

UMSZer

COLLABORATORS

	<i>TITLE :</i> UMSZer		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		December 8, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	UMSZer	1
1.1	UMSZer.guide	1
1.2	UMSZer.guide/Copyright	2
1.3	UMSZer.guide/Vorwort	2
1.4	UMSZer.guide/Einführung	3
1.5	UMSZer.guide/ZConnect	3
1.6	UMSZer.guide/ZConnect-FAQ	6
1.7	UMSZer.guide/Konfiguration	7
1.8	UMSZer.guide/zer.addressbook	9
1.9	UMSZer.guide/zer.analyze	9
1.10	UMSZer.guide/zer.binary	9
1.11	UMSZer.guide/zer.bincheck	10
1.12	UMSZer.guide/zer.download	10
1.13	UMSZer.guide/zer.ebpuffer	10
1.14	UMSZer.guide/zer.empfangsbest	11
1.15	UMSZer.guide/zer.errlog	11
1.16	UMSZer.guide/zer.footerfilter	12
1.17	UMSZer.guide/zer.gatedemon	12
1.18	UMSZer.guide/zer.gateprefix	12
1.19	UMSZer.guide/zer.headers	12
1.20	UMSZer.guide/zer.killpath	14
1.21	UMSZer.guide/zer.idchange	14
1.22	UMSZer.guide/zer.idmagic	15
1.23	UMSZer.guide/zer.ignorebin	15
1.24	UMSZer.guide/zer.mailfolder	16
1.25	UMSZer.guide/zer.mapsname	16
1.26	UMSZer.guide/zer.namemode	17
1.27	UMSZer.guide/zer.ndosaddress	17
1.28	UMSZer.guide/zer.nodosconv	17
1.29	UMSZer.guide/zer.post	17

1.30	UMSZer.guide/zer.realname	18
1.31	UMSZer.guide/zer.recfilt	18
1.32	UMSZer.guide/zer.refchange	19
1.33	UMSZer.guide/zer.reportuser	19
1.34	UMSZer.guide/zer.routes	20
1.35	UMSZer.guide/zer.serverbox	20
1.36	UMSZer.guide/zer.telefon	20
1.37	UMSZer.guide/zer.timezone	20
1.38	UMSZer.guide/zer.upload	21
1.39	UMSZer.guide/zer.username	21
1.40	UMSZer.guide/zer.uucpgate	23
1.41	UMSZer.guide/zer.zconnect	24
1.42	UMSZer.guide/zer.znetzabs	24
1.43	UMSZer.guide/Erweiterbare Konfigurationselemente	25
1.44	UMSZer.guide/Konfigurationsbeispiel	26
1.45	UMSZer.guide/Attribute	27
1.46	UMSZer.guide/ZerImport	28
1.47	UMSZer.guide/Aufruf des Importers	30
1.48	UMSZer.guide/Fehlermeldungen des Importers	30
1.49	UMSZer.guide/ZerExport	32
1.50	UMSZer.guide/Aufruf des Exporters	32
1.51	UMSZer.guide/Der Servercall	33
1.52	UMSZer.guide/Vergangenheit	35
1.53	UMSZer.guide/Zukunft	37
1.54	UMSZer.guide/Support und Dank	37

Chapter 1

UMSZer

1.1 UMSZer.guide

UMSZer 2.22 public

by Marc Jackisch

Rubensstraße 23

40789 Monheim

Germany

e-mail: mtj@tron.gun.de

Copyright	Weitergabe, Garantien
Vorwort	Ein neues Paket...
Einführung	Was ist UMSZer, was kann es?
ZConnect	Bemerkungen dazu
Konfiguration	Einstellungen bevor es richtig läuft.
Attribute	Mail-Attribute, die einzeln einstellbar sind
ZerImport	Der Importer
ZerExport	Der Exporter
Der Servercall	Ein Skript als Beispiel
Vergangenheit	Zusammenfassung der letzten Folgen.
Zukunft	Madame Olga spricht...
Support und Dank	Bugreports, etc. und Credits

Achtung, bewußter Styleguide-Verstoß: ich habe in meinem Leben zuviel Mathematik gemacht, um meine Versionsnummern Styleguide-konform zu halten. Nullen am Ende sind bedeutungslos, es gelten numerische Vergleiche für Dezimalzahlen. Daher gilt für UMSZer, daß beispielsweise mit 2.2 und 2.20 die gleiche Programmversion gemeint ist und daß 2.2 eine höhere Versionsnummer als 2.15 ist.

1.2 UMSZer.guide/Copyright

Copyright

Copyright (C)1993-94 by Marc Jackisch

UMSZer ist Mailware (öfter mal was neues). Wer UMSZer benutzt, sollte dem Autor doch mal eine kurze Mail zukommen lassen. (Es gibt wohl nichts frustrierenderes, als Software zu veröffentlichen und *keinerlei* Echo darauf zu erhalten.)

Es ist gestattet, Kopien dieser Anleitung zu erstellen und zu verbreiten, sofern der Inhalt unverändert bleibt oder Veränderungen mit dem Autor abgesprochen sind.

Es wird keinerlei Gewährleistungspflicht für die vollständige Funktionsfähigkeit der in diesem Dokument beschriebenen Programme übernommen. Sie nutzen diese Programme auf eigenes Risiko.

Der Verkauf von dem UMSZer-Paket oder Teilen davon zusammen mit anderen kommerziellen Programmen ist nicht ohne schriftliche Genehmigung des Autors gestattet.

Es darf keinerlei Geld am Vertrieb des UMSZer-Paketes oder Teilen davon verdient werden. Es ist gestattet, eine Kopiergebühr zur Deckung der Unkosten (Diskette, Laufwerk, etc.) zu erheben, solange diese den Betrag von DM 5,- nicht übersteigt! Eine Veröffentlichung des Programmes im Rahmen von Zeitschriften muß mit mir abgestimmt werden und bedarf meiner ausdrücklichen Erlaubnis.

Wer mehr als DM 5,- für eine Kopie dieses Programmes bezahlt hat, ist betrogen worden und sollte sich mit ihrer/seiner örtlichen Verbraucherzentrale in Verbindung setzen!

1.3 UMSZer.guide/Vorwort

Vorwort

Im September 1993 kam im Usenet (de.alt.comm.ums) der Aufruf der UMSZer Autoren Matthias Scheler und Markus Stipp, daß jemand gesucht wird, der die Weiterentwicklung und den Support von UMSZer übernimmt. Selbst erst kurz UMS-User habe ich sofort auf diesen Aufruf geantwortet und hatte zwei Tage später die Sources in meinem Downloadpuffer liegen...

Um es kurz zu sagen: UMSZer - wie es jetzt aussieht - hat nicht viel mit dem Bisherigen UMSZer zu tun, daher auch die Versionsnummer 2.x, sondern es basiert viel mehr auf meinen eigenen vorherigen Programmen.

Ich habe versucht, zumindest die Konfiguration kompatibel zu V 1.x zu halten (was allerdings nicht heißen soll, daß das Verhalten in jedem Punkt gleich ist).

Allerdings sollte sich jeder User das Konfigurationskapitel genau durchlesen, manche Elemente sind nicht mehr zwingend, manche fallen ganz weg und viele sind dazugekommen. Andere sind zwar noch vorhanden, haben aber ihr Verhalten geändert.

Ergo: ein Betrieb mit einer V 1.x Konfiguration ist zwar möglich, aber nicht zu empfehlen.

Die Hauptbedeutung für den Anwender liegt also in folgenden Punkten:

- * Die Konfiguration sollte neu gestaltet werden.
- * Flames, Vorschläge, etc. gehen an einen anderen Autoren.

An dieser Stelle sein den Programmierern der Version 1, Matthias Scheler und Markus Stipp gedankt. Ohne Eure Arbeit hätte meine garnicht erst begonnen.

1.4 UMSZer.guide/Einführung

Einführung

UMSZer ist ein Zusatz-Paket zum "Universal Message System" (UMS), daß es ermöglicht, UMS als Z-Netz Point zu nutzen. Das Paket besteht aus einem Exporter und einem Importer. Der Exporter (ZerExport) stellt einen Zerberus Netcallpuffer zusammen, der Importer (ZerImport) sortiert einen empfangenen Puffer ein.(1)

Dabei wird ganz besonders Usenet-Gate-Software unterstützt, d. h. Nachrichtenköpfe aus dem Usenet werden vom Importer korrekt analysiert und für UMS umgesetzt, der Exporter kann dagegen solche Nachrichtenköpfe zusammenstellen. Dabei nimmt UMSZer auf bestimmte Bretter Rücksicht, man kann also die Generierung solcher Köpfe für bestimmte Bretter individuell einstellen.

----- Fußnoten -----

(1) Die Mailverwaltung geht dabei über UMS mit einem beliebigen Newsreader. Dieser Hinweis nur, damit niemand denkt, es würde sich hier um ein komplettes Point-Paket handeln, wie man es von anderen Autoren kennt.

1.5 UMSZer.guide/ZConnect

ZConnect

Ab Version 2.2 unterstützt UMSZer auch den neueren Standard ZConnect. Aus der Sicht eines Point-Programms handelt es sich dabei hauptsächlich

um ein neues, flexibleres Pufferformat, welches starke Ähnlichkeiten zu RFC aufweist, und damit die Mailverwaltung um einiges komfortabler macht (Realnames, Reply-Chaining, etc.)

ZConnect-FAQ Was ich seit V2.2 häufiger gefragt werde.

Leider hat UMS nicht die Möglichkeit, ZConnect komplett zu unterstützen, da viele Informationen in ZConnect völlig neu sind, und dafür keine entsprechenden Felder im UMS-Nachrichtenformat existieren (etwa Postadresse, Telefon, Mail-Kommentare). Solcherlei wird dann -- je nachdem -- ignoriert, in den Kommentar eingetragen oder besonders gekennzeichnet in den Nachrichtentext geschrieben. Ich hoffe hier eine Lösung gefunden zu haben, die für alle akzeptabel sein sollte.

