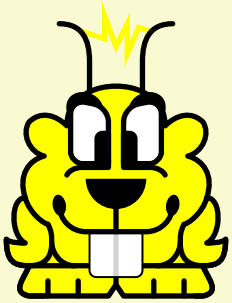
Hilfe für den Mail- und Newsserver Hamster Classic

Erarbeitet durch mehrere Autoren

PDF-Version: Steffen Hoffmann

Letzte Änderung:
24. Januar 2014*



Das folgende Dokument wurde ursprünglich durch mehrere Autoren als Hilfe für den Mail- und Newsserver Hamster Classic[a] im WinHelp32-Format erarbeitet,[b] siehe auch »Über die deutsche Hamsterhilfe« und unter Danksagungen. Ich habe die Vorlage in das TEX-Format übertragen und daraus die PDF-Version erstellt. Dazu waren jedoch einige kleinere Änderungen notwendig. Vor allem funktionieren alle Popups nicht mehr. Sie wurden zum größten Teil in ein Glossar umgewandelt. *Siehe zusätzlich auch Tipps zur Benutzung der PDF-Version dieser Hilfe.*

Wie der gesamte Hamster und mit ihm die originale Hilfe steht auch die PDF-Version unter der MIT-Lizenz.

Steffen Hoffmann

[a] Hamster Classic: http://tglsoft.de/freeware_hamster.html
[b] Herunterzuladen unter: http://www.usenet-abc.de/wiki/Team/HamHelp

Kurzinformation über den Hamster

Hamster ist ein lokaler Mail- und Newsserver für Windows-Plattformen (alle Windowsversionen der Windows-9x- und der Windows-NT-Familie, nicht aber für Windows 1.x–3.x*). Diese Freeware richtet sich an alle, die komfortabel Mail und News mit einem einzigen Programm verwalten wollen. Der Hamster sammelt Mail und News von mehreren Servern ein und stellt sie zusammen lokal zum Abholen per Client (Mailreader/Newsreader) bereit. Ideal ist er somit für Netzwerke, da mehrere Benutzer eingerichtet werden können, die auf den Hamster zugreifen können und der Datenbestand bei News nur einmal vorhanden sein muss.*

* Das entspricht Version 2014-01-09 der Hamsterhilfe im WinHelp32-Format.

Inhalt

Über die deutsche Hamster-Hilfe . . 4

Einführung 4

Installation 5

Menüs 7

Datei 7

Globale Gruppenlisten neu erzeugen 7

Aufräumen 7

News-History erneuern 7

Zähler/Protokoll löschen 8

Aktualisiere Menüeinträge 8

Kennworte neu laden 8

Beenden 8

Einstellungen 8

Grundeinstellungen 8

Optik 8

Protokolle 10

Online-Menü 11

Interne Gruppen 12

SSL 13

Verschiedenes 14

Benutzerverwaltung & Passworte 15

Lokale Benutzer 15

Aliase für (lokale) Benutzer 18

Passworte 18

Lokale Server 19

Allgemeines 19

NNTP 21

POP3 24

IMAP 25

SMTP 27

Fernzugriff 29

Automatische Abläufe 30

Allgemeines 30

Aktionen 31

Lade Artikel über ihre MID . 32

News: Server, Gruppen & Ladeaufträge 32

Einstellungen 32

Newsserver 33

Newsgruppen 36

News-Ladeaufträge 37

News: Killfile-Protokoll/-Einstellungen 38

Submenü Datei 40

Submenü Einträge 41

Reiter Kill-/Scorefile 41

Mail: Server konfigurieren 42

POP3-Einstellungen 42

POP3-Mailserver 43

SMTP-Einstellungen 47

SMTP-Mailserver 47

Mail: Killfile-Protokoll 49

Aktiviere/Deaktiviere erweiterte Einstellungen 52

Spezielle Dateien 52

Konfiguration neu laden 52

Online 52

1–9: … 52

Alle Server (News+Mail) … . . . 53

⟨Einzelner Server⟩ 53

DFÜ-Netzwerk 53

DFÜ-Verbindung schließen 53

Alle Aufträge beenden 53

Skript 53

⟨Verfügbare Skripte⟩ 53

Skripte und Module verwalten . . 53

Alle Skripte beenden 54

Lokale Server 54

Start/Stop News-Server (NNTP) . 54

Start/Stop Mail-Server (POP3) . 54

Start/Stop Mail-Server (SMTP) . 54

Start/Stop Mail-Server (IMAP) . . 54

Start/Stop Telnet-Server 55

News-/Mail-Verzeichnisse 55

Hauptfenster & seine Kontextmenüs 58

Protokolle für NewsJobsListe . . 58

Protokolle für ausgewählte Aufträge 58

Kontextmenü 58

Kopiere Einzelzeile in die Zwischenablage 58

Kopiere Protokoll in die Zwischenablage 58

Protokoll(e) leeren 59

hs2-Shell 59

Dateien & Verzeichnisse 61

Dateien im Hauptverzeichnis 61

Hamster.ini 63

[Directories] 64

[Setup] 65

[Main] 78

[Actions] 79

Sonstige Sektionen 83

IPAccess.hst 87

Scores.hst 89

MailFilt.hst 96

MAlias.hst 106

Zeichensatzkonvertierungsdateien 108

Password.!!! 109

Dateien in »⟨Hauptverzeichnis⟩\Server« . . . 110
Dateien in »⟨Hauptverzeichnis⟩\Server\⟨Servername⟩« . *110*
Server.ini . . . 110
Dateien in »⟨Hauptverzeichnis⟩\Groups« . . . 114
News.out . . . *114*
News.err . . . *114*
Dateien in »⟨Hauptverzeichnis⟩\Groups\⟨Newsgruppenname⟩« . . . *115*
Dateien in »⟨Hauptverzeichnis⟩\Mails« . . . 117
Dateien in »⟨Hauptverzeichnis⟩\Mails\Mail.out« . . . *117*
Dateien in »⟨Hauptverzeichnis⟩\Mails\admin« . . . *117*
Dateien in »⟨Hauptverzeichnis⟩\Mails\⟨Benutzer⟩« . . . *117*
Dateien in »⟨Hauptverzeichnis⟩\Logs« . . . 119

Skripte & Programmierung . . . 120
Hamsterskripte (»hs2«) . . . 120
Anweisungen und Kommentare . *121*
Präprozessor-Anweisungen 122
Variablen, Konstanten und Datentypen . . . *123*
Variablen . . . 123
Konstanten . . . 125
Datentypen . . . 125
Ausdrücke und Operatoren . . . *126*
Boolesche Ausdrücke . . . 129
Sprünge . . . *130*
Verzweigungen . . . *131*
Schleifen . . . *133*
Prozeduren und Funktionen . . . *135*
Eigene Prozeduren u. Funktionen . . . 136
Eingebaute Prozeduren . . . 139
Eingebaute Funktionen (allgemein) . . . 141
Eingebaute Funktionen (Server) . . . 181
Nebenläufige Skripte . . . *205*
Nebenläufigkeit . . . 205
Synchronisation . . . 205
Synchronisation durch Events 206
Module . . . *212*
Definition . . . 212
Gebrauch von Modulen . . . 214
Aufruf per Action . . . *214*
Hamster-Event-Objekte (Win32) . . 215
Hamster-Mutex-Objekte (Win32) . . 216
OLE-Schnittstelle für andere Programme . . . 217

Sonstiges . . . 236
Datenfluss von Newsartikeln . . . 236
Fehlermeldungen & Statuscodes . 236
Häufiger vorkommende Fehler . *237*
Hamster-Protokoll . . . 237
Hamster-Meldungen . . . 237
WinSock-Fehler . . . 237
WinSock-Fehler . . . *239*
Fehler RAS/DFÜ . . . *240*
Win32-System-Fehler . . . *243*
Win32-Netzwerk-Fehler . . . *263*
SSL-Fehler . . . *269*
Server-/Protokoll-Antwortcodes *274*
Fernsteuerung des Hamsters . . . 276
Grundlagen & Installation . . . *276*
Syntax . . . *277*
Kommandozeilenoptionen . . . 279
Linkliste . . . 280
Versionsinformationen . . . 281
Danksagungen . . . 285
Lizenzen, Maintainer . . . 286

FAQ: Häufige Fragen & Probleme . . 287

Glossar . . . 396

Register . . . 399
Gesamtindex . . . 399
Optionen der »Hamster.ini« . . . 403
Hamsterscript-Anweisungen . . . 406
Anweisungen der OLE-Schnittstelle 409

Tipps zur Benutzung der PDF-Version dieser Hilfe:

Wenn es in Ihrem PDF-Reader möglich ist und nicht bereits vorhanden, ist die Aktivierung von zusätzlichen Navigationssymbolen sehr sinnvoll. Im vermutlich am häufigsten eingesetzten *Adobe Reader* geschieht das am einfachsten dadurch, dass nach Rechtsklick auf die Werkzeugleiste im anschließend erscheinenden Kontextmenü unter dem Menüpunkt »Seitennavigation« die gewünschten Symbole aktiviert werden, vor allem »Vorherige Ansicht« und »Nächste Ansicht« dürften nützlich sein. *(Die genaue Bezeichnung ist abhängig von der Version des Adobe Readers. In der »normalen« Reader-Version können die Symbole auch über das Menü »Anzeige« eingeblendet werden, in der Plugin-Version funktioniert nur die oben beschriebene Methode.)*

Die gesamte Hilfe ist mit Linkankern versehen, die es ermöglichen, mit dem internen PDF-Viewer im Firefox[a] oder einem beliebigen Webbrowser und dem Adobe-Reader-Plugin *(soweit bekannt, mit keinem anderen PDF-Reader/-Reader-Plugin!)* auf einen bestimmten Teil der Hilfe zu verweisen. Die Linkanker sind als Tooltip zu erkennen beim Überfahren mit der Maus über den linken Rand neben einem Abschnitt, einer Option oder einem Befehl (in anderen PDF-Readern sind eventuell kleine Markierungen zu sehen). Fehlt ein Tooltip, gibt es auch keinen Linkanker.

Beispielsweise sind diese Tipps zur Benutzung der PDF-Hilfe mit dem Linkanker »`pdftipp`« versehen. Dann könnte man auf sie verweisen mit

`http://www.example.org/Hilfe_zum_Hamster_Classic.pdf#pdftipp` *oder lokal*
`file:///⟨lokaler/Pfad/zur/Datei⟩/Hilfe_zum_Hamster_Classic.pdf#pdftipp`
(»/« als Verzeichnistrenner, funktioniert ***nicht*** *mit dem* Internet Explorer*!).*

Ebenfalls ist es möglich, auf eine Seite zu verweisen mit folgender Syntax

`http://www.example.org/Hilfe_zum_Hamster_Classic.pdf#page.⟨Seitenzahl⟩`

Dass sich dabei der Fokus so ändert, dass die gesamte Seite angezeigt wird, ist Absicht. *(Auch das ist außer im* Internet Explorer *lokal möglich. Wird stattdessen »page=⟨Seitenzahl⟩« verwendet, also mit »=« anstelle des Punktes, bleibt der Fokus unverändert.)*

[a] Auch erhältlich für Google Chrome, SRWare Iron und Opera ab Vs. 15: PDF-Viewer im Chrome Web Store.

Über die deutsche Hamster-Hilfe

Aufgebaut zusammen mit der Entwicklung des ursprünglichen Hamsters von Jürgen Haible (`http://www.elbiah.de/`), massiv erweitert und bis Oktober 2005 bearbeitet von Jörn Weber (`http://www.joernweber.de/`), seitdem bis August 2010 bearbeitet von Michael Jaritz (`http://zielgra.de/`), aktuell gepflegt durch Steffen Hoffmann (`http://speravir.website.org/`).

Einführung

Zum Einstieg in das Programm empfehlen sich folgende Hilfethemen: Die Installation beschreibt die ersten Schritte zur Programmnutzung. Unter der Menüstruktur finden Sie die Erläuterungen zu den einzelnen Menüpunkten; für die *Konfiguration* gehen Sie zunächst am besten durch die verschiedenen Punkte des Menüs »Einstellungen«.

Zum Thema Sicherheit finden Sie Hilfe im FAQ-Bereich dieser Hilfe-Datei.

Hinweise auf zusätzliche Informationen, Beispiel-Skripte und komplette Zusatzprogramme sind unter Links für Informationen und Hilfen zum Hamster zu finden.

Wichtige Änderungen zwischen den Hamster-Programmversionen finden Sie unter Versionsinformationen (eine genaue Auflistung der Änderungen ist unter »`changes_en.txt`« bzw. »`changes_de.txt`« im Hamsterverzeichnis zu finden).

Um Aktivitäten zu automatisieren, ist eine Skriptsprache, Hamsterscript, Version 2 (hs2), eingebaut. Falls diese Möglichkeit nicht reicht, stehen passende Programmierobjekte für den Zugriff aus Perl, VB-Script, Delphi oder sonstigen Windows-Programmiersprachen zur Verfügung. Zum einen die Hamster-Event-Objekte (Win32) und die Hamster-Mutex-Objekte (Win32), aber vor allem auch die OLE-Schnittstelle für andere Programme, welche praktisch alle relevanten Hamster-Funktionen der eingebauten Skriptsprache enthält.

Informationen zum Aufbau des Hamsterordners finden Sie unter »Dateien und Verzeichnisse« (in dieser Hilfe wird weitgehend die klassische Bezeichnung »Verzeichnis« anstelle von »Ordner« verwendet).

Für spezielle Konfigurationsdateien gibt es noch die Hilfethemen »`IPAccess.hst`« (Zugriffe auf den Hamster auf Basis der IP-Regeln), »`Scores.hst`« (Score-/Killfile für eingehende Newsartikel), »`MailFilt.hst`« (Score-/Killfile für eingehende Mails), »`Hamster.ini`« (Hamster-Einstellungen, die nur teilweise in den Konfigurationsmenüs enthalten sind) und »`Server.ini`« (für server-spezifische Einstellungen).

Eine Auflistung möglicher Fehlermeldungen – sortiert nach Fehlerbereichen – finden Sie unter Fehlercodes und -meldungen. Häufig auftretende Fragen werden im Abschnitt FAQ erläutert.

Somit fehlen nur noch Danksagungen und Lizenz und Urheberrechte. Für weitere Informationen zur Version des Hamsters (Maintainer, Autoren, Download usw.) siehe auch das Hamster-Menü »Hilfe« → »Info«.

Installation

1. Legen Sie einen neuen Ordner für den Hamster an, z. B. »`c:\hamster\`«.

   Es wird dringend geraten, den Hamsterordner außerhalb von »`%ProgramFiles%`« anzulegen – bei 64-bit-Windows-Varianten auch außerhalb von »`%ProgramFiles(x86)%`« –, weil sonst die Funktionalität des Hamsters gewöhnlich massiv behindert wird:

   Durch die Rechteverwaltung in allen auf Windows NT aufbauenden Windows-Versionen bis inkl. Windows XP wird dem einfachen Benutzer der Schreibzugriff standardmäßig verweigert, er kann aber durch einen Administrator Schreibrechte verliehen bekommen. Seit Windows Vista wird

durch die Benutzerkontensteuerung (UAC, »User Account Control«) auch dem Administrator der Schreibzugriff zunächst verwehrt und muss jedes Mal bestätigt werden.

2. Entpacken Sie das heruntergeladene Archiv in dieses neue Verzeichnis.

3. Starten Sie die Datei »`hamster.exe`« im neuen Verzeichnis und wählen Sie im erscheinenden Dialog die gewünschte Sprache an. Beachten Sie, dass der Hamster, um den OLE-Server beim System anzumelden, beim ersten Start unter Windows XP und jüngeren Versionen einmalig Admin-Rechte benötigt. Diese kann man erlangen, indem man ihn nach Klick mit der rechten Maustaste über den Kontextmenüpunkt »Ausführen als« startet, *seit Windows Vista muss dies auch in einem »Administrator«-Konto gemacht werden!* Damit läuft der Hamster. Die benötigten Unterverzeichnisse u. ä. legt er beim ersten Starten von selbst an. Das Hamster-Hauptfenster sehen Sie, wenn Sie auf das gelbe Nagetier im Systray doppelklicken.

   Für die erste Konfiguration gehen Sie am besten durch die verschiedenen Punkte des Menüs »Einstellungen«.

4. Wenn Sie eine bestehende Hamster-Installation updaten wollen, brauchen Sie nur gegebenenfalls den laufenden Hamster zu beenden und dann das neue Archiv in das Hamster-Verzeichnis zu entpacken. Alle Dateien können überschrieben werden, ohne dass Einstellungen verloren gehen. *Ausnahme:* Sollten Sie die Sprachdatei »`hamster_XX.dat`« bearbeitet haben, würden Sie diese Änderungen verlieren (»XX« steht für ein Sprachenkürzel, »de« für deutsch).

   Siehe zusätzlich auch den FAQ-Beitrag »Hamster aktualisieren (Update)«.

Setup für das Passwort des Administrators:

Für den Zugriff der News- und E-Mail-Reader ist es erforderlich, ein Passwort für den automatisch angelegten Nutzer »admin« zu vergeben.

1. Gehen Sie auf »Einstellungen« → »Benutzerverwaltung & Passworte« ⇒ »Lokale Benutzer«. Diese Seite enthält eine Liste mit den Namen aller zur Zeit eingetragenen lokalen Benutzer.

2. Wählen Sie den Benutzer »admin« an und rufen Sie den Bearbeitungsdialog über den Button »Bearbeiten« auf (Achten Sie darauf, dass Sie nicht die Gruppe »Admin, Administration« wählen).

3. Links sehen Sie ein Passwort-Feld und daneben einen Button »Ändern«, mit dem Sie ein neues Passwort festlegen können. Das Passwort muss zwei Mal eingegeben werden. Nach der Eingabe muss im »Passwort«-Feld die Meldung »{Gesetzt}« angezeigt werden. Das Passwort selbst wird nicht angezeigt.

4. Mit OK speichern Sie die Änderungen. In der Liste der lokalen Benutzer sollte dann rechts vom »admin« ein »{Gesetzt}« zu sehen sein (statt »{Leer}«).

Menüs

Bitte beachten Sie, dass einige Einstellungen nur für erfahrene Anwender gedacht sind. Diese Einstellungen sind nur zugänglich, wenn der Menüpunkt »*Aktiviere spezielle Einstellungen*« aktiviert wurde. Anfängern wird empfohlen, sich vor Benutzung dieser Optionen genau über deren Funktion zu informieren.

Menü: Datei

Globale Gruppenlisten neu erzeugen

Die Liste der News(gruppen)-Beschreibungen (».\Server\alldescs.txt«) wird mit diesem Menüpunkt erneuert. Diese Liste wird aus den jeweiligen Gruppenbeschreibungslisten aller Server gebildet.

***Warnung:* Während dieser Aktion sollte der Hamster nach Möglichkeit nicht anderweitig genutzt werden, da ein Absturz o. ä. mit einer gewissen Wahrscheinlichkeit die gerade geöffnete Datei ernsthaft schädigen kann.**

Aufräumen

Mit dieser Funktion können alle veralteten Daten entfernt und die Dateien aufgeräumt/verkleinert werden. Ab wann die Daten veraltet sind, kann im Menü »Einstellungen« → »Lokale Server« eingestellt werden (unter den einzelnen lokalen Servern). In »Einstellungen« → »Automatische Abläufe« ⇒ »Allgemeines« können Sie auch einstellen, dass der Hamster das automatisch erledigt.

***Warnung:* Während dieser Aktion sollte der Hamster nach Möglichkeit nicht anderweitig genutzt werden, da ein Absturz o. ä. mit einer gewissen Wahrscheinlichkeit die gerade bearbeitete Datei ernsthaft schädigen kann.**

News-History erneuern

Diese Funktion erstellt die »History«-Dateien neu. Sie enthalten die Message-IDs aller Artikel und verhindern damit das mehrfache Laden des gleichen Artikels. Eine Neuerzeugung ist nur dann sinnvoll, wenn die Erkennung plötzlich Fehler machen sollte oder man durch Schäden an einer Gruppendatei in die Verlegenheit kommt, etliche Postings noch ein zweites Mal holen zu müssen.

Falls man den Gruppen eine geringe Haltezeit (wenige Tage) verordnet hat, ist es auch bei einer neuerzeugten History durchaus wahrscheinlich, dass einzelne Artikel doppelt eintreffen, da der erste

Artikel schon nicht mehr in der Gruppendatei war und somit auch seine Message-ID nicht mehr in die History eingebaut werden konnte.

Achtung! Die Erstellung dieser Datei kann erhebliche Zeit in Anspruch nehmen.

Zähler/Protokoll löschen

Die Menüunterpunke erlauben es, wahlweise nur das Protokoll-Fenster (inkl. Warnungen & Fehler) zu leeren oder die Anzeigen in der Statuszeile zurückzusetzen (Anzahl empfangener Mails, Postings/Artikel, Bytes) oder beides auf einmal.

Innerhalb von Hamsterskripten kann diese Funktion auch gezielt aufgerufen werden, Näheres ist unter den entsprechenden Hilfeabschnitten zu finden.

Aktualisiere Menüeinträge

Aktualisiert die verschiedenen dynamisch erzeugten Menüs. Dies betrifft speziell das Menü »Online« und das Menü »Skripte«. Falls z. B. ein neues Skript im entsprechenden Verzeichnis angelegt wurde, ruft dieser Menüpunkt die Hamstererkennung noch einmal auf, so dass auch das neue Skript aufgerufen werden kann bzw. eine Bearbeitung über den entsprechenden Menüpunkt möglich ist.

Kennworte neu laden

Aktualisiert die Passwörter aus der Passwort-Datei. Alle bisher vom Hamster anderweitig gemerkten Passwörter gehen dabei verloren.

Beenden

Außer dem Mausklick auf den Menüeintrag funktioniert auch die Tastenkombination Strg + Q.

Menü: Einstellungen

Grundeinstellungen Strg + G

Optik

Sprache auswählen

Hier kann man seine bevorzugte Sprache einstellen. Sollten weitere Übersetzungen hinzukommen, reicht das Kopieren der Sprachdatei in das Hamsterverzeichnis – die neue Sprache

sollte dann direkt im Auswahldialog zu finden sein. Eine Sprachdatei hat einen Dateinamen im Format »`hamster_XX.dat`«, wobei »XX« für den Ländercode steht (»de« für deutsch).

Hamster:

Start als Tray-Icon

Diese Option bestimmt, ob der Hamster beim Starten sofort minimiert in der Taskleiste abgelegt wird oder das Hauptfenster geöffnet werden soll.

Startbildschirm anzeigen

Ist diese Option aktiv, wird beim Starten des Hamsters ein Hinweisfenster eingeblendet.

Fenster schließen minimiert Hamster

Ist diese Option aktiv, wird der Hamster beim Schließen des Hauptfensters nicht beendet, sondern steht weiterhin in der Taskleiste zur Verfügung. Mit einem Doppelklick kann das Hauptfenster wieder angezeigt werden.

Hinweis, wenn unversandte Mails/Artikel

Ist diese Option aktiv und befinden sich noch unversandte Mails oder Artikel im Postausgang, wird ein entsprechender Hinweis eingeblendet, wenn der Hamster beendet werden soll.

Sichtbare Menüeinträge

Hier können Sie einstellen, welche Menüs des Hamsters angezeigt werden sollen.

Hauptfenster: hs2-Shell für direkte Eingabe

Hier können Sie die hs2-Shell, eine zusätzliche Eingabezeile für die direkte Eingabe von hs2-Befehlen, im Hauptfenster aktivieren.

Protokollanzeige im Hauptfenster

Nicht mehr Zeilen anzeigen als:

Diese Option legt fest, wie viele Zeilen die Protokoll-Datei enthalten darf, bevor der Hamster die ältesten Einträge löscht, um Platz für die neuesten Protokollzeilen zu erhalten. Der Standard von 500 Zeilen sollte für die meisten Anwendungsfälle ausreichend sein.

Folgende Einträge anzeigen:

Hier kann ausgewählt werden, welche Meldungstypen in das Bildschirmprotokoll geschrieben werden sollen. Die Farben der einzelnen Typen entsprechen dabei der Farbe der Meldungen innerhalb des Hamsters.

Hinweis: Die weiteren Anzeigen im Hauptfenster (Aufträge, Warnungen & Fehler, Einzelner Thread) werden aus dem Protokollfenster heraus gefüllt, die Abschaltung der entsprechenden Meldungstypen beeinflusst somit auch die Aussagekraft dieser Listen.

Protokolle aktualisieren:

Mit dieser Einstellung kann die Häufigkeit der Aktualisierungen in den Log-Fenstern des Hamsters justiert werden, um ein Flimmern der Fenster zu vermeiden. Diese Einstellung ist abhängig von der Verarbeitungsgeschwindigkeit des Rechners (Standard: 0,5 s).

Aktiviere »Warnungen und Fehler« nach der ersten Warnung bzw. dem ersten Fehler

Hier kann festgelegt werden, ob das Log-Fenster »Warnungen und Fehler« ständig oder nur bei Bedarf angezeigt werden soll.

Protokolle

Hauptprotokoll

Anzahl der Protokoll-Dateien:

Beim Starten des Hamsters wird jedes Mal eine neue Log-Datei (».\Logs\0.log«) angelegt und die bestehenden Dateien werden so umbenannt, dass der bisherige Name um »1« erhöht wird (aus »0.log« wird »1.log« usw.). Diese Option legt fest, wie viele Exemplare davon maximal aufbewahrt werden (Standard: 14).

Folgende Einträge protokollieren:

Hier kann ausgewählt werden, welche Meldungstypen in die Protokoll-Datei geschrieben werden sollen (Standard: bis inkl. »Detail«). Die Farben der einzelnen Typen entsprechen dabei der Farbe der Meldungen innerhalb des Hamsters. Im gespeicherten Protokoll fällt die farbliche Kennzeichnung weg.

Vorsicht bei der Verwendung der Option Debug! Es werden hierbei große Datenmengen auf der Festplatte abgelegt, da die Anzahl der Meldungen sich zumindest verdoppelt.

Öffne Protokoll:

Hier können das aktuelle und die 3 nächstjüngsten Protokolle mit dem Editor geöffnet werden, der im Menü »Einstellungen« → »Grundeinstellungen« ⇒ »Verschiedenes« eingestellt ist.

Zusätzliche Protokolle

Hier können sie die maximal zulässige Anzahl der Zeilen in den verschiedenen Protokolldateien festlegen (Standard: 0 = »keine Begrenzung«) und die Protokolldateien ggf. mit dem Editor bearbeiten, der im Menü »Einstellungen« → »Grundeinstellungen« ⇒ »Verschiedenes« eingestellt ist.

Online-Menü

Menüeintrag:

Hier kann man bis zu neun Einträge für das »Online«-Menü definieren. Jeder Eintrag definiert dabei entweder einen einzelnen Server oder eine Liste von Servern oder eine Trennzeile. Als Server stehen alle externen News-, POP3- und SMTP-Server zur Verfügung. Im Falle von News-Servern bildet das Verschicken eigener Artikel und das Abholen neuer Artikel eine logische Einheit (Grundlage ist NNTP).

Glossar: POP3 SMTP NNTP

Solange kein Eintrag angewählt ist, stehen die weiteren Eingabefelder noch nicht zur Verfügung.

Zusätzliche Schalter:

Hoch
: Setzt den aktuell gewählten Eintrag in der Liste um eins höher, d. h. er wechselt die Position mit dem darüber liegenden Menüpunkt.

Runter
: Setzt den aktuell gewählten Eintrag in der Liste um eins tiefer, d. h. er wechselt die Position mit dem darunter liegenden Menüpunkt.

Trenner
: Setzt den Menütitel des aktuell gewählten Eintrags auf »-«, was den Menüpunkt als solchen deaktiviert, aber die Unterteilung der restlichen Menüpunkte erlaubt.

Löschen
: Löscht den Menütitel des aktuell gewählten Eintrags in der Liste, so dass er nicht mehr im Online-Menü auftaucht und die Zuordnung der Server gelöscht wird.

Titel:

Angezeigter Text im »Online«-Menü. Ein einzelner Bindestrich (»-«) als Text steht dabei für einen Menütrenner, ein leerer Titel steht für einen ungenutzten Menüpunkt. Falls es sich weder um einen Trenner noch um einen ungenutzten Menüpunkt handelt, kann man die gewünschten Server einstellen:

Verfügbare/Ausgewählte Server:

Hier kann aus der Liste der vorhandenen Server eine Auswahl zusammengestellt werden. Zur Verfügung stehen hier nur Server, die über das Menü »Einstellungen« → »News: Server, Gruppen & Ladeaufträge« ⇒ »Newsserver« bzw. »Einstellungen« → »Mail: Server konfigurieren« ⇒ »POP3-Mailserver« oder … ⇒ »SMTP-Mailserver« eingerichtet wurden.

Rechts stehen die bislang ausgewählten, links die ungenutzten Server. Mittels Doppelklick auf einen Eintrag oder über die Schalter »→ « und »← « kann die Auswahl wie gewünscht geändert werden. Bei Aktivierung des damit konfigurierten Online-Menüeintrages werden dann alle für

diesen Punkt ausgewählten Server angewählt und Artikel ausgetauscht, Mails empfangen oder Mails verschickt – je nach Art des Servers.

Interne Gruppen

Hier erfolgt die Zuordnung von internen Meldungen des Hamsters zu lokalen Gruppen. Um neue lokale Gruppen zuordnen zu können, müssen diese zuvor eingerichtet worden sein: entweder hier in diesem Dialog durch Mausklick auf das untere Auswahlfeld »Neue lokale Gruppe erzeugen« oder über den Menüpunkt »Einstellungen« → »News: Server, Gruppen & Ladeaufträge« ⇒ »Newsgruppen«.

Die möglichen internen Meldungen umfassen folgende Kategorien:

- Kopien von erfolgreich verschickten Postings: Alle Postings, welche vom externen Newsserver angenommen wurden.
- Postings, die nicht verschickt werden konnten: Diese Postings sind vom Newsserver abgelehnt worden, die entsprechende Fehlermeldung wird vom Hamster an den Anfang des Artikeltextes eingefügt.
- Tägliche Gruppenstatistiken: Diese umfassen zur Zeit das tägliche »[Hamster] Infos«-Posting, welches die abonnierten Gruppen inkl. Artikelzahl (einmal nach Name, einmal nach Artikelzahl sortiert) und die Liste der »unbenutzten« Gruppen umfasst.
- Hinweise auf Cancels in lokalen Gruppen: Falls in lokalen Gruppen mal Artikel gecancelt werden (= Posting mit Löschaufforderung), kommen die Bestätigungen in der hier eingestellten Gruppe an.
- Hinweise auf neue Gruppen der Newsserver: Getrennt nach Newsserver werden alle neuen Gruppen inkl. Beschreibung (Tagline) aufgeführt.
- Empfangene Postings für unbekannte Gruppe(n): Hier landen alle Postings, deren Header nur auf Gruppen verweist, welche nicht im Hamster abonniert sind. Solche Postings kommen z. B. dann zustande, wenn man Artikel anhand einer Message-ID nachladen lässt, die sich auf ein Posting in einer ansonsten nicht genutzten Gruppe bezieht.

  Glossar: Header
- Hinweise auf Lade-Fehler (gelöschte Gruppen): Falls Fehler beim Zugriff auf externe Newsgruppen auftreten, werden die entsprechenden Fehlermeldungen in diese Gruppe umgeleitet. In der Mehrzahl der Fälle bedeutet eine entsprechende Fehlermeldung, dass die Gruppe z. B. aufgrund einer Gruppenaufteilung oder -verschiebung vom benutzten externen Server genommen wurde.

Nach einer Erstinstallation wird alles in »internal.misc« gespeichert. Die Einstellung kann im obersten Feld »Standardgruppe« geändert werden.

Wie oben bereits erwähnt, wird mit dem Schalter »Neue lokale Gruppe erzeugen« eine neue hamster-interne Gruppe erzeugt, welcher dann direkt eine Aufgabe zugeordnet werden kann.

SSL

Achtung! Um SSL in Verbindung mit dem Hamster nutzen zu können, werden zwei zusätzliche DLL-Dateien benötigt; mehr dazu siehe in der SSL-FAQ.

Glossar: SSL

Zertifikate

Schlüsselpaar (für lokale Server notwendig):

Hier kann optional die Datei mit dem privaten X.509-Schüsselpaar im PEM-Format angegeben werden, mit welchem sich die lokalen Server beim Clienten authentifizieren.

X.509 PEM

Kennwort für privaten Schüssel:

Hier muss das zu dem privaten X.509-Schlüsselpaar gehörige Passwort angegeben werden.

Pfad mit Zertifikaten zwecks Überprüfung

Hier ist der Pfad zu dem Verzeichnis anzugeben, in dem sich die lokal vorhandenen Zertifikate zur Authentifizierung der Remote-Server (Server der Provider) befinden.

Datei mit Zertifikaten zwecks Überprüfung

Hier ist die Datei anzugeben, in dem sich die lokal vorhandenen X.509-Zertifikate zur Verifikation der Remote-Server (Server der Provider) befinden. Diese Option kann alternativ zur vorhergehenden Option »Pfad mit Zertifikaten zwecks Überprüfung« verwendet werden.

Verschlüsselung:

Hier kann ausgewählt werden, welche Verschlüsselungsstärke der Hamster akzeptieren darf. Empfohlen wird für symmetrische Verschlüsselung eine Verschlüsselungsstärke von mindestens 128 Bit. Die Schlüssellängen 56 Bit und 64 Bit entsprechen nicht mehr dem Stand der Technik und sind angreifbar.

Protokolle:

Hier kann eingestellt werden, welche Handshake-Protokolle verwendet werden dürfen. Als zuverlässig gelten zur Zeit SSL, Version 3, und TLS, Version 1. TLS ist das von der IETF (Internet Engineering Task Force) favorisierte Verfahren.

TLS

Externe Datei mit Zufallswerten für den PRNG:

Hier kann eine externe Datei mit Zufallswerten zum Initialisieren des Zufallsgenerators angegeben werden. Die Angabe einer externen Quelle zum Initialisieren des Zufallsgenerators ist nur empfohlen, wenn eine Quelle zur Verfügung steht, die besser ist als die der OpenSSL-Dateien. Nähere Infos hierzu sind auf der OpenSSL-Homepage (`http://www.openssl.org/`) erhältlich.

Erweiterte Einstellungen:

Hier können weitere sicherheitsrelevante Einstellungen vorgenommen werden. Diese sollten nur dann verändert werden, wenn man genau weiß, was man tut. Nähere Infos hierzu sind auf der OpenSSL-Homepage (`http://www.openssl.org/`) erhältlich.

Verschiedenes

Skript-Sprache

Endung hsc registrieren, um per Doppelklick starten zu können

Hier können Sie den Hamster beim Betriebssystem registrieren, damit zukünftig bei einem Doppelklick auf den Dateinamen der Hamster aktiviert wird, der das Skript dann automatisch ausführt.

Editor für die Bearbeitung von Textdateien

Dieses Feld legt fest, welche Anwendung für die Bearbeitung der Hamster-Dateien, wie z. B. Skripte und Konfigurationsdateien, verwendet werden soll (Standard: »`notepad.exe`«). Im ersten Feld ist der Programmname (Name der EXE-Datei ggf. inklusive Pfad, sofern nicht im Suchpfad von Windows enthalten) einzugeben, im zweiten die Parameterliste, die zumindest aus »%1« bestehen sollte, da ansonsten der Dateiname nicht übergeben würde.

Standard-Timeouts für alle externen Server

In diesen Feldern kann ein Timeout in Sekunden für alle Verbindungen zu externen Servern und für die Abarbeitung der Kommandos durch diese externen Server eingestellt werden (Standard: jeweils 120 s).

Die Einstellungen für einzelne Server (in deren Konfigurationsdialog) haben Vorrang vor diesen Einstellungen!

Anzahl paralleler Aufträge begrenzen

Hier können Sie die Anzahl der vom Hamster parallel zu verarbeitende Prozesse begrenzen, um die vom Hamster erzeugte Systemlast und Belastung der Anschlussleitung zu verringern (Standard: »0« = unbegrenzt).

Lokale Zeichensatztabelle

Glossar: Header

Hier legen Sie den für die MIME-Codierung von Headern zu verwendenden lokalen Zeichensatz fest. Er sollte mit dem tatsächlich in Windows eingestellten Zeichensatz übereinstimmen (Standard: »windows-1252«).

Benutzerverwaltung & Passworte Strg + P

Lokale Benutzer

Hier können die Benutzer für die lokalen Server konfiguriert werden. Diese können in verschiedenen Gruppen verwaltet werden.

Benutzername und Passwort werden für alle lokalen Server verwendet (mit den unten erwähnten Einschränkungen).

Für das Hinzufügen eines neuen lokalen Benutzers wählen Sie eine Gruppe, und dann den Schalter »**Neuer Nutzer**«. Die Voreinstellungen der gewählten Gruppe werden übernommen und die Gruppeneinstellungen als Standardwerte des Benutzers voreingestellt. Diese Einstellungen können jedoch durch Markieren des entsprechenden Feldes im jeweiligen Dialog überschrieben werden. Nach Eingabe des Benutzernamens wird automatisch in den weiter unten erläuterten Bearbeitungsdialog gewechselt.

Achtung: Mailboxnamen sollten nur Buchstaben und Zahlen, aber keine Sonder- oder Leerzeichen, enthalten und RFC 2821/2822 genügen. Folgende Zeichen sind zulässig:

[»!« | »#« | »$« | »&« | »'« | »+« | »-« | »_« |
»=« | ».« | »0«...»9« | »A«...»Z« | »a«...»z«]

Für das Hinzufügen einer neuen lokalen Benutzergruppe wählen Sie den Schalter »**Neue Gruppe**«. Der Name muss eindeutig sein.

Der Schalter »**Bearbeiten**« wechselt in den Bearbeitungsdialog des aktuellen Benutzers bzw. der gewählten Gruppe.

Mit dem Schalter »**Löschen**« löschen Sie die gewählte Gruppe oder den gewählten Benutzer. Sie können beim Löschen einer Gruppe wahlweise die Benutzer mitlöschen oder in die Gruppe »Admin« verschieben lassen.

Alle Schalter sind auch über das Kontextmenü erreichbar.

»**Auto-Aufklappen**« gibt an, ob die Ansicht der Gruppen/Benutzer automatisch voll aufgeklappt werden soll, oder ob man dies manuell machen will (sinnvoll bei sehr vielen Benutzern).

Besonderer Benutzer »nntpdefault«

Glossar: NNTP

Für den Zugang zum lokalen NNTP-Server ohne Authentifikation ist der Benutzer »**nntpdefault**« mit dem Passwort »*« (ein Stern) zu verwenden. Dadurch reicht es, im Newsclienten nur den Newsserver anzugeben. Sie sollten aus Sicherheitsgründen diesem Nutzer das Senden von E-Mails untersagen. Auch ist zu beachten, dass dann *jeder*, der Zugriff auf Ihren Rechner hat, News lesen und bei entsprechender Einstellung der Gruppen auch schreiben kann. Den Zugriff

auf Ihren Rechner können Sie über »Einstellungen« → »Lokale Server« ⇒ »Allgemeines« → »IPAccess.hst bearbeiten« steuern.

Bearbeitungsdialog Gruppen

In der **Beschreibung** der Gruppe kann man eine Bemerkung zur Gruppe schreiben oder eine vorhandene Beschreibung ändern.

In den »**Standard-Einstellungen für neue Benutzer**« werden die Voreinstellungen eines neu eingerichteten Benutzers der Gruppe festgelegt.

