktiviert)
7. Stimmt die CRC-Summe? (Nur wenn CRC-Check aktiviert)
8. Darf der Sender überhaupt in die Area senden?

3.2.2 Deaktivieren von diversen Checks

-!CRC Die Ueberpruefung der Stimmigkeit der CRC-Summe (CRC-Zeile im TIC) wird hiermit deaktiviert.

-!DATE Die Ueberpruefung der Stimmigkeit des TimeStamps der Datei (Date-Zeile im TIC) wird hiermit deaktiviert.

-!SIZE Die Ueberpruefung der Stimmigkeit der DateiGroesse (Size-Zeile im TIC) wird hiermit deaktiviert.

3.2.3 Der Dupe-Check

3.2.4 Replaces im OutBound

3.2.5 Besonderheiten von erzeugten TICs

`Size <FileSize>` steht immer drin, auch wenn es im eigehenden TIC nicht drinstand.

3.3 ANNOUNCE

Hiermit werden alle Announcements geschrieben.

3.4 AREASBBS

Es wird eine AREAS.BBS-Datei erzeugt.

Format:

```
AREASBBS [<AREAS.BBS-Datei>] [<MailGruppe/Division>]
         [<MailGruppe/Division #2>]
```

Wenn kein AREAS.BBS-Dateiname angegben wird, wird eine Datei `AREAS.BBS` im Mailscan-Verzeichnis erzeugt. Es duerfen dann aber auch keine Gruppen an- gegeben werden.

MailGruppe/Division gibt den Namen der MailGruppe bzw. MailDivision an, die in die Datei mit aufgenommen werden soll(en). Es koennen mehrere Gruppen/Divs angegeben werden. Werden keine angeben, werden alle Echos in der Areas.BBS gelistet.

3.5 MAILAREALIST und FILEEAREAIST

Heisst z.Z. MALIST und FALIST

`FILEAREALIST` funktioniert analog zu `MAILAREALIST`.

`MAILAREALIST <AKA|Username> <Dateiname> [#]`

ist nur erlaubt, wenn eine AKA angegeben wurde. Dann werden nur die Areas gelistet, mit der der User mit dieser AKA Zugriff hat.

`<Dateiname>` gibt den Dateinamen an, in den die Liste geschrieben werden soll.

3.6 SCANFIX

Scannt den Netmailfolder nach *fix-Mails an das eigene System ab.

3.7 IMPORT

Importiert Configs von externen Programmen

3.7.1 IMPORT AREASBBS

Importiert eine `AREAS.BBS`. Die dort genantnen AKAs muessen schon konfiguriert sein.

3.7.2 IMPORT FS_LINKS

Importiert ein von Filescan erzeugtes ASCII-Link-File.

3.7.3 IMPORT SQUALID

Importiert eine `SQUALID.CFG`. Es werden zur Zeit nur die User-Configs und die Adressen uebernommen. Alles andere wird als Fehler ins Log geschrieben

3.8 BBS

Kommandos fuer die FILES.BBS-Dateien au dem System

3.8.1 BBS IMPORT

Importiert `file_id.diz` von allen Files, die zwar in dem gegeben Verzeichnis drin sind, aber nicht in der dortigen FILES.BBS.

3.8.2 BBS MOVE

Syntax:

```
BBS MOVE <Quellpfad, mit Datei(en)> <ZielPfad, ohne Dateiangabe>
```

Beispiel:

```
BBS MOVE z:\files\misc\mp3s\*.* z:\files\sound\mp3s\unsorted
```

Verschiebt alle Dateien von `z:\files\misc\mp3s` nach `z:\files\sound\mp3s\unsorted` und schreibt dort gleich eine neue Files.BBS. Die urspruengliche `FILES.BBS` wird nicht verschoben, Dateien wie `FILES.~~~`, `FILES.BAK` usw. schon.

Achtung: Falls ein `Files.BBS`-Eintrag schon vorhanden ist, wird dieser überschrieben (?).