ZConnect hat jedoch nicht nur Vorteile: bei der Definition des ZConnect-Formates wurde wiederum bewiesen, daß sinnlose Einschränkungen einfach aus dem Ärmel geschüttelt werden können ohne näher begründet zu werden. Vor allem denke ich dabei an die weitere Verwendung des sehr eingeschränkten MS-Dos Zeichensatzes und die vollkommen unverständliche Formatfestlegung für die Message-Id. Dabei hält sich die ZConnect Definition leider überhaupt nicht an existierende Standards wie RFC. Da UMS hier jedoch eher in Richtung RFC tendiert, bleiben Probleme nicht aus, falls man UMS für mehrere Netze bzw. Server benutzt. Hier bietet UMSzer Userdefinierte Umwandlungen an, die dazu verwandt werden sollten sich an den ZConnect-Standard zu halten, allerdings auch zu Verstößen gegen den ZConnect-Standard führen können, wenn sie nicht geschickt eingesetzt werden. Bei den entsprechenden Konfigurationselementen wird dann jeweils näher darauf eingegangen.

Einige Prinzipien mögen jedoch erklärungs-würdig sein:

- * Laut Standard dürfen Realnames keine Umlaute enthalten! Da ich passenderweise zu den Leuten gehöre, die sich über so etwas aufregen, möchte ich alle im Sinne der Netzverträglichkeit darum bitten, die Namen ihrer Useraccounts entsprechend abzuändern oder das Konfigurationselement zer.realname zu benutzen. Weiterhin untersagt sind außerdem Klammern und 'ß'. (Da ein Verstoß dagegen zu einer illegalen Nachricht führt, kann der korrekte Transport durch das Netz dann übrigens nicht mehr garantiert werden...)
 - * Da ZerImport formell falsche Mails nicht zurückschickt, wird der 'WAB'-Header nicht benutzt und nur in den Kommentar geschrieben.
 - * Die Header-Einträge 'DDA', 'ERSETZT', 'PRIO', 'LDA', 'SPERRFRIST' werden ignoriert.
 - * Die Headers 'KOP', 'MAILER', 'OAB', 'OEM', 'POST', 'STAT', 'TELEFON', 'ZNETZ-ABS', 'ZNETZ-TEXT', 'GATE' und in bestimmten Fällen 'TYP', gehen in den Mail-Kommentar.
 - * Mail-Kommentare werden besonders gekennzeichnet in den Text gesetzt.
 - * Bei Mails an mehrere Empfänger werden bei öffentlichen Mails Hardlinks, bei privaten Mails Softlinks erzeugt. Dies führt dazu,
-

daß öffentliche Crosspostings nur einmal als gelesen markiert werden müssen um alle Instanzen zu beeinflussen; persönliche Mails dagegen, werden für jeden Empfänger einzeln verwaltet, aber nur einmal auf der Platte abgelegt.

- * Empfangsbestätigungen werden direkt vom Importer in eine zweite Pufferdatei geschrieben(1). Diese kann dann entweder an den normalen Upload-Puffer angehängt (einfach mit `'type xy >> puffer'`) oder mit einem echten ZConnect-Netcall einzeln verschickt werden.
- * Bei Mails an mehrere Empfänger bzw. Bretter werden immer nur die Parameter für den ersten Empfänger benutzt. (Es können allerdings auch verschiedene Teile von Crossposting-Ketten durch verschiedene Exporter verarbeitet werden.)
- * ZConnect unterstützt auch das Umlenken von Antworten auf öffentliche Nachrichten an eine Netzadresse. Dazu muß im Newsreader die RFC-Syntax benutzt werden, d. h. unter `'Followup-To'` muß das Wort `'poster'` eingetragen sein. Ein gleichzeitig gesetztes `'Reply-To'` wirkt dann auch auf den `'Diskussion-In'` Header und wird nicht gesondert in die Mail eingetragen.
- * Wenn der Importer anscheinend lange Zeit nicht arbeitet, keine Panik. Es könnte sein, daß er gerade riesige Mengen von Hardlinks für übertriebene Crosspostings erzeugt. (Ich habe mal eine Mail erlebt, die in 24(!) Bretter ging... In solchen Fällen werden wenigstens Hardlinks erzeugt, man muß die Mail also nur einmal lesen und sie wird überall als gelesen markiert.)
- * Alle unbekannten Einträge aus dem ZConnect-Header kommen in den Kommentar. Damit sollte die Funktionalität auch dann gewährleistet sein, wenn die technische Entwicklung des ZConnect-Standards schneller fortschreitet als die Entwicklung von UMSZer.
- * Besonderer Hinweis zur Konfiguration: das Konfig-Element `zer.username` muß so gesetzt werden, wie der Username im Puffer erscheinen soll, also üblicherweise in Kleinschrift und mit Kompletter Domainangabe.

Es wird im Kapitel Konfiguration darauf hingewiesen, daß ein Alias entsprechend dem Usernamen gesetzt werden muß, damit das Importieren von persönlichen Mails überhaupt möglich ist. Unter ZConnect muß hier natürlich ein Alias für jede mögliche Netzadresse des Users (verschiedene Domains) gesetzt werden, bei mir z. B. `'mtj@tron.gun.de'` und `'mtj@tron.zer.sub.org'`.

- * Für ZConnect habe ich nach langen Überlegungen die Adresskonvertierung für UUCP-Gates wieder zugelassen, obwohl dies nach Standard nicht nötig sein sollte (die Praxis dagegen zeigt, daß es nötig ist). Um sich hier an den Standard zu halten ist der voreingestellte Wert für ZConnect `'o'` statt wie angegeben `'UUCP.ZER'`.

----- Fußnoten -----

(1) Bei entsprechender Konfiguration kann diese auch mit dem normalen Puffer identisch sein.

1.6 UMSZer.guide/ZConnect-FAQ

ZConnect-FAQ

F:

Wie muß meine Absenderadresse aussehen?

A:

Normalerweise wird die ZConnect-Absenderadresse aus der alten Netcall3.8-Adresse + Domain gebildet. Für 90% aller User bedeutet dies, daß die Adresse in Kleinschrift umgewandelt wird und das '.zer' am Ende zu einem '.zer.sub.org' ergänzt wird. Ausnahmen gibt es nur, falls die Box eine andere Domain hat; hier hilft dann der lokale Sysop weiter.

F:

Wie adressiere ich meine Mail?

A:

Die Empfängeradresse sieht genauso aus wie oben die Absenderadresse.

F:

Wie schicke ich Mail ins Internet?

A:

Unter ZC ist es nach Standard vorgesehen, daß keine Konvertierung der Adresse mehr nötig ist. Daher sollte 'zer.uucpgate' auch nicht benutzt werden. Ausnahmen nur für Spezialanwendungen.

F:

Wie muß die Message-ID lauten?

A:

ZC schreibt vor, daß die Msg-ID der Form '<Lokal>@<Point>.<Box>' folgt, also der UMS-Lokalteil (die Nummer vor dem '@'), dann das @-Zeichen gefolgt von dem Pointnamen und dem davon durch Punkt abgetrennten Boxnamen inklusive Domain. Dabei darf der Pointname keine Sonderzeichen wie Punkte oder Underscores enthalten. Falls dies zu Problemen führt bitte den Sysop fragen.

F:

Für diese Form der Message-ID muß ich meine UMS-Domain ändern, oder?

A:

Nein, siehe zer.idchange.

1.7 UMSZer.guide/Konfiguration

Konfiguration

Das UMSZer-Paket verwendet die Konfigurationsmöglichkeiten von UMS, also Einträge in das Konfigurationsfile 'UMS.config'.

Zunächst muß der Im-/Exporter als User eingetragen werden. Dazu verwendet man den Usernamen 'Zerberus' und ein leeres Passwort '""'.(1)

Eine mögliche Basiskonfiguration wäre:

```
user
  name          = Zerberus
  password      = ""
  readaccess    = zer.#?
  writeaccess   = zer.#?
  netaccess     = %
  import        = #?
  export        = #?
  distribution   = #?
  zer.binary     = "DH0:UMS/binaries"
  zer.serverbox = "TRON.ZER"
enduser
```

Konfigurationselemente:

Name	Muß	User	Z-Usr	I	X	Z38	ZCo	Def.
zer.addressbook			*	*		*	*	
zer.analyze [...]			*	*		*	*	No
zer.binary	*		*	*		*	*	
zer.bincheck [...]			*	*		*	*	No
zer.download			*	*		*	*	
zer.ebpuffer	*		*	*			*	
zer.empfangsbest		*			*		*	
zer.errlog			*	*	*	*	*	
zer.footerfilter			*	*		*	*	
zer.gatedemon			*		*	*		
zer.gateprefix			*		*	*		
zer.headers [...]			*		*	*		
zer.killpath		*	*		*	*	*	No
zer.idchange			*	*	*	*	*	
zer.idmagic			*	*	*	*		ZER
zer.ignorebin [...]			*	*		*	*	No
zer.mailfolder [...]		*		*		*	*	
zer.mapsname			*		*	*		MAPS
zer.namemode [...]			*	*		*	*	1
zer.ndosaddress			*	*		*		~(##?)
zer.nodosconv			*	*		*		~(##?)
zer.post		*			*		*	
zer.realname		*			*		*	
zer.recfilt			*	*		*	*	Yes
zer.refchange			*	*		*	*	
zer.reportuser			*	*		*	*	sysop

zer.routes [...]				*		*		*		*		No
zer.serverbox		*				*		*		*		
zer.telefon				*				*				
zer.timezone						*		*		*		W+1
zer.upload						*		*		*		
zer.username [...]		*		*				*		*		
zer.uucpgate						*		*		*		UUCP.ZER
zer.zconnect						*		*		*		No
zer.znetzabs				*				*				

Erweiterbare Konfigurationselemente

Grundlagen

Konfigurationsbeispiel

Sollte man sich ansehen!

Die einzelnen Spalten der Tabelle bedeuten:

Name

Der Name des Konfigurationselementes. Falls dieser von '['[...] gefolgt ist, kann er erweitert werden, z. B. durch Anhängen eines Brettnamens, der durch einen Punkt von dem Konfigurationseintrag getrennt wird.

Muß

Das Element muß angegeben werden.

User

Das Element kann lokal zu einem User angegeben werden. Falls es gleichzeitig andere Möglichkeiten gibt, wird das Lokale Element benutzt.

Z-Usr

Das Element wird beim User 'Zerberus' bzw. 'ZConnect' oder global gesucht.

I

Das Element wird vom Importer gelesen

X

Das Element wird vom Exporter gelesen

Z38

Das Element hat Bedeutung, falls der Zerberus 3.8 Standard verwandt wird.

ZCo

Das Element wird im ZConnect-Modus benutzt.

Def.