Glossar: NNTP

News (NNTP)

Mit der Einstellung »**Gruppen, auf die er nur lesend zugreifen kann**« wird festgelegt, auf welche Newsgruppen des lokalen Newsservers der Benutzer lesend zugreifen darf. Mit der Einstellung »**Gruppen, in die der Benutzer auch schreiben darf**« wird festgelegt, in welche Gruppen der Benutzer posten (= Artikel schicken) darf. Für das Format beider Felder siehe den folgenden Abschnitt zur Gruppenselektion. Außerdem kann festgelegt werden, ob der Benutzer den »Newnews«-Befehl des lokalen Newsservers verwenden darf, ob er mit dem NNTP-Peering-Protokoll den Hamster feeden darf und ob er den »Mode cancel«-Befehl verwenden darf.

Gruppenselektion:

Die Gruppenselektion ist eine Liste aus Regulären Ausdrücken (siehe auch die FAQ »Hamster und Reguläre Ausdrücke«), welche z. B. auch in den Filterdateien zum Einsatz kommen. Um mehrere Ausdrücke zu verwenden, müssen diese durch Leerzeichen voneinander getrennt werden. Die Felder werden von rechts nach links ausgewertet.

Beispiele:
Selektion: `.*`
Wirkung: Alle Gruppen sind erlaubt.

Selektion: `!.*`
Wirkung: Keine Gruppe ist erlaubt.

Selektion: `!^internal\. .*`
Wirkung: Alle Gruppen, ausgenommen die lokalen Gruppen, sind erlaubt.

Selektion: `^de\. !^internal\.posted ^internal\.`
Wirkung: Alle »de.*«-Gruppen und alle internen Gruppen, ausgenommen »internal.posted«, sind erlaubt.

Mail (POP3/IMAP+SMTP)

Lokales Postfach für eingehende E-Mails: Hier wird festgelegt, ob der Benutzer E-Mails

Glossar: POP3 IMAP SMTP

über den Hamster empfangen darf und, wenn ja, ob das Postfach per POP3 oder IMAP abgerufen werden soll. **Beides zugleich ist nicht möglich!**

Ist die Einstellung für das »SMTP-after-POP3«-Verfahren für den SMTP-Server aktiviert, wird mit dem Feld »**E-Mails senden erlaubt**« festgelegt, ob der Benutzer E-Mails über den Hamster versenden darf. Falls »SMTP-after-POP3« nicht verwendet wird, ist diese Einstellung ohne praktischen Belang.

Telnet-Server

Hier wird festgelegt, ob der Benutzer den Hamster per Remote-Control-Server fernsteuern darf. Siehe auch die allgemeinen Hinweise zum **Remote-Control-Server**.

Bearbeitungsdialog Benutzer

Allgemeines

Der Benutzername kann nur einmalig beim Anlegen eingegeben werden, die Beschreibung (»Voller Name«) kann dagegen geändert werden. Das Passwort kann jederzeit mit »**Ändern**« neu festgelegt werden (der Hamster speichert die Daten in der Datei »`Password.!!!`« ab, lesen Sie dort unbedingt den Sicherheitshinweis). Sie können den Benutzer auch einer anderen Benutzergruppe zuordnen (diese Einstellung ist für den Benutzer »admin« gesperrt). Durch Mausklick auf »**…**« können Sie direkt den Bearbeitungsdialog für die ausgewählte Gruppe aufrufen, ohne den Benutzerdialog zu verlassen.

News (NNTP) NNTP

Hier können die Standard-Einstellungen für die Benutzergruppe individuell überschrieben werden. Siehe dazu oben die Ausführungen zum Bearbeitungsdialog für die Gruppen.

Mail (POP3/IMAP+SMTP)

Das Feld »**Lokale E-Mail-Adressen**« erlaubt die Anlage von lokalen E-Mail-Adressen z. B. für ein lokales Netzwerk. Alle über den lokalen SMTP-Server an diese Adresse gesendeten E-Mails werden dem Empfänger lokal ohne Umweg über den Provider zugestellt. Besitzt ein Benutzer mehrere E-Mail-Adressen, müssen diese durch ein Leerzeichen getrennt eingegeben werden, Beispiel: »john.doe jd john.doe@hamster.invalid jd@hamster.invalid john.doe@provider.example.org«.

Außerdem können die Standard-Einstellungen für die Benutzergruppe individuell überschrieben werden. Siehe dazu oben die Ausführungen zum Bearbeitungsdialog für die Gruppen.

Telnet-Server

Hier kann die Standard-Einstellung für die Benutzergruppe individuell überschrieben werden. Siehe dazu oben die Ausführung zum Bearbeitungsdialog für die Gruppen.

Aliase für (lokale) Benutzer

Hier können Sie Alias-Namen für lokale Benutzer vergeben. Die hier enthaltenen Daten werden in der Datei »`MAlias.hst`« gespeichert. Beachten Sie bitte, dass Aliasse Vorrang vor den eigentlichen Benutzernamen haben. Weiterhin hat die Mailfilter-Datei »`MailFilt.hst`« Vorrang vor den Alias-Namen. Über den POP3-Clienten eingehende E-Mails können nur über den per Mailfilter festgelegten Benutzer-Namen umgeleitet werden, siehe hierzu auch die FAQ »E-Mails umleiten«. Als Platzhalter für den FQDN des Hamsters kann die Variable »%FQDN%« verwendet werden. Alle Einträge müssen das Format »⟨Quelle⟩=⟨Ziel⟩« haben. Der Name bzw. die E-Mail-Adresse vor dem Gleichheitszeichen ist immer die umzuleitende E-Mail-Adresse, der Alias. Nach dem Gleichheitszeichen steht immer die Zieladresse. Es ist auch möglich, als Ziel einen bereits vorhandenen Alias anzugeben, die E-Mails werden dann an dessen Ziel weitergeleitet (maximal 99-fache Verschachtelung), vgl. z. B. die bereits standardmäßig vorgegebenen Definitionen, wie »news@%FQDN%=news« und »news=admin«. Man sollte darauf achten, Ringstrukturen zu vermeiden (die einfachste Ringstruktur ist »›alias1=alias2‹ + ›alias2=alias1‹«). Kommentare können mit Rautezeichen »#« oder Semikolon »;« beginnend eingefügt werden.

Glossar: POP3

FQDN

Passworte

Hier können die Benutzernamen und Passwörter für Skripte und für die DFÜ-Netzwerkverbindungen festgelegt werden. Die Passwörter und Benutzernamen für die DFÜ-Netzwerkverbindungen sind dabei nur erforderlich, wenn der Hamster die Verbindung selbst herstellen soll und diese nicht schon im entsprechenden Windows-Dialog eingegeben und abgespeichert worden sind. Siehe hierzu auch das Starten des DFÜ-Netzwerkes per Menü und den Skript-Befehl HamRasDial.

Die zusätzlichen Passwort-Sätze mit den Namen »"$1"«, »"$2"«, … sind für die Skriptverarbeitung bestimmt und ansonsten nicht sinnvoll verwendbar.

Die maximale Anzahl der für Skripte bereitgestellten Passwörter ist im Feld »**Für Skripte $1 bis $ …** « einstellbar. Der Standard ist 30 Passwörter.

Die Felder für Benutzer und Passwort können drei Zustände haben:

- »{Fragen}«: Der Hamster fragt erst bei Bedarf Benutzername und Passwort ab. Dazu müssen Sie ein »?« als Benutzername oder/und Passwort eingeben.
- »{Gesetzt}«: Der Hamster speichert die Daten in der Datei »`Password.!!!`« ab (Lesen Sie dort unbedingt den Sicherheitshinweis).
- »{Leer}«: Der Hamster verwendet keine Authentifizierung.

Zum Editieren der Felder ist der entsprechende Eintrag zu markieren und der Schalter »**Bearbeiten**« zu betätigen.

Zum Löschen von Benutzername und Passwort ist der Schalter »**Löschen**« zu verwenden. Das Feld erhält anschließend den Status »{Leer}«.

Mittels der Klappleiste »**Verschlüsselung**« ist einstellbar, auf Basis welcher Daten der Hamster Daten verschlüsseln soll. Je weniger spezifische Daten einfließen, desto leichter kann eine gestohlene Passwort-Datei von anderen genutzt werden. Zudem bedeutet eine Änderung, dass alle Passwörter noch einmal eingegeben werden müssen – inkl. der Passwörter für ***alle*** Server.

Lokale Server Strg + L

Allgemeines

FQDNs und Message-IDs

FQDN für Message-IDs:

Hier ist der FQDN (»Fully Qualified Domain Name«) für den »Domain-Part« (also alles hinter dem »@«) der Message-ID einzugeben, wenn diese vom Hamster generiert werden soll. Für weitere Informationen siehe unbedingt im Glossar zu FQDN.

Message-ID-FQDN gilt als lokale Domain

Hier können Sie einstellen, ob der FQDN für die Message-ID als lokale Mailadresse bei der Verteilung der am lokalen SMTP-Server eingehenden Mails behandelt werden soll.

Glossar: SMTP

Message-ID verschlüsseln, sofern vom Hamster erzeugt

Hier können Sie einstellen, ob der »Local-Part« (also alles vor dem »@«) der vom Hamster generierten Message-ID kryptographisch gesichert werden soll. Dieses verhindert die Erkennung der Bildungsregel der Message-ID und schützt somit vor Fremd-Cancel, bei dem diese Bildungsregel der Message-ID ausgenutzt wird.

FQDN für spezielle Header, Peering etc.:

Hier ist der FQDN für »Path:«- und »Received:«-Header einzugeben, wenn diese vom Hamster generiert werden sollen. Dieser Eintrag wird dann auch für die Begrüßungstexte des Hamsters verwendet.

Header

Lokale Mails

Lokale Mailauslieferung bei Empfänger Domains, die dieser RegExp entsprechen:

Hier ist der »Domain-Part« (also alles hinter dem »@«) für die lokalen E-Mail-Adressen in Form eines Regulären Ausdrucks (RegExp, siehe auch die FAQ »Hamster und Reguläre Ausdrücke«) einzugeben. Adressen mit diesem »Domain-Part« werden als lokale E-Mail-Adressen betrachtet und nicht an den E-Mail-Server des Providers gesendet.

Der »Local-Part« (also alles vor dem »@«) dieser Adressen muss auf eine gültige lokale E-Mail-Adresse verweisen, die in den Benutzerdaten der einzelnen Benutzer (Menü

»Einstellungen« → »Benutzerverwaltung & Passworte« ⇒ »Lokale Benutzer«) angegeben wurde. E-Mails an lokale Adressen werden direkt unter Umgehung des Providers zugestellt. Existiert zu dem hier angegebenen »Domain-Part« keine lokale E-Mail-Adresse, wird die E-Mail dem Administrator zugestellt.

Die Voreinstellung ist:

```
.*hamster|\[?127\.0\.0\.1\]?|.*invalid
```

Beispiele für konkrete Adressen, welche diesen Bedingungen genügen:

```
.*hamster           irgendwer@irgendwas.hamster | irgendwer@hamster
\[?127\.0\.0\.1\]?  irgendwer@127.0.0.1 | irgendwer@[127.0.0.1]
.*invalid           irgendwer@irgendwas.invalid | irgendwer@invalid
```

Hinweis: E-Mail-Adressen ohne »Domain-Part« werden vom Hamster ***immer*** als lokale E-Mail-Adressen betrachtet.

Envelope-From: Absenderadresse für automatisch generierte E-Mails:

Hier können Sie einstellen, mit welcher E-Mail-Adresse als Absender Fehlermeldungen des Hamsters verschickt werden sollen (Standard: »admin«). Weiterhin wird der Eintrag dieses Feldes für das News-To-Mail-Gateway und die externe Weiterleitung von E-Mails verwendet.

Beachten Sie bitte, dass dieses Feld eine gültige öffentliche E-Mail-Adresse enthalten sollte, wenn das News-To-Mail-Gateway oder die externe Weiterleitung von Mails verwendet werden sollen.

Vom Hamster erzeugte Info-E-Mails nur an lokale Benutzer schicken

Hier können Sie einstellen, dass die automatisch generierten E-Mails ausschließlich im lokalen Netz verschickt werden (Standard: aktiviert). Ist der Info-E-Mail kein lokaler Benutzer zuzuordnen, wird sie an den Administrator verschickt.

Ist diese Einstellung deaktiviert, werden die Info-E-Mails an den externen (vermeintlichen) Absender verschickt, was von Spammern ausgenutzt werden kann.

Sicherheit

Hier können Einstellungen für die Zugriffssteuerung der lokalen Server bearbeitet werden, die in der Datei »IPAccess.hst« abgespeichert werden.

IPAccess.hst

Achtung! Diese Einstellungen sind sicherheitsrelevant. Nehmen Sie nur Änderungen vor, wenn Sie genau ihre Wirkung kennen. Lesen Sie auch die FAQ zum Thema »Hamster und Sicherheit«.

Hier kann die Datei »IPAccess.hst« für die Zugriffssteuerung der lokalen Server bearbeitet werden. Lesen Sie zum Verständnis die Erläuterungen zu dieser Datei!

Der Hamster besitzt einen Filter, welcher Zugriffe von IP-Paketen mit einer bestimmten IP-Adresse erlaubt oder unterbindet. Die einzelnen Zeilen dieser Datei können Sie nach dem Markieren mit den rechten Schaltern bearbeiten.

Im Bearbeitungsdialog gibt es drei verschiedene Typen von Einträgen:

- Kommentare

  Hier können Sie beliebige Bemerkungen eintragen.

- Serverzugriff

  Hier können Sie festlegen, welcher Server von welcher IP-Adresse aus mit welchen Rechten benutzt werden darf. Als Servernamen sind die Einträge »NNTP«, »SMTP«, »POP3«, »IMAP«, »RECO«, »MAIL« und »ALL«zulässig. Der Eintrag »MAIL« meint den POP3-, SMTP- und IMAP-Server gemeinsam. Der Eintrag »ALL« meint alle Server.

  Glossar: NNTP SMTP POP3 IMAP ReCo

  Als Zugriffsrecht sind die Einträge

  NA : Kein Zugriff (»No Access«)
  RO : Nur Lesen erlaubt (»Read Only«)
  WO : Nur Schreiben erlaubt (»Write Only«)
  RW : Lesen und Schreiben erlaubt (»Read & Write Allowed«)
  ALL: Keine Einschränkungen (»All Allowed«)

  möglich. Als dritter Parameter wird eine einzelne IP-Adresse oder ein IP-Adressbereich erwartet.

- Besondere Behandlung der lokalen IP-Adresse

  Da ein Rechner mehrere Netzwerkinterfaces und somit mehrere IP-Adressen haben kann, können Sie hier festlegen, welche von mehreren IP-Adressen im Rechner maßgebend für die Behandlung aller lokalen IP-Adressen ist. Die Voreinstellung ist die IP-Adresse »127.0.0.1« (der lokale Rechner = »localhost«).

Die Voreinstellung im Hamster ist diese:

```
ALL,NA,LOCAL,127.0.0.1
ALL,RW,127.0.0.1
ALL,RW,192.168.0.0,192.168.255.255
ALL,NA,0.0.0.0,255.255.255.255
```

NNTP

An folgende IP binden:

Hier wird eingestellt, an welches Netzwerkinterface der lokale NNTP-Server gebunden werden soll. Die Standard-Einstellung ist »127.0.0.1« (der lokale Rechner). Diese Voreinstellung schützt zuverlässig vor Eindringlingen.

Bevor Sie diese Einstellung ändern, sollten Sie sich genau über die Funktionsweise des TCP/IP-Stacks und der damit verbundenen Sicherheitsprobleme informieren.

Glossar: TCP

Port:

Die Port-Nummer legt den Port des Servers fest, d. h. unter welchem TCP/IP-Port der Client auf den lokalen Newsserver zugreifen kann. Voreingestellt ist der NNTP-Standardport 119.

Port NNTP

SSL-Benutzung:

Hier kann eingestellt werden, ob für den lokalen Server SSL/TLS[1] auf dem konfigurierten Port verwendet werden soll (mittels SSL-Handshake über das »STARTTLS«-Kommando). Zusätzliches NNTP over SSL (NNTPS) auf separatem Port ist nicht möglich. (Standard: »Kein TLS«; *nicht verwirren lassen davon, dass in den Optionen nur von TLS die Rede ist, das wurde in einer späteren Beta-Version korrigiert*.)

SSL TLS

Maximal erlaubte Anzahl an parallelen Client-Verbindungen:

Hier können Sie die Anzahl der maximal gleichzeitig zulässigen Verbindungen zwischen Hamster und den Readern aller User einstellen. Beachten Sie dabei, dass einige Reader mehrere Verbindungen gleichzeitig aufbauen und andererseits der Arbeitsspeicher Ihres Rechners auch Grenzen hat. Stellen Sie diesen Wert so ein, dass einerseits der Arbeitsspeicher des Rechners nicht überlastet wird und andererseits die Reader an den lokalen Servern möglichst wenig behindert werden. (Standard: 12 Verbindungen)

Maximal erlaubte Anzahl an parallelen Client-Verbindungen pro IP:

Hier können Sie die Anzahl der maximal gleichzeitig zulässigen Verbindungen zwischen Hamster und einem Clienten festlegen. Beachten Sie dabei, dass einige Reader mehrere Verbindungen gleichzeitig aufbauen und andererseits der Arbeitsspeicher Ihres Rechners auch Grenzen hat. Stellen Sie diesen Wert so ein, dass einerseits der Arbeitsspeicher des Rechners nicht überlastet wird und andererseits die Reader an den lokalen Servern möglichst wenig behindert werden. (Standard: 6 Verbindungen)

Maximale Größe für ausgehende Artikel:

Hier kann man angeben, wie groß ausgehende Postings maximal sein dürfen. Standard sind 2 097 152 Bytes (= 2 MiB). Eine Eingabe von »0« schaltet den Filter ab.

Speicherdauer

Hier kann für alle Gruppen die Voreinstellung für die Aufbewahrungszeit der Artikel in Tagen (Standard: »7«) und die Maximalanzahl von Artikeln je Gruppe (Standard: »0« = keine Begrenzung) konfiguriert werden. In den individuellen Gruppeneinstellungen (unter »Einstellungen« → »News: Server, Gruppen & Ladeaufträge« ⇒ »Newsgruppen«) können davon abweichende Werte eingestellt werden.

Zusätzlich kann eingestellt werden, wie weit zurück die News-Historie und die Einträge im Killfile-Log gespeichert werden (Standard: jeweils 7 Tage).

[1] Beeinflussung der erlaubten Version in »Einstellungen« → »Grundeinstellungen« ⇒ »SSL« → Protokolle

Glossar: Header

Header anpassen

Message-ID erzeugen, wenn noch keine vorhanden
Ist dieses Feld aktiv, generiert der Hamster selber die Message-ID, falls sie nicht bereits vom News-Reader gesetzt worden ist. Der »Local-Part« der Message-ID wird dann aus der aktuellen Zeit, der Task-Nummer und einem fortlaufenden Zähler pro Zeiteinheit gebildet. Mit einem ebenso eindeutigen FQDN ergibt sich somit eine Adresse, die den technischen Ansprüchen an Message-IDs genügt (d. h. für mindestens zwei Jahre eindeutig ist).

FQDN

Datumsheader Date erzeugen, wenn noch keiner vorhanden ist
Ist dieses Feld aktiv, erzeugt der Hamster den »Date«-Header, wenn er noch nicht vorhanden ist.

Erzeugen oder ggf. Erweitern des "User-Agent"-Headers
Ist dieses Feld aktiv, ergänzt der Hamster im Header jedes gesendeten Artikels seine Versionsnummer im User-Agent-Header oder erzeugt diesen, wenn er noch nicht vorhanden ist.

Erzeugen oder ggf. Erweitern des "Path"-Headers
Ist dieses Feld aktiv, ergänzt der Hamster im »Path«-Header jedes gesendeten Artikels den lokalen Domain-Namen oder erzeugt diesen Header, wenn er noch nicht vorhanden ist. Dieser muss dazu im Menü »Einstellungen« → »Lokale Server« ⇒ »Allgemeines« eingetragen sein. Andernfalls lehnt der lokale NNTP-Server alle zu postenden Artikel ab, da ansonsten defekte Artikel erzeugt würden.

NNTP

Zusätzliche Einstellungen

Cross-Posting auch in unbekannte Gruppen erlauben
Hier können Sie den lokalen Usern gestatten, in dem Hamster unbekannte Gruppen zu posten. Als unbekannt werden alle Gruppen betrachtet, für die kein Remote-Server für das Holen von Newsartikeln eingetragen ist.

Normalerweise werden Crosspostings, die an zumindest eine dem Hamster unbekannte Newsgruppe gehen, automatisch zurückgewiesen. Mit dieser Einstellung kann das Senden (= Posten) in unbekannte Gruppen erlaubt werden – ob der externe Newsserver den Artikel akzeptiert, ist damit natürlich noch nicht sicher.

Folgende Einschränkungen gelten auch bei aktivierter Einstellung:

- Zumindest eine Zielgruppe muss bekannt sein.
- Maximal zwei unbekannte Gruppen, sofern kein »Followup-To«-Header gesetzt ist.
- Maximal fünf unbekannte Gruppen, wenn ein »Followup-To«-Header gesetzt ist.

Unbekannte Message-IDs in Datei »`GetMIDs.txt`« eintragen
Ist dieses Feld aktiviert, trägt der Hamster Anfragen nach Artikeln mit unbekannter Message-ID in die Datei »`GetMIDs.txt`« ein, und lädt diese Artikel bei der nächsten Online-Sitzung automatisch nach. Ist die zu diesem Artikel gehörige Gruppe nicht im Hamster

vorhanden, wird der Artikel in der Gruppe abgelegt, die in »Einstellungen« → »Grundeinstellungen« ⇒ »Interne Gruppen« dafür eingerichtet ist (Standard: »internal.misc«).

Neue Artikel direkt in die lokale Datenbank speichern
Ist dieses Feld aktiv, kopiert der Hamster jeden lokal geposteten Artikel sofort lokal in die jeweilige Gruppe unter Umgehung des Umweges über den Provider. Diese Funktion ist nur zu empfehlen, wenn der Hamster als Feeder verwendet werden soll, oder eine zeitnahes lokales Einordnen der Artikel unbedingt erforderlich ist.

Achtung: Diese Funktion darf nur benutzt werden, wenn zum Provider ausgehende Artikel nicht mit Tools wie Korrnews oder Skripten nachbearbeitet werden. In diesem Fall kommt es zu unterschiedlichen Versionen der Artikel in der lokalen (Hamster-)Datenbank und im Usenet.

Glossar: Header

Für diese Funktion sind das Generieren von »Message-ID:«- und »Path:«-Header und ein lokaler Domain-Name zwingend erforderlich. Dieser muss dazu im Menü »Einstellungen« → »Lokale Server« ⇒ »Allgemeines« eingetragen sein. Andernfalls lehnt der lokale NNTP-Server alle zu postenden Artikel ab, da ansonsten defekte Artikel erzeugt würden.

NNTP

Cancel-Nachrichten
Hier können Sie verschiedene spezielle Funktionen aktivieren.

POP3

An folgende IP binden:
Hier wird eingestellt, an welches Netzwerkinterface der lokale POP3-Server gebunden werden soll. Die Standard-Einstellung ist »127.0.0.1« (der lokale Rechner). Diese Voreinstellung schützt zuverlässig vor Eindringlingen.

POP3

Bevor Sie diese Einstellung ändern, sollten Sie sich genau über die Funktionsweise des TCP/IP-Stacks und der damit verbundenen Sicherheitsprobleme informieren.

TCP

Port:
Die Port-Nummer legt den Port des Servers fest, d. h. unter welchem TCP/IP-Port der Client auf den lokalen POP3-Server zugreifen kann. Voreingestellt ist der POP3-Standardport 110.

Port

SSL-Benutzung:
Hier kann eingestellt werden, ob für den lokalen Server SSL/TLS[1] auf dem konfigurierten Port verwendet werden soll (mittels SSL-Handshake über das »STLS«-Kommando). Zusätzliches POP3 over SSL (POP3S) auf separatem Port ist nicht möglich. (Standard: »Kein TLS«; *nicht verwirren lassen davon, dass in den Optionen nur von TLS die Rede ist, das wurde in einer späteren Beta-Version korrigiert.*)

SSL
TLS

[1] Beeinflussung der erlaubten Version in »Einstellungen« → »Grundeinstellungen« ⇒ »SSL« → Protokolle

Maximal erlaubte Anzahl an parallelen Client-Verbindungen:

Hier können Sie die Anzahl der maximal gleichzeitig zulässigen Verbindungen zwischen Hamster und den Readern aller User einstellen. Beachten Sie dabei, dass einige Reader mehrere Verbindungen gleichzeitig aufbauen und andererseits der Arbeitsspeicher Ihres Rechners auch Grenzen hat. Stellen Sie diesen Wert so ein, dass einerseits der Arbeitsspeicher des Rechners nicht überlastet wird und andererseits die Reader an den lokalen Servern möglichst wenig behindert werden. (Standard: 10 Verbindungen)

Maximal erlaubte Anzahl an parallelen Client-Verbindungen pro IP:

Hier können Sie die Anzahl der maximal gleichzeitig zulässigen Verbindungen zwischen Hamster und einem Clienten festlegen. Beachten Sie dabei, dass einige Reader mehrere Verbindungen gleichzeitig aufbauen und andererseits der Arbeitsspeicher Ihres Rechners auch Grenzen hat. Stellen Sie diesen Wert so ein, dass einerseits der Arbeitsspeicher des Rechners nicht überlastet wird und andererseits die Reader an den lokalen Servern möglichst wenig behindert werden. (Standard: 5 Verbindungen)

Maximale Größe für empfangene E-Mails:

Hier kann man angeben, wie groß empfangene E-Mails maximal sein dürfen. Standard sind 2 097 152 Bytes (= 2 MiB). Eine Eingabe von »0« schaltet den Filter ab.

Diese Angabe bezieht sich ausschließlich auf den lokalen Mailserver, entsprechende Einstellungen für eingehende Mails von externen Mailservern werden unter »Einstellungen« → »Mail: Server konfigurieren« ⇒ »POP3-Einstellungen« angeboten bzw. sind über die Mailfilter-Datei »`MailFilt.hst`« nachbildbar.

Zulässige Anmeldeverfahren:

Glossar: SASL POP3 SSL

Hier können Sie festlegen, welche SASL-Mechanismen am lokalen POP3-Server zugelassen sind. Bitte beachten Sie, dass die SASL-Mechanismen PLAIN und EXTERNAL nur im Zusammenhang mit SSL zu Verfügung stehen.

Achtung: Das Authentifizierungsverfahren LOGIN verschleiert die Kennwörter nur durch Base64-Codierung. Verwenden Sie nach Möglichkeit DIGEST-MD5, CRAM-MD5 oder CRAM-SHA1, oder aber SSL.

Base64

Anzahl Tage, bis die Daten aufgeräumt werden dürfen

Hier kann eingestellt werden, wie weit zurück die Einträge in der Mailhistory gespeichert werden (Standard: 14 Tage).

IMAP

An folgende IP binden:

IMAP

Hier wird eingestellt, an welches Netzwerkinterface der lokale IMAP-Server gebunden werden soll. Die Standard-Einstellung ist »127.0.0.1« (der lokale Rechner). Diese Voreinstellung schützt zuverlässig vor Eindringlingen.

Bevor Sie diese Einstellung ändern, sollten Sie sich genau über die Funktionsweise des TCP/IP-Stacks und der damit verbundenen Sicherheitsprobleme informieren.

Glossar: TCP

Port:

Die Port-Nummer legt den Port des Servers fest, d. h. unter welchem TCP/IP-Port der Client auf den lokalen IMAP-Server zugreifen kann. Voreingestellt ist der IMAP-Standardport 143.

Port
IMAP

SSL-Benutzung:

Hier kann eingestellt werden, ob für den lokalen Server SSL/TLS[1] auf dem konfigurierten Port verwendet werden soll (mittels SSL-Handshake über das »STARTTLS«-Kommando). Zusätzliches IMAP over SSL (IMAPS) auf separatem Port ist nicht möglich. (Standard: »Kein TLS«; *nicht verwirren lassen davon, dass in den Optionen nur von TLS die Rede ist, das wurde in einer späteren Beta-Version korrigiert*.)

SSL
TLS

Maximal erlaubte Anzahl an parallelen Client-Verbindungen:

Hier können Sie die Anzahl der maximal gleichzeitig zulässigen Verbindungen zwischen Hamster und den Readern aller User einstellen. Beachten Sie dabei, dass einige Reader mehrere Verbindungen gleichzeitig aufbauen und andererseits der Arbeitsspeicher Ihres Rechners auch Grenzen hat. Stellen Sie diesen Wert so ein, dass einerseits der Arbeitsspeicher des Rechners nicht überlastet wird und andererseits die Reader an den lokalen Servern möglichst wenig behindert werden. (Standard: 10 Verbindungen)

Maximal erlaubte Anzahl an parallelen Client-Verbindungen pro IP:

Hier können Sie die Anzahl der maximal gleichzeitig zulässigen Verbindungen zwischen Hamster und einem Clienten festlegen. Beachten Sie dabei, dass einige Reader mehrere Verbindungen gleichzeitig aufbauen und andererseits der Arbeitsspeicher Ihres Rechners auch Grenzen hat. Stellen Sie diesen Wert so ein, dass einerseits der Arbeitsspeicher des Rechners nicht überlastet wird und andererseits die Reader an den lokalen Servern möglichst wenig behindert werden. (Standard: 10 Verbindungen)

Maximale Größe für eingehende E-Mails:

Hier kann man angeben, wie groß eingehende E-Mails maximal sein dürfen. Standard sind 2 097 152 Bytes (= 2 MiB). Eine Eingabe von »0« schaltet den Filter ab.

Diese Angabe bezieht sich ausschließlich auf den lokalen Mailserver, entsprechende Einstellungen für eingehende Mails von externen Mailservern werden unter »Einstellungen« → »Mail: Server konfigurieren« ⇒ »POP3-Einstellungen« angeboten bzw. sind über die Mailfilter-Datei »`MailFilt.hst`« nachbildbar.

Deaktiviere spezielle Anmelde-Methoden:

Hier können verschiedene Anmeldemethoden deaktiviert werden.

Achtung: Sind sowohl SASL (das sicherste Verfahren) als auch LOGIN (verschleiert Passwörter nur durch Base64-Codierung) deaktiviert und zugleich die Benutzung von SSL abgeschaltet, so werden die Passwörter im Klartext übertragen!

SASL
Base64

[1] Beeinflussung der erlaubten Version in »Einstellungen« → »Grundeinstellungen« ⇒ »SSL« → Protokolle

SMTP

An folgende IP binden:

Hier wird eingestellt, an welches Netzwerkinterface der lokale SMTP-Server gebunden werden soll. Die Standard-Einstellung ist »127.0.0.1« (der lokale Rechner). Diese Voreinstellung schützt zuverlässig vor Eindringlingen.

Glossar: SMTP

Bevor Sie diese Einstellung ändern, sollten Sie sich genau über die Funktionsweise des TCP/IP-Stacks und der damit verbundenen Sicherheitsprobleme informieren.

TCP

Port:

Die Port-Nummer legt den Port des Servers fest, d. h. unter welchem TCP/IP-Port der Client auf den lokalen SMTP-Server zugreifen kann. Voreingestellt ist der SMTP-Standardport 25.

Port

SSL-Benutzung:

Hier kann eingestellt werden, ob für den lokalen Server SSL/TLS[1] auf dem konfigurierten Port verwendet werden soll (mittels SSL-Handshake über das »STARTTLS«-Kommando). Zusätzliches SMTP over SSL (SMTPS) auf separatem Port ist nicht möglich. (Standard: »Kein TLS«; *nicht verwirren lassen davon, dass in den Optionen nur von TLS die Rede ist, das wurde in einer späteren Beta-Version korrigiert.*)

SSL
TLS

Maximal erlaubte Anzahl an parallelen Client-Verbindungen:

Hier können Sie die Anzahl der maximal gleichzeitig zulässigen Verbindungen zwischen Hamster und den Readern aller User einstellen. Beachten Sie dabei, dass einige Reader mehrere Verbindungen gleichzeitig aufbauen und andererseits der Arbeitsspeicher Ihres Rechners auch Grenzen hat. Stellen Sie diesen Wert so ein, dass einerseits der Arbeitsspeicher des Rechners nicht überlastet wird und andererseits die Reader an den lokalen Servern möglichst wenig behindert werden. (Standard: 3 Verbindungen)

Maximal erlaubte Anzahl an parallelen Client-Verbindungen pro IP:

Hier können Sie die Anzahl der maximal gleichzeitig zulässigen Verbindungen zwischen Hamster und einem Clienten festlegen. Beachten Sie dabei, dass einige Reader mehrere Verbindungen gleichzeitig aufbauen und andererseits der Arbeitsspeicher Ihres Rechners auch Grenzen hat. Stellen Sie diesen Wert so ein, dass einerseits der Arbeitsspeicher des Rechners nicht überlastet wird und andererseits die Reader an den lokalen Servern möglichst wenig behindert werden. (Standard: 3 Verbindungen)

Anmeldeverfahren:

SMTP AUTH

Dieses Feld aktiviert das »SMTP Authentication«-Verfahren für den lokalen SMTP-Server. Es darf nur verwendet werden, wenn der Client es auch tatsächlich unterstützt, andernfalls wird eine Fehlermeldung erzeugt.

SMTP-AUTH

[1] Beeinflussung der erlaubten Version in »Einstellungen« → »Grundeinstellungen« ⇒ »SSL« → Protokolle

Mails an lokale Benutzer auch ohne Anmeldung erlauben

Mit dieser Einstellung können Sie E-Mails an ausschließlich lokale Benutzer ohne Authentifizierung auch dann gestatten, wenn SMTP-AUTH gefordert ist. Diese Methode ist zu empfehlen, wenn der Hamster als SMTP-MX-Server verwendet werden soll. Vertrauenswürdige Benutzer können nach Authentifizierung den Hamster als Smarthost bzw. Mailrelay verwenden, während für unbekannte Benutzer eine Einlieferung von Mails an lokale Benutzer weiterhin möglich ist, aber gleichzeitig ein unbefugtes Weiterleiten von Mails dieser unbekannten Benutzer unterbunden wird.

Glossar: SMTP SMTP-AUTH

SMTP-after-POP3

Mit dem Feld »SMTP-after-POP3« kann das entsprechende Verfahren für den lokalen SMTP-Server aktiviert werden. Dieses technisch recht einfache Verfahren sichert den SMTP-Server zumindest einigermaßen gegen unbefugte Benutzung: E-Mails können dann nur noch an den SMTP-Server gesendet werden, wenn sich der Benutzer in den vergangenen 5 Minuten (Standard – die Zeitdauer ist in der »`hamster.ini`« änderbar: »local.smtpafterpop3.period« in der Sektion »[Setup]«) am POP3-Server durch Abholung neuer Mails authentifiziert hat.

POP3

Da im Hamster auch »SMTP-AUTH« aktiviert werden kann, sollten Sie »SMTP-after-POP3« nach Möglichkeit vermeiden.

Zulässige Anmeldeverfahren:

Hier können Sie festlegen, welche SASL-Mechanismen am lokalen SMTP-Server zugelassen sind. Bitte beachten Sie, dass die SASL-Mechanismen PLAIN und EXTERNAL nur im Zusammenhang mit SSL zu Verfügung stehen.

SASL

Achtung: Das Authentifizierungsverfahren LOGIN verschleiert die Kennwörter nur durch Base64-Codierung. Verwenden Sie nach Möglichkeit DIGEST-MD5, CRAM-MD5 oder CRAM-SHA1, oder aber SSL.

Base64

Weitere Einstellungen:

Mails an unbekannte lokale Benutzer bouncen/ zurücksenden …

Hier können Sie einstellen, ob am lokalen Server eingehende E-Mail an den Sender zurückgesendet werden sollen, wenn der lokale Empfänger unbekannt ist, oder ob sie an den Administrator weitergeleitet wird (Standardeinstellung). Kann von Spammern missbraucht werden!

Maximale Größe für ausgehende E-Mails:

Hier kann man angeben, wie groß ausgehende E-Mails maximal sein dürfen. Standard sind 2 097 152 Bytes (= 2 MiB). Eine Eingabe von »0« schaltet den Filter ab. Allerdings ist der Hamster nicht für Mails von 20 MB oder vergleichbaren Größen gedacht (und selbst wenn das technisch geht, können übergroße Mails durchaus Transportprobleme bekommen bzw. beim Empfänger mit einem freundlichen »Mailquote exceeded« zurückgewiesen werden).

Header hinzufügen/ändern

Message-ID erzeugen, wenn noch keine vorhanden

Ist dieses Feld aktiv, generiert der Hamster selber die Message-ID, falls sie nicht bereits vom Mail-Reader gesetzt worden ist. Der »Local-Part« der Message-ID (also alles vor dem »@«) wird dann aus der aktuellen Zeit, der Task-Nummer und einem fortlaufenden Zähler pro Zeiteinheit gebildet. Mit einem ebenso eindeutigen FQDN ergibt sich somit eine Adresse, die den technischen Ansprüchen an Message-IDs genügt (d. h. für mindestens zwei Jahre eindeutig ist). Diese Eindeutigkeit ist im Gegensatz zu Message-IDs bei Newsservern weniger wichtig, aber zumindest im Zusammenhang mit Mailinglisten u. ä. erwünscht.

Glossar: FQDN

Erzeuge zusätzlichen "X-Posting-Agent"-Header

Ist dieses Feld aktiv, »verewigt« sich der Hamster im Header jeder gesendeten Mail in Form eines zusätzlichen »X-Posting-Agent: Hamster/XXX«-Headers, wobei »XXX« die Versionsnummer darstellt (z. B. »2.1.0.11«).

Header

Erzeuge zusätzlichen "Received"-Header

Ist dieses Feld aktiv, »verewigt« sich der Hamster im Header jeder gesendeten Mail in Form eines zusätzlichen »Received«-Headers und schreibt die IP-Adresse und, falls bekannt, die User-ID des Einlieferers mit.

Beispiel:

```
Received: from localhost (HELO local.example.invalid) [127.0.0.1]
  by hamster.example.invalid (192.168.1.1)  (userid 5)
  with ESMTP (Classic Hamster Version 2.1 Build 2.1.0.11) ; Wed, 20 Oct 2010
```

Fernzugriff

An folgende IP binden:

Hier wird eingestellt, an welches Netzwerkinterface der lokale Remote-Control-Server gebunden werden soll. Die Standard-Einstellung ist »127.0.0.1« (der lokale Rechner). Diese Voreinstellung schützt zuverlässig vor Eindringlingen.

Bevor Sie diese Einstellung ändern, sollten Sie sich genau über die Funktionsweise des TCP/IP-Stacks und der damit verbundenen Sicherheitsprobleme informieren.

TCP

Port:

Die Port-Nummer legt den Port des Servers fest, d. h. unter welchem TCP/IP-Port der Client auf den lokalen Remote-Control-Server zugreifen kann. Voreingestellt ist der Telnet-Standardport 23.