Falls die Datei selbst im Zielverzeichnis existiert, wird sie umbenannt.

3.9 KILL

Syntax:

```
entweder Kill <filename> [<filename>] ...
    oder Kill <Echoname> [<Loeschungsgrund>]
```

Man kann hiermit Mailareas Ąber die Kommandozeile loeschen. Es wird allerdings nur der Config-Eintrag geloescht und nicht die Messagebase auf der Festplatte!

Sinnvoll z.B. Verarbeiten der `FIDOKILL.GER`.

3.10 KILLFA

wie KILL nur fuer Fileareas

Kapitel 4

Die Template-Files

(. . .)

4.1 Zusaetzliche Makros

Zusaetzlich zu den unter 5.1.2 genannte Makros gehen noch folgende:

%PASSWORD% Das FIX-Passwort des Users

Kapitel 5

Allg. Benutzungsinfos

5.1 „Missbrauch“ als Textposter

Alle Dateien mit der Endung .ML im Arbeitsverzeichnis von Mailscan werden gepostet.

Bei Echomails wird die Mail bei fehlender Tearline und/oder Origin automatisch um diese(n) ergaenzt.

Format einer Datei ist

1. Informationen ueber’s Posting in Form von Tokens
2. der eigentliche Messagetext incl. Makros

Beispiel:

```
--cut:TEST.ML--
$AREA XTC.TEST
$FROMAKA 2:2471/1464
$FROMNAME Von mir
$TONAME Alle
$SUBJECT Mein eigener Betreff
Hallihallo %VORNAME%!

Ganz kleiner Test von mir!

Cya,
    %SYSOP%

--- Test-Tearline
 * Origin: Test-Origin (2:2471/1464)
```

```
--end:Test.ML--
```

5.1.1 Tokens

$SUBJECT Optional. Der Betreff der Mail

$REPLY Optional. Was bei der `@REPLY:`-Kludge stehen soll.

$ORIGIN Optional. Der Origin-Text (ohne `* Origin:`) der Mail

$FOOTER Optional. Mehrere Angaben moeglich. Gibt an, welcher ein Footer angehaengt werden soll. Diese Syntax ist bei Templates fuer *fix-Replys notwendig.

5.1.2 Die Text-Makros

%VORNAME% Ergibt den Vornamen des Empfaengers

%TONAME% Ergibt den kompletten Namen des Empfaengers

%SYSOP% Ergibt deinen konfigurierten Sysopnamen

%TOAKA% Ergibt die Ziel-AKA der Mail

%VONNAME% Ergibt den kompletten Absendernamen

%VONAKA% Liefert die Absender-AKA

5.1.3 Funktionen

Mailscan kann ein paar @-Funktionien, wie sie z.B. in MaxF'req ((c) by TSC) enthalten sind.

include Format: `@include(<Filename>)`
Damit kann man ein File im Template einbinden.

5.2 Ein paar Bemerkungen zum Schreiben der Config

Hat ein User ueber den Config-Eintrag „MAILGROUPS“ Zugriff auf eine Gruppe/Division, zu der er schon ueber seine AKA Zugriff hat, dann wird beim Neuschreiben der Name dieser Gruppe/Division bei „MAILGROUPS“ nicht mehr auf- gefuehrt. Gleiches gilt analog bei „FILEGROUPS“

Kapitel 6

Abschliessendes

6.1 Dankeschoens

Zwei ganz grosse Dankeschoens gehen an Martin Weissenberger und Frank Stephan, die die Alphas auf Herz und Nieren geprueft haben.

6.2 Copyrights

- `JAM(mbp) - Copyright 1993 Joaquim Homrighausen, Andrew Milner, Mats Birch, Mats Wallin. ALL RIGHTS RESERVED.`

- FidoNet is a registered trademark of Tom Jennigs

6.3 ToDo fuer diese Doku

Beispiele ausfuehrlicher. - Keine Kurzschreibweise von AKAs. - Komplettes USERS.CFG-Beispiel