Voreinstellung, falls das Element nicht angegeben wird.

***Wichtig:** Jeder User muß einen Alias entsprechend seinem Usernamen an der Serverbox gesetzt haben. (Also == zer.username) Ansonsten können private Mails nicht zugestellt werden.

----- Fußnoten -----

(1) Natürlich kann auch ein anderer Username und ein anderes Passwort angegeben werden, diese müssen dann aber beim Aufruf der Programme in der Kommandozeile angegeben werden.

1.8 UMSZer.guide/zer.addressbook

zer.addressbook
=====

Sample: 'zer.addressbook = <Dateiname>'

Adressbuch, welches verwandt wird um Realnames zu Netzadressen herauszufinden. Das Adressbuch hat das gleiche Format wie das Adressbuch von IntuiNews 1.0, man kann also das IN 1.0 'IntuiNews.Book' Konfigurationselement hier eintragen.

Für alle, die nicht IntuiNews benutzen (selbst Schuld ;-)), hier eine kurze Formatbeschreibung:

```
Name des Users (Realname)
Kommentar      [Wird von ZerImport ignoriert]
Netzadresse    [Diese Zeile darf mehrmals vorkommen]
               [Leerzeile, beendet die Liste der Netzadressen des Users]
```

Dabei besteht jeder Eintrag aus vier oder mehr Zeilen, je nachdem ob für den User mehrere Netzadressen angegeben wurden.

1.9 UMSZer.guide/zer.analyze

zer.analyze
=====

Sample: 'zer.analyze = <Yes, No>'

Gibt an, ob RFC-Header analysiert werden sollen. Headereinträge aus dem Usenet oder eines anderen UMSZer-Users werden nur analysiert, wenn hier 'Yes' angegeben wird.

Das Konfigurationselement ist brettabhängig. Siehe Erweiterbare Konfigurationselemente für nähere Erklärungen.

1.10 UMSZer.guide/zer.binary

zer.binary
=====

Sample: 'zer.binary = <Verzeichnisname>'

Verzeichnis, in das Binärdateien vom Importer abgelegt werden. Dabei wird jede Datei in einem Unterverzeichnis abgelegt, das dem Brettnamen

entspricht, also z. B. eine Mail 'Test' im Brett
'/Z-NETZ/RECHNER/AMIGA/ALLGEMEIN' mit 'UMS:binex' als Exportverzeichnis
unter 'UMS:binex/Z-NETZ/RECHNER/AMIGA/ALLGEMEIN/Test'

1.11 UMSZer.guide/zer.bincheck

zer.bincheck
=====

Sample: 'zer.bincheck = <Yes, No>'

Wenn dieses Flag auf 'Yes' gesetzt wird, wird in jeder als binär
deklarierten Nachricht nach 0-Bytes gesucht. Falls keine gefunden
werden, wird die Nachricht als normaler Text eingelagert. Sind
Nullbytes enthalten, wird mit ihr verfahren wie mit jeder anderen
Binärnachricht.

Das Konfigurationselement ist brettabhängig. Siehe
Erweiterbare Konfigurationselemente für nähere Erklärungen.

1.12 UMSZer.guide/zer.download

zer.download
=====

Sample: 'zer.download = <Verzeichnisname>'

ZerImport sucht einer Datei namens 'PUFFER' in dem angegebenen
Verzeichnis, falls kein Dateiname beim Aufruf angegeben wurde.

1.13 UMSZer.guide/zer.ebpuffer

zer.ebpuffer
=====

Sample: 'zer.ebpuffer = <Dateiname>'

Gibt den Dateinamen des Puffers an, in dem ZerImport für ZConnect die
Empfangsbestätigungen erstellt. Der Puffer wird genau wie bei ZerExport
immer im Append-Modus geöffnet, also kann hier entweder eine
eigenständige Empfangsbestätigungsdatei stehen oder auch der Name des
normalen Upload-Puffers (falls nicht gleichzeitig etwas von ZerExport
exportiert wird).

1.14 UMSZer.guide/zer.empfangsbest

zer.empfangsbest
=====

Sample: `zer.empfangsbest = <"|Netzadresse>`

Dieses Konfigurationselement ist gleichzeitig ein Flag und eine Einstellung. Wird das Konfigurationselement lokal zu einem User angegeben, wird für private Mails eine Empfangsbestätigung angefordert, falls bei der Mail das Attribut `receipt-request` gesetzt ist. (Siehe Attribute) Wenn der String leer ist, geht diese automatisch an den Absender der Mail, ansonsten an die angegebene Netzadresse; wird `zer.empfangsbest` nicht angegeben, werden nie Empfangsbestätigungen angefordert.

Beispiel:

```
zer.empfangsbest = ""
```

1.15 UMSZer.guide/zer.errlog

zer.errlog
=====

Sample: `zer.errlog = <Fehler-Level-Liste>`

Achtung, nur für Freaks ;-)

Dieses Konfigurationselement ermöglicht die Beeinflussung des Loglevels für bestimmte UMS-Fehler. Normalerweise werden Fehler 100-108 auf Level 4, 200-216 auf Level 3 und 300-308 auf Level 2 in das Logfile eingetragen, was z. B. bei einer Writeaccess-Konfiguration, die Crosspostings verhindert zu vielen unerwünschten Fehlermeldungen `No Writeaccess` führen kann.

Über `zer.errlog` können nun für einzelne Fehlermeldungen die Loglevel vom User festgelegt werden. Das Konfigurationselement besteht dabei aus einer Liste von Fehlernummern und zugehörigen Leveln der Form:

```
<Fehlernummer>=<Loglevel>
```

Dabei werden mehrere Einträge durch einen senkrechten Strich `|` voneinander getrennt. Beispiel:

```
zer.errlog = "102=8|216=1"
```

sorgt dafür, daß der Fehler `No Writeaccess` (102) auf Level 8 (uninteressante Nachricht) in das Logfile eingetragen wird und der Fehler `Filesystem Full` (216) auf Level 1 (höchst schwerwiegend).

1.16 UMSZer.guide/zer.footerfilter

zer.footerfilter
=====

Sample: `zer.footerfilter = <Muster>`

Gibt ein Muster an, um Footer wie `-- TheAnswer 2.0` beim Einsortieren abzuschneiden.

Beispiel:

```
zer.footerfilter = "--#?(MPoint|THE ANSWER V?.)#?"
```

1.17 UMSZer.guide/zer.gatedemon

zer.gatedemon
=====

Sample: `zer.gatedemon = <Zerberus-Adresse>`

Bei manchen Arten von Internet-Gates wird E-Mail an eine Dämon-Adresse geschickt, und die erste Zeile der Mail enthält die Empfängeradresse.

ZerExport verwaltet solcherlei automatisch und falls `zer.gatedemon` angegeben ist, wird die Mail an die Adresse des Dämon geschickt und die Empfängeradresse in die erste Zeile der Mail geschrieben.

Wird `zer.gatedemon` nicht angegeben, wird das normale UUCP-Gate benutzt.

1.18 UMSZer.guide/zer.gateprefix

zer.gateprefix
=====

Sample: `zer.gateprefix = <Zeichenkette>`

Falls die Mail an einen Dämon versandt wird, kann man hier ein Präfix für die Empfängerzeile angeben, etwa: `To: `

1.19 UMSZer.guide/zer.headers

zer.headers
=====

Sample: `zer.headers = "<Flag><Flag>..."`

Mit dieser Option können UUCP-artige Header im Nachrichtentext erzeugt werden. Voreingestellt ist, daß keine Header erzeugt werden (also `'zer.headers = ""'`). Folgende Flags können angegeben werden:

`'m'`

Es wird die MsgId aus der UMS-Messagebase (also die original-Message-Id, Siehe auch `zer.idchange`) in die Nachricht geschrieben:

Message-Id: <67077247@p6.f600.n247.fido.de>

`'r'`

Es wird die Referece-Id aus der UMS-Messagebase in die Nachricht geschrieben:

References: <1C57EE5BTA@A-LINK-H.zer.sub.org>

`'o'`

Es wird die Organization (Origin) aus der UMS-Messagebase in die Nachricht geschrieben:

Organization: My Home is my Castle

`'n'`

Es wird der RealName des Absenders mit in die Nachricht geschrieben. Das bewirkt, daß ein anderer Z-Netz-Point, der UMS benutzt diesen beim From im Newsreader angezeigt bekommt.

Realname: Marc Jackisch

`'p'`

Es wird eine Programm-Id in die Nachricht geschrieben, an der man erkennen kann, mit welchem Programm (Version) die Nachricht exportiert wurde. Diese Option ist für Leute gedacht, die den Footer von The Answer II gut fanden :-). (Hier ist es schaltbar und ich bedanke mich bei allen Usern, die ihn einschalten und so für UMSZer Reklame machen.)

X-Mailer: UMSZer V2.22 public

`'t'`

Ein entsprechendes Reply-To Feld wird erzeugt, falls möglich mit Realname und der Adresse:

Reply-To: Marc Jackisch <mtj@tron.gun.de>

`'f'`

Es wird ein Followup-To Header-Eintrag erzeugt. (Dabei wird der Newsgroup-Name aus der UMS-MB verwandt, also ist das Element momentan nicht besonders sinnvoll...)

Follwup-To: zer.USENET.DE.ALT.COMM.UMS

Die Flags bewirken für andere UMS-Benutzer, daß der RealName im From-Feld des Users angezeigt wird und daß es UMS ermöglicht wird, Reply-Links zu erzeugen. (Natürlich nur, falls der andere User `'zer.analyze'` richtig gesetzt hat.

Außerdem werden solche Header von manchen Arten von UUCP-Gates analysiert und benutzt.

Das Konfigurationselement ist brettabhängig. Siehe Erweiterbare Konfigurationselemente für nähere Erklärungen.

Die Erzeugung solcher Header kann für einzelne Mails durch das Attribut 'norfc' abgeschaltet werden (siehe Attribute).

1.20 UMSZer.guide/zer.killpath

zer.killpath
=====

Sample: 'zer.killpath = <Yes, No>'

Wenn dieses Flag auf 'Yes' gesetzt wird, löscht der Exporter die Pfadnamen von Binärdateien im Nachrichtenbetreff, also z. B. 'UMSZer.lha' statt 'Dev:arcs/zerberums/UMSZer.lha'.