Port

SSL-Benutzung:

Hier kann eingestellt werden, ob für den lokalen Server SSL/TLS[1] auf dem konfigurierten Port verwendet werden soll. Zusätzliche SSL-verschlüsselte Verbindungen auf separatem

SSL
TLS

[1] Beeinflussung der erlaubten Version in »Einstellungen« → »Grundeinstellungen« ⇒ »SSL« → Protokolle

Port neben einer unverschlüsselten Verbindung auf dem eingestellten Port sind nicht möglich. (Standard: keine SSL-Nutzung)

Maximal erlaubte Anzahl an parallelen Client-Verbindungen:

Hier können Sie die Anzahl der maximal gleichzeitig zulässigen Verbindungen zwischen Hamster und den Readern aller User einstellen. Beachten Sie dabei, dass einige Reader mehrere Verbindungen gleichzeitig aufbauen und andererseits der Arbeitsspeicher Ihres Rechners auch Grenzen hat. Stellen Sie diesen Wert so ein, dass einerseits der Arbeitsspeicher des Rechners nicht überlastet wird und andererseits die Reader an den lokalen Servern möglichst wenig behindert werden. (Standard: 1 Verbindung)

Maximal erlaubte Anzahl an parallelen Client-Verbindungen pro IP:

Hier können Sie die Anzahl der maximal gleichzeitig zulässigen Verbindungen zwischen Hamster und einem Clienten festlegen. Beachten Sie dabei, dass einige Reader mehrere Verbindungen gleichzeitig aufbauen und andererseits der Arbeitsspeicher Ihres Rechners auch Grenzen hat. Stellen Sie diesen Wert so ein, dass einerseits der Arbeitsspeicher des Rechners nicht überlastet wird und andererseits die Reader an den lokalen Servern möglichst wenig behindert werden. (Standard: 1 Verbindung)

Zulässige Anmeldeverfahren:

Glossar: SASL

Hier können Sie festlegen, welche SASL-Mechanismen am lokalen Remote-Control-Server zugelassen sind.

Achtung: Das Authentifizierungsverfahren PLAIN verschlüsselt die Kennwörter gar nicht, und LOGIN verschleiert sie nur durch Base64-Codierung. Verwenden Sie nach Möglichkeit DIGEST-MD5, CRAM-MD5 oder CRAM-SHA1, oder aber SSL.

Base64

Hinweis: Siehe auch die allgemeinen Erläuterungen zum Remote-Control-Server.

Automatische Abläufe

Allgemeines

Hier können Sie angeben, ob der Hamster beim Programmstart bestimmte Aufgaben automatisch erledigen soll.

Beim Aufruf des Hamsters:

Automatisches Aufräumen beim ersten Programmstart am Tag

Mit diesem Menüpunkt können Sie einstellen, dass der Hamster einmal täglich zu Programmbeginn alle veralteten Daten aus der Datenbank entfernt.

Nach überflüssigen Verzeichnissen suchen

Mit diesem Menüpunkt können Sie einstellen, dass der Hamster bei jedem Programmstart nach nicht mehr von Servern benutzten Verzeichnissen sucht und Sie warnt. Diese Verzeichnisse können Sie dann manuell löschen.

Folgende lokale Server starten:

Hier können Sie festlegen, welche lokalen Server der Hamster bei Programmbeginn automatisch starten soll (Standard: aktiviert NNTP-, SMTP-, POP3- und OLE-Server, deaktiviert IMAP- und Remote-Control-Server).

Glossar: NNTP SMTP POP3 IMAP Remote-Control-Server

Einmal am Tag:

Statistiken erzeugen

Hier können Sie festlegen, ob der Hamster einmal pro Tag eine Statistik über die Newsgruppen erstellen und in die interne Gruppe posten soll, die in »Einstellungen« → »Grundeinstellungen« ⇒ »Interne Gruppen« dafür eingerichtet ist (Standard: »internal.misc«).

News-Filterdatei auf abgelaufene Einträge prüfen:

Hier legen Sie fest, ob der Hamster aus dem News-Score-File einmal täglich veraltete Einträge entfernen oder auskommentieren soll.

Mail-Filterdatei auf abgelaufene Einträge prüfen:

Hier legen Sie fest, ob der Hamster aus dem Mail-Score-File einmal täglich veraltete Einträge entfernen oder auskommentieren soll.

Mails/Postings im Mülleimer nach wieviel Tagen löschen

Hier können Sie einstellen, nach wieviel Tagen Mail- oder Postingdateien aus dem Mülleimer (vgl. Menü »Lokale Server« → »News-/Mail-Verzeichnisse«) entfernt *und damit endgültig gelöscht* werden.

Info-Postings für jede Gruppenneueinrichtung eines Newsservers:

Durch das Eintragen eines Regulären Ausdrucks (siehe auch die FAQ »Hamster und Reguläre Ausdrücke«) in diesem Feld legen Sie fest, über welche Gruppenneueinrichtungen der Hamster Sie durch ein Posting in der internen Gruppe informieren soll, die in »Einstellungen« → »Grundeinstellungen« ⇒ »Interne Gruppen« dafür eingerichtet ist. Für den Regulären Ausdruck ist der Wert ».*« und für die interne Gruppe »internal.misc« voreingestellt.

Aktionen

Hier können Sie bestimmten Ereignissen die Ausführung eines bestimmten Programms, Skriptes oder einzelner hs2-Skriptzeilen zuordnen.

Unter Aktionen oder »Actions« werden im Hamster die Möglichkeiten zusammengefasst, bei bestimmten Aktionen automatisch ein externes Programm und/oder ein Hamster-Skript ausführen zu lassen. Der dazu gehörende Abschnitt in der »`hamster.ini`« heißt »[Actions]«.

Zu jeder möglichen Aktion gibt es im Dialog rechts oben einen Button »Erläuterungen«, so dass die Aktionen weitgehend selbsterklärend sind.

Wenn Sie eine **externe Anwendung starten** wollen, so müssen Sie als Parameter mindestens »%1« angeben für den Dateinamen des aktuellen Artikels bzw. der aktuellen E-Mail, bei Leerzeichen im Namen oder Pfad mit Anführungszeichen: »"%1"« (sollte bei den vom Hamster generierten Dateien nur auftreten, wenn Sie manuell die Pfade geändert haben).

Zur Auswahl unter »**Starte (hs2-)Skript**« stehen nur »hs2«-Skripte, die der Hamster im Skriptsuchpfad findet.

Unter »**Starte Skriptzeilen**« können Sie einzelne »hs2«-Befehle ausführen, ohne gleich ein Skript dafür schreiben zu müssen.

Achtung: Bei der Verwendung von »hs2«-Befehlen in action-gesteuerten Skripten oder Skriptzeilen, welche auf Mail- oder News-Transport-Action reagieren, sollten keine News- oder Mailtransportbefehle verwendet werden, um Deadlocks zu vermeiden.

Lade Artikel über ihre MID

Hier können Sie die Liste der Message-IDs der Artikel ändern, welche wegen einer Anforderung eines dem Hamster unbekannten Artikels durch einen Newsreader beim nächsten Serverkontakt auf den jeweiligen Servern gesucht und nachgeladen werden sollen. Die Message-ID kann dabei auf Wunsch automatisch aus der Zwischenablage entnommen werden. Ein manuelles Eintragen der MID ist ebenfalls möglich. Dabei ist das Format der MID zu beachten (»`<Message-ID>`«, d.h. inkl. der spitzen Klammern). Um Artikel zu laden, die bereits in der Newshistory stehen, kann man ein Ausrufezeichen an den Anfang der entsprechenden Zeile setzen (»`!<Message-ID>`«).

Es handelt sich hierbei um ein Front-End für die (für jeden Server separat existierende) »`Get-MIDs.txt`«.

News: Server, Gruppen & Ladeaufträge Strg + N

Einstellungen

News

Artikel-Limit (Standard / leere Gruppe):

Hiermit wird voreingestellt, wie viele Artikel pro Newsgruppe maximal bei einem einzelnen Download vom Newsserver geladen werden dürfen. Dabei wird das allgemeine Limit und das Limit bei einer erstmalig benutzten (= leeren) Gruppe unterschieden. Sofern man einen Newsserver mit sehr langer Haltezeit verwendet, sollte zumindest bei »leere Gruppe« ein sinnvolles Limit eingestellt werden (Standard: jeweils die 100 ältesten Artikel).

Ladeaufträge teilen, wenn sehr viele neue Artikel geladen werden sollen

Mit der Teilung großer Ladeaufträge kann erreicht werden, dass bei einer Unterbrechung

der Verbindung nur der unterbrochene Teilauftrag wiederholt werden muss (Standard: jeweils deaktiviert).

Es kann eingestellt werden, ab welcher Artikelanzahl eine Aufteilung erfolgt (Standard: 300 Artikel) und wie groß diese Teilaufträge dann sind (Standard: 250 Artikel).

Zusätzliche Einstellungen:

Beim Ausführen von Newsjobs automatisch Serverinformationen abfragen
: Hiermit stellen Sie ein, ob beim Ausführen eines Jobs zum Austausch von News auch automatisch ein Job ausgeführt wird, welcher die Liste der Newsgruppen und deren Beschreibung holt.

Newsserver

Hier werden alle externen Newsserver eingetragen und konfiguriert, von denen die Artikel geholt werden sollen bzw. über welche die eigenen Artikel wieder verschickt werden sollen.

Neu

Zum Hinzufügen eines neuen Servers dient der Schalter »**Neu**«. Geben Sie den Namen des Newsservers und bei Bedarf, durch ein Komma abgetrennt, noch den Port des Newsservers an (Beispiel: »`news.example.com,119`« oder kurz »`news.example.com`«). Für Standardports bekannter Dienste kann auch der übliche Name eingetragen werden, wie er in der Datei »`services`« zu finden ist (im Beispiel: »`news.example.com,nntp`«). Bei Windows 95 bis Windows ME liegt diese Datei im Windows-Verzeichnis, bei Windows NT und Nachfolgern im Verzeichnis »`%SYSTEMDIR%\drivers\etc`« (normalerweise »`c:\windows\system32\drivers\etc`«).

Glossar: Port

Die Liste der auf diesem Newsserver verfügbaren Newsgruppen wird bei der nächsten Online-Verbindung des Newsservers automatisch geladen.

Sollte ein externer Server mit demselben Namen bereits existieren *(gleichgültig, ob News- oder Mailserver)*, wird nachgefragt, ob ein Alias angelegt werden soll. Dieses erlaubt z. B. die sinnvolle Nutzung mehrerer Accounts. Der Alias ist durch ein nachgestelltes »/XX« erkennbar, wobei »XX« für beliebigen Text steht (z. B. eine fortlaufende Numerierung).[1]

Einstellungen

Zum Ändern der Einstellungen eines Newsservers dient der Schalter »**Einstellungen**«. Hier können Sie im Feld »NNTP-Server-Port« einstellen, welchen TCP-Port der Hamster beim Provider benutzen soll (man kann in diesem Feld Eingaben machen – solange man im Dialogmenü bleibt, merkt sich der Hamster alle bisherige Eingaben, und diese sind dann über das Auswahlfeld anwählbar). Für die Portbezeichnung gilt dasselbe, was oben zur Neueinrichtung ausgeführt wurde. Sie können per Schalter die »`server.ini`« bearbeiten und die Gruppenliste öffnen

NNTP
Port
TCP

[1] Das ist eine Folge des hamsterinternen Systems der Speicherung in Verzeichnissen, die den Namen des Servers tragen.

(beide werden in dem Editor geöffnet, der im Menü »Einstellungen« → »Grundeinstellungen« ⇒ »Verschiedenes« eingestellt ist).

Zudem legen Sie hier Anmeldenamen und Passwort für den Zugriff fest oder löschen die aktuellen Angaben wieder (»**Ändern**« bzw. »**Löschen**«). Die Felder für Benutzer und Passwort können drei Zustände haben:

- »{Fragen}«: Der Hamster fragt erst bei Bedarf Benutzername und Passwort ab. Dazu müssen Sie ein »?« als Benutzername oder/und Passwort eingeben.
- »{Gesetzt}«: Der Hamster speichert die Daten in der Datei »`Password.!!!`« ab (Lesen Sie dort unbedingt den Sicherheitshinweis).
- »{Leer}«: Der Hamster verwendet keine Authentifizierung.

Ebenfalls kann hier das Timeout für die Verbindung zum Server und für die Serverkommandos eingestellt werden. Soll ein Server erst einmal ungenutzt bleiben, muss man ihn nicht gleich löschen, sondern kann ankreuzen, dass dieser Server temporär deaktiviert sein soll.

Einschränkungen

Der Server kann als schreibgeschützt definiert werden, so dass keine Postings mehr über diesen Newsserver abgeschickt werden (können). Sie können auch Ihre eigenen Beiträge in ein eigenes Verzeichnis ablegen lassen. Über einen Schieberegler können Sie die Anzahl der Threads (= gleichzeitige Newsjobs) festlegen zwischen 0, d. h. es wird nichts abgeholt, und 4, Standard ist die Stellung auf »1 Thread«. Bei mehr als einem Thread sollten Sie zuerst Ihren Newsprovider kontaktieren, wieviel Threads den Nutzern erlaubt sind. Im letzten Auswahlfeld können Sie für Spezialfälle den Servertyp ändern: Neben dem Standard »Vollwertiger NNTP-Server« können Sie einstellen »Kein Abholen von News möglich, nur Gruppenliste«, was zum kurzzeitigen Deaktivieren unter Erhaltung einer aktuellen Gruppenliste sinnvoll ist, und »Pseudo-Server«, was sinnvoll ist, wenn man die Artikel per UUCP weiterverarbeiten will; zum Abonnieren von Gruppen muss, um die Gruppenliste laden zu können, bei letzterem Typ aber temporär auf einen der zwei anderen Servertypen umgestellt werden, oder die zu abonnierenden Gruppen müssen direkt in die Datei »`Pulls.hst`« eingetragen werden.

Glossar: NNTP

Beim nächsten Abholen von News:

Sie können ankreuzen, dass beim nächsten Online-Gehen die Gruppenliste und/oder die Gruppenbeschreibungen (»Taglines«) nachgeladen werden.

SSL-Einstellungen

Hier können Sie einstellen, ob Sie die Verbindung zum Provider kryptographisch mit dem SSL-Verfahren sichern wollen.

SSL

Achtung! Um SSL in Verbindung mit dem Hamster nutzen zu können, werden zwei zusätzliche DLL-Dateien benötigt; mehr zum Thema siehe in der SSL-FAQ.

Es stehen hier 4 Nutzungsarten zur Verfügung:

- **Keine SSL/TLS-Nutzung**
- **SSL/TLS bei sicherem Port immer benutzen** – In diesem Fall wird eine SSL-Verbindung ohne SSL-Handshake hergestellt (NNTPS), der Standardport dafür ist 563. Falls die Herstellung der SSL-Verbindung scheitert, wird keine TCP-Verbindung hergestellt.
- **SSL/TLS nutzen, wenn möglich (Standardport)** – Es wird nur dann SSL benutzt, wenn der Provider es anbietet,[1] andernfalls wird eine unsichere Verbindung hergestellt. **Diese Methode ist nicht zu empfehlen.**
- **SSL/TLS immer verwenden (Standardport)** – Es wird immer versucht, eine SSL-Verbindung mittels SSL-Handshake[1] herzustellen. Falls die Herstellung der SSL-Verbindung scheitert, wird keine TCP-Verbindung hergestellt.

Glossar: Port, TLS, TCP

Es stehen mehrere Verfahren zur Überprüfung der X.509-Server-Zertifikate zur Verfügung:

X.509

- **Keine Überprüfung** – **Nicht zu empfehlen, da so keinerlei Sicherheit über die Identität des Servers besteht.**
- **Zertifikat überprüfen, wenn vorhanden** – Falls das Zertifikat nicht vom Server geliefert wird, wird trotzdem eine Verbindung hergestellt. **Achtung! Die Verbindung wird, falls kein Server-Zertifikat vorliegt, zwar verschlüsselt, aber der Server nicht authentifiziert.**
- **Zertifikat immer überprüfen** – Liefert der Server kein Zertifikat, dann wird keine Verbindung hergestellt.
- **Zusätzlich Serverzertifikat lokal überprüfen** – In diesem Fall wird nur eine Verbindung zum Server hergestellt, wenn sich dessen Zertifikat im lokalen Zertifikate-Verzeichnis oder in der lokalen Zertifikate-Datei befindet. **Dieses ist die sicherste Methode, fragen Sie Ihren Provider nach seinem öffentlichen Server-Zertifikat.**

Für die lokale Prüfung des Server-Zertifikates können Sie im Feld »Überprüfung über Datei:« eine Datei mit einem Zertifikat im PEM-Format angeben.

PEM

Ladeaufträge

Über »**Ladeaufträge**« können Sie die von diesem Server abonnierten Gruppen verwalten. Um *alle* Gruppen zu verwalten, sollten Sie besser auf die Seite »News-Ladeaufträge« wechseln.

Löschen

Zum Entfernen eines bestehenden Newsservers aus der Liste ist der Schalter »**Löschen**« zu betätigen. Die Gruppen selbst bleiben dabei existent; sofern die Gruppen bislang ausschließlich über diesen Server bezogen wurden, werden sie vom Hamster zukünftig als lokale Gruppen behandelt.

[1] Über das »STARTTLS«-Kommando.

Bevorzugter Post-Server:

Normalerweise werden eigene Artikel über den Server versandt, über den auch neue Artikel bezogen werden. Werden News-Gruppen aber von mehreren Servern bezogen, ist die Auswahl nicht mehr eindeutig. Da der Hamster vorläufig Artikel immer nur über einen Server hinausschickt, muss er einen Server auswählen. Dies geschieht nach der internen Reihenfolge, außer, der hier eingestellte bevorzugte Post-Server ist ebenfalls ein passender Kandidat. In dem Fall wird der bevorzugte Post-Server verwendet ungeachtet der internen Reihenfolge.

Zum Auswählen des bevorzugten Servers müssen Sie nur den Server in der Liste markieren und den Schalter »**Auswahl**« betätigen.

Wähle Ladeaufträge für Newsserver ...

Auf der linken Seite sind alle zur Zeit vom aktuellen Server abonnierte Gruppen zu sehen, rechts ist seine gesamte Gruppenliste. Über Doppelklicks oder die passenden Schalter können Gruppen abonniert bzw. wieder abbestellt werden.

Die Gruppenliste rechts ist dabei nach verschiedenen Möglichkeiten filterbar: Entweder mit einem Regulären Ausdruck (siehe auch die FAQ »Hamster und Reguläre Ausdrücke«), einem einfachen Suchausdruck, dem Beginn des Gruppennamens oder nach den Initialen der gesuchten Gruppe.

Mit »**OK**« wird die geänderte Auswahl im Hamster gespeichert.

Newsgruppen

Diese Liste enthält alle lokalen (blau gefärbt) und alle von externen Newsservern bezogenen Newsgruppen (schwarz gefärbt). Moderierte und »Nur lesen«-Gruppen enthalten einen zusätzlichen grünen Hinweistext. Über die Seite »News-Ladeaufträge« oder über die Newsserver-Einstellungen selber können neue Gruppen abonniert werden. In dieser Liste wird jede Gruppe nur einmal angezeigt, egal, von wie vielen Newsservern sie geholt wird.

Die internen Gruppen sind dabei nicht löschbar, nach einer Neueinrichtung ist zumindest die interne lokale News-Gruppe »internal.misc« vorhanden.

Neue lokale Gruppen werden mit dem Schalter »**Lokale Gruppen**« hinzugefügt.

Mit dem Schalter »**Einstellungen**« können die Voreinstellungen der Gruppen überschrieben werden: Dies betrifft die Gruppenbeschreibung, den Gruppentyp, den Server, bei welchem die eigenen Artikel gepostet werden (falls diese Gruppe bei mehreren Server abonniert worden ist). Hier kann gegebenenfalls auch eingestellt werden, dass die Artikel dieser Gruppe nicht zum Provider gepostet, sondern zu anderen Servern gefeedet werden sollen oder an eine Gruppenmoderation oder eine Mailingliste gesendet werden sollen.

Folgende Gruppentypen stehen zur Verfügung:

- »Posten erlaubt« = y
  Dieses ist die Voreinstellung für reguläre Newsgruppen.
- »Nur Lesen« = n
  Normale Newsgruppe, die nur gelesen werden darf.
- »Moderiert« = m
  Moderierte Newsgruppe. Es muss eine E-Mail-Adresse für die Moderation eingegeben werden.
- »Gateway« = g
  Mailingliste. Es muss eine E-Mail-Adresse für den Listengenerator angegeben werden. Siehe hierzu auch die FAQ »Mailinglisten in Newsgruppen konvertieren«.

Weitere Einstellungen betreffen das Bearbeiten der »`Data.ini`«-Datei über »**.ini bearbeiten**« und individuelle Einstellungen für die maximale Anzahl an neuen Postings je Ladevorgang und die Haltedauer in Tagen bei Anwendung der Funktion »**Aufräumen**«. Als Voreinstellung für die letzten beiden Angaben werden die Daten aus der Unterseite »Einstellungen« bzw. aus den Einstellungen des lokalen Newsservers verwendet.

Durch Betätigen des Schalters »**Neue News**« kann die markierte Newsgruppe mit den in Pull befindlichen Servern abgeglichen werden. Das heißt, es werden alle auf den Servern der Provider verfügbaren Artikel geholt. Zum Aufräumen von einzelnen Gruppen kann der Schalter »**Aufräumen**« verwendet werden.

Zum Löschen von Gruppen kann der Schalter »**Löschen**« verwendet werden, die entsprechenden Ladeaufträge werden dabei automatisch mit gelöscht, und alle im Hamster für die Gruppe gespeicherten Postings werden dabei natürlich auch gelöscht. Das dazugehörige Verzeichnis im Gruppenpfad ist dann leer. Manchmal kann das Verzeichnis selbst nicht automatisch mitgelöscht werden, insbesondere wenn es durch ein anderes Programm oder einen anderen Hamster-Thread geöffnet ist. Wenn Sie dagegen eine Gruppe nur in den Ausgangszustand versetzen möchten, können Sie den Schalter »**Zurücksetzen**« verwenden. Siehe dazu auch den FAQ-Beitrag »Unterschied zwischen ›Gruppe zurücksetzen‹ und ›Gruppe löschen‹«.

News-Ladeaufträge

Diese Liste enthält alle Ladeaufträge, d. h. alle Gruppenabonnements aller eingestellten Newsserver, sortiert nach dem Gruppennamen.

Hier kann man mit dem Schalter »**Neu**« neue Gruppen abonnieren. Siehe hierzu auch »Abonnierte Gruppen verwalten«.

Es ist möglich, eine Gruppe bei mehr als einem Server auszuwählen. Durch diese Redundanz erhöht sich zwar die Menge der zu ladenden Daten, aber gleichzeitig verringert sich die Wahrscheinlichkeit, dass Artikel fehlen. Zudem können temporäre Ausfälle eines Newsservers ganz gut kompensiert werden.

Zum Löschen von Ladeaufträgen (= Server/Gruppen-Kombinationen) ist der Schalter »**Löschen**« zu betätigen.
Hinweis: Der Schalter löscht nicht die Gruppe selbst, sondern nur das Abonnement vom entsprechenden Server. Falls auch das letzte Abonnement gelöscht wurde, wird die Gruppe automatisch zu einer lokalen Gruppe.

Newsserver: Wähle Gruppen ...

Dazu werden im oberen Bereich des entsprechenden Dialoges die gewünschten Gruppen markiert und anschließend im unteren Teil die gewünschten Gruppen/Server-Kombinationen gewählt. Mit »**OK**« werden die Gruppen dann endgültig abonniert. Um die obere Auswahl übersichtlicher zu gestalten, ist die Gruppenliste (inkl. der Beschreibungen) nach verschiedenen Möglichkeiten filterbar: Entweder mit einem Regulären Ausdruck (siehe auch die FAQ »Hamster und Reguläre Ausdrücke«), einem einfachen Suchausdruck, dem Beginn des Gruppennamens oder nach den Initialen der gesuchten Gruppe – die letzte Auswahl wird dabei automatisch gespeichert. In der unteren Liste kann, mit einem Regulären Ausdruck gefiltert oder über eine Liste, die Auswahl des gewünschten Servers getroffen werden.

News: Killfile-Protokoll/-Einstellungen Strg + K

Die Fenster zur Bearbeitung und Kontrolle des News-Score- bzw. -Killfiles

Einträge

Dieses Fenster zeigt den Inhalt der Datei »`killfile.log`« an. Sie enthält Verweise auf alle aufgrund des News-Score-/Killfiles nicht geladenen Artikel, deren Score-Wert zwischen −1 und dem einstellbaren Wert (siehe Fenster »Einstellungen«) liegen. Ein Verweis besteht immer aus der Kombination Newsserver + Message-ID, somit kann ein und derselbe Artikel durchaus mehrfach auftauchen, wenn man Gruppen von mehreren Servern parallel bezieht. Mit einem Klick auf eine Spaltenüberschrift können die Verweise nach der entsprechenden Spalte sortiert werden. Sollte einem diese Sortierung zu ungenau sein, kann man den Menüpunkt »Datei« → »Zusatzsortierung bearbeiten« verwenden, welcher eine Angabe eines zweiten und dritten Sortierkriteriums ermöglicht. Mit einem Doppelklick auf den Artikel erhält man weitere Informationen zu diesem Artikel. Über das **Kontext-Menü** ist ein Zugriff auf weitere Funktionen möglich.

Einstellungen

Im zweiten Fenster wird der Score-Wert angegeben, für den noch geloggt werden soll. Dies muss ein negativer Wert sein, der maximal vierstellig sein darf. Ist man sich über die Funktion des Score-/Killfiles noch nicht völlig sicher, empfiehlt sich der Wert −9999 (Standard und kleinstmöglicher Wert). Gibt man hier z. B. −9000 an, werden die nicht geladenen Artikel mit Score-Werten von −9000 bis −1 im ersten Fenster angezeigt. Die nicht geladenen Artikel mit Score-Werten unterhalb von −9000 (also z. B. −9001) werden unter »Einträge« nicht mehr angezeigt.

Außerdem kann eingestellt werden, wie lange die Einträge in der Datei »`killfile.log`« aufbewahrt werden. Voreingestellt sind 7 Tage, die Eingabe von »0« deaktiviert diese Funktion, d. h. sie werden nur gelöscht, wenn man dies manuell im Fenster »Einträge« durchführt. Sonst wird nur über den Menüpunkt »Datei« → »Aufräumen« gelöscht oder wenn man das Aufräumen automatisch ausführen lässt (Menü »Einstellungen« → »Automatische Abläufe« ⇒ »Allgemeines«).

Kill-/Scorefile

In diesem Fenster ist ein einfacher Editor für das News-Score-/Killfile zu sehen.

Im Menü stehen folgende Punkte zur Auswahl:

***Datei* →**

Zusatzsortierung bearbeiten

Ermöglicht die Angabe des zweiten und dritten Sortierkriteriums.

Alle Änderungen speichern Strg + S

Speichert alle Einstellungen und Anforderungen für das Nachladen von Artikeln und Threads ab.

Alle Änderungen verwerfen

Verwirft alle Einstellungen und Anforderungen für das Nachladen von Artikeln und Threads.

Schließen

Beendet die Bearbeitung des Killfiles.

***Einträge* →**

Zeige Details

Zeigt Details zu einem markierten Artikel an.

Wähle alle Strg + A

Markiert alle Einträge im Killfile zur gemeinsamen Bearbeitung.

Nachträglich doch laden Strg + M

Speichert die Message-ID in der Datei »`GetMIDs.txt`« des zugehörigen Servers. Der Artikel wird dann automatisch bei der nächsten Onlineverbindung mit diesem Server geholt, sofern er dort nicht schon gelöscht (expired) ist.

Löschen Strg + K

Löscht den oder die markierten Einträge im Killfile endgültig.

Gesamten Thread dauerhaft laden Strg + Alt + M

Hier können zu einem Artikel gehörige Threads komplett nachgeladen werden.

Gesamten Thread dauerhaft löschen Strg + Alt + K

Hier können zu einem Artikel gehörige Threads komplett gelöscht werden.

Scorefile-Test Strg + T
Testet den Artikel mit den aktuellen Filtereinstellungen und zeigt dessen Ergebnis an. Siehe hierzu auch die Beschreibung zur Datei »Scores.hst«.

Kontextmenü:

Wähle alle
Markiert alle Zeilen.

Nachträglich doch noch laden
Speichert die Message-ID in der Datei »GetMIDs.txt« des zugehörigen Servers. Der Artikel wird dann automatisch bei der nächsten Onlineverbindung mit diesem Server geholt, sofern er dort nicht schon gelöscht (expired) ist.

Gesamten Thread dauerhaft löschen/nachladen
Hier können komplette Threads nachgeladen bzw. gelöscht werden.

Löschen
Entfernt alle markierten Einträge.

Scorefile-Test
Testet den Artikel mit den aktuellen Filtereinstellungen und zeigt dessen Ergebnis an. Siehe hierzu auch die Beschreibung zur Datei »Scores.hst«.

Datei (Submenü)

Dieses Dialog-Menü bietet folgende Möglichkeiten:

Zusatzsortierung bearbeiten
Ermöglicht die Angabe eines zweiten und dritten Sortierkriteriums. Die Zusatzsortierung ist numerisch codiert anzugeben:

Glossar: Header

0 = Server	6 = »Date:«-Header
1 = Gruppe	7 = »Message-ID:«-Header
2 = Score	8 = »REF:«-Header
3 = Nummer	9 = Größe des Artikels in Byte
4 = Betreff	10 = Größe des Artikels in Zeilen
5 = »From:«-Header	11 = »Xref:«-Header

Alle Änderungen speichern Strg + S
Mit Betätigen dieses Menüpunktes werden alle Änderungen im Scorefile und in der Datei »GetMIDs.txt« gespeichert.

Alle Änderungen verwerfen
Mit Betätigen dieses Menüpunktes werden alle Änderungen im Scorefile und in der Datei »GetMIDs.txt« verworfen und der vorherige Zustand wiederhergestellt.

Schließen

Mit Betätigen dieses Menüpunktes wird das Score- bzw. Killfile-Menü verlassen.

Einträge (Submenü)

Dieses Dialog-Menü bietet folgende Möglichkeiten:

Zeige Details

Zeigt Detailinformationen zu dem markierten Artikel und dem dazugehörigen Scorewert an.

Wähle alle Strg + A

Markiert alle Einträge des Score- bzw. Killfiles zur gemeinsamen Bearbeitung.

Nachträglich doch laden Strg + M

Speichert die Message-ID in der Datei »`GetMIDs.txt`« des zugehörigen Servers. Der Artikel wird dann automatisch bei der nächsten Onlineverbindung mit diesem Server geholt, sofern er dort nicht schon gelöscht (expired) ist.

Löschen Strg + Alt + K

Löscht den Artikel endgültig aus dem Score- bzw. Killfile.

Gesamten Thread dauerhaft nachladen Strg + Alt + M

Entfernt alle Antworten auf die markierten Postings und speichert die Message-IDs in der Datei »`GetMIDs.txt`« des zugehörigen Servers. Der Thread wird dann automatisch bei der nächsten Onlineverbindung mit diesem Server geholt, sofern er dort nicht schon gelöscht (expired) ist. Außerdem wird ein Eintrag in das Scorefile geschrieben, damit Postings in diesem Thread von nun an sofort geladen werden. Dabei wird geprüft, ob schon ein entsprechender Eintrag existiert. Der Abschnitt im Scorefile und der Wert, der dem Thread zugewiesen wird, kann über die Einstellungen »score.RetrieveThread.section« und »score.RetrieveThread.value« in der »`hamster.ini`« (Sektion »[Setup]«) eingestellt werden.

Gesamten Thread dauerhaft löschen Strg + K

Entfernt alle Antworten auf die markierten Postings endgültig aus dem Scorefile.

Scorefile-Test Strg + K

Testet den Artikel mit den aktuellen Filtereinstellungen und zeigt dessen Ergebnis an. Siehe hierzu auch die Beschreibung zur Datei »`Scores.hst`«.

Kill-/Scorefile (Reiter)

Ein einfacher Editor für das News-Score-/Killfile (siehe auch: »Datei: Scores.hst«).

Um die Datei mit dem gewohnten Editor zu bearbeiten, kann aus dem Hamster-Hauptfenster heraus der Menüpunkt »Einstellungen« → »Spezielle Dateien« → »News-Score-/Killfile« genutzt werden.

Welcher Editor verwendet wird, kann über »Einstellungen« → »Grundeinstellungen« ⇒ »Verschiedenes« eingestellt werden.

Mail: Server konfigurieren Strg + M

POP3-Einstellungen

Mails vom externen POP3-Server laden:

E-Mails auf Server belassen/nicht löschen

Hier bestimmt man für alle POP3-Server (ohne individuelle abweichende Einstellungen), ob die E-Mails auf dem Server belassen werden sollen, anstatt sie wie üblich zu löschen. Diese Einstellung ist für die ersten Tests durchaus sinnvoll, auf Dauer ist es aber nicht empfehlenswert, da die Kapazität der meisten elektronischen Postfächer begrenzt ist und neue Mails bei einem vollen Postfach üblicherweise »bouncen«, was den Absender nicht unbedingt erfreut.

Glossar: POP3

Bereits geladene E-Mails kein zweites Mal laden

Diese Einstellung ergibt nur mit dem vorigen Punkt Sinn, wenn dort eingestellt ist, dass die E-Mails auf dem Server belassen werden, und erlaubt es für diesen Fall, jede E-Mail trotzdem nur einmal zu holen.

Erzeuge zusätzlichen "Received"-Header

Diese Einstellung ermöglicht es, im »Received:«-Header einer Mail mitzuschreiben, über welchen Server die E-Mail in den Hamster gelangt ist.

Header

Mails=>News-Gateway

Automatisch Header "Followup-To: poster" setzen

Hier können Sie entscheiden, ob der Hamster den Header »Followup-To: poster« für Artikel setzt, welche als Mail beim Hamster eingehen und in einer lokalen Newsgruppe verwaltet werden sollen. Dieses wird für lokale Newsgruppen empfohlen, welche Mailinglisten handhaben und für den Rückweg nicht das News-to-Mail-Gateway des Hamsters, sondern den SMTP-Clienten des Newsreaders nutzen wollen.

SMTP

"!Envelope From" über das Standard-Hamster-"From" statt über den News-"From"-Header bilden

Hier können Sie festlegen, wie der »!Envelope From«-Header für das News-to-Mail-Gateway gebildet wird. Die Verwendung des News-»From«-Headers ist zu empfehlen, wenn mehrere User gemeinsam den Hamster verwenden.

Eingehende Mails filtern:

Die folgenden Optionen erlauben die Behandlung übergroßer Mails. Diese Überprüfung wird noch vor dem Testen der Mailfilter-Datei durchgeführt und sorgt dafür, dass die betreffende übergroße Mail gar nicht erst geladen wird. Ein Wert von »0« steht dafür, dass die Überprüfung nicht durchgeführt werden soll.

Hinweis an Admin, wenn Mail größer als ... Bytes
: Falls dieser Grenzwert überschritten wird, erfolgt eine Mail an den Benutzer»admin«, die Mail bleibt dabei weiterhin auf dem Mailserver bestehen (Standard: »0« = keine Filterung).

Ignoriere Mail, wenn größer als ... Bytes
: Falls dieser Grenzwert überschritten wird, wird die Mail nicht geladen, bleibt dabei weiterhin auf dem Mailserver bestehen und es erfolgt kein Hinweis an den Admin oder ähnliche Userkonten (Standard: »0« = keine Filterung).

Lösche Mail, wenn größer als ... Bytes
: Falls dieser Grenzwert überschritten wird, wird die Mail nicht nur nicht geladen, sondern auch auf dem Mailserver gelöscht – im Gegensatz zu News ist die Mail damit endgültig verloren, daher sollte diese Option wirklich nur dann verwendet werden, wenn man sich über die Auswirkungen im Klaren ist (Standard: »0« = keine Filterung).

Bearbeite Score-File "MailFilt.hst"
: Hier kann die Mail-Filterdatei »`MailFilt.hst`« bearbeitet werden.

POP3-Mailserver

Glossar: POP3

Hier werden die POP3-Server eingetragen, von welchen der Hamster E-Mails abholen soll.

Port

Zum Hinzufügen eines neuen POP3-Servers ist der Schalter »**Neu**« zu betätigen. Geben Sie im folgenden Dialog den Servernamen und bei Bedarf, durch ein Komma getrennt, auch noch den Server-Port ein (Beispiel: »`pop.example.com,110`« oder kurz »`pop.example.com`«). Für Standardports bekannter Dienste kann auch der übliche Name eingetragen werden, wie er in der Datei »`services`« zu finden ist (im Beispiel: »`pop.example.com,pop3`«). Bei Windows 95 bis Windows ME liegt diese Datei im Windows-Verzeichnis, bei Windows NT und Nachfolgern im Verzeichnis »`%SYSTEMDIR%\drivers\etc`« (normalerweise »`c:\windows\system32\drivers\etc`«).

Sollte ein externer Server mit demselben Namen bereits existieren *(gleichgültig, ob News- oder Mailserver)*, wird nachgefragt, ob ein Alias angelegt werden soll. Dieses erlaubt z. B. die sinnvolle Nutzung mehrerer Accounts. Der Alias ist durch ein nachgestelltes »/XX« erkennbar, wobei »XX« für beliebigen Text steht (z. B. eine fortlaufende Numerierung).[1]

[1] Das ist eine Folge des hamsterinternen Systems der Speicherung in Verzeichnissen, die den Namen des Servers tragen.

Einstellungen

Glossar: POP3 Port TCP

Zum Ändern der Einstellungen eines POP3-Servers dient der Schalter »**Einstellungen**«. Hier können Sie im Feld »POP3-Server-Port« einstellen, welchen TCP-Port der Hamster beim Provider benutzen soll (man kann in diesem Feld Eingaben machen – solange man im Dialogmenü bleibt, merkt sich der Hamster alle bisherige Eingaben, und diese sind dann über das Auswahlfeld anwählbar). Für die Portbezeichnung gilt dasselbe, was oben zur Neueinrichtung ausgeführt wurde. Sie können per Schalter die »`Server.ini`« bearbeiten (sie wird in dem Editor geöffnet, der im Menü »Einstellungen« → »Grundeinstellungen« ⇒ »Verschiedenes« eingestellt ist).

Zudem legen Sie hier Anmeldenamen und Passwort für den Zugriff fest oder löschen die aktuellen Angaben wieder (»**Bearbeiten**« bzw. »**Leeren**«). Die Felder für Benutzer und Passwort können drei Zustände haben:

- »{Fragen}«: Der Hamster fragt erst bei Bedarf Benutzername und Passwort ab. Dazu müssen Sie ein »?« als Benutzername oder/und Passwort eingeben.
- »{Gesetzt}«: Der Hamster speichert die Daten in der Datei »`Password.!!!`« ab (Lesen Sie dort unbedingt den Sicherheitshinweis).
- »{Leer}«: Der Hamster verwendet keine Authentifizierung.

Ebenfalls kann hier das Timeout für die Verbindung zum Server und für die Serverkommandos eingestellt werden. Soll ein Server erst einmal ungenutzt bleiben, muss man ihn nicht gleich löschen, sondern kann ankreuzen, dass dieser Server temporär deaktiviert sein soll.

Weiterhin kann eingestellt werden, wie die E-Mails nach dem Laden vom Server zu behandeln sind. (Löschen/nicht Löschen/globale Vorgabe), Standardeinstellung ist hier die globale Vorgabe für alle Server aus dem Menü »Einstellungen« → »Mail: Server konfigurieren« ⇒ »POP3-Einstellungen«.

Standardbenutzer

Hier können Sie angeben, welchem lokalen Benutzer die E-Mails von diesem Server zugeordnet werden sollen. Hierbei werden alle lokalen Benutzer in der Form »⟨Benutzername⟩ = ⟨Beschreibung⟩« zur Auswahl angeboten, z. B. »`admin = Hamster-Administrator`«.

Ebenso möglich ist hier die Angabe, welcher Abschnitt der Mailfilter-Datei »`Mailfilt.hst`« für diesen Server verwendet werden soll. Dieser Abschnitt kann gegebenenfalls gleich mittels Schalter bearbeitet werden.

Authentifizierung

Ermöglicht es, einen bestimmten Authentifizierungsmechanismus zu erzwingen.