1.21 UMSZer.guide/zer.idchange

zer.idchange
=====

Sample: 'zer.idchange = <Umwandlung>'

Multi-Netz-Betrieb führt bei Mitbenutzung des Z-Netz zu gewissen Problemen mit der Message-ID, da diese für ZConnect und die meisten Netcall3.8 Systeme in einer bestimmten Form vorliegen muß (die sich dann üblicherweise nicht mit den anderen benutzten Netzen vertägt.) Spätestens beim Mehrfach-Point-Betrieb mit ZConnect geht dann nichts mehr.

Aus diesem Grunde wurde das Konfigurationselement 'zer.idchange' eingeführt, wodurch der Exporter eine benutzerdefinierte Umwandlung der Message-Id vornimmt. Die Umwandlung besteht aus einem String der Form

<Original-Ende>=<Neues Ende>

wobei das Originalende üblicherweise der Domain-Adress der UMS-MB entsprechen wird. Dieses wird vor dem Exportieren durch das angegebene neue Ende ersetzt.

Zusätzlich wird das Element auch zur Erzeugung korrekter Message-IDs für Empfangsbestätigungen vom Importer gelesen.

Beispiel: die Domain-Adress lautet 'tron.gun.de', also haben alle Message-Ids die Form '...@tron.gun.de'. Ein Netcall3.8 System verlangt nun, daß die ID auf '@TRON' endet. Dies kann durch

zer.idchange = "tron.gun.de=TRON"

erreicht werden.

Ein ZConnect-System dagegen benutzt nur den Lokalteil und hängt

dann den Pointnamen und die Domain an diesen an. Dies führt zu der Form 'lokal@point.do.main', welche man der Konsistenz halber (und vorsichtshalber, denn man weiß ja nie...;) folgendermaßen erzeugen kann:

```
zer.idchange = "tron.gun.de=mtj.tron.gun.de"
```

Werden diese Konfigurationselemente lokal zum jeweiligen Zerberus-User angegeben, sollte ein Multi-Point oder Multi-Netz-Betrieb kein Problem mehr sein.

Zu beachten ist noch, daß die Internet-Header-Erzeugung (Siehe zer.headers) die Original-Id benutzt.

Die Änderungen der ID durch 'zer.idchange' sollten im Sinne einer funktionierenden Kommentarverkettung durch zer.refchange in Referenzen wieder rückgängig gemacht werden. Hier etwa:

```
zer.refchange = "@TRON=@tron.gun.de|@mtj.tron.gun.de=@tron.gun.de"
```

1.22 UMSZer.guide/zer.idmagic

```
zer.idmagic  
=====
```

Sample: 'zer.idmagic = <Zeichenkette>'

Da es offensichtlich Mailboxen gibt, bei denen auf eine korrekte Message-Id Wert gelegt wird (wenn solcherlei auch selten vorkommt) werden Mails an mehrere User oder in mehrere Bretter unter eindeutigen Message-Ids exportiert und Referenzen auf solche Mails beim Importieren wieder zurückverwandelt. Dabei wird die Neue Message-Id aus der bei 'zer.idmagic' angegebenen Zeichenkette, ihrer Nummer aus der UMS-Message-Base und der Originalen Message-Id zusammengesetzt. Damit hier keine Verwechslungen entstehen sollte eine Zeichenkette eingestellt werden, die in Kombination mit der Domain-Adresse eindeutig ist, z. B. die Point-Seriennummer oder der Username.

Bei 'zer.idmagic = "Magie"', einer Message-Id '123456789@TRON' und Mailnummern 1001, 1002 und 1003 an Verschiedene User werden also folgende IDs erzeugt:

```
1001 => 123456789@TRON  
1002 => Magie1002.123456789@TRON  
1003 => Magie1003.123456789@TRON
```

1.23 UMSZer.guide/zer.ignorebin

```
zer.ignorebin  
=====
```

Sample: 'zer.ignorebin = <Yes, No>'

Wird dieses Element auf 'Yes' gesetzt, wird ein eventuell gesetztes Binärflag in der Message ohne weitere Prüfung ignoriert. Sinnvoll z. B. für Internet-Bretter, bei denen man sicher ist, daß nur Mails im 'uuencode' Format versandt werden.

Das Konfigurationselement ist brettabhängig. Siehe Erweiterbare Konfigurationselemente für nähere Erklärungen.

1.24 UMSZer.guide/zer.mailfolder

```
zer.mailfolder
=====
```

Sample: 'zer.mailfolder.<Adresse> = <Folder>'

Über dieses Konfigurationselement ist es möglich, mehrere Netzadressen für einen Account in verschiedenen Foldern zu verwalten. Ein Box-Sysop hat beispielsweise oft mehrere Adressen, wie 'Peter' (zufällige Benennung?) und 'Sysop', was nichts an der Tatsache ändert, daß es sich um *eine* Person handelt, die auch nur *einen* UMS-Account benutzen will.

Um bei diesem Beispiel zu bleiben:

```
zer.mailfolder.sysop = "Sysop-Folder"
zer.mailfolder.sysop@tron.gun.de = "Sysop-Folder"
```

schiebt direkt beim importieren alle persönlichen Mails, die an 'sysop' oder 'sysop@tron.gun.de' adressiert sind, in den Folder 'Sysop-Folder'. (1) Alle anderen Mails gehen nach wie vor in die normale 'Mail'-Gruppe.

Die Regelung des Absenders geht dann über zer.username.

----- Fußnoten -----

(1) Hier ist genau wie bei den Aliases zu beachten, daß ein Eintrag für jede mögliche Adressierungsart vorhanden ist.

1.25 UMSZer.guide/zer.mapsname

```
zer.mapsname
=====
```

Sample: 'zer.mapsname = <Name>'

Einfach der Name von MAPS. Damit wird bewirkt, daß bei Mail an MAPS keine RFC-Header erzeugt werden.

1.26 UMSZer.guide/zer.namemode

zer.namemode
=====

Sample: `zer.namemode = <a|l|u|z>`

Bestimmt die Generierung von Absendernamen aus Absenderadressen (leider noch nötig, solange es noch Leute gibt, die nicht mit UMS arbeiten und so ihre Realnames in der Message hinterlassen...)

`a` bewirkt, daß die gesamte Adresse angegeben wird, `l` nur den Login-Namen, wobei anhängende Gatenamen nicht entfernt werden, `u` nur den Usernamen ohne eventuelle Gates und `z` schneidet von der Adresse nur das `.ZER` am Ende ab.

Das Konfigurationselement ist brettabhängig. Siehe Erweiterbare Konfigurationselemente für nähere Erklärungen.

1.27 UMSZer.guide/zer.ndosaddress

zer.ndosaddress
=====

Sample: `zer.ndosaddress = <Muster>`

Ähnlich wie `zer.nodosconv`, allerdings für persönliche Mail und Absenderadressen, z. B. `zer.ndosaddress = "~(#?.ZER)"` um alle Mails, die nicht aus dem Z-Netz stammen unkonvertiert zu lassen.

1.28 UMSZer.guide/zer.nodosconv

zer.nodosconv
=====

Sample: `zer.nodosconv = <Muster>`

In allen Brettern, deren (konvertierter) Name dem Muster entspricht, wird die Zeichenkonvertierung abgeschaltet. Nützlich z. B., falls ein Gate Umlaute nicht in den IBM-Standard umwandelt.

Beim exportieren wird hier auch auf das Einfügen von CR beschränkt.

1.29 UMSZer.guide/zer.post

```
zer.post
=====
```

```
Sample: `zer.post = <Postadresse>`
```

Durch Angabe dieses Konfigurationselementes bei einem User wird in seinen Mails der ZConnect-Header 'POST' erzeugt. (Natürlich nur im ZConnect-Modus.) Nach ZConnect-Standard sollen hier die verschiedenen Zeilen der Adresse durch ';' getrennt werden, bitte halten sie sich daran (sonst bekomme ich wieder Flames, von wegen UMSZer würde nicht funktionieren...)

Beispiel:

```
zer.post = "Rubensstrasse 23; 40789 Monheim; Germany"
```

1.30 UMSZer.guide/zer.realname

```
zer.realname
=====
```

```
Sample: `zer.realname = <Name>`
```

Normalerweise wird als Realname zum Puffer-Aufbau der Name des UMS-Accounts benutzt. Falls dies unerwünscht sein sollte (z. B. bei Umlauten im Accountnamen, die ja unter ZConnect verboten sind), kann durch dieses Konfigurationselement ein anderer Name angegeben werden.

Beispiel:

```
zer.realname = "Marc Jackisch"
```

1.31 UMSZer.guide/zer.recfilt

```
zer.recfilt
=====
```

```
Sample: `zer.recfilt = <Yes, No>`
```

Dieses Konfigurationselement ist nur sinnvoll, falls 'zer.analyze' eingeschaltet ist.

Normalerweise werden 'Received'-Zeilen in der RFC-Headeranalyse ignoriert, da sie für den normalen User nicht von weiterem Interesse sind. Dies lässt sich durch Einstellen von 'zer.recfilt' auf 'No' verhindern; in diesem Fall werden die 'Received'-Zeilen mit in den Kommentar übernommen.

1.32 UMSZer.guide/zer.refchange

zer.refchange
=====

Sample: 'zer.refchange = <Umwandlungsliste>'

Wenn man schon RFC-Header analysiert, sollte man auch möglichst viel herausholen... Das 'refchange' Konfigurationselement ermöglicht die Umwandlung von 'References' im Internet-Header bzw. 'BEZ' in ZConnect-Headern nach userdefinierbaren Funktionen. Z. B. werden References von Internet-Usern auf Zerberus-Mails meistens mit dem Ende '.zer.sub.org' versehen, was dazu führt, daß die Referenz nicht erkannt wird; auch am Gate kann dieses vorkommen, etwa '@tron.gun.de' statt der Original-ID '@TRON'. Andererseits werden von ZConnect-Systemen nur die Lokalteile der ID benutzt (alles vor dem @) und Point- und Domainnamen werden angehängt; hier kann man für Bezüge auf eigene Mails den Original-Domain-Namen wieder herstellen.

In das Konfiguartionselement wird eine Liste von Umwandlungen der Form

<Original-Ende>=<Neues Ende>

eingetragen, einzelne Umwandlungen werden dabei durch einen senkrechten Strich '|' getrennt. Wenn nun bei einer Reference das Original-Ende gefunden wird, wird es durch das neue Ende ersetzt. Das neue Ende darf dabei leer sein. Beispiel:

```
zer.refchange =  
".zer.sub.org=|@TRON=@tron.gun.de|@mtj.tron.gun.de=@tron.gun.de"
```

schneidet das Ende '.zer.sub.org' einfach ab und ersetzt sowohl '@mtj.tron.gun.de' als auch '@TRON' durch '@tron.gun.de'.

Dabei möge bitte beachtet werden, daß die Überprüfung der Enden von links nach rechts stattfindet, was bei Überschneidungen zu Problemen führen kann, etwa wenn das Originalende 'tron.gun.de' vor 'mtj.tron.gun.de' angegeben ist, wird die erste Umwandlung stattfinden. Ich empfehle daher, die Zeichenketten falls möglich mit einem führenden '@' zu versehen, damit Eindeutigkeit gewährleistet ist.

1.33 UMSZer.guide/zer.reportuser

zer.reportuser
=====

Sample: 'zer.reportuser = <Name>'

Gibt den Namen eines Users an, der benachrichtigt wird falls eine Mail nicht zugestellt werden kann.

1.34 UMSZer.guide/zer.routes

zer.routes
=====

Sample: `zer.routes = <Yes|No>`

Bestimmt, ob Route-Strings als Kommentar generiert werden sollen oder nicht.

Das Konfigurationselement ist brettabhängig. Siehe Erweiterbare Konfigurationselemente für nähere Erklärungen.

1.35 UMSZer.guide/zer.serverbox

zer.serverbox
=====

Sample: `zer.serverbox = <Boxname.ZER>`

Dieses Feld wird zur generierung des Absenders und zur Adressanalyse im Adressbuch benötigt, z. B. `zer.serverbox = "TRON.ZER"``

1.36 UMSZer.guide/zer.telefon

zer.telefon
=====

Sample: `zer.telefon = <String>`

Angabe einer Telefonnummer. Wird dieses Konfigurationselement angegeben, erzeugt der Exporter im ZConnect-Modus einen `TELEFON`-Header. Hier hat der geneigte User mal wieder selbst für die Einhaltung der ZConnect-Richtlinien zu sorgen: Es wird ein Kennbuchstabe (V für Voice, B für Mailbox oder F für Fax), gefolgt von einer Telefonnummer in Internationaler Schreibweise und bei Voice-Nummern mit Anrufbeantworter einem nachgestellten Q angegeben. Mehrere Nummern werden durch Semikolon oder Leerzeichen getrennt angegeben; bei kombinierten Nummern werden alle passenden Kennbuchstaben vorangestellt. Beispiele aus der ZC-Dokumentation (nein, ich werde meine Telefonnummer nicht in die Doku schreiben! ;-)):

```
zer.telefon = "V+49-521-561345Q F+49-521-561785 B+49-521-193004"
zer.telefon = "VF+49-521-562342Q"
```

1.37 UMSZer.guide/zer.timezone


```
zer.timezone  
=====
```

Sample: `'zer.timezone = <Zeitzone>'`

Die Zeitzone für die ZConnect-Nachrichten in der Form `'[S/W][+/-]<Offset>'`, also S oder W für Sommer- oder Winterzeit und der Offset zur Greenwich-Meantime. Für Deutschland also entweder `'W+1'` für Winterzeit oder `'S+2'` für die Sommerzeit. (Noch etwas mehr zum Umstellen...)

`'zer.timezone'` wird auch von ZerImport zum erstellen von Empfangsbestätigungen benutzt.

1.38 UMSZer.guide/zer.upload

```
zer.upload  
=====
```

Sample: `'zer.upload = <Verzeichnisname>'`

In diesem Verzeichnis wird die `'PUFFER'`-Datei abgelegt, falls kein Dateiname an ZerExport übergeben wurde.

1.39 UMSZer.guide/zer.username

```
zer.username  
=====
```

Sample: `'zer.username = <Username>'`

Muß bei jedem User eingetragen werden. Dies ist der Username, den die entsprechende Person an der Serverbox hat. Dabei ist zu beachten, daß der Username so angegeben werden muß, wie er im Point-Puffer erscheinen soll, also für Z3.8-Systeme nur der Login-Name, z.B. `'MTJ'` und für ZConnect-Systeme der Username mit der kompletten Adresse, z.B. `'mtj@tron.zer.sub.org'` oder `'mtj@tron.gun.de'`.

Und extra nochmal, da es so viele falsch machen: die Angabe von `'zer.username'` reicht **nicht**, damit der Importer persönliche Mails zustellen kann, weil er dazu **nur** Alias-Einträge und den Namen des Useraccounts benutzt. Es muß **unbedingt** ein Alias mit dem gleichen Inhalt, wie `'zer.username'` angegeben werden!

Bei dem Namen des Konfigurationselementes kann auch noch ein Boxname angehängt werden, der dem Eintrag `zer.serverbox` entspricht und von `'zer.username'` durch einen Punkt getrennt wird. Dieser wird im Zweifelsfall bevorzugt genommen. Eine weitere mögliche Erweiterung ist der Name eines Mailfolders, dazu unten mehr.

Für einfachen Point-Betrieb mit einer Serverbox ist dies nicht nötig!

Durch diese Erweiterung kann mit mehreren Zerberus-Usern (unter verschiedenen Namen) ein Pointbetrieb mit mehreren Servern konfiguriert werden; dabei muß 'zer.serverbox' natürlich jeweils lokal zum jeweiligen Zerberus angegeben werden und der Benutzer selber muß entsprechende Aliase gesetzt haben.

Zur Erweiterung von Konfigurationselementen, siehe auch Erweiterbare Konfigurationselemente.

Beispiel: Username 1 ist 'MTJ@TRON.ZER', Username 2 ist 'MARC@SONSTWO.ZER', Username 3 ist 'm.jackisch@woanders.zer.sub.org' für ein ZConnect-System:

```
user
  name          = Zerberus
  ...
  zer.serverbox = TRON.ZER
enduser

user
  name          = ZerSonstwo
  ...
  zer.serverbox = SONSTWO.ZER
enduser

user
  name          = ZCWoanders
  ...
  zer.serverbox = woanders.zer.sub.org
  zer.zconnect  = "Yes"
enduser

user
  name          = Marc Jackisch
  alias         = MTJ
  zer.username  = MTJ
  alias         = MARC
  zer.username.SONSTWO.ZER = MARC
  alias         = m.jackisch@woanders.zer.sub.org
  zer.username.woanders.zer.sub.org
                = m.jackisch@woanders.zer.sub.org
  ...
enduser
```

Zum Aufruf des Importers oder Exporters unter dem Namen 'ZerSonstwo' oder 'ZCWoanders' müssen dann Username und Passwort auf der Kommandozeile angegeben werden. Siehe hierzu Aufruf des Importers und Aufruf des Exporters.

Als ob das noch nicht kompliziert genug wäre, ist noch eine weitere Erweiterung möglich. An die Form 'zer.username.<Box>' kann noch der Name eines Folders angehängt werden. (Die Angabe der Box ist dabei Pflicht, auch wenn dies sonst nicht nötig sein sollte.) Dieser wird

von dem Bisherigen durch zwei Minuszeichen getrennt, also
`'zer.username.<Box>--<Folder>'`.

Die Ergänzung um einen Folder bewirkt die genaue Umkehrung des Verhaltens von `zer.mailfolder` beim Exportieren. Mail, die in verschiedenen Foldern geschrieben wird, kann verschiedene Absenderadressen tragen. Als Beispiel, passend zu `'zer.mailfolder'`:

```
zer.username.tron.gun.de--Sysop-Folder = "sysop@tron.gun.de"
```

exportiert alle Mails im Sysop-Folder unter der Netzadresse
`'sysop@tron.gun.de'`

1.40 UMSZer.guide/zer.uucpgate

```
zer.uucpgate
=====
```

Sample: `'zer.uucpgate = <Netzadresse>'`

Dieses Feld legt den Gateway ins UUCP fest. Sollte normalerweise auf `'UUCP.ZER'` gestellt werden. Soll das Z-Netz als UUCP-Gate benutzt werden, so müssen die export-Rechte des "Zerberus"-Users richtig gesetzt werden, z. B.: `'Export = "#?(.zer|.de|.sub.org)''`. Für ZConnect sollte keine Konvertierung nötig sein, also ist im ZC-Modus der voreingestellte Wert `'o'`.

Eine InterNet oder Subnet Adresse wird durch Konvertierung in Großschrift und Voranstellen von Backslashes vor Großbuchstaben im Loakteil der Adresse folgendermaßen gewandelt:

```
Jackisch@radium.informatik.uni-bonn.de
-> \JACKISCH%RADIUM.INFORMATIK.UNI-BONN.DE@UUCP.ZER

mtj@tron.GUN.de
-> MTJ%TRON.GUN.DE@UUCP.ZER
```

Da dies jedoch nicht für jede Art von Gate-Software passend ist, gibt es die Möglichkeit, an den eingestellten String ein durch Komma getrenntes Flag anzuhängen. Verwandte Flags sind bisher `'o'` für keinerlei Adresswandlung (es wird also nicht einmal der Gate-Name angehängt), `'n'` für Adresskonvertierung ohne Klein/Großschrift zu verändern und `'h'` für eine Adresswandlung wie oben beschrieben, ohne jedoch eine Gate-Adresse zu benutzen. Eine weitere Möglichkeit ist `'d'` für den voreingestellten Modus.

Falls die Mails für Importer und Exporter unterschiedlich behandelt werden sollen, kann ein zweites mit Komma abgetrenntes Flag nur für den Exporter angegeben werden, etwa `'UUCP.ZER,h,d'`.

Für Flag `'o'` sähe ein Konfigurationseintrag etwa so aus:

```
zer.uucpgate = "UUCP.ZER,o"
oder einfach
zer.uucpgate = ",o"
```

wobei keine Adresswandlung stattfindet (eher für UUCP mit
Zerberus-Übertragungsprotokoll geeignet), für das Flag 'n' wäre ein
Beispieleintrag

```
zer.uucpgate = "UUCP.ZER,n"
```

und die Wandlung der obigen Beispiele:

```
Jackisch@radium.informatik.uni-bonn.de  
-> Jackisch%radium.informatik.uni-bonn.de@UUCP.ZER  
  
mtj@tron.GUN.de  
-> mtj%tron.GUN.de@UUCP.ZER
```

entsprechend der Eintrag für das Flag 'h':

```
zer.uucpgate = "UUCP.ZER,h"  
oder einfach  
zer.uucpgate = ",h"
```

mit der Wandlung:

```
Jackisch@radium.informatik.uni-bonn.de  
-> \JACKISCH@RADIUM.INFORMATIK.UNI-BONN.DE  
  
mtj@tron.GUN.de  
-> MTJ@TRON.GUN.DE
```

ZerImport wandelt beim importieren in der anderen Richtung. (Wenn zwei
Flags angegeben werden, wird von ZerImport das erste benutzt.)