Verwendung des APOP-Verfahrens zur Authentifizierung:
Um das APOP-Verfahren zu nutzen (erlaubt die verschlüsselte Übermittlung von Name und Passwort, bei POP3 werden diese Daten normalerweise unverschlüsselt übertragen),

ist dem Passwort das Präfix »APOP:«voranzustellen. Für die zwangsweise Verwendung des herkömmlichen Authentifizierungsverfahrens kann das Präfix »PASS:« verwendet werden. Dies ist jedoch nur erforderlich, falls »PASS:« oder »APOP:« Bestandteile des eigentlichen Passworts sind. Bei den Präfixen ist auf Großschreibung zu achten.

Beispiele für Eingaben im Passwort-Feld:

`secret`	→ Benutzt: USER/PASS-Standardeinstellung
`APOP:secret`	→ Benutzt APOP
`PASS:APOP:secret`	→ Benutzt USER/PASS mit Passwort »APOP:secret«.
`PASS:PASS:secret`	→ Benutzt USER/PASS mit Passwort »PASS:secret«.

Verwendung des SASL-Verfahrens zur Authentifizierung:

Glossar: SASL POP3

Um das SASL-Verfahren zu nutzen (erlaubt die verschlüsselte Übermittlung von Name und Passwort, bei POP3 werden diese Daten normalerweise unverschlüsselt übertragen), ist dem Passwort das Präfix des jeweiligen SASL-Mechanismus oder das Präfix »SASL:« für die automatische Auswahl des Mechanismus voranzustellen. Für die zwangsweise Verwendung des herkömmlichen Authentifizierungsverfahrens kann das Präfix »PASS:« verwendet werden. Dies ist jedoch nur erforderlich, falls »PASS:« Bestandteil des eigentlichen Passworts ist. Bei den Präfixen ist auf Großschreibung zu achten.

Unterstützt werden die SASL-Mechanismen LOGIN, PLAIN, DIGEST-MD5, CRAM-MD5, CRAM-SHA1,DIGEST-MD5 und EXTERNAL. Sie können die Auswahlmöglichkeit einschränken, indem Sie durch Mausklick auf »**...** « einen zusätzlichen Dialog aufrufen und unerwünschte oder dem Server unbekannte Mechanismen deaktivieren.

Beispiele für Eingaben im Passwort-Feld:

`CRAM-MD5:secret`	→ Benutzt CRAM-MD5 zwangsweise als Authentifizierungs-Mechanismus.
`SASL:secret`	→ Benutzt SASL zur automatischen Auswahl des Authentifizierungs-Mechanismus.

Verwendung des veralteten POP-AUTH-Verfahrens zur Authentifizierung:

Um das POP-AUTH-Verfahren für die Authentifizierung nutzen zu können, muss dem Passwort das Präfix »AUTH:« vorangestellt werden. Dabei ist auf Großschreibung des Präfixes zu achten.

Beispiel für Eingaben im Passwort-Feld:

`AUTH:secret`	→ Benutzt POP-AUTH zur automatischen Auswahl des Authentifizierungs-Mechanismus.

Achtung! Dieser Mechanismus sendet die Passwörter im Klartext, falls kein kryptografischer Mechanismus zwischen Server und Client vereinbart werden kann.

SSL-Einstellungen

Hier können Sie einstellen, ob Sie die Verbindung zum Provider kryptographisch mit dem SSL-Verfahren sichern wollen.

Glossar: SSL

Achtung! Um SSL in Verbindung mit dem Hamster nutzen zu können, werden zwei zusätzliche DLL-Dateien benötigt; mehr zum Thema siehe in der SSL-FAQ.

Es stehen hier 4 Nutzungsarten zur Verfügung:

- **Keine SSL/TLS-Nutzung**
- **SSL/TLS bei sicherem Port immer benutzen** – In diesem Fall wird eine SSL-Verbindung ohne SSL-Handshake hergestellt (POP3S), der Standardport dafür ist 995. Falls die Herstellung der SSL-Verbindung scheitert, wird keine TCP-Verbindung hergestellt.

  Port TLS TCP
- **SSL/TLS nutzen, wenn möglich (Standardport)** – Es wird nur dann SSL benutzt, wenn der Provider es anbietet,[1] andernfalls wird eine unsichere Verbindung hergestellt. **Diese Methode ist nicht zu empfehlen.**
- **SSL/TLS immer verwenden (Standardport)** – Es wird immer versucht, eine SSL-Verbindung mittels SSL-Handshake[1] herzustellen. Falls die Herstellung der SSL-Verbindung scheitert, wird keine TCP-Verbindung hergestellt.

Es stehen mehrere Verfahren zur Überprüfung der X.509-Server-Zertifikate zur Verfügung: X.509

- **Keine Überprüfung** – **Nicht zu empfehlen, da so keinerlei Sicherheit über die Identität des Servers besteht.**
- **Zertifikat überprüfen, wenn vorhanden** – Falls das Zertifikat nicht vom Server geliefert wird, wird trotzdem eine Verbindung hergestellt. **Achtung! Die Verbindung wird, falls kein Server-Zertifikat vorliegt, zwar verschlüsselt, aber der Server nicht authentifiziert.**
- **Zertifikat immer überprüfen** – Liefert der Server kein Zertifikat, dann wird keine Verbindung hergestellt.
- **Zusätzlich Serverzertifikat lokal überprüfen** – In diesem Fall wird nur eine Verbindung zum Server hergestellt, wenn sich dessen Zertifikat im lokalen Zertifikate-Verzeichnis oder in der lokalen Zertifikate-Datei befindet **Dieses ist die sicherste Methode, fragen Sie Ihren Provider nach seinem öffentlichen Server-Zertifikat.**

Für die lokale Prüfung des Server-Zertifikates können Sie im Feld »Überprüfung über Datei:« eine Datei mit einem Zertifikat im PEM-Format angeben. PEM

Löschen

Einen bestehenden POP3-Server können Sie über den Schalter »**Löschen**« aus dem Hamster entfernen. POP3

[1] Über das »STLS«-Kommando.

SMTP-Einstellungen

Glossar: SMTP

Hier können Sie Optionen für das Versenden von E-Mails festlegen:

Der Wert »**Maximale Sendeversuche je E-Mail**« legt die maximale Anzahl der Versuche fest, welche der Hamster unternimmt, um eine E-Mail an den Server des Provider zu senden (Standard: »3«).

Wenn das Feld »**E-Mail löschen, nachdem alle Sendeversuche erfolglos waren**« markiert ist, wird die E-Mail gelöscht, falls die festgelegte Anzahl von Versuchen, die E-Mail zu versenden, überschritten wurde.

SMTP-Mailserver

Hier werden die Mailserver eingetragen, über welche E-Mails per SMTP oder heutzutage meist SMTP-AUTH verschickt werden sollen. SMTP-AUTH

Zum Hinzufügen eines SMTP-Servers ist der Schalter »**Neu**« zu betätigen. Geben Sie im folgenden Dialog den Servernamen und bei Bedarf, durch ein Komma getrennt, auch noch den Server-Port ein (Beispiel »`mail.example.com,25`« oder kurz »`mail.example.com`«). Für Standardports bekannter Dienste kann auch der übliche Name eingetragen werden, wie er in der Datei »`services`« zu finden ist (im Beispiel: »`mail.example.com,smtp`«). Bei Windows 95 bis Windows ME liegt diese Datei im Windows-Verzeichnis, bei Windows NT und Nachfolgern im Verzeichnis »`%SYSTEMDIR%\drivers\etc`« (normalerweise »`c:\windows\system32\drivers\etc`«). Port

Sollte ein externer Server mit demselben Namen bereits existieren *(gleichgültig, ob News- oder Mailserver)*, wird nachgefragt, ob ein Alias angelegt werden soll. Dieses erlaubt z. B. die sinnvolle Nutzung mehrerer Accounts. Der Alias ist durch ein nachgestelltes »/XX« erkennbar, wobei »XX« für beliebigen Text steht (z. B. eine fortlaufende Numerierung).[1]

Einstellungen

Zum Ändern der Einstellungen eines SMTP-Servers dient der Schalter »**Einstellungen**«. Hier können Sie im Feld »SMTP-Server-Port« einstellen, welchen TCP-Port der Hamster beim Provider benutzen soll (man kann in diesem Feld Eingaben machen – solange man im Dialogmenü bleibt, merkt sich der Hamster alle bisherige Eingaben, und diese sind dann über das Auswahlfeld anwählbar). Für die Portbezeichnung gilt dasselbe, was oben zur Neueinrichtung ausgeführt wurde. Sie können per Schalter die »`Server.ini`« bearbeiten (sie wird in dem Editor geöffnet, der im Menü »Einstellungen« → »Grundeinstellungen« ⇒ »Verschiedenes« eingestellt ist). TCP

Ebenfalls kann hier das Timeout für die Verbindung zum Server und für die Serverkommandos eingestellt werden. Soll ein Server erst einmal ungenutzt bleiben, muss man ihn nicht gleich löschen, sondern kann ankreuzen, dass dieser Server temporär deaktiviert sein soll.

[1] Das ist eine Folge des hamsterinternen Systems der Speicherung in Verzeichnissen, die den Namen des Servers tragen.

Glossar: SMTP-AUTH

Zudem können Sie einstellen, ob für den Server eine Anmeldung nötig ist (SMTP-AUTH). Wenn diese Funktion aktiviert ist, legen Sie hier Anmeldenamen und Passwort für den Zugriff fest oder löschen die aktuellen Angaben wieder (»**Bearbeiten**« bzw. »**Leeren**«). Die Felder für Benutzer und Passwort können drei Zustände haben:

- »{Fragen}«: Der Hamster fragt erst bei Bedarf Benutzername und Passwort ab. Dazu müssen Sie ein »?« als Benutzername oder/und Passwort eingeben.
- »{Gesetzt}«: Der Hamster speichert die Daten in der Datei »`Password.!!!`« ab (Lesen Sie dort unbedingt den Sicherheitshinweis).
- »{Leer}«: Der Hamster verwendet keine Authentifizierung.

Verwendung des SASL-Verfahrens zur Authentifizierung:

SASL SMTP

Um das SASL-Verfahren zu nutzen (erlaubt die verschlüsselte Übermittlung von Name und Passwort, bei SMTP werden diese Daten normalerweise unverschlüsselt übertragen), ist dem Passwort das Präfix des jeweiligen SASL-Mechanismus oder das Präfix »SASL:« für die automatische Auswahl des Mechanismus voranzustellen. Für die zwangsweise Verwendung des herkömmlichen Authentifizierungsverfahrens kann das Präfix »PASS:« verwendet werden. Dies ist jedoch nur erforderlich, falls »PASS:« Bestandteil des eigentlichen Passworts ist. Bei den Präfixen ist auf Großschreibung zu achten.

Unterstützt werden die SASL-Mechanismen LOGIN, PLAIN, DIGEST-MD5, CRAM-MD5, CRAM-SHA1,DIGEST-MD5 und EXTERNAL. Sie können die Auswahlmöglichkeit einschränken, indem Sie durch Mausklick auf »**...** « einen zusätzlichen Dialog aufrufen und unerwünschte oder dem Server unbekannte Mechanismen deaktivieren.

Beispiele für Eingaben im Passwort-Feld:

`CRAM-MD5:secret` → Benutzt CRAM-MD5 zwangsweise als Authentifizierungs-Mechanismus.

`SASL:secret` → Benutzt SASL zur automatischen Auswahl des Authentifizierungs-Mechanismus.

SSL-Einstellungen

SSL

Hier können Sie einstellen, ob Sie die Verbindung zum Provider kryptographisch mit dem SSL-Verfahren sichern wollen.

Achtung! Um SSL in Verbindung mit dem Hamster nutzen zu können, werden zwei zusätzliche DLL-Dateien benötigt; mehr zum Thema siehe in der SSL-FAQ.

Es stehen hier 4 Nutzungsarten zur Verfügung:

- **Keine SSL/TLS-Nutzung**
- **SSL/TLS bei sicherem Port immer benutzen** – In diesem Fall wird eine SSL-Verbindung ohne SSL-Handshake hergestellt (SMTPS), der Standardport dafür ist 465. Falls die Herstellung der SSL-Verbindung scheitert, wird keine TCP-Verbindung hergestellt.

Port TLS TCP

Glossar: Port SSL TLS

- **SSL/TLS nutzen, wenn möglich (Standardport)** – Es wird nur dann SSL benutzt, wenn der Provider es anbietet,[1] andernfalls wird eine unsichere Verbindung hergestellt. **Diese Methode ist nicht zu empfehlen**.
- **SSL/TLS immer verwenden (Standardport)** – Es wird immer versucht, eine SSL-Verbindung mittels SSL-Handshake[1] herzustellen. Falls die Herstellung der SSL-Verbindung scheitert, wird keine TCP-Verbindung hergestellt.

TCP

Es stehen mehrere Verfahren zur Überprüfung der X.509-Server-Zertifikate zur Verfügung:

X.509

- **Keine Überprüfung** – **Nicht zu empfehlen, da so keinerlei Sicherheit über die Identität des Servers besteht.**
- **Zertifikat überprüfen, wenn vorhanden** – Falls das Zertifikat nicht vom Server geliefert wird, wird trotzdem eine Verbindung hergestellt. **Achtung! Die Verbindung wird, falls kein Server-Zertifikat vorliegt, zwar verschlüsselt, aber der Server nicht authentifiziert.**
- **Zertifikat immer überprüfen** – Liefert der Server kein Zertifikat, dann wird keine Verbindung hergestellt.
- **Zusätzlich Serverzertifikat lokal überprüfen** – In diesem Fall wird nur eine Verbindung zum Server hergestellt, wenn sich dessen Zertifikat im lokalen Zertifikate-Verzeichnis oder in der lokalen Zertifikate-Datei befindet. **Dieses ist die sicherste Methode, fragen Sie Ihren Provider nach seinem öffentlichen Server-Zertifikat.**

Für die lokale Prüfung des Server-Zertifikates können Sie im Feld »Überprüfung über Datei:« eine Datei mit einem Zertifikat im PEM-Format angeben.

PEM

Löschen

Zum Entfernen eines SMTP-Servers aus dem Hamster ist der Schalter »**Löschen**« zu betätigen.

SMTP

SMTP-Server, welcher standardmäßig für ausgehende Mail verwendet werden soll:

Der hier eingestellte SMTP-Server wird zum Verschicken von Mails verwendet, wenn mehr als ein SMTP-Server eingetragen ist. Zum Auswählen des Servers ist dieser in der Liste zu markieren und der Schalter »**Wählen**« zu betätigen.

Mail: Killfile-Protokoll ⇧ + Strg + K

Die Fenster zur Bearbeitung und Kontrolle des Mail-Score- bzw. -Killfiles

Einträge

Im ersten Fenster ist das Killfile-Protokoll zu sehen. Hier werden die gelöschten und ignorierten Mails übersichtlich nach Accountname sortiert angezeigt. Die Liste enthält alle

[1] Über das »STARTTLS«-Kommando.

Mails, die aufgrund der Filterregeln in der »`Mailfilt.hst`« gelöscht oder nicht geladen und zugleich mittels »Log(...)«-Befehl in das Protokoll aufgenommen wurden. Darüber hinaus werden hier auch Mails angezeigt, die aufgrund einer Größenbeschränkung in der »`hamster.ini`« ignoriert bzw. gelöscht wurden (»mail.filter.mailsize.ignore/mailfilter.mailsize.delete« im Abschnitt »[Setup]«).

Es ist zu beachten, dass einige wichtige Teilfunktionen des Killfile-Protokolls (wie das Nachladen und Löschen von Mails) nur funktionieren, wenn der verwendete Mailserver UIDL (»Unique ID Listing) unterstützt. Siehe dazu auch den entsprechenden Abschnitt im FAQ-Beitrag zu »Mail und News mit Telnet« unter »Mail abholen mit Telnet – POP3«.

Es werden Informationen bezüglich Typ (Filter-Ignore bzw. Filter-Kill), Mailserver, Account (Benutzer), UIDL, Score, Subject, From, To, CC, Datum, MID, References, Bytes und Zeilen angezeigt. Man kann einzelne Spalten durch Zusammenziehen auf Spaltenbreite Null auch unsichtbar machen. Die verdeckten Informationen der Spalten stehen über den Menüpunkt »Einträge« → »Zeige Details« bzw. das Kontextmenü weiterhin zur Verfügung.

Mit einem Klick auf eine Spaltenüberschrift können die Verweise nach der entsprechenden Spalte sortiert werden. Sollte einem diese Sortierung zu ungenau sein, kann man den Menüpunkt »Datei« → »Zusatzsortierung bearbeiten« verwenden, welcher eine Angabe eines zweiten und dritten Sortierkriteriums ermöglicht.

Mit einem Doppelklick auf die Mail erhält man weitere Informationen zu diesem Artikel. Über das Kontext-Menü ist ein Zugriff auf weitere Funktionen möglich. Die Funktionen des Kontext-Menüs stehen auch über das Menü »Einträge« zur Verfügung.

Einstellungen

Hier können Sie einige erweitere Einstellungen vornehmen:

Glossar: Header

TOP-Command – Header plus wieviel Zeilen holen?
Diese Einstellung bestimmt, wieviel Zeilen des Bodys einer E-Mail geholt werden (Standard: 20), siehe dazu auch die Ausführungen im Abschnitt zu diesem Kommando im FAQ-Beitrag zu »Mail und News mit Telnet« unter »Mail abholen mit Telnet – POP3«.

Zusätzliche Informations-Header in hereinkommende/gefilterte Mails einfügen:
Gibt Ihnen die Möglichkeit, einige informative Header in die E-Mails einzufügen oder diese Funktion abzuschalten. In der Voreinstellung werden alle Header gesetzt.

Filterdatei

Dieses Fenster stellt einen einfachen Editor für das Mail-Score-/Killfile dar (siehe auch: »Datei: MailFilt.hst«).

Um die Datei mit dem gewohnten Editor zu bearbeiten, kann aus dem Hamster-Hauptfenster heraus der Menüpunkt »Einstellungen« → »Spezielle Dateien« → »Mail-Score-/Killfile« genutzt werden.

Welcher Editor verwendet wird, kann über »Einstellungen« → »Grundeinstellungen« ⇒ »Verschiedenes« eingestellt werden.

Im Menü stehen folgende Punkte zur Auswahl:

Datei →

Zusätzliche Sortierung ...
Ermöglicht die Angabe des zweiten und dritten Sortierkriteriums.

Alle Änderungen speichern Strg + S
Speichert alle Einstellungen und Anforderungen für das Nachladen bzw. Löschen der Mails ab.

Alle Änderungen verwerfen
Verwirft alle Einstellungen und Anforderungen für das Nachladen bzw. Löschen der Mails.

Schließen
Beendet die Bearbeitung des Killfiles. Falls Änderungen durchgeführt wurden, die aber noch nicht gespeichert wurden, erfolgt eine entsprechende Nachfrage.

Einträge →

Zeige Details ...
Zeigt alle Details zu einem markierten Artikel an. Neben den Informationen aus sämtlichen Spalten der tabellarischen Übersicht enthält die Aufstellung auch das aus dem Datumswert der Mail berechnete Alter (»Age«).

Alle Auswählen Strg + A
Markiert alle Einträge im Killfile zur gemeinsamen Bearbeitung.

Zum Nachladen markieren Strg + M
Die UIDL der Mail wird zum Nachladen markiert und die Mail bei der nächsten Onlineverbindung automatisch geholt. Ist mindestens eine der ausgewählten Mails bereits gelöscht (Filter-Kill), so erscheint eine entsprechende Warnung. Der Eintrag verschwindet nach der Festlegung der auszuführenden Aktion aus der Liste.

Zum Löschen vom Server markieren Strg + K
Die UIDL der Mail wird zum Löschen markiert und die Mail bei der nächsten Onlineverbindung automatisch gelöscht (sofern dies nicht über die Mailserver-Optionen verboten wird). Ist mindestens eine der ausgewählten Mails bereits gelöscht (Filter-Kill), so erscheint eine entsprechende Warnung. Der Eintrag verschwindet nach der Festlegung der auszuführenden Aktion aus der Liste.

Log-Eintrag löschen Strg + E
Löscht den Eintrag aus der Liste, ohne eine Aktion festzulegen. Handelt es sich um eine bereits gelöschte Mail (Filter-Kill), so ist dies das übliche Vorgehen (über die anderen Menüpunkte erzeugt man unnötige Warnungen). Handelt es sich um eine lediglich ignorierte

Mail, so wird die Mail auf dem Server verbleiben. Hat man in den Mailserver-Optionen festgelegt, dass Mails nur einmal geladen werden (»[x] kein zweites Mail laden«), so erhält man darüber auch keine weitere Mitteilung. Wählt man diese Option jedoch ab, so erhält man beim nächsten Mal erneut eine Mitteilung und der Eintrag taucht wieder in der Liste auf.

(*Hinweis:* Wenn man bei einer Mail zwei Notifies hat und von denen einen zum Löschen und einen zum Nachladen anwählt, so wird die Mail geladen!)

Das Kontextmenü zeigt noch einmal die Punkte des Menüs »Einträge«.

Aktiviere/Deaktiviere erweiterte Einstellungen

Hier können zusätzliche Optionen des Hamster sichtbar gemacht werden, welche entweder ziemlich speziell sind oder ohne fortgeschrittene Kenntnisse noch nicht sinnvoll verwendbar sind. Dies umfasst sowohl Menüpunkte des Einstellungs-Menüs als auch verschiedene Felder und Unterseiten in den diversen Einstellungsdialogen. Innerhalb der Dialoge steht meistens die Möglichkeit zur Verfügung, diese verborgenen Optionen temporär anzuzeigen.

Achtung! Wenn Sie diese erweiterten Optionen ändern möchten, so sollten Sie sich vorher über deren Funktion in der Hilfe informieren und sich den Originalzustand merken.

Spezielle Dateien

Mit dem im Menü »Einstellungen« → »Grundeinstellungen« ⇒ »Verschiedenes« eingestellten Texteditor können hier die Konfigurationsdateien des Hamsters bearbeitet werden.

Konfiguration neu laden F5

Mit diesem Menüpunkt können alle Einstellungen der Konfigurationsdateien erneut geladen werden.

Menü: Online

1–9: ...

Zeigt die im Menü »Einstellungen« → »Grundeinstellungen« ⇒ »Online-Menü« definierten Servergruppen unter dem eingestellten Menünamen an. Eine sinnvolle Benutzung setzt natürlich eine Internetverbindung voraus, eine automatische Einwahl ist nicht vorgesehen.

Alle Server (News+Mail) / Alle NNTP-Server / Alle POP3-Server ⇧ + Strg + A

Glossar:
NNTP
POP3
SMTP

Diese Funktion startet **alle** vorhandenen NNTP-, POP3- und SMTP-Server, beziehungsweise die **ausgewählte Gruppe** der Server (NNTP und POP3).

⟨Einzelner Server⟩

Diese Funktion startet **einzelne** vorhandene Server des jeweiligen Servertyps.

DFÜ-Netzwerk

Diese Funktion startet vorhandene DFÜ-Netzwerkverbindungen. Zur sinnvollen Nutzung sollten Benutzername und Passwort (zumindest »?« für manuelle Eingabe) per Menüpunkt »Einstellungen« → »Benutzerverwaltung & Passworte« ⇒ »Passworte« oder im RAS-Client von Windows eingegeben worden sein.

Die Einwahl wird nicht als Thread durchgeführt, somit kann der Hamster bei Einwahlproblemen kurzfristig »hängen«, dies deutet *nicht* auf eine Fehlfunktion hin!

DFÜ-Verbindung schließen

Diese Funktion schließt eine vorhandene DFÜ-Netzwerkverbindung.

Alle Aufträge beenden

Diese Funktion beendet sofort **alle** aktiven Prozesse im Hamster.

Menü: Skript

⟨Verfügbare Skripte⟩

Startet das ausgewählte Skript. Angezeigt wird jeweils der Dateiname ohne die Erweiterung »`.hsc`«.

Skripte und Module verwalten F4

Startet den integrierten Skriptmanager. Zur Syntax der Skripte siehe die Erläuterungen zur Skriptsprache »**hs2**«.

In der Standardeinstellung erzeugt dieses Fenster keinen eigenen Eintrag in der Windows-Taskleiste. Durch folgenden Eintrag in der »`hamster.ini`«, Abschnitt »[EditScripts]«, können Sie das ändern: »OwnTaskbarEntry=1«.

Für bequemes und vor allem übersichtliches Editieren von Skripten und Scorefiles existieren Syntaxschemata. Das sind Schablonen, die den jeweiligen Editor dazu bringen, die verschiedenen Bestandteile der Syntax einer Skriptsprache auch verschieden(farbig) darzustellen (Syntaxhighlighting). Hinweise dazu finden Sie im Hamsterarchiv, siehe »Links für Informationen, weitere Hilfe und Zusatztools zum Hamster«.

Alle Skripte beenden

Beendet sofort **alle** aktiven Skripte.

Menü: Lokale Server

Start/Stop News-Server (NNTP) F6

Glossar: NNTP

Dieses Menü startet bzw. stoppt den lokalen NNTP-Server, so dass im letzteren Fall vom Newsreader weder Postings verschickt noch abgeholt werden können. Im Menü »Einstellungen« → »Automatische Abläufe« ⇒ »Allgemeines« kann eingestellt werden, dass der Start dieses Servers nicht automatisch mit dem Hamster-Start erfolgt.

Start/Stop Mail-Server (POP3) F7

POP3

Dieses Menü startet bzw. stoppt den lokalen POP3-Server, so dass im letzteren Fall vom Mailreader keine Mails mehr empfangen werden können. Im Menü »Einstellungen« → »Automatische Abläufe« ⇒ »Allgemeines« kann eingestellt werden, dass der Start dieses Servers nicht automatisch mit dem Hamster-Start erfolgt.

Start/Stop Mail-Server (SMTP) F8

SMTP

Dieses Menü startet bzw. stoppt den lokalen SMTP-Server, so dass im letzteren Fall vom Mailreader keine Mails mehr an den Hamster verschickt werden können. Im Menü »Einstellungen« → »Automatische Abläufe« ⇒ »Allgemeines« kann eingestellt werden, dass der Start dieses Servers nicht automatisch mit dem Hamster-Start erfolgt.

Start/Stop Mail-Server (IMAP) F9

IMAP

Dieses Menü startet bzw. stoppt den lokalen IMAP-Server, so dass im letzteren Fall vom Mailreader keine Mails mehr vom Hamster empfangen werden können. Im Menü »Einstellungen« → »Automatische Abläufe« ⇒ »Allgemeines« kann eingestellt werden, dass der Start dieses Servers automatisch mit dem Hamster-Start erfolgt.

Start/Stop Telnet-Server

Dieses Menü startet bzw. stoppt den lokalen Telnet-Server, welcher den Hamster fernsteuert. Im Menü »Einstellungen« → »Automatische Abläufe« ⇒ »Allgemeines« kann eingestellt werden, dass der Start dieses Servers automatisch mit dem Hamster-Start erfolgt.

Siehe auch die allgemeinen Erläuterungen zum Remote-Control-Server.

News-/Mail-Verzeichnisse F3

Dieses Menü öffnet ein Fenster, das eine Übersicht über die beiden Ordner »`News.Out`« und »`News.Err`«, die Mailverzeichnisse (also in der Voreinstellung alle Ordner unterhalb von »⟨`Hauptverzeichnis`⟩`\Mails`«) und den Mülleimer bietet sowie das Editieren oder Löschen der enthaltenen Dateien ermöglicht.

In der Standardeinstellung erzeugt dieses Fenster keinen eigenen Eintrag in der Windows-Taskleiste. Durch folgenden Eintrag in der »`hamster.ini`«, Abschnitt »[EditDirs]«, können Sie das ändern: »OwnTaskbarEntry=1«.

Anhand der Symbole in der oberen Leiste kann man erkennen, ob ein Ordner »`Msg`«- bzw. »`Err`«-Dateien enthält, auch wird die Zahl der enthaltenen Dateien in Klammern angezeigt, getrennt nach »`*.msg`« und »`*.err`«. Ein Klick auf ein Symbol zeigt die Dateien (E-Mails) des jeweiligen Kontos an. Bei IMAP-Konten beziehen sich die Zähler *ausschließlich* auf die Dateien des Posteingangs (INBOX). Dateien in Unterordnern werden zwar mit ihrem Relativpfad angezeigt, aber nicht mitgezählt. Durch Setzen der Option »ShowIMAPSubDirs=0« in der »`hamster.ini`«, Sektion »[EditDirs]«, werden die Dateien der Unterordner ausgeblendet.

Übersicht über die Symbole (große Variante):

Ordner	leer	gefüllt
»`News.out`«		
»`News.err`«		
»`Mail.out`«		
Mailordner		
Mülleimer		

Zunächst werden die enthaltenen Dateien in einer speziellen Detaildarstellung angezeigt. Sie können (in der Standardeinstellung) durch Mausklick auf die Spalten die Sortierung ändern (Standard: aufsteigend nach Dateiname). Welche Spalten angezeigt werden und in welcher Reihenfolge, können Sie in der »`hamster.ini`« ändern, siehe Sektion »[EditDirs]«in Hamster.ini, sonstige Sektionen (»mail.ShowHeaders« bzw. »news.ShowHeaders«).

Durch Einfachklick auf eine Datei wird auf der rechten Seite eine einfache Vorschau geöffnet. Durch Drücken der Esc-Taste wird die Dateivorschau wieder geschlossen.

Durch Doppelklick auf eine Datei (oder den entsprechenden Punkt im Menü »Datei« oder im Kontextmenü) wird sie standardmäßig in dem Editor geöffnet, der im Menü »Einstellungen« → »Grundeinstellungen« ⇒ »Verschiedenes« eingestellt ist. Sie können in der »`hamster.ini`«, Sektion »[Setup]«, speziell für »Msg«- bzw. »Err«-Dateien einen eigenen Editor festlegen, siehe dazu unter »Editor.<ext>.app«.

Änderungen an den Dateien sollten nur dann erfolgen, wenn Sie sich über deren Aufbau wirklich im Klaren sind!

Glossar: POP3 IMAP

Innerhalb der beiden Newsordner und des Mülleimers sowie innerhalb der Ordner für POP3-Konten, des Mail.out-Ordners und des Mülleimers, also aller Ordner *außer* denen für IMAP-Konten, können Sie die Dateien per Drag & Drop verschieben.

Löschen von Dateien

Neben dem Verschieben per Maus in den Mülleimer gibt es für die beiden Newsordner und die Ordner für POP3-Konten auch die Möglichkeit, das Löschen per Tastenkombination auszulösen:

- Entf verschiebt in den Mülleimer.
- ⇧ + Entf verschiebt ohne Nachfrage.
- Strg + Entf löscht sofort. Die Datei wird auch *nicht* in den Windows-Papierkorb verschoben.
- ⇧ + Strg + Entf löscht direkt ohne Sicherheitsnachfrage. *Seien Sie vorsichtig damit!*

(D. h. mit der Umschalttaste ⇧ unterdrückt man jeweils die Nachfrage.)

Wie lange die Dateien im Mülleimer erhalten bleiben, können Sie unter »Einstellungen« → »Automatische Abläufe« ⇒ »Allgemeines« einstellen.

Achtung! Wegen der speziellen technischen Implementation ist das Löschen innerhalb der Ordner von IMAP-Konten nicht möglich. Nutzen Sie dafür die Möglichkeiten Ihres IMAP-Clienten.

Verfügbare Menüeinträge

Datei →

Bietet verschiedene Möglichkeiten zur Manipulation der Dateien.

Editor Strg + E
Öffnet ausgewählte Datei im eingestellten Editor, siehe oben.

Löschen Entf ***/ Kopieren*** Strg + T ***/***
Verschieben Strg + M ***/ Umbenennen*** F2
Verschiedene Aktionen mit ausgewählter Datei *(nicht in IMAP-Ordnern!)*, siehe auch oben.

Explorer starten
Öffnet den Windows-Explorer in dem gerade aktivierten Mail- oder Newsordner.

Bearbeiten →
Bietet verschiedene Funktionen, um die ausgewählte Datei oder deren markierten Teil zu decodieren.

Alle auswählen Strg + A
Markiert alle Dateien im ausgewählten Ordner.

ROT-13 Strg + R
Zur Entschlüsselung »ROT13«-codierter Passagen. Für eine ausführliche Information siehe unter http://de.wikipedia.org/wiki/ROT13.

Ansicht →
Alle Dateien anzeigen Strg + 0 **/**
Nur Postings/E-Mails (*.msg) Strg + 1 **/**
Nur unverschickbare Postings (*.err) Strg + 2
Wählt aus, welcher Dateityp angezeigt wird. Ist die Anzeige aller Dateien ausgewählt, so werden sie nur in einer einfachen Liste angezeigt.

Kleine Icons verwenden
Wird diese Option aktiviert, ist bei einem leeren Ordner kein Symbol zu sehen und bei einem gefüllten nur ein kleiner gelber Punkt.

Aktuelle als Standard-Sortierung speichern
Ist erst aktiv, wenn man die bisher vorgegebene Sortierung geändert hat.

Vorschau →
Vorschau aktiviert Strg + P
Hier kann man die Vorschau deaktivieren.

Festbreitenschrift Strg + F
Voreingestellt ist für die Vorschau eine proportionale Schriftart (Standard: »Microsoft Sans Serif« in Schriftgröße 8). Für viele Fälle ist aber die Ansicht in einer Festbreitenschrift, d. h. nichtproportionalen Schriftart, sinnvoller. Das kann hier aktiviert werden (Vorgabe ist »Courier New« in Schriftgröße 9).

Gesamter Header Strg + H
In der Standardeinstellung werden nur die wichtigsten Header-Zeilen angezeigt. Nach Aktivierung ist der gesamte Headerblock zu sehen.

Glossar: Header

Automatischer Umbruch Strg + W
Nach Aktivierung dieser Option werden überlange Zahlen automatisch umbrochen, sonst wird eine Laufleiste eingeblendet.

Textattribute →

Formatieren und Originalzeichen unterdrücken /
Formatieren, aber Originalzeichen beibehalten /
Als normalen Text anzeigen

Hier können Sie auswählen, wie Text dargestellt wird, der durch Pseudoformatierung ausgezeichnet ist (»Textattribute«). Voreinstellung ist »Formatieren und Originalzeichen unterdrücken«:

Formatieren und Originalzeichen unterdrücken:	*Formatieren, aber Originalzeichen beibehalten:*	*Als normalen Text anzeigen (keine Formatierung):*
Fetter Text	***Fetter* *Text***	*Fetter* *Text*
Kursiver Text	*/Kursiver/ /Text/*	/Kursiver/ /Text/
Unterstrichener Text	_Unterstrichener_ _Text_	_Unterstrichener_ _Text_

Das **Kontextmenü** bietet noch einmal die wichtigsten Funktionen des Dateimenüs.

Hauptfenster und dazugehörige Kontextmenüs

Protokolle für NewsJobsListe

Sofern noch Aufträge in der Newsjobsliste auf ihre Ausführung warten, werden diese im Unterfenster »Offene Newsjobs« nach Priorität sortiert angezeigt. Pro Zeile wird dabei Auftragstyp, Zusatzdaten des Auftrags und Priorität angezeigt. Über die Buttons oberhalb der eigentlichen Liste kann die Gesamtliste geleert, der aktuelle Auftrag gelöscht oder seine Priorität geändert werden.

Protokolle für ausgewählte Aufträge

Hier können Sie sich das Log für den markierten Auftrag in einem separaten Fenster anschauen.

Kontextmenü im Hauptfenster

Kopiere Einzelzeile in die Zwischenablage

Hier können Sie eine einzelne Zeile des Logs in die Zwischenablage kopieren.

Kopiere Protokoll in die Zwischenablage

Hier können Sie das gesamte Protokoll in die Zwischenablage kopieren.

Protokoll(e) leeren

Mit diesem Menüpunkt können Sie das gesamte Protokoll löschen.

hs2-Shell Alt + 2

In die hs2-Shell können einzelne Hamsterskript-Anweisungen eingegeben und mit Drücken auf Enter↵ ausgeführt werden. Mit Mausklick auf »>« öffnet sich ein Menü, das eine reiche Auswahl aus den Hamsterskript-Anweisungen bietet.

Bei Eingabe von Anweisungs-Sequenzen ist anstelle eines Zeilenumbruchs das Zeichen »¶« einzugeben, das Sie mit Strg + T erhalten:

```
HamNewsJobsPost("news.example.com")¶HamNewsJobsStart
```

Die letzten 100 ausgeführten Anweisungen werden in der Datei »ShellHis.log« gespeichert, die ihrem Typ entsprechend im Log-Verzeichnis zu finden ist. Sie können das Abspeichern der Shell-History auch abschalten, indem Sie in der »hamster.ini« in der Sektion »[Main]« den Schlüssel »ShellHist.Log« mit dem Wert »0« eintragen.

hs2Shell.hsm

Wenn im Moduleverzeichnis (siehe Schlüssel »Modules« in der »hamster.ini«, Sektion »[Directories]«) eine Moduldatei mit dem Namen »hs2Shell.hsm« gefunden wird, dann wird sie in die hs2-Shell eingebunden,[1] und alle dort definierten Befehle stehen zur Verfügung, was vor allem für umfangreiche Test-Routinen sehr nützlich ist.

Sie können die Datei »hs2Shell.hsm« anlegen oder bearbeiten über das Menü »Skript« → »Skripte und Module verwalten …« ⇒ »Allgemeines« → »Standardmodul für hs2-Shell«; sie wird beim Speichern im richtigen Pfad abgelegt.

Wie in jedem Modul darf in der »hs2Shell.hsm« ein »#!initialize«- und ein »#!finalize«-Bereich vorkommen. Es jedoch zu beachten, dass das Modul bei jeder Ausführung der Shell neu geladen wird.

Beispiele:

Bsp. 1:
```
# hs2shell.hsm
#!hs2
Sub pullXmplNews()
   HamNewsJobsPull("news.example.org")
   HamNewsJobsStart
EndSub

Sub postXmplNews()
   HamNewsJobsPost("news.example.org")
```

[1] Technisch gesehen durch eine Load-Anweisung zu Beginn des temporären Skriptes: »#!load hs2Shell.hsm«.

```
   HamNewsJobsStart
EndSub
```

Aufruf in der hs2-Shell: `pullXmplNews` *oder* `postXmplNews` *oder* `postXmplNews¶HamWaitIdle(10000)¶pullXmplNews`

Bsp. 2:
```
# hs2shell.hsm
#!hs2
Sub NewsXChange( $server )
   Print( "Starte News-Austausch mit: " + $server )
   HamNewsJobsPost( $server )
   HamNewsJobsStart
   HamWaitIdle( 10000 )
   HamNewsJobsPull( $server )
   HamNewsJobsStart
EndSub
```

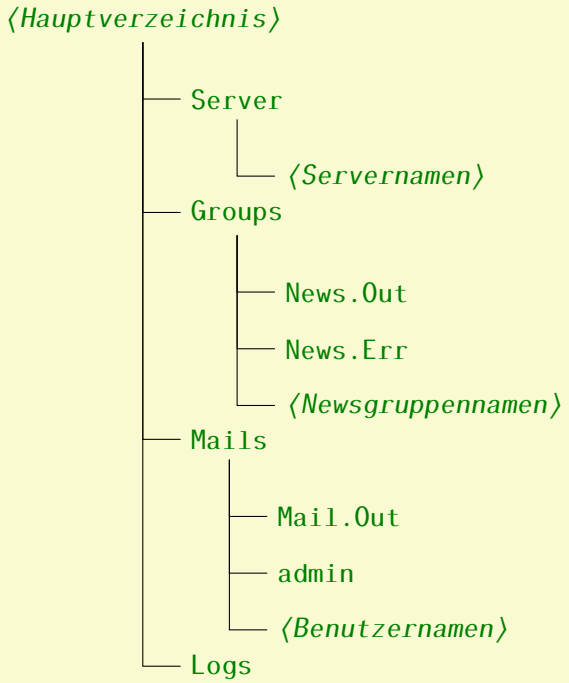
Aufruf in der hs2-Shell:
`NewsXChange("news.example.com")` *oder* `NewsXChange("news.example.net")` *oder* `NewsXChange("news.example.org")`

Wie in Bsp. 2 zu sehen ist, erfolgt die Wertzuweisung zu der Variablen, die in der Prozedur deklariert wurde, über die hier als Input-Box wirkende Eingabezeile. Das funktioniert auch dann, wenn für die Prozedur im Parameter mehr als eine Variable deklariert wurde.