1.41 UMSZer.guide/zer.zconnect

```
zer.zconnect  
=====
```

Sample: 'zer.zconnect = <Yes|No>'

Dieses Konfigurationselement legt fest, in welchem Hauptmodus UMSZer
arbeitet. Wird es auf 'Yes' gesetzt, werden Pufferdateien im
ZConnect-Format erwartet bzw. erzeugt; andernfalls folgt UMSZer dem
Zerberus3.8-Standard.

1.42 UMSZer.guide/zer.znetzabs

```
zer.znetzabs  
=====
```

Sample: 'zer.znetzabs = <Adresse>'

Ein weiterer ZConnect-Header. Dies gibt eine Adresse für das Z-Netz an,
falls sich die Adressierung des Users unterscheiden sollte. Wird
normalerweise automatisch von Gates erzeugt und sollte im Allgemeinen
nicht angegeben werden.

1.43 UMSZer.guide/Erweiterbare Konfigurationselemente

Erweiterbare Konfigurationselemente

Die Konfigurationselemente, denen in der Tabelle ein '[...]' folgt, sind brettabhängig bzw. erweiterbar, d. h. durch Anhängen eines Zusatzes (in den meisten Fällen ein Brettname oder ein Teil davon) kann der Wirkungsbereich beschränkt werden. Dabei muß die Erweiterung durch einen Punkt vom eigentlichen Namen des Konfigurationselementes getrennt werden. Der Punkt gilt auch innerhalb von Gruppennamen als Trennzeichen, z. B.

```
zer.namemode.zer.Z-NETZ.RECHNER.AMIGA = "z"
```

stellt den Namensmodus 'z' für alle Z-Netz/Rechner/Amiga-Gruppen ein. Die Suche nach dem "richtigen" Konfigurationselement geht dabei von dem kompletten Namen kürzer werdend bis zum Konfigurationselement selber, also wenn z. B. 'zer.namemode' für die Gruppe 'zer.Z-NETZ.RECHNER.AMIGA.ALLGEMEIN' gesucht ist, werden nacheinander folgende Einträge überprüft:

```
zer.namemode.zer.Z-NETZ.RECHNER.AMIGA.ALLGEMEIN
zer.namemode.zer.Z-NETZ.RECHNER.AMIGA
zer.namemode.zer.Z-NETZ.RECHNER
zer.namemode.zer.Z-NETZ
zer.namemode.zer
zer.namemode
```

Der erste gefundene Eintrag wird verwandt.

Der Gruppenname für Persönliche Messages ist 'Mail'.

Beispiele:

```
zer.namemode.zer.Z-NETZ = "l" => Namemode 'l' für das gesamte Z-Netz
zer.namemode.zer.Z-NETZ.RECHNER = "u" => für Z-NETZ.RECHNER.#?
zer.namemode.Mail = "z" => für Persönliche Mail
zer.namemode = "a" => für alles Andere
```

```
zer.routes = "Yes" => Route-Strings generieren
zer.routes.zer.USENET = "No" => ... aber nicht für Usenet-Gruppen
```

```
zer.bincheck = "Yes" => Binärcheck ein
zer.bincheck.zer.FRAS.BIN = "No" => aber in FRAS.BIN.#? aus
```

```
zer.ignorebin = "No" => Binärflag nicht ignorieren
zer.ignorebin.zer.USENET = "Yes" => außer im Usenet
```

Eine Ausnahme bei den Erweiterungen bildet das Konfigurationselement 'zer.username'. Wenn hier der Name einer Serverbox angehängt wird, muß dieser **komplett** angegeben werden und darf **nicht** an einem Punkt abgeschnitten werden. Der Anhang muß dann **exakt gleich** wie das Element 'zer.serverbox' lauten, etwa:

```

zer.serverbox = TRON.ZER
[...]
zer.username.TRON.ZER = "MTJ"

```

1.44 UMSZer.guide/Konfigurationsbeispiel

Konfigurationsbeispiel

```
=====
```

Als Ausgangspunkt für eine eigene Konfiguration hier ein Beispiel, mit allen Zerberus-Relevanten Konfigurationselementen. Zunächst allgemeines und der User (hier eingestellt für die Benutzung des Zerberus und des ZConnect-Users):

```

AKA = mtj#?
DOMAIN-ADDRESS = tron.gun.de

user
  name                = "Marc Jackisch"
  sysop
  alias               = "Marc"
  alias               = sysop
  alias               = "MTJ"
  alias               = "mtj@tron.gun.de"
  password            = "xy"
  readaccess          = #?
  writeaccess         = #?
  netaccess           = #?
  zer.username        = "MTJ"
  zer.username.TRON.gun.de = "mtj@tron.gun.de"
  zer.killpath        = "Yes"
enduser

```

Der Zerberus-User für ein Netcall3.8-System:

```

user
  name                = Zerberus
  password            = ""
  readaccess          = zer.#?
  writeaccess         = zer.#?
  netaccess           = %
  import              = #?
  export              = #?
  distribution        = #?
  zer.serverbox       = "TRON.ZER"
  zer.uucpgate        = "USENET.ZER"
  zer.headers         = "nmp"
  zer.headers.zer.USENET = "rp"
  zer.binary          = "HD:UMS/zer-bin"
  zer.ignorebin       = "No"
  zer.ignorebin.zer.USENET = "Yes"
  zer.bincheck        = "Yes"
  zer.analyze         = "Yes"

```

```

zer.addressbook          = "HD:UMS/reader/AddressBook"
zer.refchange            = "@TRON=@tron.gun.de|.zer.sub.org="
zer.namemode             = "z"
zer.idmagic              = "MJ01"
zer.footerfilter         =
    "(--#?MPoint|--#?THE ANSWER V|Zodiac's Point|#?CrossPoint)#?"
zer.idchange             = "tron.gun.de=TRON"
enduser

```

Alternativ dazu ein ZConnect-User:

```

user
  name                   = ZConnect
  password               = ""
  readaccess             = zer.#?
  writeaccess            = zer.#?
  netaccess              = %
  import                 = #?
  export                 = #?
  distribution           = #?
  zer.zconnect           = "Yes"
  zer.timezone           = "W+1"
  zer.serverbox          = "tron.gun.de"
  zer.binary             = "HD:UMS/zer-bin"
  zer.analyze            = "Yes"
  zer.addressbook        = "HD:UMS/reader/AddressBook"
  zer.idchange           = "tron.gun.de=mtj.tron.gun.de"
  zer.refchange          =
    "@TRON=@tron.gun.de|@mtj.tron.gun.de=@tron.gun.de"
  zer.namemode           = "z"
  zer.idmagic            = "MJ01"
  zer.footerfilter       =
    "(--#?MPoint|--#?THE ANSWER V|Zodiac's Point|#?CrossPoint)#?"
enduser

```

1.45 UMSZer.guide/Attribute

Attribute

Was bei anderen Point-Programmen über Mail-Requester oder ähnliches geht, nämlich die Einstellung verschiedener Flags, geht mit UMSZer über den Eintrag von 'Attributen' im Newsreader. Somit können auch hier einige Einstellungen individuell für jede Mail gesetzt werden. Mehrere Attribute können dabei in beliebiger Reihenfolge durch Leerzeichen getrennt angegeben werden.(1)

Die nun folgenden Attribute sind UMSZer bekannt. Es folgt jeweils eine kurze Beschreibung der Wirkung eines Attributs:

'crash'

Wird ignoriert ('crash' ist hier nur angegeben, da IntuiNews 1.1 es bei 'file-attach' automatisch hinzufügt.)

'file-attach'

Dabei gibt der Betreff der Mail eine Datei an, die beim

Exportieren mit in den Puffer geschrieben wird. Unter Zerberus 3.8 wird *nur* die angegebene Datei in den Puffer geschrieben und ein eventueller Text ignoriert. Unter ZConnect wird ein eventueller Text als Kommentar davorgeschrieben. In beiden Fällen wird die erzeugte Message als 'binär' markiert.

'norfc'

Bei Z3.8 wird für diese Mail die erzeugung von RFC-Headern im Text unterdrückt. Unter ZConnect wird dies ignoriert.

'receipt'

Wird vom Importer für Empfangsbestätigungen gesetzt.

'receipt-request'

Unter ZConnect wird zu der Mail eine Empfangsbestätigung angefordert. (Nur, falls es sich um eine persönliche Nachricht handelt, für News nie.) Unter Z3.8 wird dies ignoriert.

----- Fußnoten -----

(1) Es wird zum Erstellungsszeitpunkt dieser Dokumentation darüber diskutiert, in IntuiNews einen frei definierbaren Attribut-Requester einzubauen, so daß vielleicht früher oder später beliebige Attribute per Mausklick gesetzt werden können. Unter IN 1.1 müssen die meisten leider noch per Hand eingegeben werden.

1.46 UMSZer.guide/ZerImport

ZerImport

Der Mail-Importer ZerImport ist dafür verantwortlich, einen Netcallpuffer zu analysieren und in die UMS Message-Base einzutragen. Dabei wurde bei der Programmierung des Importers besonders viel Wert auf die Verwendbarkeit an einem Internet- (Usenet) Gate gelegt. ZerImport kann soweit es möglich ist, Internet-Header Einträge am Anfang einer Mail analysieren und unterstützt dabei alle Möglichkeiten von UMS. Außerdem kann ZerImport Absenderadressen in echte Internet-Adressen wandeln und - soweit vorhanden - die Realnames der Absender beim Eintragen der Message verwenden.

ZerImport hat folgende Eigenschaften:

- * Erkennung von Binärdateien. Jeder hat damit Erfahrung: oft ist (meistens nach Gates) unnötigerweise das Binärflag in einer Datei gesetzt. ZerImport kann in solchen Fällen auf Wunsch überprüfen, ob es sich wirklich um eine Binärdatei handelt, indem die Message durchsucht wird. Werden darin Nullbytes gefunden, nimmt ZerImport an, daß es sich um eine Binärdatei handelt. Ansonsten ist es eindeutig eine Textdatei und wird als solche weiterverarbeitet.
- * Auslagern von Binärdateien. Binärdateien werden von ZerImport ausgelagert. (UMS arbeitet mit 0-terminierten Strings, kann also keine Binaries innerhalb der Message-Base verwalten.) Anstelle der

ursprünglichen Mail wird in die Messagebase eine Kurze Notiz eingetragen, unter welchem Dateinamen die Binärdatei zu finden ist und welche Größe sie hat. Dabei verwendet ZerImport eindeutige Namen, indem Zahlen an den Dateinamen angehängt werden, falls er schon vorhanden sein sollte. Existierende Dateien werden somit nie überschrieben. (1)

- * Analyse von Internet-Headern. Falls der Text der Nachricht mit einem Internet-Header beginnt, kann dieser analysiert und für UMS verwandt werden. Da diese Header-Einträge normalerweise der abgesandten Originalmail mehr entsprechen, als die Z-Netz-Header, werden sie im Zweifelsfall bevorzugt benutzt.
- * Textkonvertierung nach Brettern und Absendern an/abschaltbar, z. B. an Gates, die Umlaute nicht in das IBM-Format konvertieren. Bei abgeschalteter Konvertierung werden nur noch die CR-Zeichen (ASCII 13) aus dem Text entfernt.
- * Unterstützung von mehreren Text-Kodierschemata. Momentan ZC-Standard, ISO, Quoted-Printable.
- * Erweiterte Dupe-Erkennung. Wenn die Message ID bereits existiert, wird festgestellt, ob es sich wirklich um ein Dupe handelt, indem sämtliche Felder der Message mit der bereits Existierenden verglichen werden. Sollte sich der Text unterscheiden, versucht ZerImport ein Softlink zu kreieren, falls sich andere Felder unterscheiden (Crosspostings) wird ein Hardlink erstellt. Nur wenn *alle* Felder gleich sind, wird die Message als Dupe aussortiert. (2)
- * Adressbuchunterstützung. Wird der Absender einer Nachricht anhand des Adressbuchs erkannt, wird als Realname der Adressbucheintrag verwandt. Das Adressbuchformat ist kompatibel zu IntuiNews 1.1.
- * Reference-Umwandlung zur besseren Erzeugung wiedererkennbarer Message-IDs.
- * Automatischer Versand von Empfangsbestätigungen.

Aufruf des Importers	Wie sage ich's meinem Kinde?
Fehlermeldungen des Importers	Was und warum

----- Fußnoten -----

(1) Da die Binärdateien ausgelagert werden, *bevor* die Nachricht in die UMS-MB geschrieben wird, sollte man möglichst vermeiden, einen Netcallpuffer mehrmals einzulagern. Dies könnte zu überflüssigen Binärdateien führen.

(2) Hier kommt leider noch eine kleine Unstimmigkeit (sollte behoben werden): Falls eine Message schon einmal als Hardlink einsortiert wurde, wird ein weiteres Hardlink erzeugt. Dieses ist allerdings im Newsreader nicht zu sehen, sondern fällt nur durch einen zusätzlichen Eintrag unter 'Newsgroups' im Message-Header auf (eine Newsgroup wird dann zweimal angegeben).

1.47 UMSZer.guide/Aufruf des Importers

Aufruf des Importers
=====

Der Importer ZerImport wird von der Shell mit folgender Schablone gestartet:

```
ZerImport FILE,PRETEND/S,DEBUG/S,VERBOSE/S,NAME/K,PASS/K,SERVER/K,ZCO/S
```

Die Bedeutung der Parameter:

'FILE'

Dieser Parameter gibt den Namen des Netcallpuffers an. Falls FILE nicht gesetzt wird, sucht ZerImport in einem voreingestellten Verzeichnis nach einer Datei 'PUFFER'.

'PRETEND/S'

Pretend mode. Der Importer importiert keine Mails, sondern gibt nur die Mail-Header in die Shell aus.

'DEBUG/S'

Debug mode. Alle möglichen sinnlosen Nachrichten werden auf dem Bildschirm ausgegeben. Für Anwender uninteressant.

'VERBOSE/S'

Verbose mode. Im Falle eines Schreibfehlers wird der Nachrichtenkopf ausgegeben.

'NAME/K'

Der Username, unter dem sich der Zerberus-Importer in die UMS-MB einloggen soll

'PASS/K'

Das zugehörige Passwort.

'SERVER/K'

Zur Angabe von bestimmten UMS-Servern.

'ZCO/S'

Wird dieser Schalter gesetzt, erzeugt ZerImport am Anfang von Brettnamen den String 'zco' statt wie sonst 'zer'.

1.48 UMSZer.guide/Fehlermeldungen des Importers

Fehlermeldungen des Importers
=====

Diese Liste erhebt keinesfalls den Anspruch auf Vollständigkeit! Es ist auch nicht garantiert, daß Vorschläge zur Beseitigung funktionieren. Ich möchte alle User bitten, mir Bescheid zu geben, falls eine hier

nicht genannte Fehlermeldung auftritt, oder eine Fehlermeldung durch Anwendung anderer Methoden beseitigt wurde. Nur so kann diese Liste vielleicht jemals komplett werden!

Fehler in ReadArgs()

Die Argumente wurden nicht korrekt angegeben, überprüfen Sie die Kommandozeile, mit der ZerImport gestartet wurde.

Konnte Puffer-Datei ... nicht öffnen!

Ein Fehler ist beim Öffnen der Pufferdatei aufgetreten. Existiert der Angegebene Dateiname überhaupt?

Login als ... auf Server ... schiefgegangen!

Entweder der angegebene Server existiert nicht, oder der Username oder das Passwort sind fehlerhaft.

Konnte Library ... V... nicht öffnen

Kein Speicher oder die Library ist in der verlangten Version wirklich nicht vorhanden. Sehen Sie sich nach dem neuesten UMS-Paket um!

Konnte Konfigurationselement ... nicht lesen

Ein Konfigurationselement wurde nicht gefunden, entweder es fehlt oder vielleicht ein Tippfehler?

Konnte DOMAIN-ADDRESS nicht finden!

Sollte eigentlich von UMS selber schon bemängelt worden sein.

Parse-Pattern Fehler für ...

Das angegebene Muster ist kein korrektes AmigaDOS-Namensmuster.

Kein (freier) Speicher mehr

Der Rechner hat Speicherprobleme... entweder ein bisschen weniger das Multitasking nutzen oder ihm doch mal ein bisschen mehr Speichererweiterung schenken.

Kann User ... nicht finden

Der Username existiert nicht. Haben Sie an den Alias, der Ihren Usernamen an der Serverbox angibt gedacht?

Fehler beim schreiben von Nachricht ...

Sollte von einer UMS-Fehlermeldung gefolgt werden. Vielleicht liegt es an den Import-Rechten des Zerberus-Users.

Fehler beim lesen von Nachricht ...

Die angegebene Nachricht konnte nicht aus der UMS-MB gelesen werden, eine Fehlermeldung sollte folgen. Könnte an dem 'readaccess' des Zerberus-Users liegen. (Der Importer liest innerhalb des Dupechecks auch Nachrichten zu Vergleichszwecken aus.)

Fehler beim Erstellen eines Hardlink/Softlink

Wiederum sollte eine UMS-Fehlermeldung folgen.

Fehler ... beim Erstellen des Verzeichnisses ...

Beim Auslagern von Binärdateien ist der angegebene Fehler aufgetreten.

1.49 UMSZer.guide/ZerExport

ZerExport

Der Mail-Exporter ZerExport ist dafür verantwortlich, aus Mail, die vom User geschrieben wurde, einen Netcallpuffer zusammenzustellen.

Wichtig:

Der Puffer wird immer im 'append'-Modus geöffnet, damit vorher exportierte Mails nicht verlorengehen. Daher muß ein Puffer immer nach einem erfolgreichen Netcall gelöscht werden. Wird dies unterlassen, werden Mails mehrmals verschickt; wird der Puffer vor einem erfolgreichen Netcall gelöscht, gehen Mails verloren!

Ein 'Servercall'-Skript, sollte also folgenden Mechanismus benutzen:

ZerExport <Upload-Puffer> aufrufen

Upload-Puffer packen

Servercall ausführen

Wenn der Servercall erfolgreich war (und nur dann):

Upload-Puffer löschen

Download-Puffer entpacken und per ZerImport einsortieren

Die wichtigsten Fähigkeiten des Exporters in Kürze:

- * Wandlung von Internet-Adressen. Der Exporter kann Adressen, die *nicht* auf '.ZER' bzw. '.zer.sub.org' enden, in Internet-Mail verwandeln. Dabei werden mehrere Arten von Gate-Software unterstützt.
- * Einlagern von Binärdateien. Wenn die Mail das Attribut 'file-attach' gesetzt hat, wird die zum Betreff korrespondierende Binärdatei versandt. Unter Z3.8 wird ein eventuell vorhandener Text ignoriert, unter ZConnect wird er als Kommentar der Mail vorangestellt. Beim Einlagern wird der Pfad der Mail entfernt, falls das Konfigurationselement 'zer.killpath' auf 'Yes' gesetzt ist.
- * Erzeugung von RFC-Headern nach Gruppen einstellbar. Dabei werden bei Mails an MAPS oder Mails die das Attribut 'receipt-request' gesetzt haben keine Header generiert.
- * Automatische Erzeugung Eindeutiger Message-IDs für Crosspostings unter Z3.8.

Aufruf des Exporters Nun mach, ey!

1.50 UMSZer.guide/Aufruf des Exporters

Aufruf des Exporters

=====

Der Exporter ZerExport wird von der Shell mit folgender Schablone gestartet:

```
ZerExport FILE,PRETEND/S,NAME/K,PASS/K,SERVER/K,BIT/K/N,ZCO/S
```

Die Bedeutung der Parameter:

'FILE'

Dieser Parameter gibt den Namen des Netcallpuffers an. Falls FILE nicht gesetzt wird, erzeugt ZerExport in einem voreingestellten Verzeichnis eine Datei 'PUFFER'.

'PRETEND/S'

Pretend mode. Der Exporter ruft immer die Funktion CannotExport() auf, das Exported Flag wird also nicht gesetzt. (Dieser Schalter ist nur für Debugging gedacht und sollte vom Anwender nicht benutzt werden.)