Dateien & Verzeichnisse

Dateien und Verzeichnisse

Verzeichnisübersicht *(Standardeinstellung, siehe Hinweis)*:



(In der Hamsterhilfe wird weitgehend die klassische Bezeichnung »Verzeichnis« anstelle von »Ordner« verwendet.)

Hinweis: Seit Version 1.3.21.0 sind die Pfade per »hamster.ini« änderbar (Sektion »[Directories]«). Skript-Autoren sollten daher die entsprechenden OLE-/hs2-Funktionen und nicht die festen Pfade nutzen.

Dateien im Hauptverzeichnis

Hinweis: Auch der Pfad zu diesen Dateien ist änderbar!

Hamster.exe
: Die ausführbare Hauptdatei.

Hamster.ini
: In dieser Datei speichert Hamster die meisten Einstellungen, sofern sie nicht serverspezifisch sind oder einen komplexeren Aufbau haben, der eine eigene Datei rechtfertigt. Diese Datei wird spätestens beim ersten Hamsterstart erzeugt.

Hamster.hlp
: Hamsters englische Hilfedatei.

Hamster_de.hlp
: Hamsters deutsche Hilfedatei. Die PDF-Version dazu lesen Sie gerade.

`Hamster_XX.dat`
: Sprachdateien. Die beiden »X« sind Platzhalter für die jeweilige Landeskennung (»de« steht für die deutschsprachige Version).

`Changes_en.txt`
: Versions-History in Englisch.

`Changes_de.txt`
: Versions-History in Deutsch (ab Version 1.3.21.xx).

`Groups.hst`
: Liste der Newsgruppen im Hamster.

`Pulls.hst`
: Liste der Ladeaufträge für Newsgruppen.

`Server.hst`, `SrvPOP3.hst`, `SrvSMTP.hst`
: Liste der NNTP-, POP3- und SMTP-Server.

Glossar:
NNTP
POP3
SMTP

`Password.!!!`
: Passwortdatei.

`Accounts.!!!`
: Lokale Benutzer.

`Groups.!!!`
: Lokale Benutzergruppen.

`IPAccess.hst`
: Zugangsberechtigungen auf IP-Basis. Erlaubt den lokalen Zugriff, standardmäßig auch den Zugriff aus dem lokalen Netzwerk und bei entsprechenden Änderungen auch aus dem Internet.

`Scores.hst`
: Scorefile für News. Erlaubt es, Artikel anhand der sogenannten Overview-Informationen schon vor dem eigentlichen Einliefern zu löschen. Wird nicht automatisch angelegt, Erstellen und Bearbeiten z. B. mittels Menü »Einstellungen« → »Spezielle Dateien« → »News-Score-/Killfile«.

`MailFilt.hst`
: Filterdatei für E-Mails. Erlaubt auch die Verteilung auf mehrere Benutzer anhand der konkreten Mails oder das Importieren von Mailinglisten in entsprechende lokale Newsgruppen. Sie wird nicht automatisch angelegt, Erstellen und Bearbeiten z. B. mittels Menü »Einstellungen« → »Spezielle Dateien« → »Mail-Score-/Killfile«.

`MAlias.hst`
: Datei zum Umleiten von E-Mail-Adressen und Erstellen von virtuellen E-Mail-Adressen.

`*.cnv` (Zeichensatzkonvertierungsdateien)
: Dateien zum Konvertieren der Zeichensätze von Artikeln bzw. Mails in den lokalen Zeichensatz.

`Ham.exe`
: Kommandozeilen-Tool (für Details geben Sie »`ham help`« in die Eingabeaufforderung ein).

`*.hsc, demo*.hsc`
: (Beispiel-)Hamster-Skripte. Falls in der »`hamster.ini`« der Pfad für die Skripte umgestellt worden ist, sollten zumindest die eigenen Skripte natürlich in das entsprechende Verzeichnis kopiert werden.

`*.hsm`
: Hamsterskript-Module. Falls in der »`hamster.ini`« der Pfad für die Module umgestellt worden ist, sollten diese natürlich in das entsprechende Verzeichnis kopiert werden.

`*.pl, demo*.pl`
: (Beispiel-)Perl-Skripte

`*.vbs, demo*.vbs`
: (Beispiel-)VBScript-Skripte (»Windows Scripting Host«/WSH)

`Hamst_xy.ico`
: Die Hamstericons im Tray sind durch ein eigenes Icons ersetzbar, die Benennung ist »`Hamst_xy.ico`«, wobei **`x`** für »1« (kein laufender Thread) oder »2« (mind. ein Thread aktiv) steht und **`y`** für »a« (alles OK), »b« (mind. eine Warnung) oder »c« (mind. ein Fehler).

Hamster.ini

Diese INI-Datei enthält alle wesentlichen Hamster-Einstellungen, sofern diese nicht server-/gruppenspezifisch sind oder aufgrund ihrer Komplexität in gesonderten Dateien gespeichert werden. Die hier aufgeführten Einstellungen sind nur teilweise über die Einstellungsmenüs erreichbar und müssen somit bei Bedarf direkt in der INI-Datei geändert werden. Die INI-Datei ist hierbei in verschiedene Sektionen gegliedert. Jede Sektion beginnt mit dem in eckigen Klammern gefassten Namen. Folgende Sektionen sind vorhanden:

`[Directories]`
: Dieser Abschnitt erlaubt die freie Konfiguration des Hamster-Verzeichnisbaumes. Änderungen können ggf. die Funktionalität älterer Zusatzprogramme oder Skripte beeinträchtigen.

`[Setup]`
: Die Sektion Setup ist die allgemeine Sektion für alle Einstellungen, welche keine eigene Sektion besitzen.

[Main]

Die Sektion Main legt das Layout der Hamster-Oberfläche fest.

[Actions]

Unter Actions werden im Hamster die Möglichkeiten zusammengefasst, bei bestimmten Aktionen automatisch ein externes Programm und/oder ein Hamster-Skript ausführen zu lassen.

Weiterhin gibt es noch einige andere Sektionen, die in der Hilfe zusammengefasst sind:

[DlgGroupSelect] / [EditDirs] / [EditScripts] / [KillList] / [MailKillList] / [Marker] / [Online-Menu] / [ScoreEdit] / [SelPulls] / [SSL] / [Stats]

Sektion [Directories]

```
[Directories]
; Dieser Abschnitt erlaubt die freie Konfiguration des Hamster-Verzeichnisbaumes.
; Änderungen können ggf. die Funktionalität älterer Zusatzprogramme oder Skripte
; beeinträchtigen.

; Mit dieser Angabe kann man die Programmdatei getrennt von der Konfiguration und den
; Daten auf dem Rechner ablegen, oder man kann zu Testzwecken als Entwickler o. ä.
; eine »hamster.exe« auf die komplette Installation eines bereits an anderer Stelle
; installierten Hamsters zugreifen lassen und so dessen Daten und sonstige Dateien
; (Hilfedatei, Skripte, *.cnv, *.hst usw.) nutzen.
; In demselben Verzeichnis wie die »hamster.exe« muss dann eine »hamster.ini« abgelegt
; werden, die außer dem Sektionsbezeichner »[Directories]« ausschließlich den folgen-
; den Eintrag besitzt.
Main=⟨Pfad zu Konfigurationsdateien und Daten⟩

; Die folgenden Verzeichnisse können sowohl absolut (mit Laufwerksangabe) als auch re-
; lativ zum Hauptverzeichnis angegeben werden.
Server=⟨Pfad für Serververz., Standard: »Server«⟩
Groups=⟨Pfad für Newsgruppenverz., Standard: »Groups«⟩
News.out=⟨Pfad für ausgehende News, Standard: »Groups/News.Out«⟩
News.err=⟨Pfad für nicht gepostete Artikel. Standard: »Groups/News.Err«⟩
Mails=⟨Pfad für Mailverz., Standard: »Mails«⟩
Mail.out=⟨Pfad für ausgehende Mails, Standard: »Mails/Mail.Out«⟩
Logs=⟨Pfad für das Protokollverzeichnis, Standard: »Logs«⟩
Scripts=⟨Pfad für Skripte (*.hsc), Standard: Hauptverzeichnis⟩
Modules=⟨Pfad für Skriptmodule (*.hsm), Standard: Skriptpfad⟩
Trash=⟨Pfad für zum Löschen vorgesehene Artikel, Standard: »Trash«⟩
Charsets=⟨Pfad für Zeichensatz-Konvertierungstabellen (*.cnv), Standard: Hauptver-
        zeichnis⟩
ScriptsRC=⟨Pfad für Skripte, die der ReCo-Server verwenden kann, Standard: Skriptpfad⟩
```

Glossar: ReCo

Sektion [Setup]

```
[Setup]
; Die Sektion Setup ist die allgemeine Sektion für alle Einstellungen, die keine
; eigene Sektion besitzen.

; Einstellungen für Hamster-Interna und GUI

; Diese Einstellung gibt an, ob in der Ansicht der Gruppen/Benutzer im Menü
; »Einstellungen → Benutzerverwaltung & Passworte ⇒ Lokale Benutzer« die Treeview
; automatisch aufgeklappt werden soll, oder ob man dieses beim Erreichen des Dialogs
; manuell machen will (sinnvoll bei vielen Benutzern).
config.local.treefull=1

; Legt den Editor für die Bearbeitung von Textdateien und seine Parameter fest. Der
; Standard ist (ohne Eintrag in die »hamster.ini«) das Windows-Notepad (»notepad.exe«)
; mit Parameter »%1«.
editor.app=notepad.exe
editor.params=%1

; Auf der Endung ».⟨ext⟩« basierend können verschiedene Editor-Aufrufe definiert wer-
; den, um z. B. für die Endung ».msg« einen separaten Editor zu nutzen. Der zweite
; Eintrag ist für eventuell notwendige Parameter (mind. »%1«). Falls kein spezieller
; Aufruf definiert ist, wird der Standard-Editoraufruf verwendet.
editor.⟨ext⟩.app=
editor.⟨ext⟩.params=%1

; Anzahl der verwendeten chunkbits für die Log-Dateien. ACHTUNG! Die Anzahl der hier-
; für erforderlichen Dateien wächst quadratisch mit der Anzahl der Bits:
; »0« = eine Datei, »2« = vier Dateien, ... , »8« = 256 Dateien.
; Standard ist 4 Bit und somit 16 Dateien.
; Um Änderungen durchzuführen (nicht empfohlen):
;   1. Hamster schließen.
;   2. »history.chunkbits« ändern.
;   3. »hist*.dat« im »Groups«-Verzeichnis löschen.
;   4. History über Menü (neu) erzeugen.
history.chunkbits=4

; Zwischenspeichern der Sprachdateien:
; x=1 (Standard): Einstellungen werden nur beim Start oder bei einem Sprachwechsel
; eingelesen,
; x=0 (für Übersetzer/Entwickler): Direkttest für das Lesen aus der entsprechenden
; Sprachdatei, fehlende Einträge werden automatisch in die INI-Datei geschrieben.
Language.cached=x

; Update der Sprachdateien:
; x=1: Wenn zusätzlich »Language.cached« auf 1 steht, werden fehlende Einträge in der
; in der englischen Sprachdatei ebenfalls automatisch gefüllt (Standard: 0).
Language.Update.en.dat=x

; Aus Performance-Gründen ist die automatische Ergänzung der Sprachdateien standard-
; mäßig deaktiviert, Reaktivierung mit:
language.savefile=1
```

```
; Hier kann das Dateinamenschema für die Haupt-Logdateien eingestellt werden, der
; String muss dabei »%d« für die Nummer bzw. »%.xd« enthalten, wobei x für die
; Anzahl der Stellen steht, die ggf. mit Nullen aufgefüllt werden.
; Beispiel: »log.FormatFileName=Ham_%.4d.log« beginnt mit »Ham_0000.log«.
log.FormatFileName=

; Schaltet die Warnung »Connection lost«/»Verbindung verloren« ab.
; Workaround ist für unsauber programmierte Reader, wie z. B. Outlook Express.
Log.IgnoreConnectionLost=1

; Logausgaben in den Sub-Fenstern »ges. Protokoll« (»logfilter.ViewXXX«) und
; »Aufträge (Threads)« (»logfilter.ThreadXXX«) bzw. in den Logdateien
; (»logfilter.FileXXX«) können unterdrückt werden. In den Sprachdateien
; »hamster_xx.dat« stehen Kandidaten entsprechend dem Typ »key=Wert«; mit
; »logfilter.YYYByKey« wird die Angabe links vom Gleichheitszeichen ausgewertet,
; mit »logfilter.YYYByText« die Angabe rechts vom Gleichheitszeichen.
; Beispiel:
; Beim Start eines Skripts erschienen immer vier Zeilen.
;   Sys {fff78aed} Starte Skript: E:\HAMSTER\Scripts\Misc\e-time.hsc, Warten=0
;   Sys {fffae921} {script e-time.hsc} Start
;   Sys {fffae921} {script e-time.hsc} Ende
;   Sys {fff78aed} Skript E:\HAMSTER\Scripts\Misc\e-time.hsc beendet.
; Mit »logfilter.ViewByKey=^Skript\.(Start|End)$« kann man zwei Zeilen unterdrücken.
; (In der »hamster_de.dat« heißen die zwei Einträge »Skript.Start=Start« und
; »Skript.End=Ende«.)
; Ergebnis:
;   Sys {fff78aed} Starte Skript: E:\HAMSTER\Scripts\Misc\e-time.hsc, Warten=0
;   Sys {fff78aed} Skript E:\HAMSTER\Scripts\Misc\e-time.hsc beendet.
; Siehe auch die FAQ »Hamster und Reguläre Ausdrücke«.
; ACHTUNG: Jede der unten aufgeführten Zeilen darf nur einmal genutzt werden. Zur Un-
; terdrückung mehrerer Logeinträge müssen diese eventuell in einem Regulären Ausdruck
; vereint werden (durch ODER-Verknüpfung wie im obigen Bsp.).
logfilter.ViewByKey=⟨RegEx⟩
logfilter.ViewByText=⟨RegEx⟩
logfilter.ThreadByKey=⟨RegEx⟩
logfilter.ThreadByText=⟨RegEx⟩
logfilter.FileByKey=⟨RegEx⟩
logfilter.FileByText=⟨RegEx⟩

; Änderung der Dateierweiterungen für E-Mail und News.
; HINWEISE: Diese Einstellungen sind weitgehend inkompatibel zu älteren Classic-
; Varianten, erst recht zum Hamster »Playground«. Die Umstellung bedingt ein manuelles
; Anpassen aller vorhandenen E-Mail- und News-Dateien!
; Zudem MUSS die Endung für »mail.ext.err« anders sein als für »mail.ext«, da die
; »Err«-Dateien im selben Verzeichnis wie die zu sendenden Mails gespeichert werden!
mail.ext=msg
mail.ext.err=err
news.ext=msg
news.ext.err=msg
```

```
; Bestimmt, ob abgelaufene Filtereinträge gelöscht (1) oder nur auskommentiert (0)
; werden sollen. Letzteres setzt ein »# Expired! # « vor die Originalzeile. In beiden
; Fällen wird ein Eintrag in die Logdatei gemacht. Standard: 0.
Mail.ScoreFile.ExpiredEntries.Delete=0
News.ScoreFile.ExpiredEntries.Delete=0

; Nachfrage beim Beenden, wenn noch Mails oder News zu verschicken sind.
main.AskIfOutboxNotEmpty=1

; Ausschalten einzelner Menüpunkte im Hamster-Menü.
; Die Namen der auszuschaltenden Menüpunkte müssen mit Kommata getrennt werden und
; den internen Menünamen entsprechen, die man am einfachsten durch Nachschauen in der
; »hamster_de.dat«, Sektion »[THamsterMainWindow/Form]«, finden kann: Einfach das »MI.«
; davor wegnehmen.
; Beispiel: Main.HideMenuItems=mnuFileRebuildGlobalLists, mnuFilePurge, mnuFileRebuild
Main.HideMenuItems=

; Legt den für die Passwort-Datei zu verwendenden Schlüssel fest:
;   0 = Anmeldename und PC-Konfiguration werden im Schlüssel verwendet (Standard),
;   1 = Der Anmeldename wird nicht im Algorithmus verwendet,
;   2 = Weder Anmeldename noch PC-Konfiguration werden im Schlüssel verwendet.
; ACHTUNG: Nach Ändern dieser Variable ist ein Neustart und die erneute Eingabe aller
; Passworte erforderlich. Bei Einstellung »2« kann die Passwort-Datei auf jedem Rechner
; verwendet werden.
password.codebase=0

; Maximalzahl enthaltener Postings nach dem Purgen/Aufräumen. Wert gilt als Vorein-
; stellung für alle Gruppen, kann aber für jede Gruppe individuell geändert werden,
; Standardwert: 0.
purge.articles.keepcount=0

; Die Haltezeit in Tagen für Artikel seit dem Pullen. Wert gilt als Voreinstellung für
; alle Gruppen, kann aber für jede Gruppe individuell geändert werden.
; Standardwert: 0.
purge.articles.keepdays=0

; Hier wird festgelegt, ob der Hamster täglich einmal den Datenbestand beim Start
; putzen soll: 0 = nicht putzen (Standard), 1 = putzen.
purge.daily=0

; Datum des letzten automatischen Putzens beim Start,
; Format: YYYYMMDD.
purge.lastdailypurge=

; Anzahl der Tage, bis die Daten in der News-History gelöscht werden.
purge.history.keepdays=7

; Anzahl der Tage, bis die Daten in der Mail-History gelöscht werden.
purge.mhistory.keepdays=14

; Mit diesem Parameter wird die tägliche Löschung/Auskommentierung der abgelaufenen
; Einträge abgeschaltet (0) / eingeschaltet (1), Standard ist 1.
PurgeMailScorefile.daily=1
```

```
; Legt die Anzahl der Zeilen fest, ab der eine Log-Datei beim Aufräumen um die ältes-
; ten Einträge gekürzt wird.
purge.mailinlog.keeplines=0
purge.mailoutlog.keeplines=0
purge.newsoutlog.keeplines=0
purge.rasdiallog.keeplines=0

; Expire in Tagen für Delete-Thread-Einträge im News-Scorefile.
score.DeleteThread.expire=

; Die Sektion im News-Scorefile, in die beim Löschen eines Threads der Eintrag ge-
; schrieben werden soll. Muss mit den eckigen Klammern angegeben werden.
; Standard: »[*]«.
score.DeleteThread.section=

; Der Wert, der verwendet werden soll, wenn im News-Scorefile ein zu löschender Thread
; eingetragen wird. Das »=«-Zeichen kann verwendet werden. Standard: »=-9999«.
score.DeleteThread.value=

; Expire in Tagen für Retrieve-Thread-Einträge im News-Scorefile.
score.RetrieveThread.expire=

; Die Sektion im News-Scorefile, in die beim Nachladen eines Threads der Eintrag ge-
; schrieben werden soll. Muss mit den eckigen Klammern angegeben werden.
; Standard: »[*]«.
score.RetrieveThread.section=

; Der Wert, der verwendet werden soll, wenn im News-Scorefile ein nachzuladender
; Thread eingetragen wird. Das »=«-Zeichen kann verwendet werden. Standard: »=+9999«.
score.RetrieveThread.value=

; Höhe einer hs2-Listbox in Pixeln.
Script.Listbox.Height=

; Timeout für die Übergabe von Skripten einer zweiten Instanz an die erste Instanz.
StartUp.ScriptTimeout=30

; Begrüßungsbildschirm anzeigen, 1=Ja 0=Nein.
startup.splash=1


; Lokale Server: Verschiedenes

; Schaltet Hamsters »User-Agent:«-Header ab.
AddUserAgent=0

; Schaltet Hamsters »X-Posting-Agent:«-Header für News bzw. Mails ab.
; Standard ist ein Wert 0=abgeschaltet für News und 1=eingeschaltet für Mails.
mail.addxhamster=1
news.addxhamster=0

; Schaltet Hamsters »Received:«-Header ab für Mails.
mail.addreceived=0
```

Glossar:
Header

Glossar: FQDN

```
; Hier ist der FQDN für den »Domain-part« (alles hinter dem »@«) der Message-ID ein-
; zutragen, wenn diese vom Hamster generiert werden soll.
FQDN=

; Hier ist der FQDN für »Path:«- und »Received:«-Header einzutragen, wenn diese vom
; Hamster generiert werden sollen. Dieser Eintrag wird dann auch für die Begrüßungs-
; texte des Hamsters verwendet.
FQDN2=

; Einstellung des lokalen Zeichensatzes
local.codepage=windows-1252

; Datenmenge, die in einem Rutsch gesendet werden darf. Standardeinstellung:
local.limit.bytessend=1024

; Anzahl der Fehlversuche, wenn Sendeversuche mehrfach komplett scheitern.
; Standardeinstellung:
local.limit.sendfails=20

; Wartezeit in Millisekunden bis zum nächsten Versuch, wenn Sendeversuch komplett
; scheitert.
local.timewait.sendfails=100

; Legt fest, ob der FQDN für die Message-IDs eine lokale Domain ist.
local.mail.LocalMIDFQDN=1

; Verschleierung des vom Hamster bei Message-IDs benutzten »Local-Parts« (alles vor
; dem »@«). 0=Ausgeschaltet (Standard), 1=Eingeschaltet.
mid.crypt=0

; Standard-Account für den Postmaster.
local.postmaster=admin

; Standard-Account für den Newsmaster.
local.usenet=admin

; SMTP-after-POP3: Zeit in Sekunden, die zwischen Authentifikation am lokalen
; POP3-Server und dem SMTP-Zugriff maximal vergehen darf,
; Standard: 300 s.
local.smtpafterpop3.period=300

; Timeout in Minuten, nach dessen Ablauf ohne Client-Aktivitäten die Verbindung zwi-
; schen Client und Hamster beendet werden soll.
local.timeout.inactivity=60

; Zeit in Millisekunden für Timeout nach »QUIT«-Befehl.
local.timeout.quitdelay=100

; Schaltet den »Followup-To:«-Header für das Mail-To-News-Gate ab.
mail.flupforgate=1
```

Header

POP3
SMTP

Glossar: Header

```
; Schaltet für das News-To-Mail-Gateway die Quelle der E-Mail-Adresse des
; »Envelope-From:«-Headers um:
; 1 = E-Mail-Adresse aus dem Menü »Einstellungen« → »Lokale Server« ⇒ »Allgemeines«
;     verwenden,
; 0 = »From:«-Header des Newsartikels für den »Envelope-From:«-Header verwenden.
mail.GateUseHamsterEnvelope=1

; Zusatz für den »Local-Part« der Message-ID (alles vor dem »@«) zum Kennzeichnen der
; über das News-To-Mail-Gateway ausgehenden Mails.
news2mail.MIDExtension=n2m-g.

; FQDN-Liste für das Entfernen von Message-IDs von lokal eingelieferten Mails. Dies
; erlaubt das Löschen falsch vom Mailreader generierter Message-IDs. Ein ähnlicher
; Mechanismus für News-Postings wird aus Sicherheitsgründen (Dupes) nicht unterstützt.
mail.removemids=fqdn1@mid.example.com,fqdn2@mid.example.net,fqdn3@mid.example.org,...

; Verwenden der originalen Subject-Header für die Header der Info-Mails des Mail-
; filters.
mail.infomail.addheadertosubject=0

; Dieser Wert legt fest, ob E-Mails mit Fehlermeldungen des Hamsters extern versendet
; werden dürfen, oder ob sie immer dem Admin zugestellt werden.
; Standard ist der Wert 1 = »alle E-Mails an den Admin senden«.
SendInfoMailLocalOnly=1

; Anzahl der für E-Mails und News maximal zu verwendenden Tasks.
MaxTasks=10

; DCOM-Server ab- (0) bzw. anschalten (1, Standard).
Startup.OLEServer=1

; Verwenden eines alternativen Namens für den DCOM-Server.
ComServername=Hamster


; Lokaler Newsserver

; Authentifizierung für den IHAVE-Befehl des lokalen Newsservers abschalten.
; 0 schaltet Authentifizierung ab, 1 schaltet Authentifizierung ein (Standard).
; Achtung! Das Abschalten der Authentifizierung ist ein Sicherheitsrisiko!
ihave.auth=1

; Legt die internen Gruppennamen fest, Standard ist für alle Kategorien
; »internal.misc«.
internalgroup.default=⟨Gruppe für alle nicht zugewiesenen Kategorien⟩
internalgroup.cancelnotify=⟨Gruppe für Cancels in lokalen Gruppen⟩
internalgroup.newgroups=⟨Gruppe für Meldungen über neue Gruppen⟩
internalgroup.postok=⟨Gruppe für erfolgreich versendete Artikel⟩
internalgroup.posterrors=⟨Gruppe für fehlerhaft versendete Artikel⟩
internalgroup.pullerrors=⟨Gruppe für Fehlermeldungen beim Laden neuer Artikel⟩
internalgroup.statistics=⟨Gruppe für die tägliche Gruppenstatistik⟩
internalgroup.unknowngroup=⟨Gruppe für Artikel mit unbekannter Gruppe⟩
```

```
; Automatischer Start des lokalen Newsservers bei Programmstart, Standard ist 1 = ja.
local.autostart.nntp=1

; Limit für Zeilenlänge und Textlänge der lokal empfangenen Artikel, siehe hierzu auch
; die FAQ »Hamster und große E-Mails«. Der Wert 0 schaltet das Limit ab.
local.limit.linelen.nntp=1000
local.limit.textsize.nntp=2097152

; Dieser Wert legt fest, ob der NNTP-Befehl »LIST« am lokalen Newsserver eine Anmel-
; dung mit Username und Passwort erfordert.
local.list.require.auth=0

; Hier kann der Hamster so eingestellt werden, dass das Feeden von Artikeln mit unbe-
; kannten Gruppen möglich ist (Wert = 1), diese landen dann in der entsprechenden in-
; ternen Gruppe. Auch Artikel ohne »FROM:-«Header sind dann zugelassen.
local.nntp.feedjunk=0

; Dieser Wert legt fest, ob der Newsmaster bei fehlerhaftem Posten von Usenet-Artikeln
; benachrichtigt werden soll.
local.nntp.infomail=1

; Speicherung von abgewiesenen Postings in lokale Gruppen (Ja=1 [Standard], Nein=0).
local.nntp.infopostfail=1

; Speicherung von erfolgreich abgeschickten Postings in lokale Gruppen
; (Ja=1 [Standard], Nein=0).
local.nntp.infopostok=1

; Hier kann die Bindung des lokalen Newsservers an das lokale Netzwerkinterface ein-
; gestellt werden. Der Standardwert ist die IP-Adresse 127.0.0.1.
; Falls der lokale Server in einem lokalen Netz angeboten werden soll, muss dieser
; Wert entweder auf die IP-Adresse der Rechner-Netzwerkkarte oder auf die IP-Adresse
; 0.0.0.0 (meint alle Interfaces des Rechners) geändert werden.
; ACHTUNG, diese Einstellung ist sicherheitsrelevant!
local.nntp.serverbind=127.0.0.1

; Timeout in Millisekunden, bis der User sich nach dem Aufbau der Verbindung am loka-
; len Newsserver angemeldet haben muss, Standard: 10000.
local.nntp.timeout.login=10000

; Lokalen Newsserver mit SSL/TLS benutzen:
;   0: kein SSL verwenden (Standard),
;   1: SSL verwenden, falls Client dies wünscht,
;   2: SSL erzwingen: Login ohne SSL ist nicht möglich.
local.nntp.tlsmode=0

; Einstellung, wie der »XOver«-Befehl auf Zeilenwechsel in Headern reagieren soll.
; »0« wandelt in Leerzeichen, »1« (Standard) löscht CR/LF.
local.nntp.XOVERdelCRLF=1

; TCP-Port des lokalen Newsservers.
local.port.nntp=119
```

Glossar: NNTP

Header

SSL TLS

TCP

```
; Limit für die Anzahl der Clienten am lokalen Newsserver. Wert 0 schaltet das Limit
; ganz aus.
MaxLocalNNTPServers=12

; Limit für Verbindungen von einem Rechner zum lokalen Newsserver. Wert 0 schaltet das
; Limit ganz aus.
MaxLocalNNTPServersPerIP=6

; Fügt den FQDN, falls vorhanden, dem »Path:«-Header beim Posten eines Artikels am lo-
; kalen Newsserver des Hamsters zu.
news.addpath=1

; Legt den Namen für den X-Trace-Header fest, welcher den lokal eingelieferten Arti-
; keln hinzugefügt wird; Beispiel: »news.addxhtrace=X-Hamster-Trace« ergibt
; »X-Hamster-Trace: x.y.z 963151230 236 127.0.0.1 (1 ec 2000.07.09 14:00:30)«.
news.addxhtrace=

; Generieren des Date-Headers für einen Artikel beim Einliefern an den lokalen News-
; server, falls er nicht schon durch den Newsreader gesetzt wurde.
news.GenerateDate=0

; Cancel-/Supersedes-Nachrichten ausführen, welche von außen per Feed hereinkommen.
news.feededcancel=0
news.feededsupersedes=0

; From-/Sender-Header der Cancel-/Supersedes-Nachrichten mit dem Original vergleichen.
news.verifycancel=1
news.feededcancelverify=1

; Artikel auch löschen, wenn das Original eine Control-Nachricht ist.
; VORSICHT: Dadurch können »Beweise« vernichtet werden.
news.cancelcontrolmsg=0
news.feededcancelcontrolmsg=0

; Aktiviert die direkte Speicherung von am lokalen Server geposteten Artikeln auch für
; nichtlokale Gruppen.
news.localinjection=0


; Lokaler IMAP-Server

; Ersetzen von 8-Bit-Zeichen durch Fragezeichen in Envelope-Antworten des lokalen
; IMAP-Servers, um auch fehlerhafte E-Mails mit Readern lesen zu können, welche diese
; E-Mails andernfalls ignorieren würden (z. B. Outlook Express), 1 = Zeichen ersetzen.
imap.7bit=0

; Hier kann man zur Umgehung eines Outlook-XP-Fehlers eine Verzögerung der vom lokalen
; IMAP-Server gesendeten Antworten in Millisekunden einstellen.
imap.delay=0

; »LOGIN«-Befehl des lokalen IMAP-Servers abschalten.
imap.LOGINDisable=1
```

Glossar: FQDN, Header, IMAP

Glossar:
SASL
IMAP
SSL
TLS

```
; Den SASL-Mechanismus »AUTH=LOGIN« des IMAP-Servers deaktivieren. Dieser ist ohne
; SSL/TLS unsicher. Sollte/Darf nur aktiviert werden, wenn SSL erzwungen wird.
imap.SASL.LOGINDisable=1

; Workaround für einen IMAP-Protokoll-Bug in den Netscape-Messenger-4.x-Versionen.
imap.NcBrain=0

; Automatischer Start des lokalen IMAP-Servers bei Programmstart,
; Standard ist 0 = nein.
local.autostart.imap=0

; »ID«-Befehl für den lokalen IMAP-Server abschalten.
local.imap.id=1

; Hinweis: »local.imap.sasl« existiert derzeit nicht.

; Hier kann die Bindung des lokalen IMAP-Servers an das lokale Netzwerkinterface ein-
; gestellt werden. Der Standardwert ist die IP-Adresse 127.0.0.1.
; Falls der lokale Server in einem lokalen Netz angeboten werden soll, muss dieser
; Wert entweder auf die IP-Adresse der Rechner-Netzwerkkarte oder auf die IP-Adresse
; 0.0.0.0 (meint alle Interfaces des Rechners) geändert werden.
; ACHTUNG, diese Einstellung ist sicherheitsrelevant!
local.imap.serverbind=127.0.0.1

; Timeout in Millisekunden, bis der User sich nach dem Aufbau der Verbindung
; am lokalen IMAP-Server angemeldet haben muss, Standard: 10000.
local.imap.timeout.login=10000

; Lokalen IMAP-Server mit SSL/TLS benutzen:
;   0: kein SSL verwenden (Standard),
;   1: SSL verwenden, falls Client dies wünscht,
;   2: SSL erzwingen: Login ohne SSL ist nicht möglich.
local.imap.tlsmode=0

; Limit für die Textlänge der lokal empfangenen Artikel (für IMAP gibt es kein Zeilen-
; längenlimit), siehe hierzu auch die FAQ »Hamster und große E-Mails«. Der Wert 0
; schaltet das Limit ab.
local.limit.textsize.imap=2097152

; TCP-Port des lokalen IMAP-Servers.
local.port.imap=143

; Limit für die Anzahl der Clienten am lokalen IMAP-Server. Wert 0 schaltet das Limit
; ganz aus.
MaxLocalIMAPServers=10

; Limit für Verbindungen von einem Rechner zum lokalen IMAP-Server. Wert 0 schaltet
; das Limit ganz aus.
MaxLocalIMAPServersPerIP=10
```

TCP

Glossar: POP3

```
; Lokaler POP3-Server

; Automatischer Start des lokalen POP3-Servers bei Programmstart, Standard ist 1 = ja.
local.autostart.pop3=1

; Limit für Zeilenlänge und Textlänge der lokal empfangenen Artikel, siehe hierzu auch
; die FAQ »Hamster und große E-Mails«. Der Wert 0 schaltet das Limit ab.
local.limit.linelen.pop3=1000
local.limit.textsize.pop3=2097152

; Legt den zu nutzenden SASL-Mechanismus (RFC 2222) für den lokalen POP3-Server fest.
; Es werden die Mechanismen LOGIN, PLAIN, CRAM-MD5, CRAM-SHA1 unterstützt.
local.pop3.sasl=LOGIN PLAIN CRAM-MD5 CRAM-SHA1

; Hier kann die Bindung des lokalen POP3-Servers an das lokale Netzwerkinterface ein-
; gestellt werden. Der Standardwert ist die IP-Adresse 127.0.0.1.
; Falls der lokale Server in einem lokalen Netz angeboten werden soll, muss dieser
; Wert entweder auf die IP-Adresse der Rechner-Netzwerkkarte oder auf die IP-Adresse
; 0.0.0.0 (meint alle Interfaces des Rechners) geändert werden.
; ACHTUNG, diese Einstellung ist sicherheitsrelevant!
local.pop3.serverbind=127.0.0.1

; Timeout in Millisekunden, bis der User sich nach dem Aufbau der Verbindung am loka-
; len POP3-Server angemeldet haben muss, Standard: 10000.
local.pop3.timeout.login=10000

; Lokalen POP3-Server mit SSL/TLS benutzen:
;   0: kein SSL verwenden (Standard),
;   1: SSL verwenden, falls Client dies wünscht,
;   2: SSL erzwingen: Login ohne SSL ist nicht möglich.
local.pop3.tlsmode=0

; TCP-Port des lokalen POP3-Servers.
local.port.pop3=110

; Limit für die Anzahl der Clienten am lokalen POP3-Server. Wert 0 schaltet das Limit
; ganz aus.
MaxLocalPOP3Servers=10

; Limit für Verbindungen von einem Rechner zum lokalen POP3-Server. Wert 0 schaltet
; das Limit ganz aus.
MaxLocalPOP3ServersPerIP=5

; Hier kann man zur Umgehung eines Outlook-XP-Fehlers eine Verzögerung der vom lokalen
; POP3-Server gesendeten Antworten in Millisekunden einstellen.
pop3.delay=0


; Lokaler SMTP-Server

; Automatischer Start des lokalen SMTP-Servers bei Programmstart, Standard ist 1 = ja.
local.autostart.smtp=1
```

SASL

SSL
TLS

TCP

SMTP

```
; Limit für Zeilenlänge und Textlänge der lokal empfangenen Artikel, siehe hierzu auch
; die FAQ »Hamster und große E-Mails«. Der Wert 0 schaltet das Limit ab.
local.limit.linelen.smtp=1000
local.limit.textsize.smtp=2097152

; Bounce senden, wenn Mails für unbekannte User am lokalen SMTP-Server eintreffen.
local.mail.BounceIfUnknownUser=0

; Nicht autorisierten Usern am lokalen SMTP-Server trotz eingeschaltetem SMTP-AUTH
; das Einliefern von Mails für lokale Empfänger gestatten.
local.mail.reqnotauth=0

; TCP-Ports des lokalen SMTP-Servers.
local.port.smtp=25

; Legt den zu nutzenden SASL-Mechanismus (RFC 2222) für den lokalen SMTP-Server fest.
; Es werden die Mechanismen LOGIN, PLAIN, CRAM-MD5, CRAM-SHA1 unterstützt.
local.smtp.sasl=LOGIN PLAIN CRAM-MD5 CRAM-SHA1

; Hier kann die Bindung des lokalen SMTP-Servers an das lokale Netzwerkinterface ein-
; gestellt werden. Der Standardwert ist die IP-Adresse 127.0.0.1.
; Falls der lokale Server in einem lokalen Netz angeboten werden soll, muss dieser
; Wert entweder auf die IP-Adresse der Rechner-Netzwerkkarte oder auf die IP-Adresse
; 0.0.0.0 (meint alle Interfaces des Rechners) geändert werden.
; ACHTUNG, diese Einstellung ist sicherheitsrelevant!
local.smtp.serverbind=127.0.0.1

; Timeout in Millisekunden, bis der User sich nach dem Aufbau der Verbindung am loka-
; len SMTP-Server angemeldet haben muss, Standard: 10000.
local.smtp.timeout.login=10000

; Lokalen SMTP-Server mit SSL/TLS benutzen:
;   0: kein SSL verwenden (Standard),
;   1: SSL verwenden, falls Client dies wünscht,
;   2: SSL erzwingen: Login ohne SSL ist nicht möglich.
local.smtp.tlsmode=0

; Abschalten der Prüfung auf korrekten »!MAIL FROM:«-/»!RCPT TO:«-Header beim Empfan-
; gen von E-Mails:
;   1 = Die spitzen Klammern um den Absender/Empfänger im »!MAIL FROM:«/»!RCPT TO:«
;       werden benötigt (Standardeinstellung),
;   0 = »!MAIL FROM:« bzw. »!RCPT TO:« wird auch ohne die spitzen Klammern toleriert.
mail.smtpneedbrackets=1

; Limit für die Anzahl der Clienten am lokalen SMTP-Server. Wert 0 schaltet das Limit
; ganz aus.
MaxLocalSMTPServers=3

; Limit für Verbindungen von einem Rechner zum lokalen SMTP-Server. Wert 0 schaltet
; das Limit ganz aus.
MaxLocalSMTPServersPerIP=3
```

Glossar: SMTP

SMTP-AUTH

TCP

SASL

SSL
TLS

Header

Glossar: SMTP

```
; Hier kann man zur Umgehung eines Outlook-XP-Fehlers eine Verzögerung der vom loka-
; len SMTP-Server gesendeten Antworten in Millisekunden einstellen.
smtp.delay=0


; Lokaler ReCo-Server

; Automatischer Start des ReCo-Servers bei Programmstart, Standard ist 0 = nein.
local.autostart.reco=0

; TCP-Port des lokalen ReCo-Servers.
local.port.reco=23

; Legt den zu nutzenden SASL-Mechanismus (RFC 2222) für den ReCo-Server fest. Es wer-
; den die Mechanismen LOGIN, PLAIN, CRAM-MD5, CRAM-SHA1 unterstützt.
local.reco.sasl=LOGIN PLAIN CRAM-MD5 CRAM-SHA1

; Hier kann die Bindung des lokalen ReCo-Servers an das lokale Netzwerkinterface ein-
; gestellt werden. Der Standardwert ist die IP-Adresse 127.0.0.1.
; Falls der lokale Server in einem lokalen Netz angeboten werden soll, muss dieser
; Wert entweder auf die IP-Adresse der Rechner-Netzwerkkarte oder auf die IP-Adresse
; 0.0.0.0 (meint alle Interfaces des Rechners) geändert werden.
; ACHTUNG, diese Einstellung ist sicherheitsrelevant!
local.reco.serverbind=127.0.0.1

; Timeout in Millisekunden, bis der User sich nach dem Aufbau der Verbindung am loka-
; len ReCo-Server angemeldet haben muss, Standard: 10000.
local.reco.timeout.login=10000

; ReCo-Server mit SSL-Verschlüsselung starten: 0 = nein (Standard), 1 = ja.
local.reco.usetls=0

; Limit für die Anzahl der Clienten am lokalen ReCo-Server. Wert 0 schaltet das Limit
; ganz aus.
MaxLocalRECOServers=1

; Limit für Verbindungen von einem Rechner zum lokalen ReCo-Server. Wert 0 schaltet
; das Limit ganz aus.
MaxLocalRECOServersPerIP=1


; Externe Server: Verschiedenes

; Timeout für die Remote-Server, d. h. Zeitspanne in Sekunden, nach der Verbindungs-
; und Abfrageversuche zu externen Servern als gescheitert betrachten werden,
; Standard: 120 s.
Remote.Timeout.Connect=120
Remote.Timeout.Command=120
```

```
; Externe Newsserver

; Benachrichtigungen über neue Gruppen per RegExp beschränken. Eine Beschränkung auf
; Newgroups für die Hierarchien »de.ALL« und »hamster.ALL« sähe so aus:
; »FilterNewGroupsInfo=^(de|hamster)\.«.
; Siehe auch die FAQ »Hamster und Reguläre Ausdrücke«.
FilterNewGroupsInfo=.*

; Mit »MaxUnknownGroupsInRe« kann das Limit von max. 2 unbekannten Gruppen in einem
; Posting ohne Follow-up-To für Antworten etwas höher gesetzt werden. Wird der Wert
; zu hoch gesetzt (>10), wird der Standardwert 2 verwendet.
MaxUnknownGroupsInRe=2

; Dieser Eintrag gibt an, ob Hamster den Overview in mehrere Teile spalten soll, oder
; ob er alles gleichzeitig lädt. Dieser Eintrag ist sinnvoll, wenn man größere NGs
; komplett lädt (>2000 Artikel pro Ladegang) oder eine teure und langsame Verbindung
; hat.
news.makeparts=0

; Dieser Eintrag gibt die kleinste Overview-Größe an, die in Teilen geladen wird. Er
; ist nur wirksam, wenn »news.makeparts« auf 1 gesetzt wurde.
news.sizeofparts.max=700

; Dieser Eintrag gibt die Standardgröße für Teile des Overview an. Er ist nur wirksam,
; wenn »news.makeparts« auf 1 gesetzt wurde.
news.sizeofparts.min=500

; Schaltet das automatische Einstellen von Jobs zum Überprüfen des »Active« bei der
; Ausführung der »hs2«-Befehle »HamNewsPost« und »HamNewsPull« ab.
nntp.autogetserverinfos=0

; Abschalten des automatischen Verwerfens (Wert=1) von Newsjobs nach Scheitern aller
; laufenden Threads für einen externen Newsserver.
nntp.dropresidualjobs=0

; Abschalten des Befehls »MODE READER« des NNTP-Clienten. 1 schaltet den Befehl ein
; (Standard), 0 schaltet den Befehl ab.
nntp.ModeReader=1

; Externe POP3-Server

; Globaler Filter für das Laden von E-Mails auf Grundlage der Antwort des »List«-
; Befehls vom Provider: Ist die E-Mail größer als der angegebene Wert (in Bytes), wird
; sie entsprechend der Kategorie behandelt.
; Als Kategorie stehen »ignore« = Ignorieren, »kill« = Löschen und »notify« = Info an
; den »admin« zur Verfügung. Der Wert »0« schaltet den Filter ab.
; Diese Einstellungen haben Vorrang vor der Mail-Score-Datei »MailFilt.hst«.
mail.filter.mailsize.ignore=0
mail.filter.mailsize.delete=0
mail.filter.mailsize.notify=0

; Anzahl der für den Mailfilter als Vorschau zu ladenden Body-Zeilen.
mail.filter.toplines=20
```

Glossar: NNTP

POP3

Glossar:
SMTP

```
; Externe SMTP-Server

; Anzahl der Versuche für den Mailversand an die eingestellten externen SMTP-Server.
sendmail.attempts.max=10

; Aktion des Hamsters, falls die Mail nach der eingestellten Anzahl von Versuchen
; immer noch nicht verschickt werden konnte: 0 = Umbenennen, 1 = Löschen.
sendmail.attempts.del=0
```

Sektion [Main]

```
[Main]
; Die Sektion Main legt das Layout der Hamster-Oberfläche fest.

; Erlaubt das Einstellen der Farben für die Logfile-Darstellung im Hauptfenster.
; ⟨xxx⟩ steht für »debug«, »detail«, »info«, »system«, »warning« oder »error«, und
; ⟨yyy⟩ für »text« (Vordergrundfarbe) oder »brush« (Hintergrundfarbe). Zudem wird
; »color.else.brush« für die leeren Protokollzeilen verwendet. Als Farbwerte sind die
; Delphi-Konstanten (»clBlack«, »clWhite«, »clBlue«, »clWindowText«, ...) oder Hex-
; Werte (in der Form $BBGGRR von »$000000« bis »$FFFFFF«) erlaubt. Bei fehlerhaften
; Angaben werden die Windows-Standardwerte verwendet.
color.⟨xxx⟩.⟨yyy⟩=

; Tab für das Fenster »Warnungen/Fehler«, 0 = kein farbiger Tab.
ColoredTabs=1

; Die konkreten Farben für das Fenster »Warnungen/Fehler«, sofern die Vorgabe nicht
; gefällt (Delphi-Konstanten, wie »clBlack«, »clWhite«, oder Hex-Werte im Format
; $BBGGRR):
TabColorErrorText=clWhite;
TabColorErrorBrush=clRed;
TabColorWarningText=clBlack;
TabColorWarningBrush=clYellow;

; Ein Wert=0 schaltet die Hints beim Überfahren mit der Maus ab.
Logs.ShowHints=1

; Legt fest, ob eine gesonderte Eingabezeile für hs2 im Hauptfenster des Hamsters an-
; gezeigt werden soll.
; 1=Fenster oberhalb des Hauptfensters anzeigen.
; 2=Fenster unterhalb des Hauptfenster anzeigen.
ShowShell=0

; Legt fest, ob für die hs2-Eingabezeile im Hauptfenster des Hamsters die letzten
; Eingaben (max. 100) in einer Datei »ShellHis.log« gespeichert werden;
; 0 = Deaktivieren, 1 = Speichern (Standard).
ShellHist.Log=1

; Taskbareintrag konfigurieren. Folgende Platzhalter sind zulässig: »%Hamster%« für
; Produktname, »%Version%« für Versionsnummer und »%Readonly%« für Nur-Lese-Modus.
; Standard bei Nichtangabe: »%Hamster% V%Version% %Readonly%«.
TaskbarTitle=Hamster
```

```
; Hauptfenstertitel konfigurieren. Folgende Platzhalter sind zulässig: »%Hamster%« für
; Produktname, »%Version%« für Versionsnummer und »%Readonly%« für Nur-Lese-Modus.
; Standard bei Nichtangabe: »%Hamster% V%Version% %Readonly%«.
Title=

; Festlegen der Statusinformation für die Statuszeile und Hinweiszeile des Tray-Icons:
;   %TA = Anzahl aktiver Aufträge,
;   %JO = Anzahl aller noch offenen Aufträge,
;   %NO = Anzahl Newsbeiträge zum Versenden (News.Out),
;   %MO = Anzahl Mails zum Versenden (Mail.Out),
;   %NI = Anzahl neuer Artikel von externen Newsservern (%NI=%NL+%NH+%NK+%NM),
;   %NM = Anzahl der per Message-ID nachgeladenen Artikel,
;   %NL = Anzahl davon geladener Artikel,
;   %NH = Anzahl Artikel, die bereits in der History vorhanden waren,
;   %NK = Anzahl durch Einträge im Kill-/Score-File nicht geladener Artikel,
;   %MI = Anzahl geladener Mails,
;   %BY = übertragene Bytes,
;   %KB = übertragene Kibibytes,
;   %US = Sekunden-Anteil der Hamster-Uptime,
;   %UM = Minuten-Anteil der Hamster-Uptime,
;   %UH = Stunden-Anteil der Hamster-Uptime
;   %UD = Tage-Anteil der Hamster-Uptime,
;   %UT = zeigt die Uptime in der Form Stunden:Minuten:Sekunden an,
;   %M:⟨Username⟩% = Mails im Postfach des Users ⟨Username⟩.
; Individuelle Counter »%X0« bis »%X9« sind in der Statusleiste integrierbar und in
; hs2 und DCOM/COM-Schnittstelle per »XCounter(⟨Nr⟩)« abfragbar, per
; »ClearXCounter(⟨Nr⟩ [, ⟨BisNr⟩])« löschbar, per »SetXCounter(⟨Nr⟩, ⟨Wert⟩)« ein-
; stellbar, per »IncXCounter(⟨Nr⟩ [, ⟨Wert⟩])« oder »DecXCounter(⟨Nr⟩ [, ⟨Wert⟩])«
; veränderbar.
BottomlineFormat=Tasks/Jobs=%TA/%JO   OutBox: N/M=%NO/%MO   News=%NI
HintFormat=Tasks/Jobs=%TA/%JO  N/M=%NO/%MO  Arts=%NI  Mails=%MI Admin=%M:admin%
```

Sektion [Actions]

```
[Actions]
; Unter Actions werden im Hamster die Möglichkeiten zusammengefasst, bei bestimmten
; Aktionen automatisch ein externes Programm und/oder ein Hamster-Skript ausführen zu
; lassen.

; Zuerst ein praktisches Beispiel:

; news.out.exec=Korrnews.exe
; news.out.pars=Type:news Filename:"%1"

; Dieser Eintrag (inklusive einiger mehr) wird bei Bedarf von »Korrnews« automatisch
; angelegt und hat folgende Aufgabe: Wenn man über seinen Newsreader ein neues Pos-
; ting an den Hamster schickt, speichert er es bis zum Abschicken als Datei im
; »News.out«-Verzeichnis. Obige Aktion sorgt dafür, dass er anschließend das Programm
; »Korrnews.exe« aufruft und als Parameter u. a. auch den Namen der Datei mit dem Pos-
; ting übergibt, so dass Korrnews gezielt dieses Posting nachbearbeiten kann.
```

```
; Folgende Grundtypen stehen für Aktionen zur Auswahl:
;---------------
; Mail:
;   mail.in.* für eingehende Mails,
;   mail.out.* für ausgehende Mails,
;   mail.getheader für das Filtern von Mails während des Preloads mittels Skripten,
;     bevor sie durch den eigentlichen Mailfilter bearbeitet werden,
;   mail.internal.* für vom Hamster generierte Mails,
;   mail.local.* für Mails an andere lokale Benutzer.
; Über mail.* kann man eine Aktion für alle Mailarten gleichzeitig vorsehen, existiert
; eine speziellere Aktion, hat diese Vorrang. Speziell für eingehende Mails gibt es
; noch eine weitere account-spezifische Variante, die weiter unten erläutert wird.
;---------------
; News:
;   news.out.* für ausgehende Postings,
;   news.local.* für Postings, die in lokale Newsgruppen geschickt werden.
;     Im Falle eines Crosspostings in eine lokale und eine externe Newsgruppe würde
;     die lokale Variante über den »news.local.*«-Teil behandelt und die Variante für
;     den »News.Out«-Ordner über den »news.out.*«-Teil.
;   news.in.nntp für alle per NNTP hereinkommenden News-Artikel.
;     ACHTUNG, die Nutzung dieser Action kann das Laden der Artikel deutlich verlang-
;     samen.
; Mittels news.* ist auch direkt eine gemeinsame Behandlung aller Postings definier-
; bar, wie bei Mails hat die spezifische Variante allerdings Vorrang.
;---------------
; DUN (DFÜ):
; Für Ereignisse, welche der Hamster beim DFÜ-Netzwerk auslöst, existieren die
; Action-Typen DUN.BeforeDial, DUN.DialFailed, DUN.Connected und DUN.HangUp.
;---------------
; Sonstiges:
;   group.doubleclick.* erlaubt es, bei einem Doppelklick auf eine Newsgruppe im
;     News-Konfigurationsdialog spezifisch zu reagieren, als Parameter wird der
;     Gruppenname übergeben.
;   trayicon.click.*/trayicon.doubleclick.*/trayicon.middleclick.*/
;   trayicon.middledoubleclick.*: Erlaubt spezifische Aktionen beim Einfach-
;     oder Doppelklick mit der linken oder mittleren Maustaste auf das Trayicon.
;   SearchMID kann als Reaktion auf einen nicht per Message-ID gefundenen Artikel
;     genutzt werden.
;   atEveryHour mit den Parametern Stunde und Tag der Woche startet stündlich,
;     wobei das Ende aktiver Threads abgewartet wird.

; Aufbau einer Aktion:
;   *.exec: Dateiname (ggf. inkl. Pfad) der aufzurufenden Anwendung.
;   *.pars: Parameter für die aufzurufende Anwendung, »%1« erlaubt die Übergabe des
;     Dateinamens (bzw. des Gruppennamens) und »%0« die Übergabe des Aktionstyps.
;   *.script: Aufruf eines hs2-Skripts. Sollten sowohl Anwendung als auch Skript
;     angegeben werden, werden beide ausgeführt -- die Anwendung wird dabei zuerst
;     gestartet. Ein aufgerufenes Skript bekommt automatisch zwei Parameter:
;     »ParamStr(1)« enthält den Aktionstyp als String (entspricht »%0« bei »*.pars«),
```

Glossar: NNTP

```
;     »ParamStr(2)« den Dateinamen bzw. Gruppennamen (entspricht »%1« bei »*.pars«).
;   *.ScriptLines=⟨hs2-Zeile⟩: ermöglicht das direkte Ausführen von hs2-Befehlen
;     durch eine Action. Sollte besser über den Menüdialog
;     »Einstellungen« → »Automatische Abläufe« ⇒ »Aktionen« geändert werden.
;   *.wait: Warteverhalten. Es gibt vier Varianten: 0 (Standard) wartet gar nicht,
;     1 wartet erst den Aufruf der Anwendung ab (sofern Anwendung definiert), dann
;     den Ablauf des Skripts (sofern Skript definiert) und setzt erst dann die weitere
;     interne Ausführung im Hamster fort, 2 wartet nur auf die Anwendung und 3 nur auf
;     das Skript. Im Falle von »news.local.*« wird »wait« automatisch auf 1 gesetzt.
;   *.ExeStartType: Beim Aufruf eines externen Programmes kann dieses versteckt im
;     Hintergrund aufgerufen werden, Einstellung mittels Menüdialog
;     »Einstellungen« → »Automatische Abläufe« ⇒ »Aktionen« oder hier in der
;    INI-Datei per »⟨Actionname⟩.ExeStartType=0/1« (normal, versteckt), Standard 0.

; Account-spezifische Aktionen:
; Um vor allem bei eingehenden Mails eine höhere Flexibilität zu erhalten, sind ac-
; count-spezifische Mail-Aktionen möglich. Der Aufbau ist dabei »Typ.⟨Accountname⟩.*«,
; d. h. statt »mail.in.exec« und »mail.in.pars« wird für die Behandlung des Accounts
; »test« »mail.in.test.exec« und »mail.in.test.pars« verwendet. Auch auf Account-Ebene
; ist eine allgemeine Mailbehandlung möglich: »mail.test.*« behandelt ggf.
; »mail.in.test.*«, »mail.local.test.*« und »mail.internal.test.*«.
; Wenn weder ein account-spezifischer spezieller Maileintrag existiert noch ein ac-
; count-spezifischer allgemeiner Maileintrag, wird auf die normalen Einträge zurückge-
; griffen.

news.out.exec=
news.out.pars=
news.out.script=
news.out.wait=
news.in.nntp.exec=
news.in.nntp.pars=
news.in.nntp.script=
news.in.nntp.wait=
news.preprocess.exec=
news.preprocess.pars=
news.preprocess.script=
news.preprocess.wait=
news.local.exec=
news.local.pars=
news.local.script=
news.local.wait=
news.gateway.exec=
news.gateway.pars=
news.gateway.script=
news.gateway.wait=
mail.out.exec=
mail.out.pars=
mail.out.script=
mail.out.wait=
mail.local.exec=
```

```
mail.local.pars=
mail.local.script=
mail.local.wait=
mail.in.exec=
mail.in.pars=
mail.in.script=
mail.in.wait=
mail.internal.exec=
mail.internal.pars=
mail.internal.script=
mail.internal.wait=
mail.getheader.exec=
mail.getheader.pars=
mail.getheader.script=
mail.getheader.wait=
group.doubleclick.exec=
group.doubleclick.pars=
group.doubleclick.script=
group.doubleclick.wait=
trayicon.doubleclick.exec=
trayicon.doubleclick.pars=
trayicon.doubleclick.script=
trayicon.doubleclick.wait=
trayicon.click.exec=
trayicon.click.pars=
trayicon.click.script=
trayicon.click.wait=
trayicon.middledoubleclick.exec=
trayicon.middledoubleclick.pars=
trayicon.middledoubleclick.script=
trayicon.middledoubleclick.wait=
trayicon.middleclick.exec=
trayicon.middleclick.pars=
trayicon.middleclick.script=
trayicon.middleclick.wait=
startup.exec=
startup.pars=
startup.script=
startup.wait=
shutdown.exec=
shutdown.pars=
shutdown.script=
shutdown.wait=
status.doubleclick.exec=
status.doubleclick.pars=
status.doubleclick.script=
status.doubleclick.wait=
status.click.exec=
status.click.pars=
status.click.script=
```

```
status.click.wait=
dun.beforedial.exec=
dun.beforedial.pars=
dun.beforedial.script=
dun.beforedial.wait=
dun.dialfailed.exec=
dun.dialfailed.pars=
dun.dialfailed.script=
dun.dialfailed.wait=
dun.connected.exec=
dun.connected.pars=
dun.connected.script=
dun.connected.wait=
dun.hangup.exec=
dun.hangup.pars=
dun.hangup.script=
dun.hangup.wait=
searchmid.exec=
searchmid.pars=
searchmid.script=
searchmid.wait=
ateveryhour.exec=
ateveryhour.pars=
ateveryhour.script=
ateveryhour.wait=

; Timeout für externe Programme, welche per »Actions« aufgerufen werden.
Timeout.ExecInit=60
Timeout.ExecRun=180
```

Sonstige Sektionen

```
[DlgGoupSelect]
; In diesem Bereich speichert Hamster die zuletzt benutzten Einstellungen des Auswahl-
; fensters für das Abonnieren von Newsgruppen unter dem Menüpunkt News/Ladeaufträge.


[EditDirs]
; Hier werden die Einstellungen des Browsers für Mails und News (Menü
; »Lokale Server« →  »News-/Mail-Verzeichnisse«) abgespeichert.

; Hier kann ausgewählt werden, welche Dateien im Vorschaufenster angezeigt werden
;  sollen:  0 zeigt nur »*.msg«, mit einigen Headern (Standard),
;           1 zeigt nur »*.err«, mit einigen Headern,
;           2 zeigt alle Dateien, aber ohne Header.
ShowFileTypes=

; Hier kann ausgewählt werden, nach welchen Spalten in der Übersicht sortiert werden
; soll. Die Werte bedeuten bei »news.*«: 0=Filename, 1=Subject, 2=Score, etc.,
; bei »mail.*« abweichend 0=Filename, 1=Newsgroup, 2=Subject, etc.
news.DefColumnSort=0
mail.DefColumnSort=0
```

Glossar: Header

```
; Hier kann ausgewählt werden, ob auf- oder absteigend sortiert werden soll,
; 0 = absteigend, 1 = aufsteigend.
news.DefColumnSortUp=1
mail.DefColumnSortUp=1

; Hier werden die Spaltenbreiten gespeichert (in der INI-Datei eine Zeile).
ColWidths=Default=95 Newsgroups=140 Subject=274 Date=201 ...
          ... KB=57 Lines=40 To=139 From=152 Score=68

; Legt fest, ob eine Vorschau des Artikels gezeigt werden soll.
ShowDirectly=1

; Hier können die Header ausgewählt werden, welche im Vorschaufenster
; des Browsers für Mails und News angezeigt werden sollen.
mail.ShowHeaders=
; z. B. mail.ShowHeaders=Subject, From, To, Date
news.ShowHeaders=
; z. B. news.ShowHeaders=Subject, From, Newsgroups, Followup-To, Message-ID, Date

; Anzeige aller Header (Standard = 0).
Preview_AllHeader=1

; Legt fest, wie viele Zeilen eines Artikels maximal angezeigt werden.
Preview_MaxLines=300

; Legt fest, ob die Textzeilen eines Artikels automatisch umbrochen werden.
Preview_Wordwrap=1

; Vorschau in Festbreitenschrift (Standard = 0)
Preview_FixedPitchFont=1

; Textattribute in der Vorschau:
;   0 = Anzeige als normaler Text,
;   1 = Formatieren, aber Originalzeichen beibehalten,
;   2 = Formatieren und Originalzeichen unterdrücken (Standard).
Preview_TextAttr=

; Update-Intervall der Anzeige im Verzeichnis-Browser in Millisekunden.
UpdateIntervalms=1000

; Darstellung der Verzeichnisse per Menü zwischen großen und kleinen Symbole umstel-
; len, Standard = 0 (große Symbole).
UseSmallIcons=0

; Legt die anzuzeigenden Postfächer und deren Reihenfolge im Browser fest. Bei Stan-
; dardeinstellung (keine Angabe) werden alle Postfächer angezeigt. Vgl. »NoMailboxes«
Mailboxes=

; Erlaubt als Gegenstück zu »Mailboxes« eine gezielte Unterdrückung von Postfächern in
; der Anzeige.
NoMailboxes=
```

Glossar: Header

Glossar:
IMAP

```
; Entscheidet, ob bei IMAP-Ordnern alles oder nur das Hauptverzeichnis angezeigt wer-
; den soll, Standard ist 1 = alles anzeigen.
ShowIMAPSubDirs=1

; Erlaubt das Abschalten des Mülleimers, Standard ist 1
; ACHTUNG! Artikel, die in den Mülleimer gezogen werden, werden nicht endgültig ge-
; löscht, sondern nur bis zur endgültigen Löschung zwischengelagert.
ShowTrash=1

; Löschen von Dateien im Mülleimer nach wieviel Tagen.
EmptyTrashAfterXDays=

; Zuletzt benutzte relative Breite des Vorschaufensters.
AspectListToPreview=

; Zuletzt angezeigtes Verzeichnis,
; z. B. LastDir=e:\hamster\mails\mail.out\
LastDir=

; Eigener Taskbareintrag für Verzeichnis-Dialog.
OwnTaskbarEntry=1


[EditScripts]
; Hier werden die Einstellungen des Dialogs für Skripte und Module (Menü
; »Skript« →  »Skripte und Module verwalten«) abgespeichert.

; Eigener Taskbareintrag für Skript-Bearbeitungs-Dialog.
OwnTaskbarEntry=1


[KillList]
; In diesem Bereich merkt sich Hamster Größe und Position des Fensters für das
; Kill-Log-Fenster, die Lage der Spalten und die Sortierung der Einträge.


[MailKillList]
; In diesem Bereich merkt sich Hamster Größe und Position des Fensters für das
; Kill-Log-Fenster, die Lage der Spalten und die Sortierung der Einträge.


[Marker]
; Hier merkt sich Hamster, ob die globalen Listen für die Newsgruppen erneuert werden
; müssen, falls neue Gruppen dazugekommen sind. Der Eintrag dort muss nicht editiert
; werden, da Hamster das selbst macht und nur zum Merken benutzt.

GlobalList=0
```

```
[Online-Menu]
; Hier stehen die Einträge, die im Menüpunkt »Online« → »1-9 ...« abrufbar sind.

Title1=News abholen
Server1=news.example.com,nntp;nntps.example.org,563
Title2=Mail Hauptpostfach abholen
Server2=pop3s.example.org,995


[ScoreEdit]
; In diesem Bereich merkt sich Hamster Größe und Position des Fensters für das
; Editieren des Scorefiles.


[SelPulls]
; In diesem Bereich speichert Hamster Größe und Position des Auswahlfensters für das
; Abonnieren der Newsgruppen unter dem Menüpunkt Newsserver/Ladeaufträge.


[SSL]
; Hier werden die grundlegenden SSL-Einstellungen abgespeichert.

; Datei zum Initialisieren des Zufallsgenerators.
RandFile=

; Datei für das private SSL-Schlüsselpaar der lokalen Server.
PrivateKeyPair=

; Pfad für die lokal installierten Einzel-Zertifikate.
CaPath=

; Sammeldatei für lokal installierte Zertifikate.
CaFile=

; Überprüfung von Clientzertifikaten durch lokale Server:
;   0 = keine Überprüfung (Standard),
;   1 = Überprüfung nur, falls Zertifikat vorhanden,
;   2 = immer Überprüfung,
;   3 = immer Überprüfung & Vergleich mit lokaler Kopie.
VerifyLevel=0

; Optionen für die OpenSSL-Erweiterung.
UseSSLv3=1
UseTLSv1=1
LowEncryption=0
MediumEncryption=1
HighEncryption=1
UseRC2=1
UseRC4=1
UseIDEA=0
UseDES=1
Use3DES=1
UseMD5=1
UseSHA1=1
```

Glossar: SSL

```
; Kompatibilitäts-Einstellungen.
; Bitte nur aktivieren, wenn unbedingt nötig, da potentiell unsicher!
UseADH=0
UseExportAlgorithms=0
UseServerModeV2=0


[Stats]
; Hier merkt sich Hamster diverse Einträge für statistische Zwecke, die nur im
; Ausnahmefall editiert werden sollten.

LastInfoMsg=03.10.2001
Load=421295
Hist=83047
Kill=485146
```

IPAccess.hst

Achtung! Diese Einstellungen sind sicherheitsrelevant. Nehmen Sie nur Änderungen vor, wenn Sie genau ihre Wirkung kennen. Lesen Sie auch die FAQ zum Thema »Hamster und Sicherheit«.

Zweck der Datei »IPAccess.hst« ist es, den Hamster vor unbefugten Zugriffen aus dem Internet zu schützen. Dieses geschieht auf Basis der Internet-Adresse des Clients. Es kann festgelegt werden, welche IP-Adressen/-Adressräume vom Hamster akzeptiert werden und welche er zurückweisen soll.

Die Datei existiert in der Standardeinstellung des Hamsters nicht. Sie muss also von Hand oder durch eine Änderung im Menü-Dialog »Einstellungen« → »Lokale Server« ⇒ »Allgemeines« → »IPAccess.hst« angelegt werden.

Format:

Einfacher ASCII-Text. Kommentarzeilen beginnen mit dem Rautezeichen (»#«) oder dem Semikolon (»;«). Eine Zeile kann drei oder vier durch Kommata getrennte Elemente enthalten.

Syntax:

⟨Server⟩, ⟨Berechtigung⟩, ⟨IP-Adress-Bereich⟩

⟨Server⟩

Dieses Element gibt an, für welchen Server die Zeile gelten soll. Dies erlaubt eine Differenzierung je nach angebotenem Dienst (z. B. zur unterschiedlichen Handhabung ankommender und abgehender Mail).

Erlaubte Schlüsselworte:

Glossar:
NNTP
POP3
SMTP
IMAP
ReCo

NNTP: Lokaler NNTP-Server
POP3: Lokaler POP3-Server
SMTP: Lokaler SMTP-Server
IMAP: Lokaler IMAP-Server
RECO: Lokaler ReCo-Server
MAIL: Lokaler POP3-, SMTP- und IMAP-Server gemeinsam (entspricht somit »POP3« + »SMTP« + »IMAP«)
ALL: Alle lokalen Server gemeinsam (entspricht somit »MAIL« + »NNTP« + »RECO«)

⟨Berechtigung⟩

Dieses Element beschreibt die zu vergebenden Zugriffsrechte.

Berechtigungsart:

RW: Schreiben und Lesen (»*Read & Write*«)
WO: Nur Schreiben (»*Write Only*«)
RO: Nur Lesen (»*Read Only*«)
NA: Kein Zugriff (»*No Access*«)
ALL: Keine Einschränkungen (»*All Allowed*«)

⟨IP-Adress-Bereich⟩

IP-Adress-Bereich oder einzelne IP-Adresse, für welche die Zeile gelten soll. Wird ein Adressbereich angegeben, sind Anfangs- und End-Adresse des Bereiches durch ein Komma getrennt anzugeben.

Bitte beachten Sie, dass die IP-Adressen vom Absender der IP-Pakete gefälscht werden können, und daher dieser Filter keinen 100%igen Schutz gegen Eindringlinge bietet!

Sonderbehandlung der lokalen Adressen:
Das Schlüsselwort »LOCAL« im dritten Feld (IP-Adress-Bereich) legt fest, dass der Hamster alle lokalen IP-Adressen (also z. B. die beim Öffnen einer Internet-Verbindung mittels DFÜ-Netzwerk vom Provider [dynamisch] zugewiesene Adresse) wie die im vierten Feld angegebene IP-Adresse behandeln soll.

In diesem speziellen Fall müssen die Felder eins und zwei trotzdem die korrekte Syntax aufweisen, ihr konkreter Inhalt wird allerdings ignoriert, es gelten ausschließlich die Einstellungen des 4. Parameters.

Beispiel:

```
# alle lokalen IP-Adressen sollen behandelt werden wie 127.0.0.1:
ALL, NA, LOCAL, 127.0.0.1
```

Hinweise:

- Die Datei wird nur einmal beim Starten des Hamsters geladen. Nach Änderungen ist es somit notwendig, den Hamster zu beenden und erneut zu starten.
- Die Liste wird von oben nach unten abgearbeitet. Sobald eine Zeile zutrifft, wird deren Berechtigung verwendet und alle weiteren Zeilen werden ignoriert.
- Der Hamster ist für die private Nutzung zu Hause voreingestellt. Er erlaubt durch diese Voreinstellung den Zugriff über die IP-Adressen 127.0.0.1 und 192.168.0.0 bis 192.168.255.255. Alle lokalen IP-Adressen werden wie die IP-Adresse 127.0.0.1 behandelt. Das entspricht folgendem Inhalt (allerdings existiert die Datei, wie oben erwähnt, zunächst nicht):

```
ALL,NA,LOCAL,127.0.0.1
ALL,RW,127.0.0.1
ALL,RW,192.168.0.0,192.168.255.255
ALL,NA,0.0.0.0,255.255.255.255
```

Beispiele:

```
# Alle lokalen Adressen werden wie 127.0.0.1 behandelt:
ALL, NA, LOCAL, 127.0.0.1

# Voller Zugriff auf alle Server vom eigenen Rechner:
ALL, RW, 127.0.0.1

# Voller Zugriff auf alle Server von allen IP-Adressen im Intranet
# (=lokales Netzwerk):
ALL, RW, 192.168.0.0, 192.168.255.255

# Voller Zugriff auf alle Server von zwei speziellen IP-Adressen:
ALL, RW, 11.22.33.44  # John Doe
ALL, RW, 44.33.22.11  # Jane Doe

# Nur Lese-Zugriff auf den NNTP-Server (News) von allen IP-Adressen:
NNTP, RO, 0.0.0.0, 255.255.255.255

# Kein Zugriff auf irgendeinen Server von allen IP-Adressen:
# (drückt den Nutzwert des Hamsters erheblich, sofern vorher
# keine Ausnahmen definiert wurden ...)
ALL, NA, 0.0.0.0, 255.255.255.255
```

Glossar: NNTP

Scores.hst

Die in der Datei »Scores.hst« festgelegten Regeln entscheiden, welche Newsartikel geladen und welche ignoriert werden. Für die Behandlung von E-Mails siehe unter »MailFilt.hst«.

Wählen Sie die Filterkriterien sorgsam aus, andernfalls kann es zu Verlusten von eventuell wichtigen Nachrichten kommen. Schreiben Sie deshalb keine Filter aus anderen Quellen kritiklos ab und verwenden nur Filter, die Sie inhaltlich verstehen.

Eventuelle Fehlerquelle: Die Datei muss wirklich »`Scores.hst`« heißen, nicht »`Scores.txt`« oder »`Scores.hst.txt`« o. ä. Wird die Datei mit Notepad gespeichert, ist die Chance auf einen falschen Namen groß: Beim Abspeichern darauf achten, dass der Name in Anführungszeichen gesetzt wird (»`"Scores.hst"`«).

Hinweise:

Das angewendete Filterverfahren wird allgemein als »Scoring« bezeichnet. Jeder Artikel erhält einen fiktiven Wert (den »Score«), der mit dem Durchlaufen des Filters sinken und steigen kann. Jeder Artikel beginnt den Filterdurchlauf mit dem Wert Null. Ist dieser Wert nach dem Filterdurchlauf größer oder gleich Null, wird der Artikel geladen; ist er negativ, wird der Artikel ignoriert.

Wird ein Artikel ignoriert, so wird er – falls er einen einstellbaren Wert (Standard: -9999) nicht unterschreitet – im Killfile-Log verzeichnet. Später kann man sich über das Menü »Einstellungen« → »News: Killfile-Protokoll/-Einstellungen« eine Übersicht über die ungeladenen Artikel verschaffen und einzelne Artikel eventuell doch noch nachladen lassen.

Teile der Filter-Datei können mit folgender Präprozessor-Anweisung in eine separate Datei ausgelagert werden:

```
#!Include ⟨Dateiname⟩
```

Lange Zeilen können mittels Unterstrich (»_«) am Zeilenende auf mehrere kurze aufgeteilt werden.

Filterregeln können mit einem Kommentar versehen werden. Der Kommentar muss mit dem Rautezeichen (»#«) beginnen, auf welches aber kein Ausrufezeichen folgen darf.
Ein »#« darf auch hinter einem *kompletten* Filterausdruck einen Kommentar einleiten. Wenn ein Filterausdruck auf mehrere Zeilen aufgeteilt wird, darf ein Kommentar also auch nur als Abschluss in der letzten Zeile des Filterausdrucks stehen.

Nach dem Bearbeiten und Speichern der Datei »`Scores.hst`« ist es sinnvoll, im Hamster die Konfiguration neu zu laden über das Menü »Einstellungen« → »Konfiguration neu laden« oder durch Drücken der Taste F5 (wenn der Hamster wieder im Fokus ist), weil dann unter anderem auch eine Überprüfung der Filterdatei erfolgt. Bestenfalls sieht das dann so aus:

```
10:14:52 Sys {f84} Teste News-Filterdatei
10:14:52 I   {f84} News-Filterdatei: Test OK
```

»Teste News-Filterdatei« bedeutet, es wurde eine Datei namens »`Scores.hst`« gefunden. **»Test OK« bedeutet nicht unbedingt, dass alles wirklich in Ordnung ist, es sagt nur aus, dass der**

Hamster keine syntaktischen Fehler in den einzelnen Zeilen finden konnte. Ansonsten sieht es vielleicht so aus:

```
9:59:11 WAR {f84} Score-File "D:\Hamster\Scores.hst", Zeile 13 wird ignoriert:
                  =-9999 from "Spammer" +@newsgroups:"hamster.test"
9:59:11 WAR {f84} Grund: "=-9999 from "Spammer" +@newsgroups:"hamster.test"" =>
                  invalid field "newsgroups:"
```

Hier soll in Zeile 13 der »`Scores.hst`« auf die Headerzeile »Newsgroups:« gefiltert werden, was so nicht möglich ist. Was die einzelnen Felder bedeuten und bewirken, wird unten beschrieben; ebenso, auf welche Weise die gewünschte Filterung doch erfolgreich durchgeführt werden kann.

Zuerst folgt die Theorie in Form der Syntaxübersicht, darauf die Definition der Abschnitte und erlaubte Score-Regeln, danach werden noch einige praktische Beispiele gezeigt.

Syntaxübersicht:[1]

```
ScoreDatei          = *( ScoreBlock / cEOL )
ScoreBlock          = ScoreAbschnitt *( ScoreRegel / cEOL )

ScoreAbschnitt      = "[" Gruppenmuster *( 1*WSP Gruppenmuster ) "]" cEOL

Gruppenmuster       = [?] [ "+" / "-" ] Suchmuster
ScoreRegel          = [ "=" ] ScoreWert [ 1*WSP "unless" ] 1*WSP ScoreAuswahl cEOL

ScoreWert           = -2147483647...+2147483647
ScoreAuswahl        = ( ScoreDefFeld 1*( 1*WSP ScoreMuster ) [ 1*WSP ScoreExpire ] ) /
                      ( ScoreDefFeld [ 1*WSP ScoreExpire ] 1*( 1*WSP ScoreMuster ) )
ScoreMuster         = ["+"/"-"] [ "@" ScoreFeld ":" ] Suchmuster
ScoreDefFeld        = [ "~" ] ScoreFeld [":"]
ScoreFeld           = ( "Number" / "Subject" / "From" / "Date" / "Message-ID" /
                        "References" / "Bytes" / "Lines" / "Xref" / "Xpost" / "Age" )
ScoreExpire         = "Expire:" Datum
Datum               = Jahr Monat Tag ; JJJJMMTT
Jahr                = 4DIGIT ; vierstellige Jahreszahl JJJJ
Monat               = 2DIGIT ; zweistellige Monatszahl MM
Tag                 = 2DIGIT ; zweistellige Tageszahl TT

Suchmuster          = RegExpMuster / EinfachesMuster
RegExpMuster        = "{" RegularExpression "}"
RegularExpression   = PCRE ; vgl. FAQ Reguläre Ausdrücke
EinfachesMuster     = MusterAlle / MusterText / MusterZahl
MusterAlle          = "*"
MusterText          = CaseSensitiveText / CaseInsensitiveText
CaseInsensitiveText = """ Text """ ; ohne Beachtung von Groß-/Kleinschreibung
CaseSensitiveText   = "'" Text "'" ; mit Beachtung von Groß-/Kleinschreibung
```

[1] Orientiert an ABNF (RFC 5234)

```
MusterZahl          = "%" ( "<" / "=" / ">" / "<>" / ">=" / "<=" ) Zahl ["kb" / "mb"]
                      ; Achtung! 1 KB = 1024 Bytes, 1 MB = 1024 KB = 1048576 Bytes.
Zahl                = *DIGIT [ "." *DIGIT ]

DIGIT               = "0"..."9" ; eine Ziffer
cEOL                = [ "#" Kommentar ] CRLF
Kommentar           = Text
Text                = ISO-TEXT ; Import von ISO 8859
CRLF =              = %x0D %x0A ; = %d13 %d10, d. h. Wagenrücklauf (carriage return) und
                                ; Zeilenvorschub (line feed)
WSP                 = Space / HTab ; White-Space
Space               = %x20 ; = %d32, d. h. Leerzeichen (space)
HTAB                = %x09 ; = %d09, d. h. Tabulator (horizontal tab)
```

Score-Abschnitte:

Jeder Abschnitt im Scorefile beginnt mit dem Gültigkeitsbereich »**[…]**« Dieser Bereich legt die Newsgruppen fest, in denen die anschließend folgenden Filterausdrücke gültig sein sollen. Somit kann je nach Gruppe oder Teilhierarchie mit unterschiedlichen Kriterien gefiltert werden:

```
[*]
# Score-Abschnitt für alle Gruppen.

[* -".announce"]
# Score-Abschnitt für alle Gruppen, außer denen, welche die Zeichenkette ».announce«
# enthalten.

["news" "usenet"]
# Score-Abschnitt für alle Gruppen, welche die Zeichenketten »news« oder »usenet«
# enthalten.

[{^news\.} {^alt\.usenet\.}]
# Score-Abschnitt für alle Gruppen, welche mit den Zeichenketten »news.« oder
# »alt.usenet.« beginnen, als Reguläre Ausdrücke geschrieben, vgl. unten.
```

Die Ausdrücke innerhalb der eckigen Klammern (»[…]«) und die Score-Ausdrücke in den eigentlichen Filterzeilen sind dabei nach dem gleichen Schema aufgebaut.