'NAME/K'

Der Username, unter dem sich der Zerberus-Exporter in die UMS-MB einloggen soll

'PASS/K'

Das zugehörige Passwort.

'SERVER/K'

Zur Angabe von bestimmten UMS-Servern.

'BIT/K/N'

Gibt ein zusätzliches Bit aus den User-Flags des Zerberus-Users zur Selektion der zu exportierenden Nachrichten an. Damit kann man durch andere Tools (z. B. 'sumsset') eine Vorauswahl treffen, etwa beim User Zerberus auf bestimmten Messages das 'UFlag3'-Bit setzen (siehe 'sumstl.doc') und als 'BIT'-Parameter 3 angeben. Die Bitnummern entsprechen dabei immer dem 'UFlagX' unter 'sumsset'. (Da hier gewisse Nebeneffekte auftreten können, da z. B. Bit 4 dem Archiv-Flag entspricht, empfiehlt es sich, nach dem Exportieren die Flags wieder zu löschen.)

'ZCO/S'

Wenn dieser Schalter gesetzt wird, erwartet ZerExport den String 'zco' am Anfang von Brettnamen, anstelle des normalen 'zer'.

1.51 UMSZer.guide/Der Servercall

Der Servercall

Damit ein Servercall korrekt abläuft, muß ein entsprechendes Skript zusammengestellt werden, daß den bei ZerExport erwähnten Mechanismus verfolgt. Angenommen, die Up-/Download Verzeichnisse heißen 'UPLOAD:' bzw. 'DOWNLOAD:', könnte das Skript etwa so aussehen:

```
.key ,

; Stack setzen, besonders ZerImport barucht ein wenig...
stack 50000

; PUFFER erstellen/Update und packen
ZerExport UPLOAD:PUFFER
if exists UPLOAD:Out.lha
    delete UPLOAD:Out.lha
endif
if exists UPLOAD:PUFFER
    lha -2 a UPLOAD:Out.lha UPLOAD:PUFFER
else
    echo "!Kein PUFFER"
    quit 5
endif

lab sbegin

; Download-Verzeichnis säubern
delete DOWNLOAD:(out.txt|PUFFER)

; Servercall
setenv Servercallergebnis 0

; Das Servercallprogramm sollte bei Erfolgreichem Servercall die
; Environment-Variable 'Servercallergebnis' auf 1 setzen. Das hier
; verwandte Programm CallZer liegt bei Version 2.2 als Goodie dem
; Originalarchiv bei.
CallZer TRON.cnf onsucc "setenv Servercallergebnis 1"

; Done.->entpacken

; Falls der Servercall nicht erfolgreich war, nochmal Versuchen
if 'getenv Servercallergebnis' eq 0
    echo "Fehler beim Servercall."
    echo "Noch ein Versuch"
    skip back sbegin
endif

; Dieser Punkt wird nur nach erfolgreichem Call erreicht.

; Letzten Puffer als 'PUFFER.bak' behalten. Dadurch, daß der
; Upload-Puffer schon an dieser Stelle gelöscht wird, können
; unter ZConnect die Empfangsbestätigungen (zer.ebpuffer)
; direkt in den nächsten Upload-Puffer geschrieben werden.

if exists UPLOAD:PUFFER.bak
    delete UPLOAD:PUFFER.bak
endif
rename UPLOAD:PUFFER as UPLOAD:PUFFER.bak

if exists DOWNLOAD:out.txt
    ; Alten Puffer löschen
    if exists DOWNLOAD:PUFFER
        delete DOWNLOAD:PUFFER
```

```

endif
; Entpacken
lha x DOWNLOAD:out.txt DOWNLOAD:
; Wenn vorhanden, dann einsortieren
if exists DOWNLOAD:PUFFER
    ZerImport DOWNLOAD:PUFFER
    skip AllRight
endif

; Wenn der angekommene Puffer leer ist, hat out.txt die Größe 1 und
; alles ist in Ordnung
if `!list DOWNLOAD:out.txt lformat %L` eq 1
    echo "Leeren Puffer erhalten."
    skip AllRight
endif
endif

echo "!Fehler beim Entpacken"
quit 5

lab AllRight

echo "!!!!!! OK !!!!!!"

```

In bisherigen Versionen der Doku stand hier ein Anfänger-Abschrecker-Text. Ich bin allerdings der Meinung, daß das Skript (mit gewissen Anpassungen der Datei- und Verzeichnisnamen) so laufen sollte und daher jetzt nur noch ein *Benutzung auf eigene Gefahr!*

Allerdings wendet sich das ganze ja an DFÜler, also sollte niemand Probleme haben in Notfällen Hilfe aus dem Netz zu bekommen.

1.52 UMSZer.guide/Vergangenheit

Vergangenheit

Eine Liste der Versionen und was dann geschah.

2.0 beta

Erste veröffentlichte Version, leider mit ein Paar üblen Bugs behaftet. (Warum tauchen Bretter vor dem Kopf bloß nicht in der Brettliste der Serverbox auf? Mich wundert, daß der Exporter überhaupt irgendwo lief.)

2.05 beta

- * Bugfix: keine Headergenerierung mehr bei Binärdateien
- * Bugfix: Der Fehler 209, der vorher überall auftrat, sollte jetzt nicht mehr vorkommen. Der leere Alias kann auch wegfallen.
- * Message-IDs werden jetzt für Mail an mehrere Personen und Crosspostings eindeutig erzeugt und Referenzen darauf wieder

zurückgewandelt.

- * Zusätzlicher Parameter 'BIT' beim Exporter. (Vorschlag von Matthias Scheler)
- * Bugfix: Importer stürzte bei leerem Puffer mit Division durch Null ab.
- * Neue Einstellung 'u' bei Konfigurationselement 'zer.namemode'. (Vorschlag von Michael Leun)

2.10 public

- * Legal-Status geändert.
- * Bugfix: Bei nicht angegebenem Realname im RFC-Header-Eintrag 'Reply-To' wurde u. U. ein Falscher Name eingetragen.
- * Bugfix: Der Footerfilter funktioniert jetzt.
- * Unbekannte RFC-Header-Einträge werden jetzt in 'Comments' aufgenommen.
- * Konfigurationselemente 'zer.analyze', 'zer.bincheck', 'zer.ignorebin', 'zer.namemode', 'zer.routes' auf Brettformat umgestellt.
- * 'zer.username' kann nun den Namen einer Box angehängt bekommen, und gilt dann nur, wenn dieser dem aktuellen 'zer.serverbox'-Eintrag entspricht. (Für Point-Betrieb an mehreren Boxen.) (Nach einem Hinweis von Henning Sauer.)
- * Flags für 'zer.uucpgate' eingeführt. (Nach Vorschlägen von Henning Sauer ('o') und Michael Leun ('n'))
- * Unterstützung des deutschen Datumsformates implementiert.
- * Fortführung der Headeranalyse des Importers über Leerzeilen hinweg hinzugefügt.

2.20 public

- * ZConnect...
- * Aus dem Header 'Program-Id' wurde 'X-Mailer'
- * Kleine Umstellung der RFC-Analyse
- * Auslagern von Binärdateien funktioniert nun auch mit Locks auf den Verzeichnissen (der einzige echte Bug von 2.1 ;) Außerdem werden Dateinamen nun gekürzt und bei laufenden Nummern ein Punkt vorangestellt.
- * Diverse Fehler in der Dokumentation beseitigt.

2.22

- * Doku-Bug: 'zer.idmagic' wird unter ZC natürlich nicht benutzt.

- * Weitere Doku-Updates. Vor allem die Kapitel über den Im- und Exporter sind endlich angepaßt.
- * Diverse Bugfixes.
- * Neue Konfig-Elemente `'zer.errlog'`, `'zer.recfilt'`
- * Konfig-Element `'zer.mailfolder'` und weitere Ergänzung von `'zer.username'`
- * `'zer.realname'`
- * Diskussionsumleitung in mehrere Bretter oder an den Absender wird jetzt unterstützt.

1.53 UMSZer.guide/Zukunft

Zukunft

Geplant ist:

- * Schaltbare Zeichenkonvertierung des Exporters
- * Konvertierbare Art des Auslagerns für Binärdateien
- * Allgemeines Gate-Definitions-Format zur Userabhängigen festlegung von beliebig vielen Gateways in andere Netze mit verschiedenen Konvertierungen, etc. Vorschläge dazu sind ausdrücklich erwünscht.
- * Was immer mir vorgeschlagen wird...

1.54 UMSZer.guide/Support und Dank

Support und Dank

Alles ist fehlerhaft... Natürlich wird UMSZer da keine Ausnahme machen. Bitte beachtet bei Bugreports folgendes:

- * Schickt mir eine möglichst genaue Beschreibung des Fehlers
 - * Fehler sollten reproduzierbar sein
 - * Bei Fehlern des Importers möglichst entsprechende Teile des Puffers mitschicken
 - * Bei Exporter-Fehlern möglichst die entsprechende Mail abspeichern und mitschicken
 - * Bei Erstinstallation oder soeben erfolgter Umstellung auf
-

ZConnect, möglichst die Konfiguration mitschicken.

Fragen zu allem was mit UMS zu tun hat, stellt man am besten in
'de.comm.software.ums'; wer keinen Usenet-Zugriff hat, sollte sich bitte
darüber klar werden, zu *welchem* Programm er Hilfe benötigt und die
entsprechenden Autoren direkt ansprechen.

Mein Dank gebührt folgenden Personen:

- * Matthias Scheler und Markus Stipp für UMSZer Version 1 sowie
programmiertechnische Unterstützung. Außerdem für IntuiNews,
was wohl der Hauptgrund ist, UMS zu benutzen.
 - * Martin Horneffer für UMS.
 - * Ganz besonders: (alphabetisch) Hartmut Goebel, Henning Hucke, Mario
Kemper, Wolfram Kuss, Michael Leun, Peter Pommeranz, Henning Sauer
und Christoph Viethen für das Beta-Testing und viele konstruktive
Vorschläge.
 - * Allen DFülern in de.alt.comm.ums und /T-NETZ/ZCONNECT/#? für die
Beantwortung meiner ewigen Fragen.
 - * Detlef Wuerkner für die Testpuffer
 - * Peter Pommeranz und Helmut Neumann für die beste Box der Welt ;-)
-