Score-Regeln:

Mit jeder Score-Regel kann sich der Gesamt-Score eines Artikels verringern oder erhöhen. Dazu wird der Zeile ein **positiver oder negativer Wert** vorangestellt:

```
+100 subject "hamster"
-100 subject "make money fast"
```

Alternativ ist auch eine feste Wertzuweisung möglich, dazu wird dem eigentlichen Wert ein **Gleichheitsszeichen** (»=«) vorangestellt:

```
=+9999 from "my.mail@address"
=-9999 from "spam.mail@address"
```

In beiden Fällen wird der Score-Wert natürlich nur dann geändert bzw. gesetzt, wenn die dem Score-Wert folgende Regel zutrifft.

Auf dem Server können dabei ohne Laden des Artikels nur die im »XOver« des Servers vorhandenen Headerfelder verarbeitet werden. Meistens sind das die folgenden Felder: »**Subject**«, »**From**«, »**Date**«, »**Message-ID**«, »**References**«, »**Bytes**«, »**Lines**« und »**Xref**« (der Hamster speichert im Verzeichnis jedes Servers den jeweiligen »XOver« in der Datei »`overview.txt`«).

Glossar: Header

Erst, wenn der Hamster den Artikel vom Server geladen hat, kann auf alle anderen Header gefiltert werden. Eine Regel für das nachträgliche Filtern muss *mit einem Fragezeichen am Zeilenanfang* beginnen. Für das Filtern nach dem Laden des Artikels stehen zusätzlich die fiktiven Header »**Header**«, »**Body**« und »**Article**« zur Verfügung. Diese drei zusätzlichen Felder liegen immer im Rohformat vor. Die Sonderfunktion zur Headerdecodierung »~« (s. unten) kann auf diese Felder nicht angewandt werden.

Zur optischen Trennung kann nach dem Headerfeld ein Doppelpunkt gesetzt werden.

Beispiele für die Anwendung verschiedener Felder:

```
+100 subject "hamster"
=+0 subject: "subject"
-100 from {no.*spam}
+500 message-id "@my.unique.fqdn" # Bitte Hinweise im Glossar zu FQDN lesen!
+100 references "@my.unique.fqdn"
-100 bytes %>10000
```

Das fiktive Feld »**Xpost**« basiert auf dem Header »**Xref**« und enthält die Anzahl der Gruppen, in welchen der Artikel erscheinen soll. Dies erlaubt gezielte Reaktionen auf Cross-Postings:

```
-10 xpost %>2     # Artikel abwerten, die in mehr als zwei Gruppen gepostet wurden.
=-9999 xpost %>5  # in mehr als fünf Gruppen gepostete Artikel in die Tonne.
```

Achtung! Der »Xref«-Header ist von der jeweiligen Implementation des Remote-Servers abhängig. Sein Inhalt ist von den tatsächlich auf diesem Server geführten Gruppen abhängig und muss nicht mit dem Inhalt des Newsgroups-Header übereinstimmen.

Das fiktive Feld »**Age**« basiert auf dem Header »**Date**« und enthält das Alter des Artikels in Tagen:

```
=-9999 age %>14   # Über zwei Wochen alte Artikel ignorieren.
```

Das »**Bytes**«-Feld bietet die Möglichkeit, auf die Größe zu filtern:

```
-9999 Bytes: %>1000000
```

Man kann auch Angaben in »KB« und »MB« machen, wobei zu beachten ist, dass die Präfixe nicht dem SI entsprechend den Faktor $10^3 = 1000$ darstellen, sondern den Faktor $2^{10} = 1024$, d. h. 1 KB = 1024 Bytes und 1 MB = 1024 KB = 1 048 576 Bytes.

Beginnt der Feldname mit der **Tilde** (»~«), wird das dazugehörende Scoremuster vor Anwendung der Regel mit dem MIME-Verfahren decodiert (siehe hierzu auch die Beschreibung der Zeichensatzkonvertierungsdateien):

```
+100 ~subject "hämstêr"
-100 ~from "jürgen"
```

Die Decodierung erfolgt immer in den lokal eingestellten Systemzeichensatz. Darin nicht enthaltene Zeichen des Artikel-Zeichensatzes werden nicht ersetzt, so dass sie durch die zufällige Entsprechung im lokalen Zeichensatz dargestellt werden. *Infolgedessen können Sie sinnvoll **nur** auf Zeichen filtern, die in Ihrem lokalen Zeichensatz enthalten sind!*

Hinweis: Die MIME-Decodierung der Felder wirkt immer auf die komplette Filterzeile, also auch auf alle Felder einer Filterzeile! Sie kann nicht auf die fiktiven Headerfelder »Header«, »Body« und »Article« angewandt werden.

Glossar: Header

Score-Muster:

Muster ohne führendes Plus- oder Minuszeichen bedeuten, dass zumindest eines von mehreren Scoremustern dem Feld entsprechen muss, damit die Regel gültig ist:

```
# Subject enthält »hamster« oder »newsserver« oder »mailserver«:
+1 subject "hamster" "newsserver" "mailserver"
```

Scoremuster mit führendem Pluszeichen bedeuten, dass diese dem Feld entsprechen müssen, damit die Regel gültig ist:

```
# Subject enthält »newsserver« oder »mailserver« oder beides, aber in jedem Fall
# zusätzlich »hamster«:
+1 subject +"hamster" "newsserver" "mailserver"
```

Scoremuster mit führendem Minuszeichen bedeuten, dass diese dem Feld *nicht* entsprechen dürfen, damit die Regel gültig ist:

```
# Subject enthält »newsserver« oder »mailserver«, aber weder »unix«, noch »linux«,
# noch »inn«:
+1 subject "newsserver" "mailserver" -"unix" -"linux" -"inn"

# From-Header ohne »@«
=-9999 from -"@"
```

Bei numerischen Vergleichen muss ein **Prozentzeichen** (»%«) vorangestellt werden:

```
# Sehr lange Postings abwerten:
-100 lines %>250
```

Um in einer Abfrage auf den Inhalt eines anderen Feldes zugreifen zu können, muss der Feldname dem eigentlichem Scoremuster mit »**@⟨Feldname⟩:**« vorangestellt werden:

```
# Score erniedrigen, sofern das Subject »help«, »urgent« oder »!!!« enthält,
# es sei denn, das From-Feld enthält »my@address« oder das Subject »SCNR«:
-1 subject "help" "urgent" "!!!" -@from:"my@address" -"SCNR"
```

Beachten Sie, dass zwischen »@⟨Feldname⟩:« und »⟨Muster⟩« *keine Leerzeichen* vorkommen dürfen: »@⟨Feldname⟩:⟨Muster⟩«. Diese Nebenregel darf auch erst *nach* der Hauptregel (im Beispiel »subject«) eingetragen werden.

Man kann Regeln auch mit einem Verfallsdatum versehen. Dafür gibt es die spezielle Anweisung »**Expire:JJJJMMTT**« (*ohne Punkte und Leerzeichen!*), die dafür sorgt, dass die Regel, auf die sie angewandt wird, ab dem Datum ⟨TT.MM.JJJJ⟩ ungültig wird und vom Hamster ignoriert wird. Wie der Hamster beim Aufräumen damit umgehen soll, kann in »Einstellungen« → »Automatische Abläufe« ⇒ »Allgemeines« eingestellt werden. Der Expire-Parameter darf höchstens einmal an beliebiger Stelle *nach* der Hauptregel eingesetzt werden, sofern zumindest ein Leerzeichen Abstand eingehalten wird.

```
# Beispiel für Verfallsdatum: Regel wird am 31.Dezember 2000 ungültig.
=-9999 References: "<3ae1.invalid>" Expire:20001231
```

Stehen Scoremuster in Anführungsstrichen (»"⟨Muster⟩"«), werden sie als einfache Zeichenketten interpretiert, stehen sie dagegen in geschweiften Klammern (»{⟨Muster⟩}«), werden sie als Reguläre Ausdrücke betrachtet, was deutlich komplexere Ausdrücke erlaubt:

```
# Spam-Spoiler ignorieren: Regel erschlägt z. B. »From: nospam@nix.example.com«,
# »From: no.spam@nada.example.net«, »From: no-spam-please@example.org«,
# »From: please.delete@this.invalid«, »From: deletethis.plz@example.com«,
# »From: remove_this@address.invalid«, »From: name_cut-this-out@example.net«.
-1 from {no.?spam} {(remove|delete|cut).*this}
```

Weitere Beispiele:

```
# Eigene Artikel haben oberste Priorität:
=+9999 From "Mein Name" # Fehltreffer möglich bei Namensgleichheiten.
=+9999 Message-ID "@mein.fqdn"

# Antworten auf eigene Artikel
=+5000 References "@mein.fqdn"

# Überlange Artikel ignorieren, sofern nicht »FAQ« im Subject steht und nicht in
# Gruppen erschienen, die »announce«, »info« oder »infos« im Namen enthalten:
[* -announce -{infos?$}]
-10 Lines %>200
-10 Bytes %>10000
+20 Subject "FAQ"

# Ignorieren von Artikeln, die in mehr als 3 Gruppen erscheinen sollen:
-10 Xpost %>3 # Alternative mit gleicher Wirkung: »-10 Xpost %>=4«
```

```
# Postings ab einer bestimmten Größe abwerten, zum Umrechnungsfaktor vgl.
# Bemerkungen oben
-5000 Bytes %>=500kb # = 512.000 Bytes
-9999 Bytes %>=1mb # = 1.048.576 Bytes

# Ignorieren von Artikeln mit »!!!« im Subject außer in Anfänger-Gruppen:
[* -newuser]
-1 Subject "!!!"

# Lange Postings eines bekannten, häufig nervenden Posters ignorieren bzw. ab einer
# bestimmten Länge kommentarlos verwerfen:
=-9999 lines: %>200 +@from:"Detlef Dampfplauderer" # Score festgesetzt.
-1234 lines: %>100 +@from:"Detlef Dampfplauderer"  # Score noch veränderbar, lässt
                                                   # Raum für weitere Regeln.

# Beispiel für das Filtern nach dem Laden der Artikel:
?+10 Supersedes: *
?=-9999 Body: "Make Mony fast"
```

MailFilt.hst

Glossar: POP3

Die Regeln in der Datei »`MailFilt.hst`« beschreiben, welche E-Mails vom POP3-Server geladen werden sollen bzw. was mit ihnen zu geschehen hat. Für die Behandlung von Newsartikeln siehe unter »Scores.hst«.

Wählen Sie die Filterkriterien sorgsam aus, andernfalls kann es zu Verlusten von eventuell wichtigen Nachrichten kommen. Schreiben Sie deshalb keine Filter aus anderen Quellen kritiklos ab und verwenden nur Filter, die Sie inhaltlich verstehen.

Eventuelle Fehlerquelle: Die Datei muss wirklich »`MailFilt.hst`« heißen, nicht »`MailFilt.txt`« oder »`MailFilt.hst.txt`« o. ä. Wird die Datei mit Notepad gespeichert, ist die Chance auf einen falschen Namen groß: Beim Abspeichern darauf achten, dass der Name in Anführungszeichen gesetzt wird (»`"MailFilt.hst"`«).

Hinweise:

Teile der Filterdatei können mit folgender Präprozessor-Anweisung in eine separate Datei ausgelagert werden:

```
#!Include ⟨Dateiname⟩
```

Lange Zeilen können mittels Unterstrich (»_«) am Zeilenende auf mehrere kurze aufgeteilt werden.

Filterregeln können mit einem Kommentar versehen werden. Der Kommentar muss mit dem Rautezeichen (»#«) beginnen, auf welches aber kein Ausrufezeichen folgen darf.

Ein »#« darf auch hinter einem *kompletten* Filterausdruck einen Kommentar einleiten. Wenn ein Filterausdruck auf mehrere Zeilen aufgeteilt wird, darf ein Kommentar also auch nur als Abschluss in der letzten Zeile des Filterausdrucks stehen.

Nach dem Bearbeiten und Speichern der Datei »MailFilt.hst« ist es sinnvoll, im Hamster die Konfiguration neu zu laden über das Menü »Einstellungen« → »Konfiguration neu laden« oder durch Drücken der Taste F5 (wenn der Hamster wieder im Fokus ist), weil dann unter anderem auch eine Überprüfung der Filterdatei erfolgt. Bestenfalls sieht das dann so aus:

```
09:47:53 Sys {de7} Teste Mail-Filterdatei
09:47:53 I   {de7} Mail-Filterdatei: Test OK
```

»Teste Mail-Filterdatei« bedeutet, es wurde eine Datei namens »MailFilt.hst« gefunden. **»Test OK« bedeutet nicht unbedingt, dass alles wirklich in Ordnung ist, es sagt nur aus, dass der Hamster keine syntaktischen Fehler in den einzelnen Zeilen finden konnte.** Ansonsten sieht es vielleicht so aus:

```
10:00:03 WAR {de7} Error in Mail-filter-file "D:\Hamster\MailFilt.hst"
10:00:03 WAR {de7} Error in line 3: noitfy(admin)
10:00:03 WAR {de7} Error description: "noitfy(" => Unknown action "noitfy"
```

Hier ist dem Hamster die Aktion »noitfy« in Zeile 3 der »MailFilt.hst« unbekannt. Richtig muss es heißen »notify(admin)«. Was das ist und wofür es gut ist, wird weiter unten erklärt.

Zuerst folgt die Theorie in Form der Syntaxübersicht, darauf zwei Abschnitte darüber, wie man über Laden bzw. Nichtladen sowie die Verteilung auf verschiedene Nutzer entscheiden kann, anschließend die Definition der Filterabschnitte und erlaubten Filterregeln, und zum Schluss werden noch einige praktische Beispiele gezeigt.

Syntaxübersicht:[1]

```
MFilterDatei        = *( MFilterBlock / cEOL )
MFilterBlock        = MFilterAbschnitt *( MFilterRegel / cEOL )
MFilterAbschnitt    = "[" Teilmuster *(1*WSP Teilmuster) "]" cEOL
MFilterRegel        = ["="] ( MLadeRegel / MZielRegel ) [1*WSP Expire] cEOL

MLadeRegel          = "load"       [ "("                  ")" ] 1*WSP MAuswahl /
                      "ignore"     [ "("                  ")" ] 1*WSP MAuswahl /
                      "kill"       [ "("                  ")" ] 1*WSP MAuswahl /
                      "notify"     [ "(" [ Benutzerliste ] ")" ] 1*WSP MAuswahl /
                      "notifyoff"                                1*WSP MAuswahl /
                      "log"        [ "(" [ Benutzerliste ] ")" ] 1*WSP MAuswahl /
                      "logoff"                                   1*WSP MAuswahl

MZielRegel          = "addaccounts" [ "("                  ")" ] 1*WSP MPrimFeld /
                      "adddefault"  [ "("                  ")" ] 1*WSP MAuswahl  /
```

[1] Orientiert an ABNF (RFC 5234)

```
                  "default"     [ "(" [ Benutzerliste ] ")" ] 1*WSP Mauswahl  /
                  "add"           "("   Benutzerliste   ")"   1*WSP MAuswahl  /
                  "set"           "("   Benutzerliste   ")"   1*WSP MAuswahl  /
                  "del"           "("   Benutzerliste   ")"   1*WSP MAuswahl  /
                  "postto"        "("    Newsgruppen    ")"   1*WSP MAuswahl  /
                  "setscore"      "("     Scorewert     ")"   1*WSP MAuswahl  /
                  "addscore"      "("     Scorewert     ")"   1*WSP MAuswahl

Expire            = "Expire:" Datum
Datum             = Jahr Monat Tag
Jahr              = 4DIGIT ; vierstellige Jahreszahl JJJJ
Monat             = 2DIGIT ; zweistellige Monatszahl MM
Tag               = 2DIGIT ; zweistellige Tageszahl TT

MPrimFeld         = (MEinzelfeld / MGruppenfeld)[:]
MAuswahl          = MHauptfeld *( 1*WSP ( Muster / MNebenfeld ) )

MHauptfeld        = [ "unless" 1*WSP ] [ "~" ] [ "*" ] ( MEinzelfeld /
                    MGruppenfeld ) [ ":" ] 1*WSP Muster
MEinzelfeld       = RFCFieldName / "Bytes" / "Top" / "Score" / "TopOnly" / "Header" / "Age"
MGruppenfeld      = "Any-Sender" / "All-Senders" / "Any-Recipient" / "All-Recipients"
MNebenfeld        = [ "+" / "-" ] "@" MEinzelfeld ":" Muster

Benutzerliste     = BenutzerSpez *( "," BenutzerSpez )
BenutzerSpez      = BenutzerName [ SubFolderSpez ] [ ":" IMAPFlagSpez ]
SubFolderSpez     = 1*( "/" Subfoldername )
IMAPFlagSpez      = *( "\" IMAPFlagname )

BenutzerName      = 1*BString
BString           = "A"..."Z" / "a"..."z" / "0"..."9" / "!" / "#" /
                    "$" / "&" / "'" / "+" / "-" / "_" / "=" / "."

Subfoldername     = 1*FString
FString           = "A"..."Z / "a..."z" / "0"..."9" / "!" / "$" / "'" /
                    "+" / "-" / "." / "=" / "@" / "^" / "_" / "´" / "~"

IMAPFlagName      = "seen" / "recent" / "answered" / "draft" / "flagged"

Scorewert         = -2147483647...+2147483647

Newsgruppen       = Newsgruppenname *( "," Newsgruppenname )

Teilmuster        = [ "+" / "-" ] Muster
Muster            = RegExpMuster / EinfachesMuster
RegExpMuster      = "{" RegularExpression "}"
RegularExpression = PCRE ; vgl. FAQ Reguläre Ausdrücke
EinfachesMuster   = MusterAlles / MusterText / MusterZahl / MusterHeader
MusterAlles       = "*"
MusterText        = CaseSensitiveText / CaseInsensitiveText
```

```
CaseSensitivetext   = "'" Text "'" ; mit Beachtung von Groß-/Kleinschreibung
CaseInsensitiveText = """ Text """ ; ohne Beachtung von Groß-/Kleinschreibung
MusterZahl          = "%" ( "<" / "=" / ">" / "<>" / ">=" / "<=" ) Zahl ["kb" / "mb"]
                      ; Achtung! 1 KB = 1024 Bytes, 1 MB = 1024 KB = 1.048.576 Bytes.
MusterHeader        = "$" RFCFieldName "$"

RFCFieldName        = RFC 5322 ; Headerfeld entsprechend RFC
Zahl                = *DIGIT [ "." *DIGIT ]
DIGIT               = "0"..."9" ; eine Ziffer

cEOL                = [ "#" Kommentar ] CRLF
Kommentar           = Text
Text                = ISO-TEXT ; Import von ISO-8859
CRLF =              = %x0D %x0A ; = %d13 %d10, d. h. Wagenrücklauf (carriage return) und
                                ; Zeilenvorschub (line feed)
WSP                 = Space / HTab ; White-Space
Space               = %x20 ; = %d32, d. h. Leerzeichen (space)
HTab                = %x09 ; = %d09, d. h. Tabulator (horizontal tab)
```

Laden oder nicht laden ...

Glossar: Header

Enthält der Filter »Ignore«- oder »Kill«-Regeln, holt der Hamster als erstes die Kopfzeilen der E-Mails und entscheidet dann, welche E-Mails komplett geladen, gelöscht oder ignoriert werden.

Sofern eine »**ignore**«-Regel im Filter anschlägt, wird die E-Mail nicht geladen, aber auf dem Server belassen. Wenn eine »**kill**«-Regel im Filter anschlägt, wird die E-Mail nicht nur nicht geladen, sondern auch noch auf dem Server gelöscht.

Sollte allerdings eine nachfolgende »**load**«-Regel zutreffen, wird die vorherige »Ignore«-/»Kill«-Anweisung dadurch wieder aufgehoben, und die E-Mail geladen.

E-Mails, auf die *keine* Regel zutrifft, werden immer geladen.

Falls die Ausführung von »Ignore«- und »Kill«-Regeln gemeldet werden soll, kann man dies über eine passende »**notify**«-Anweisung erreichen, ansonsten erfolgt in der Voreinstellung das Ignorieren und Löschen kommentarlos. Bei mehreren »Notify«-Anweisungen gilt immer nur die aktuell letzte Benutzerliste, die vorherige wird durch die jeweils nächste überschrieben. Diese Liste kann mit der Anweisung »**notifyoff**« auch wieder komplett gelöscht werden, so dass keine Benachrichtigungen versandt werden. Wird die »Notify«-Anweisung ohne Parameter verwendet, so werden alle Meldungen an den Standard-Account gesendet.

Die »**log**«-Anweisung erlaubt den Eintrag von Mails, die von einem Filter ignoriert oder gelöscht werden, in das Mail-Killfile-Protokoll; »**logoff**« schaltet das Loggen wieder ab. In der Voreinstellung ist das Mitloggen ausgeschaltet. Die Log-Einträge lassen sich per Menü »Einstellungen« → »Mail: Killfile-Protokoll« bequem dazu nutzen, ignorierte Mails endgültig zu löschen oder nachzuladen. Als Parameter wird entweder wie bei »**notify**« ein oder mehrere User oder »*«

für ein globales Protokoll angegeben; bei fehlendem oder leeren Parameter wird das Killfile-Protokoll des Standard-Accounts verwendet.

Eine alternative, aber auch wesentlich komplexere Methode, zu entscheiden, welche E-Mails geladen werden, ist das sogenannte »***Scoring***«. Hierzu vergibt man für jede E-Mail mit der Laderegel »**addscore**« oder »**setscore**«, je nach Ergebnis einer Auswahlregel, einen frei wählbaren Punktwert mit negativen oder positiven Vorzeichen. Somit kann man durch Verwendung mehrerer Auswahlregeln eine Gesamtentscheidung treffen, ob eine E-Mail geladen werden soll oder nicht. Diese Entscheidung realisiert man mit einer »**kill**«- oder »**ignore**«-Regel, welche den Wert des Pseudo-Headers »**Score:**« auf die Einhaltung eines vorgegebenen Wertes hin überprüft.

Glossar: Header

Mit der zusätzlichen Anweisung »**Expire:JJJJMMTT**« (*enthält keine Punkte und Leerzeichen!*) können Einträge mit Verfallsdatum versehen werden: Sie werden dann ab dem Datum ⟨TT.MM.JJJJ⟩ vom Hamster ignoriert. Der Expire-Parameter darf *nach* dem eigentlichen Filterbefehl höchstens einmal an beliebiger Stelle in der Definition eingesetzt werden, d. h. vor oder nach dem Filterfeld/einem Filterausdruck, sofern zumindest ein Leerzeichen Abstand eingehalten wird:

```
# Filter ist nur gültig bis zum 30. Dezember 2000:
=Ignore() ~From: "Heiner Dämlich" Expire:20001231
# Alternativer Ausdruck mit derselben Bedeutung:
=Ignore() Expire:20001231 ~From: "Heiner Dämlich"
```

Vor allem in der Testphase sollte man von den »Log«- und »Notify«-Möglichkeiten ausgiebig Gebrauch machen und sicherheitshalber erst einmal nur »Ignore«-Regeln vorsehen.

Hinweis: Die Option »Einstellungen« → »Mail: Server konfigurieren« ⇒ »POP3-Einstellungen« hat eine höhere Priorität als die Mail-Filter der »`MailFilt.hst`«.

POP3

Verteilung der Post auf lokale Nutzer

Der Hamster legt eine geladene E-Mail immer im Postfach mindestens eines Benutzers ab. Das Postfach legt der Anwender entweder durch den Filter (die Datei »`MailFilt.hst`«) fest oder, wenn dieser kein Ergebnis liefert, durch den Standardempfänger. Kann der Hamster kein gültiges Postfach ermitteln, verwendet er das »admin«-Postfach.

Zur Auswertung des Filters benutzt der Hamster eine anfangs leere Empfängerliste. Folgende Regeln beeinflussen die Liste:

- Eine gültige »**add**«-Regel fügt den angegebenen Empfänger der Empfängerliste hinzu.
- Eine gültige »**del**«-Regel löscht den angegebenen Empfänger wieder aus der Liste.
- Eine gültige »**set**«-Regel ersetzt die vorhandenen Einträge durch den angegebenen Empfänger. Mit einer leeren Benutzerliste, d. h. bei Angabe von »set()«, löscht die Regel die bisherige Liste.
- Die »**addaccounts**«-Regel testet, ob die E-Mail-Adresse des angegebenen Headers lokal bekannt ist (d. h. sie ist in den Einstellungen eines lokalen Benutzers eingestellt worden). Ist dies der Fall, wird der entsprechende lokale Benutzer der Empfängerliste hinzugefügt.

- Die »**adddefault**«-Regel fügt den Standardempfänger hinzu.
- Eine gültige »**postto**«-Regel fügt eine Newsgruppe als Empfänger ein. *(Das Headerfeld »FollowUp-To: poster« setzt der Hamster dabei automatisch.)*

Glossar: Header

Bei einem IMAP-Konto als Empfänger können zusätzlich zum Postfach noch ein IMAP-Verzeichnis und die IMAP-Flags angegeben werden (siehe Beispiele unten).

IMAP

Der Standardempfänger kann wie folgt festgelegt werden (Reihenfolge in absteigender Priorität):

- Durch eine gültige »**default**«-Regel im Filter. Mit einer leeren Benutzerliste, d. h. bei Angabe von »default()« oder »default«, löscht die Regel einen zuvor gesetzten Standardempfänger.
- Wenn der Mail-Abruf durch ein Skript erfolgt, durch den (optionalen) Parameter »⟨destuser⟩« des Skriptbefehls »HamFetchMail«.
- Durch die entsprechende Option in den Einstellungen des betreffenden POP3-Servers.

POP3

Filterabschnitte:

Jeder Abschnitt wird mit eckigen Klammern »[...]«-gekennzeichnet. Er kann einen Stern »*« oder einen Abschnittsnamen aus den Menüeinstellungen für die POP3-Server oder aus einem Skriptbefehl enthalten oder eine Kombination aus mehreren Abschnitten, vgl. Beispiele. Abschnittsnamen können gezielt für einzelne Nutzer im Menü »Einstellungen« → »Mail: Server konfigurieren« ⇒ »POP3-Mailserver« eingestellt werden. Der Stern bezeichnet den allgemeinen Abschnitt, der alle E-Mails betrifft. Zusätzlich kann ein Plus- oder Minuszeichen vorangesetzt werden, wobei für die Auswahl der Abschnitte bei der Filterung gilt:

- Von mehreren Abschnittsnamen ohne Plus- oder Minuszeichen muss mindestens einer zutreffen.
- Abschnittsnamen mit führendem Pluszeichen müssen in jedem Fall zutreffen.

  Abschnitte mit führendem Pluszeichen zu verwenden dürfte in der »`MailFilt.hst`« sehr selten sinnvoll sein, allerdings wird dieselbe Parser-Engine auch bei den Filterregeln und in der »`Scores.hst`« benutzt.

- Abschnittsnamen mit führendem Minuszeichen sind in jedem Fall ausgeschlossen.

```
[*]
# Filterregeln für alle E-Mails

[private]
# Filterregeln für »private« z. B. aus Skriptbefehl oder Einstellung für Server

[* -private]
# Filterregeln für alle E-Mails außer den privaten
```

Filterregeln:

Hinweis: Die folgende Beschreibung enthält hauptsächlich die Differenzen zwischen Mail- und Newsfilter. Für nähere Details, vor allem bei Verwendung von setscore *und* addscore*, siehe somit auch die Erläuterungen zur* »Scores.hst«.

Jede Filterregel besteht aus einem Befehl (fett markiert) und einem Filterausdruck (kursiv):

```
kill()     From: "sehr.bekannter@spammer.invalid"
=add(john) To:   "john@mail.example.com"
postto(internal.mailinglists.ehamster) To: "ehamster@egroups.com"
```

Wichtiger Hinweis: Hinter »kill«, »add«, »postto« etc. und der öffnenden Klammer darf *kein Leerzeichen* stehen, ansonsten ist die Regel technisch ungültig und wird als nicht existent betrachtet! Zwischen der schließenden Klammer und dem Suchausdruck sind beliebig viele Leerzeichen dagegen erlaubt und der Übersichtlichkeit halber auch empfohlen.

Ein **Gleichheitszeichen** (»=«) vor einer Regel bedeutet, dass, sobald diese Regel zutrifft, für die gerade bewertete E-Mail die Abarbeitung der Filterregeln an dieser Stelle abgebrochen und der bis dorthin ermittelte Stand der Filterung auf die E-Mail angewandt wird; *alle* folgenden Regeln werden ignoriert.

Der **Befehl** gibt an, welche Aktion der Filter ausführt, wenn der Filterausdruck zutrifft.

Der **Filterausdruck** besteht aus einem oder mehreren Filtermustern. Bei mehreren Filtermustern entscheidet der Aufbau der Regel darüber, ob sie teilweise oder vollständig zutreffen müssen, damit der Gesamtausdruck greift. Ein Filtermuster besteht aus einem *Feldbezeichner*, folgend meist kurz als *Feld* bezeichnet, und einem *Muster*, das auf diesen Bezeichner zutreffen muss. Jedes Muster kann als einfache Zeichenkette in Anführungsstrichen angegeben werden (»"⟨Muster⟩"«) oder als Regulärer Ausdruck in geschweiften Klammern (»{⟨Muster⟩}«), was deutlich komplexere Ausdrücke erlaubt.

Glossar: Header

Die Feldbezeichner beziehen sich entweder direkt auf einen Header des zu testenden Artikel oder auf einen der vom Hamster aus diesem Artikel generierten Spezialfelder.

Das Haupt-Filtermuster, welches in jedem Filterausdruck vorkommen *muss*, kann aus einem oder mehreren, durch Leerzeichen getrennten Teilmustern bestehen, die sich alle auf dasselbe Hauptfeld beziehen.

Neben-Filtermuster beziehen sich auf einen abweichenden Feldbezeichner. Der Nebenfeld-Bezeichner muss zu Beginn mit einem »@« markiert werden, zwischen dem Bezeichner und dem Muster *muss* ein **Doppelpunkt** »:« stehen, und es darf sich an dieser Stelle *kein Leerzeichen* befinden. Der separate Teil würde sonst als weiteres Teilmuster des Haupt-Filtermusters angesehen werden.[1]

[1] Syntaktisch gesehen ist das gesamte Neben-Filtermuster im Prinzip ein zusätzliches Teilmuster des Hauptfilters.

Den Haupt-Mustern und den Neben-Filtermustern kann optional ein **Plus-** (»+«) oder **Minuszeichen** (»-«) vorangestellt werden.

Damit der Filterausdruck greift …

Glossar: Header

- muss der Header, auf den sich das Haupt-Filtermuster bezieht, vorhanden sein.

  Bei den vom Hamster generierten Pseudoheadern (s. u.) ist das immer der Fall. In einer E-Mail kann ein Header aber auch fehlen. Laut dem RFC für SMTP sind ausschließlich die Header »From:« und »Date:« Pflicht. Die meisten Mailserver verlangen zusätzlich den »To:«-Header, nicht immer aber wird z. B. kontrolliert, ob ein »Subject:« vorhanden ist.

SMTP

- müssen *alle* Ausdrücke zutreffen, die mit einem »+« beginnen,
- darf *kein einziger* Ausdruck zutreffen, der mit einem »-« beginnt,

  und

- es muss *mindestens einer* der restlichen Ausdrücke zutreffen.

Das Schlüsselwort »**unless**« kehrt die Bedeutung des *gesamten* Filterausdrucks um.

```
=set(suspect) unless Subject: * # Mail ohne Betreff dem Konto »suspect« zuordnen
```

Spezialfelder

Der Pseudoheader »**Any-Sender**« testet alle senderspezifischen Header wie »From:«, »Apparently-From:«, »Sender:«, »Reply-To:«, »X-Sender:«, »Envelope-From:« und »X-Envelope-From:«:

```
load() Any-Sender: "boss@company.example.com"
```

Der Pseudoheader »**All-Senders**« prüft nicht alle Senderfelder einzeln, sondern – mit Zeilenvorschub (CR/LF [#13#10]) getrennt – auf einmal.

Der Pseudoheader »**Any-Recipient**« testet umgekehrt alle empfängerspezifischen Header wie: »To:«, »Apparently-To:«, »CC:«, »BCC:«, »Envelope-To:« und »X-Envelope-To:« *(nicht aber »Delivered-To:«!)*:

```
add(john) Any-Recipient: "john@mail.example.org"
add(jane) Any-Recipient: "jane@mail.example.org"
addaccounts() Any-Recipient:
```

Der Pseudoheader »**All-Recipients**« prüft nicht alle Empfängerfelder einzeln, sondern – mit Zeilenvorschub (CR/LF [#13#10]) getrennt – auf einmal.

Mit dem Spezialfeld »**Header**« kann auf den kompletten Header einer E-Mail zugegriffen werden:

```
load() Header: "Mein Name" # Mails, welche etwas mit mir zu tun haben
```

Glossar: POP3

Das Spezialfeld »**Top**« greift auf die beim Filtern vorliegenden Zeilen des Bodys zu. Ob es sich dabei um den gesamten Body handelt, ist von verschiedenen Faktoren abhängig. Unterstützt der externe Server das gleichnamige POP3-TOP-Kommando und ist dessen Verwendung im Hamster aktiviert *(in der »Server.ini«, Sektion »[POP3]«, des betreffenden POP3-Servers Schlüssel »UseTOPCommand« auf »1« gesetzt)*, lädt der Hamster nur die unter »Einstellungen« → »Mail: Killfile-Protokoll« ⇒ »Einstellungen« → »TOP-Command … « festgelegte Anzahl von Zeilen herunter, sonst wird die E-Mail vollständig geladen. Außerdem lädt der Hamster sofort die gesamte Mail, wenn deren Größe nur unwesentlich größer ist, als der vermutliche Umfang der Toplines. Bei dem Feld »**TopOnly**« handelt es sich um eine Variante dieses Feldes, bei der die Filterung *in jedem Fall* nur auf den entsprechend der Konfiguration gekürzten Body angewandt wird.

Siehe auch in der Telnet-FAQ die Ausführungen zum TOP-Kommando.

Größe der E-Mail: Vgl. LIST-Kommando in der Telnet-FAQ.

Mit einem speziellen Feld kann man die E-Mail auf ihre Größe testen:

```
ignore() Bytes: %>100000
```

Wegen des numerischen Vergleichs wird hier das **Prozentzeichen** (»**%**«) benötigt. Man kann auch Angaben in »KB« und »MB« machen, wobei zu beachten ist, dass die Präfixe nicht dem SI entsprechend den Faktor $10^3 = 1000$ darstellen, sondern den Faktor $2^{10} = 1024$, d. h. 1 KB = 1024 Bytes und 1 MB = 1024 KB = 1 048 576 Bytes.

Header

Das fiktive Feld »**Age:**« basiert auf dem Header »**Date**« und enthält das Alter der E-Mail in Tagen (auch hier wegen des numerischen Vergleichs ein »%«):

```
=ignore Age %>14 # Über zwei Wochen alte E-Mail ignorieren
```

Der Pseudoheader »**Score**« ermöglicht die fortgeschrittene Methode des Scorings, siehe auch Beispiele weiter unten. Hier ebenso »%« wegen numerischer Vergleiche.

```
set(suspect) Score: %<0 # Mail mit negativem Score dem Konto »suspect« zuordnen
```

Sonderfunktionen

Normalerweise wird bei einer Regel nur die erste passende Header-Zeile je E-Mail ausgewertet. Um auch multiple Felder komplett zu testen, muss dem Feldnamen ein **Stern** »*« vorangestellt werden:

```
add(john) *Received: "john@"
```

Beginnt der Feldname mit der **Tilde** (»~«), wird das Feld vor Anwendung der Regel mit dem MIME-Verfahren decodiert (siehe hierzu auch die Beschreibung der Zeichensatzkonvertierungsdateien):

```
load() ~To: "jürgen"
```

Die Decodierung erfolgt immer in den lokal eingestellten Systemzeichensatz. Darin nicht enthaltene Zeichen des E-Mail-Zeichensatzes werden nicht ersetzt, so dass sie durch die zufällige

Entsprechung im lokalen Zeichensatz dargestellt werden. *Infolgedessen können Sie sinnvoll **nur** auf Zeichen filtern, die in Ihrem lokalen Zeichensatz enthalten sind!*

Hinweis: Die MIME-Decodierung kann nur auf Hauptfelder angewandt werden *(Nebenfelder werden durch vorangestelltes »@« gekennzeichnet)*.

Glossar: Header

Die Spezialvariable »**XXX**« kann für Vergleiche mit dem Inhalt anderer Mail-Header herangezogen werden, wobei »XXX« für einen beliebigen Header der zu beurteilenden Mail steht:

```
ignore() From: $To$ # Ignorieren, wenn in »From:« und»To:« dasselbe steht.
```

Weitere Beispiele:

```
# E-Mails an bekannte lokale Adressen zustellen, dabei alle Empfängerfelder und die
# kompletten Received-Header überprüfen:
[*]
addaccounts() Any-Recipient:
addaccounts() *Received:

# User und Admin benachrichtigen, wenn E-Mails gelöscht oder ignoriert wurden:
Notify(user,admin)

# E-Mails eines bekannten Spammers oder von einer unerwünschten Domain löschen:
[*]
kill() Any-Sender: "grosser@spammer.example.com"
kill() Any-Sender: "@spammer-domain.example.net"

# Die E-Mails ignorieren, deren »From«-Header eine bestimmte Adresse enthält und sie
# auf dem Server liegen lassen, um sie später noch anderweitig zu lesen.
ignore() From: "geschaeftsadresse@firma.example.com"

# E-Mails eines bekannten Trolls für eine gewisse Zeit ignorieren:
=Ignore() Expire:20031231 ~From: "grêãt trøll"

# Filterabschnitt für einen bestimmten Nutzer »jemand« im Abschnitt [irgendwer]
[irgendwer]
kill() From: "feines_von_jemand@spammer.example.com"
add(jemand) To: "jemand@mail.example.org"

# Alle E-Mails vom Chef laden, es sei denn, der Betreff enthält ein bestimmtes
# Reizwort:
load() Any-Sender: "boss@company.example.com" -@Subject:"Kündigung!"

# Die E-Mails einer Mailingliste in eine lokale Newsgruppe posten und dem Admin noch
# jeweils eine Kopie der Original-Mail senden:
[*]
postto(local.hamster-ml.usehamsternet) To: "usehamsternet@egroups.com"
add(admin)                             To: "usehamsternet@egroups.com"
postto(local.hamster-ml.ehamster) To: "ehamster@egroups.com"
add(admin)                        To: "ehamster@egroups.com"
```

```
# Alle E-Mails löschen, deren »From«- und »To«-Header identisch sind. In der Regel
# schicken sich nur Spammer selber E-Mails. Doch Vorsicht: Newsletter und andere
# erwünschte E-Mails, die ebenfalls identische »From«- und »To«-Header enthalten
# können, sind dann unbedingt gesondert zu behandeln. - Der »To«-Header stimmt mit
# dem »From«-Header überein und enthält mindestens ein Zeichen (bei »{.+}« handelt
# es sich um einen Regulären Ausdruck):
kill() From: $To$ +@To:{.+}
# Anmerkung: Der Quantifier »+« im Zusatz »+@To:{.+}« ist eigentlich überflüssig,
# jedoch für ein gutes, schnelles Verständnis des Filters ungemein wertvoll.

# E-Mails ignorieren, wenn der Headerfeld-Bezeichner »Subject« selbst ausschließlich
# aus Großbuchstaben besteht (nur mit Regulären Ausdrücken möglich):
=ignore Header: {(?m-i)^SUBJECT:}

# E-Mails bis zu einer bestimmten Größe einem User zuordnen, ab einer bestimmten
# Größe aber ignorieren oder sogar löschen
add(john) Bytes: %<=10kB # = 10.240 Bytes, vgl. die Bemerkungen oben
ignore()  Bytes: %>500000
kill()    Bytes: %>=1.5MB # = 1.572.864 Bytes, vgl. die Bemerkungen oben

# Einen Scorewert für eine E-Mail festlegen
addscore(-100) ~From: "grêãt trøll"
addscore(+500) ~To: "Mein Name"
#
# Entscheiden, ob eine E-Mail anhand ihres Scorewertes gelöscht werden soll.
ignore() Score:%< 0
=kill()  Score:%<=-100

# Beispiele für IMAP-Konten:
# ==========================
# Setzt die  Empfängerliste auf das Konto »john_imap« und legt die E-Mail in der
# »INBOX« des Kontos ab:
set(john_imap) To: "john@example.com"
#
# Wie zuvor, legt die E-Mail aber in dem IMAP-Ordner »privat« ab:
set(john_imap/privat) To: "john@example.com"
#
# Fügt als zusätzlichen Empfänger den Ordner »john/privat« des Kontos »archiv_imap«
# hinzu und setzt sowohl das »/seen«- als auch das »/flagged«-Flag:
add(archiv_imap/john/privat:\seen\flagged) To: "john@example.com"
```

MAlias.hst

Diese Datei enthält eine Tabelle von virtuellen oder umzuleitenden E-Mail-Adressen. Alle Umleitungen bzw. virtuellen Adressen sind zeilenweise aufzuführen. Am Zeilenbeginn muss die umzuleitende bzw. die virtuelle Adresse stehen. Der umzuleitenden bzw. virtuellen E-Mail-Adresse (dem Alias) folgt dann, getrennt durch ein Gleichheitszeichen, die E-Mail-Adresse, auf

welche die E-Mail umzuleiten ist, beziehungsweise die E-Mail-Adresse, welche die E-Mails für den virtuellen Account erhalten soll.

Mit der Raute oder dem Semikolon am Zeilenbeginn können Kommentare eingefügt werden.

Beispiele:

```
# Alle am lokalen Server eingehenden E-Mails, welche an
# »meine.virtuelle@Adresse.invalid« gerichtet sind,
# werden an den Admin weitergeleitet:
meine.virtuelle@Adresse.invalid=admin

# Alle am Account »admin« eingehenden E-Mails werden
# an »realnamen@domain.example.org« weitergeleitet:
admin=realname@domain.example.org

# Alle an den Postmaster gerichteten Mails werden an den Admin weitergeleitet:
postmaster=admin
```

Hinweise:

- Sie können die Datei auch über das Menü »Einstellungen« → »Benutzerverwaltung & Passworte« ⇒ »Aliase für Benutzer« verwalten.
- Alle Adressen rechts vom Gleichheitszeichen dürfen sowohl lokale als auch entfernte Adressen sein. Entfernte Adressen werden vom internen Mailrouter des Hamsters im Mailausgang abgelegt und beim nächsten Versenden von E-Mails mit versandt.
- Um an umgeleitete Adressen korrekt versenden zu können, ist es erforderlich, dass im Menü »Einstellungen« → »Lokale Server« ⇒ »Allgemeines« in »Envelope-From: Absenderadresse für automatisch generierte E-Mails:« eine Absenderadresse eingetragen ist. Diese Adresse wird im »Envelope-From:«- und im »Sender:«-Header der umgeleiteten E-Mail verwendet.

  Glossar: Header
- Die E-Mail-Adressen dürfen nur aus dem Namen, dem Symbol »@« und einer Domain bestehen. Kommentare und der ausgeschriebene volle Name dürfen hierin nicht enthalten sein.
- Der Domain-Part der E-Mail-Adresse kann auch die Wildcard »%FQDN%« enthalten, welche dem eingestellten FQDN entspricht. Sofern kein FQDN vergeben ist, werden die entsprechenden Mailalias-Zeilen ignoriert.

  FQDN
- Eine Mehrfachverschachtelung der Aliasse ist bis zu 99-fach möglich (vgl. auch die Verschachtelung in den Standardeinträgen). Es sollte aber darauf geachtet werden, dass keine Ringstrukturen von Alias-Verweisen entstehen. Ggf. werden diese eventuell vorhandenen Ringstrukturen durch den Mailrouter bei einer Verschachtelungstiefe von 99 bearbeiteten Alias-Verweisen aufgebrochen.
- Wird beim Starten des Hamsters die Datei »`MAlias.hst`« nicht gefunden, legt der Hamster sie automatisch mit den folgenden Einträgen an:

```
news=admin
abuse=admin
usenet=admin
postmaster=admin
news@%FQDN%=news
abuse@%FQDN%=abuse
usenet@%FQDN%=usenet
postmaster@%FQDN%=postmaster
local-hamster-info@%FQDN%=admin
local-smtp@%FQDN%=admin
```

*.cnv (Zeichensatzkonvertierungsdateien)

Diese Dateien enthalten die Konvertierungstabellen, um die Zeichensätze von News-Artikeln und E-Mails in den lokalen 8-Bit-Windows-Zeichensatz zu konvertieren. Dieses ist insbesondere im Mailfilter und im News-Scorefile erforderlich, wenn vom ASCII-Zeichensatz abweichende Zeichensätze im Header verwendet werden.

Glossar: Header

Beispiel: `From: =?iso-8859-1?Q?=E4rger?= dich nicht <irgend.ein@spammer.invalid>`

Bei diesem Beispiel ist es wegen der »Quoted-printable«-Codierung nicht ohne weiteres möglich, auf das Wort »Ärger« zu filtern. Deshalb muss dieser Header vor dem eigentlichen Filtern decodiert werden. In den Filtern geschieht dieses mit dem Tilde-Symbol (»~«). Damit der Decoder nun die Zeichensätze der Artikel bzw. E-Mails in den lokalen Zeichensatz umsetzen kann, sind Umsetzungstabellen erforderlich. Diese Konvertierungs-Tabellen müssen für den Hamster folgenden Aufbau besitzen:

Quoted-printable

Dateiname:

`cs_⟨ZeichensatzA⟩_⟨ZeichensatzL⟩.cnv`

Hierbei ist »⟨ZeichensatzA⟩« der Zeichensatz in den Artikeln bzw. E-Mails und »⟨ZeichensatzL⟩« der lokale Zeichensatz. In Westeuropa ist auf Windows-Systemen der Zeichensatz »windows-1252« als lokaler Zeichensatz üblich. In den Artikeln bzw. E-Mails sind hauptsächlich die Zeichensätze »ISO-8859-1«, »ISO-8859-15« und »UTF-8« (als Unicode-Transferformat) üblich.

Beispiel: `cs_ISO-8859-15_WINDOWS-1252.cnv`
Diese Datei ist für die Konvertierung des »ISO-8859-15«-Zeichensatzes in den Zeichensatz »windows-1252« zuständig.

Welcher Zeichensatz als lokaler Zeichensatz verwendet werden soll, kann im Menü »Einstellungen« → »Grundeinstellungen« ⇒ »Verschiedenes« ausgewählt bzw. in der Datei »`hamster.ini`« im Abschnitt [Setup] mit der Variablen »local.codepage=⟨Zeichensatz⟩« eingestellt werden.

Dateiaufbau:

Die Konvertierungstabellen müssen folgender Regel entsprechen:

ZeichenA=ZeichenL

wobei ZeichenA = zu konvertierendes Zeichen in dezimaler Schreibweise
ZeichenL = lokales Zeichen in dezimaler Schreibweise

Wird vom Decoder für ein Zeichen kein Wertepaar gefunden, so wird es unverändert belassen. Einige Tabellen sind auch leer, da der Ausgangszeichensatz eine Untermenge des Zielzeichensatzes ist.

Beispiel:

cs_ISO-8859-15_WINDOWS-1252.cnv	*Zeichendarstellung*
164=128	€
166=138	Š
168=154	š
180=142	Ž
184=158	ž
188=140	Œ
189=156	œ
190=159	Ÿ

Diese Tabelle konvertiert den Zeichensatz »ISO-8859-15« in Artikeln und E-Mails in den Zeichensatz »windows-1252«.

Hinweis: Zum Testen einer Tabelle eignet sich die »hs2«-Funktion »DecodeMIMEHeaderString«. Mit den »hs2«-Funktionen »UTF8toUCS32« und »UTF7toUCS16« kann die UTF-Codierung einer Zeichenkette entfernt werden.

Password.!!!

In dieser Datei werden die Passwörter gespeichert.

Die Einträge in der Datei sind mit dem »Blowfish«-Algorithmus verschlüsselt. Der dafür verwendete kryptografische Schlüssel beinhaltet – je nach Einstellung unter »Einstellungen« → »Benutzerverwaltung & Passworte« ⇒ »Passworte« – u. a. einige Systemangaben (PC-Name, Betriebssystem-Version usw.) und den Anmeldenamen.

Dies erschwert zwar die Nutzung einer gestohlenen Passwort-Datei, aber die allgemeine Annahme, dass die Verschlüsselung einer mit diesem Algorithmus codierten Datei 100%ig sicher sei, *ist falsch*. Richtig ist aber, dass dieses Verfahren wesentlich sicherer ist als eine Speicherung im Klartext bzw. in mit ROT13 oder XOR verschleierten Dateien.

Überlassen Sie diese Datei niemals anderen Personen – auch dann nicht, wenn Sie Probleme mit Ihren Zugängen oder ähnliche Probleme haben!

Dateien in »⟨Hauptverzeichnis⟩\Server«

***Hinweis:** Der Pfad zum Verzeichnis ist änderbar!*

`alldescs.txt`
: Beschreibung der News(gruppen).

  Format: Gruppenname ⟨TAB⟩ Gruppenbeschreibung

`allpulls.txt`
: Falls diese Datei vorhanden ist, kann sie gelöscht werden. Die Datei wird nicht mehr verwendet.

Dateien in »⟨Hauptverzeichnis⟩\Server\⟨Servername⟩«

`Server.ini`
: Grundeinstellungen des Servers.

`Greeting.txt`
: Grußmitteilung des Servers.

`Groups.txt`
: Gruppenliste des Servers.

`Help.txt`
: Hilfedatei des Servers.

`GrpDescs.txt`
: Liste der Gruppenbeschreibungen.

`Overview.txt`
: Overview-Felder (»XOver«) des Servers.

`GetMids.txt`
: Liste der Message-IDs, die aus dem Killfile heraus nachgeladen werden sollen. Diese Datei wird beim nächsten Online-Holen abgearbeitet und anschließend gelöscht. In jeder Zeile steht eine Message-ID; sofern ein »!« vor der Message-ID ist, wird der Artikel auch dann geholt, wenn er dem Hamster an sich bereits bekannt ist (History).

Server.ini

Diese INI-Datei enthält je nach Server-Typ verschiedene Einstellungen, die nur teilweise direkt über die Dialoge geändert werden können:

```
[Setup]

; News-, POP3- und SMTP-Server:

; 1 = Server ist deaktiviert,
; 0 = Server ist aktiviert (Standard: 0).
Disabled=0

; Timeout für die Remote-Server, d. h. nach welcher Zeitspanne in Sekunden soll der
; Hamster Verbindungs-/Abfrageversuche zu externen Servern als gescheitert betrachten
; (Standard: 120 s).
remote.timeout.connect=120
remote.timeout.command=120

; Nur Newsserver:

;   1 = Wenn nur Rechte zum Holen von Artikeln auf dem Server vorhanden sind,
;   0 = Wenn auch schreibender Zugriff auf den Server möglich ist (Standard: 0).
ReadOnly=0


[NNTP]

; Nur Newsserver (NNTP):

; einige Einstellungen für SSL, siehe die Erläuterungen zum entsprechenden Menü
; unter »Einstellungen« → »News: Server, Gruppen & Ladeaufträge« ⇒ »Newsserver«
;---------------
; Modus der Nutzung:
; 0 = kein SSL/TLS (Standard), 1 = SSL/TLS bei sicherem Port immer benutzen,
; 2 = SSL/TLS auf Standardport nutzen, wenn möglich, 3 = SSL/TLS auf Standardport
;  immer verwenden:
SSLMode=3
;---------------
; Verfahren zur Überprüfung der X.509-Server-Zertifikate;
; 0 = keine Überprüfung (Standard), 1 = Zertifikat überprüfen, wenn vorhanden,
; 2 = Zertifikat immer überprüfen, 3 = zusätzlich Serverzertifikat lokal überprüfen:
SSLVerifyLevel=3
;---------------
; Für die lokale Prüfung des Server-Zertifikates können Sie eine Datei mit einem
; Zertifikat im PEM-Format angeben:
SSLCaFile=⟨Pfad zur Datei mit dem Server-Zertifikat⟩

; Eigene Beiträge in ein eigenes Verzeichnis ablegen lassen,
; 0: nein (Standard), 1: ja.
UsePostDir=0
;---------------
; Angabe das Verzeichnisses für die eigenen Beiträge:
PostDir=
```

Glossar: POP3 SMTP NNTP SSL TLS X.509 PEM

```
; Servertyp einstellen und somit mögliche Einschränkungen definieren,
; Typ = 100: Vollwertiger NNTP-Server (Standard), Typ = 10: Kein Abholen von News
; möglich, nur Gruppenliste, Typ = 0: Pseudo-Server.
Servertype=100

[Newgroups]

; Nur Newsserver:

; 1 = 4-stellige-Jahresangabe im »NEWGROUPS«-Kommando,
; 0 = 2-stellige-Jahresangabe im »NEWGROUPS«-Kommando (Standard ist 0).
4-Digit-Years=0

; »LastGMT« speichert den Zeitpunkt, an dem die Hamster-Gruppenliste zum letzten Mal
; mit der des Server abgeglichen wurde. Zum Deaktivieren des automatischen Abgleichs
; kann »NEVER« verwendet werden.
LastGMT=NEVER

; Legt fest, ob die Beschreibungen für neue Gruppen sofort geladen werden. Beim Stan-
; dardwert »0« versucht Hamster zu erkennen, ob das verwendete Kommando vom Newsserver
; unterstützt wird. Falls nicht, wird der Wert auf »2« gesetzt, was das Laden der
; Gruppenbeschreibungen deaktiviert. Bei einem Wert von »1« versucht der Hamster, die
; Beschreibungen jedes Mal zu laden.
LoadDescs=0

[Pull]

; Nur Newsserver:

; Anzahl der gleichzeitig zu startenden Aufträge für den Server.
; Hinweis: Bitte die Vorschriften des Servers diesbezüglich beachten und den Admin des
; Servers fragen, wie viel Threads zulässig sind (Wertebereich 0-4).
Threads=1

[POP3]                                                    Glossar: POP3

; Nur POP3-Mailserver:

; einige SSL-Einstellungen; Erläuterungen dazu im Abschnitt „[NNTP]“
SSLMode=3
SSLVerifyLevel=3
SSLCaFile=<Pfad zur Datei mit dem Server-Zertifikat>

; UseTopCommand=0 schaltet den »TOP«-Befehl ab. Standard ist 1.
UseTOPCommand=1

; Erzwingt ein Authentifizierungsverfahren. Mögliche Werte sind PASS, APOP, AUTH,
; SASL, und NONE. Aus Sicherheitsgründen sollte, wenn vom Server unterstützt, mög-      SASL
; lichst APOP, AUTH oder SASL verwendet werden.
ForcedAuthMech=none
```

```
; Definiert die Liste der unterstützten SASL-Mechanismen, getrennt durch Leerzeichen
; für die Authentifizierung beim Mailserver.
; Unterstützte SASL-Mechanismen: LOGIN, PLAIN, CRAM-MD5, CRAM-SHA1, DIGEST-MD5.
SASL=

; Maximale Anzahl der Mails, welche in einem Durchgang vom Mailserver geladen werden.
; Der Standardwert ist 0 und bedeutet, dass alle Mails in einem Durchgang vom Server
; geladen werden.
GetMaxMailsPerSession=0

; 1 = Hamster löscht E-Mails nicht auf dem Server,
; 0 = Hamster löscht E-Mails auf dem Server,
; ? = Hamster richtet sich nach der allgemeinen Einstellung.
LeaveOnServer=0

; Eingehende E-Mails werden standardmäßig an diesen lokalen Benutzer geschickt:
LocalUser=admin

; Filterabschnitt, der für diesen Server in der Datei »Mailfilt.hst« benutzt wird:
FilterSection=*


[SMTP]

; Nur SMTP-Mailserver:

; AUTH=1 aktiviert das SMTP-AUTH-Verfahren für den SMTP-Server (Standard 0).
AUTH=0

; Definiert die Liste der unterstützten SASL-Mechanismen, getrennt durch Leerzeichen,
; für die Authentifizierung beim Mailserver.
; Unterstützte SASL-Mechanismen: LOGIN, PLAIN, CRAM-MD5, CRAM-SHA1, DIGEST-MD5.
SASL=

; einige SSL-Einstellungen; Erläuterungen dazu im Abschnitt „[NNTP]“
SSLMode=3
SSLVerifyLevel=3
SSLCaFile=<Pfad zur Datei mit dem Server-Zertifikat>


[Statistic]

; Wird für alle externen Server (NNTP, POP3, SMTP) geführt, erfolgreich ist eine
; Einwahl dann, wenn die Authentifizierung komplett erfolgreich war, ein schlichtes
; Erreichen des Servers reicht nicht.

; Dauerzähler für erfolgreiche Verbindungen.
Connections.ok=

; Dauerzähler für gescheiterte Verbindungen.
Connections.failed=
```

Glossar: SMTP

SMTP-AUTH

SASL

NNTP
POP3

```
; Datum der letzten erfolgreichen Verbindung.
LastConnect.ok=

; Datum der letzten gescheiterten Verbindung.
LastConnect.failed=

; Falls letzte Verbindung erfolgreich: Anzahl der erfolgreichen Versuche am Stück,
; sonst 0.
LastConnect.Count.ok=

; Falls letzte Verbindung gescheitert: Anzahl der gescheiterten Versuche am Stück,
; sonst 0.
LastConnect.Count.failed=
```

Dateien in »⟨Hauptverzeichnis⟩\Groups«

***Hinweis:** Der Pfad zum Verzeichnis ist änderbar!*

`History.dat`
: Falls diese Datei vorhanden ist, kann sie gelöscht werden. Die Datei wird nicht mehr verwendet.

`Hist*.dat`
: Liste mit Artikel-Informationen (MD5-Hash der Message-ID).

`Kills.log`
: Übersicht aller wegen des Scorefiles nicht geladenen Artikel.

News.out

`*.msg`
: Artikel, die zum NNTP-Server des Providers gesendet werden sollen.

Glossar: NNTP

(*Hinweis:* Die Dateiendung kann in der »`hamster.ini`« geändert werden.)

News.err

`*.err`
: Artikel, die zum NNTP-Server des Providers gesendet werden sollten, aber auf Grund eines Fehlers nicht versandt werden konnten.

(*Hinweis:* Die Dateiendung kann in der »`hamster.ini`« geändert werden.)

Dateien in »⟨Hauptverzeichnis⟩\Groups\⟨Newsgruppenname⟩«

data.dat
: Artikel-Datenbank.

data.idx
: Index der Artikel-Datenbank.

data.ini
: Gruppenspezifische Daten und Einstellungen.

  Die hierin enthaltenen Einstellungen sollten nach Möglichkeit nicht geändert werden oder nur, wenn man genau weiß, was man tut:

```
[Setup]

; Enthält die E-Mail-Adresse der Moderation, wenn diese Gruppe moderiert ist,
; oder enthält die E-Mail-Adresse einer Mailingliste, wenn diese Gruppe ein News-
; to-Mail-Gateway ist.
Moderator=

; Enthält den Typ der Gruppe:
;   g = News-to-Mail-Gateway,
;   n = readonly (nur Lesen erlaubt),
;   m = moderiert,
;   y = normale Gruppe.
Type=

; Artikel dieser Gruppe nicht zum Provider posten, sondern zu anderen Servern
; feeden; 0 = posten (Standard), 1 = feeden.
feedonly=

; Angabe eines individuellen Postservers für die Gruppe:
postserver=

; Diese Einstellung überschreibt, wenn angegeben, die entsprechende globale
; Einstellung (Menü »Einstellungen« → »News: Server, ...« ⇒ »Einstellungen«,
; Eintrag »Artikel-Limit (Standard)« bzw. »pull.limit« in der »hamster.ini«,
; Sektion [Setup]).
; Mögliche Werte:
;   0 = kein Limit, alles laden,
;   n (positiver Wert) = die n jüngsten Artikel laden (d. h. die neuen Artikel
;                        mit den höchsten Index-Nummern),
;   n (negativer Wert) = die n ältesten Artikel laden (d. h. die neuen Artikel
;                        mit den niedrigsten Index-Nummern).
pull.limit=

; Die Haltezeit in Tagen seit dem Pullen für diese Gruppe.
; Standardwert: »hamster.ini«, [Setup] → »purge.articles.keepdays=0«
purge.articles.keepdays=
```

```
; Maximalzahl enthaltener Postings nach dem Purgen/Aufräumen für diese Gruppe.
; Standardwert: der Wert von »purge.articles.keepcount« in der »hamster.ini«,
; Sektion [Setup].
purge.articles.keepcount=

[Info]

; Kurzbeschreibung der Newsgruppe.
; Standardmäßig wird hier die in der Gruppenliste mitgelieferte Beschreibung
; eingetragen. Diese kann hier geändert werden und erscheint dann so auch in
; der Gruppenliste Ihres Newsreaders.
description=

; Datum und Uhrzeit, wann die Gruppe im Hamster angelegt wurde,
; Format: YYYYMMTTHHMMSS.
Created=

; Datum und Uhrzeit des letzten Abrufs durch ein Clientprogramm,
; Format: YYYYMMTTHHMMSS.
LastClientRead=

[Ranges]

; Enthalten die niedrigste (min) und höchste (max) verfügbare Artikelnummer im
; Hamster.
; Achtung! Änderungen dieser Einstellungen werden sehr wahrscheinlich Probleme
; im Hamster und/oder den Newsreadern von Hamsternutzern verursachen.
; Nicht mit den Einstellungen der Min-/Max-Werte für die externen Server verwech-
; seln, wie sie im übernächsten Punkt beschrieben werden.
Local.Min=
Local.Max=

; Wurden vom externen Newsserver bereits Artikel geladen? 0 = nein, 1 = ja.
⟨Server⟩.FirstPullDone=

; Enthält die Index-Nummer des beim letzten Abruf ältesten (low) und jüngsten
; (max) auf dem externen Newsserver verfügbaren Artikels sowie die Nummer des
; jüngsten geladenen Artikels (min).
⟨Server⟩.Low=
⟨Server⟩.Min=
⟨Server⟩.Max=
; Hamster benutzt diese Index-Nummern, um bei der Abfrage, ob ein Newsserver neue
; Artikel hat, nicht immer wieder den gesamten Artikelbestand des Newsservers
; durchsuchen zu müssen.
; Wird ein Newsserver zum ersten Mal angesprochen, fragt Hamster nach der höch-
; sten Indexnummer und zählt um den Wert von »pull.limit.first« zurück.
; Wenn der Eintrag »⟨Server⟩.Min« manuell verändert wird, richtet sich Hamster
; bei der nächsten Verbindung nach den geänderten Werten. Über diese Änderung
; können ältere Artikel nachgeladen werden.
; Bedenken Sie, dass durch diese Änderung bei Servern mit langer Haltezeit sehr
; lange Ladezeiten und ein Volumen über die Grenzen der Festplatte hinaus verbun-
; den sein können.
```

Dateien in »⟨Hauptverzeichnis⟩\Mails«

Hinweis:* *Der Pfad zum Verzeichnis ist änderbar!

`MHistory.dat`
: Mail-History.

Dateien in »⟨Hauptverzeichnis⟩\Mails\Mail.out«

`*.msg`
: E-Mails, welche an den SMTP-Server des Providers gesendet werden sollen.

Glossar: SMTP

`*.err`
: E-Mails, welche an den SMTP-Server des Providers gesendet werden sollten, aber auf Grund eines Fehlers nicht versandt werden konnten.

(*Hinweis:* Die Dateiendung kann in der »`hamster.ini`« geändert werden.)

Weitere manuell abgelegte Dateien mit anderer Endung in diesem Ordner werden vom Hamster ignoriert.

Dateien in »⟨Hauptverzeichnis⟩\Mails\admin«

`*.msg`
: E-Mails, die vom POP3-Server des Providers geholt und keinem normalen Benutzer zugeordnet wurden.

POP3

(*Hinweis:* Die Dateiendung kann in der »`hamster.ini`« geändert werden.)

Weitere manuell abgelegte Dateien mit anderer Endung in diesem Ordner werden vom Hamster ignoriert.

Dateien in »⟨Hauptverzeichnis⟩\Mails\⟨Benutzer⟩«

Das Hamster-Format einer E-Mail ist eine Einzeldatei, welche den Inhalt als ASCII-Text im Rohformat enthält.

`*.msg`
: E-Mails, die vom POP3-Server des Providers geholt und dem betreffenden Benutzer zugeordnet wurden.

(*Hinweis:* Die Dateiendung kann in der »`hamster.ini`« geändert werden.)

Glossar: POP3

Besonderheiten – POP3-Konto

Benennung:

Beliebig.

Der Hamster erzeugt den Namen aus einer fortlaufenden Numerierung, welche alle POP3-Konten und den Postausgang umfasst.

Weitere Dateien:

Keine. Manuell abgelegte Dateien mit einer anderen Endung werden vom Hamster ignoriert.

Besonderheiten – IMAP-Konto IMAP

Benennung:

⟨nnnn⟩`.msg`, wobei ⟨nnnn⟩ eine Zahl darstellt:

Der Name **muss** aus einer Zahl bestehen. Führende Nullen sind dabei nicht erlaubt.

Der Name/Die Zahl einer E-Mail-Datei entspricht der IMAP-ID in der Datenbank. Daher dürfen in einem IMAP-Verzeichnis E-Mails weder erzeugt, gelöscht noch umbenannt werden.
Ausnahme: Der User setzt gleichzeitig die Datenbank zurück. Dazu muss der User – bei heruntergefahrenem IMAP-Server – die Dateien »`Mails.dat`« und »`Status.dat`« im selben Verzeichnis löschen. *Dabei gehen jedoch auch alle auf dem Server gespeicherten Flags verloren!*

Weitere Dateien:

`Mails.dat`
: Speichert e-mail-bezogene Informationen.

`Status.dat`
: Speichert Informationen zum IMAP-Ordner.

`SUBSCRIBED.DAT`
: Die Datei dient als Flag, ob der IMAP-Ordner von einem Client abonniert wurde.

`NOSELECT`
: Die Datei dient als Flag, um einen IMAP-Ordner, der aus systeminternen Gründen nicht entfernt werden kann, als gelöscht zu markieren. Sie wird in der Regel nur verwendet, wenn ein zu löschender Ordner weitere nicht gelöschte Ordner enthält. Wird der Ordner von einem Client neu angelegt, wird die Datei automatisch entfernt.

Die Flag-Dateien sind 0 Byte groß, da sie ansonsten keine Daten enthalten.

Hinweise zur Fehlerbeseitigung:

- Treten im Client falsche Anzeigen über die Gesamtzahl der (ungelesenen) E-Mails auf, reicht es, die »`Status.dat`« zu löschen.
- Reicht das nicht zur Fehlerbeseitigung, oder fehlen einzelne E-Mails in der Anzeige des Clients, sollten beide Dateien »`Mails.dat`« und »`Status.dat`« gelöscht werden. *Dabei gehen auch alle auf dem Server gespeicherten Flags verloren!*
- Existiert eine »`NOSELECT`«-Datei ohne regulären Grund, kann eine nicht dem Hamster zugeordnete Datei (»`*.bak`« oder ähnliches) die Ursache sein. In jedem Fall darf das Verzeichnis manuell entfernt werden (Inhalt der Dateien ggf. vorher mit einem Texteditor überprüfen). Wird nur die »`NOSELECT`«-Datei gelöscht, ist der Ordner anschließend im Client wieder sichtbar.

Grundsätzlich dürfen Veränderungen an jeglichen Dateien nur bei heruntergefahrenem Server erfolgen.

Dateien in »⟨Hauptverzeichnis⟩\Logs«

***Hinweis:** Der Pfad zum Verzeichnis ist änderbar!*

`0.log, 1.log, 2.log, ...`
: Hamster-Protokoll. Der Dateiname kann per Eintrag in die »`hamster.ini`«, Sektion »[Setup]« geändert werden, siehe »log.FormatFileName«.

`MailIn.log`
: Protokoll über die vom POP3-Server geholten E-Mails.

Glossar: POP3

`MailOut.log`
: Protokoll über die zum SMTP-Server gesendeten E-Mails.

SMTP

`RasDial.log`
: Protokoll über die vom Hamster hergestellten DFÜ-Netzwerkverbindungen.

`ShellHis.log`
: History der zuletzt eingegebenen Befehle in der Eingabezeile für hs2 im Hauptfenster des Hamsters.

Skripte und Programmierung

Hamsterskripte (»hs2«)

Ein **Hamsterskript** (oder einfach **Skript**) enthält eine Folge von Anweisungen, die nach dem Starten des Skripts abgearbeitet werden, wodurch häufig benötigte Arbeitsgänge automatisiert ausgeführt werden können. Das erspart nicht nur immer wiederkehrende manuelle Aufrufe von Menüs und die Bearbeitung der zugehörigen Dialoge, viele Abläufe lassen sich überhaupt nur skriptgesteuert realisieren und auf individuelle Bedürfnisse zuschneiden. Skripte erlauben einfache Interaktionen mit dem Benutzer.

Skripte sind in Textdateien zu speichern, wie sie beispielsweise der Windows-Editor »Notepad« erzeugen und bearbeiten kann. Der Dateiname ist bei Hamsterskripten mit der Erweiterung »`hsc`«, bei Modulen mit der Erweiterung »`hsm`« zu versehen. (Vor allem) Skriptdateien sind im Skripte-Verzeichnis, Moduldateien im Module-Verzeichnis des Hamsters zu speichern; die Vorgabe für beide ist das Hamster-Hauptverzeichnis. In der Sektion »[Directories]« der Datei »`hamster.ini`« können Sie mit den Schlüsseln »Scripts« und »Modules« andere Pfade angeben.

Die hier beschriebene Sprache Hamsterscript, Version 2, wurde erst mit dem Hamster 1.3.17.0 eingeführt. Sie werden daher in Skriptsammlungen oder archivierten Usenet-Postings in den meisten Skripten als Einleitung die Präprozessor-Anweisung »**#!hs2**« in einer separaten Zeile finden, da frühere Hamsterversionen anderenfalls von einem Quelltext der weniger mächtigen Vorläufer-Skriptsprache (Hamsterscript, Version 1) ausgingen, die der Hamster inzwischen aber nicht mehr unterstützt.

Die folgenden Abschnitte beschreiben die Sprache der Hamsterskripte:

- Anweisungen und Kommentare
  - Präprozessor-Anweisungen
- Variablen, Konstanten und Datentypen
- Ausdrücke und Operatoren
  - Boolesche Ausdrücke
- Sprünge
- Verzweigungen
- Schleifen
- Prozeduren und Funktionen
  - Eigene Prozeduren und Funktionen
  - Eingebaute Prozeduren
  - Eingebaute Funktionen (allgemein)
  - Eingebaute Funktionen (Server)
- Nebenläufige Skripte
  - Synchronisation
  - Synchronisation durch Events
- Module
- Aufruf von Skripten per Action
- Aufruf von Skripten per Kommandozeile: Siehe Kommandozeilenoptionen des Hamsters

Anweisungen und Kommentare

Anweisungen

Ein Skript für den Interpreter des Hamsters besteht aus einer Abfolge von Anweisungen (Befehlen). Sofern **Kontrollstrukturen** wie Sprünge, Verzweigungen oder Schleifen nichts anderes vorgeben, werden die Anweisungen in der Reihenfolge ihres Auftretens im **Quelltext** ausgeführt. Die Quelltexte sind in Textdateien zu speichern, wie sie beispielsweise der Windows-Editor »Notepad« erzeugen und bearbeiten kann. Der Dateiname ist bei Hamsterskripten mit der Erweiterung »hsc«, bei Modulen mit der Erweiterung »hsm« zu versehen.

Jede Anweisung steht für sich allein in ihrer Anweisungszeile, es bedarf daher keines abschließenden Zeichens. Eine Anweisungszeile kann beliebig lang sein. Beispiel:

```
print( uppercase( "Das ist eine einzelne Zeile" ) )
```

Eine Anweisung kann sich auch über mehrere Textzeilen erstrecken, die an ihrem Ende durch einen Unterstrich (»_«) zu einer einzigen Anweisungszeile (zur »logischen Zeile«) verbunden werden. Der Unterstrich muss durch ein Leerzeichen von den vorangehenden Zeichen getrennt sein:

```
print( _
uppercase( "Das ist eine einzelne Zeile" ) _
)
```

Innerhalb eines Literals (im Beispiel: »Das ist eine einzelne Zeile«) kann eine Anweisungszeile allerdings nicht umbrochen werden.

Die Anweisungen eines Skriptes werden ausgeführt, bis der Interpreter auf eine »**quit**«-Anweisung stößt. Optional kann die »**quit()**«-Anweisung einen in Klammern einzuschließenden Zahlenwert zurückgeben.

Kommentare

Kommentare dienen allein der Orientierung des Entwicklers oder anderer menschlicher Leser des Quelltextes und werden vom Interpreter ignoriert.

Ein Kommentar wird durch die Raute »#« eingeleitet, der ihr folgende Text wird vom Interpreter nicht beachtet:

```
print( "SCHNELL REICH WERDEN" ) # und noch schneller wieder arm!
```

Ausgeführt wird hier nur die »print()«-Anweisung, der Kommentar hinter »#« hat keinerlei Einfluss. Ein Kommentar kann auch allein in einer Zeile stehen.

Keine Kommentare sind die Präprozessor-Anweisungen, die durch eine Raute und ein unmittelbar folgendes Ausrufezeichen eingeleitet werden (»**#!**«). Um Überschneidungen mit bereits implementierten oder möglichen zukünftigen Präprozessor-Kommandos zu vermeiden, darf daher ein Kommentar nach der Raute nicht mit einem Ausrufzeichen beginnen.

Kommentare in geschweiften Klammern »{}« können auch innerhalb einer Anweisung stehen, so wie hier in einer über mehrere Textzeilen reichenden Anweisung:

```
print( _ { Kommentar in einer umbrochenen Anweisungszeile }
uppercase( "Das ist eine einzelne Zeile" ) _
)
```

In Literalen (im Beispiel: »Das ist eine einzelne Zeile«) kann grundsätzlich nicht kommentiert werden, alle Zeichen des vermeintlichen Kommentars würden einschließlich der Klammern oder der Raute zum Literal selbst gezählt.

Präprozessor-Anweisungen

Präprozessor-Anweisungen sind Spezialbefehle, die vor der eigentlichen Abarbeitung des Skripts ausgeführt werden. Diese Spezialbefehle gehören immer an den Anfang eines Skripts und müssen allein in einer Zeile stehen.

`#!hs2` kennzeichnet die Verwendung der neuen Skriptsprache. Die alte Skriptsprache wird nicht mehr unterstützt. Diese Zeile muss ohne führende Leerzeichen und ohne folgende Parameter in der allerersten Zeile stehen, um wirksam zu sein.

Achtung: *Obwohl die Skripte derzeit auch ohne diese Anweisung funktionieren, sollten Sie sie an den Beginn der Skripte setzen, um mögliche Konflikte bei eventuellen Änderungen in der Zukunft zu vermeiden.*

`#!trace 1` schaltet die Anzeige aller ausgeführten Programmzeilen im Hamster-Protokoll ein, um **Tests** und das **Debuggen** (»Entwanzen«, Suchen und Entfernen von Programmierfehlern) von Skripten zu erleichtern. Siehe hierzu auch die hs2-Prozedur »trace«.

Dieser Befehl eignet sich insbesondere zum Anzeigen des Initialisierungs-Abschnittes von Modulen.

`#!debug ⟨level⟩` setzt den Debug-Level, Standard ist 0. Mit dem Debug-Level 0 werden alle »debug«-Befehle ignoriert. Der Debug-Level 255 zeigt alle »debug«-Befehle an.

`#!load ⟨module⟩.hsm` lädt das betreffende Modul und erlaubt die Benutzung der dort definierten Funktionen, siehe dazu unter Module.

`#!initialize` kennzeichnet den Beginn eines Initialisierungsabschnittes, der stets zuerst ausgeführt wird. Kann in jedem Skript verwendet werden, ist aber besonders sinnvoll in Modulen, mehr siehe deshalb dort. Darf ohne »#!finalize« verwendet werden. Außerhalb von Modulen *muss* der Initialisierungsabschnitt *nach* dem Hauptskript stehen und nach der »return«-Anweisung zusätzlich mit »quit« abgeschlossen werden.

`#!finalize` kennzeichnet den Beginn eines Finalisierungsabschnittes, der stets als letztes ausgeführt wird. Kann ebenso in jedem Skript verwendet werden, ist aber besonders sinnvoll in Modulen, mehr siehe deshalb dort. Darf *nicht ohne* »#!initialize« verwendet werden und

muss *nach* diesem auftreten, sonst gibt es in beiden Fällen eine Fehlermeldung. Außerhalb von Modulen ist der Finalisierungsabschnitt zusätzlich mit »quit« abzuschließen.

`#!menu:⟨hauptmenüpunkt⟩=⟨Menübezeichnung⟩`: Sofern das Skript im Skriptpfad bzw. einem Unterpfad gespeichert ist, wird es mit der gewählten Menübezeichnung unter dem ausgewählten Hauptmenüpunkt in das bestehende Hamster-Menü eingefügt und kann auf diese Art bequem gestartet werden. Als Hauptmenüpunkt stehen zur Verfügung: »***File***« (Menü »Datei«), »***Config***« (Menü »Einstellungen«), »***Online***« (Menü »Online«), »***Local***« (Menü »Lokale Server«), »***Tools***« (Menü »Tools«) und »***Help***« (Menü »Hilfe«). Sofern sich kein Skript für den Menüpunkt »Tools« per obiger Anweisung registriert, wird dieser Hauptmenüpunkt nicht angezeigt.

Variablen, Konstanten und Datentypen

Die Abarbeitung von Skripten dient der Verarbeitung von **Daten**. Diese werden während der Ausführung von Skripten in **Variablen** gespeichert und durch ihren **Wert** repräsentiert. Jeder Wert hat einen bestimmten **Datentyp**, der die auf allen Werten dieses Typs zulässigen Operationen charakterisiert. Der Interpreter entnimmt dem Datentyp auch Art und Größe des für den Wert zu reservierenden Arbeitsspeichers.

Variablen

Variablen sind während ihrer Lebensdauer nicht fest an einen bestimmten Datentyp gebunden und haben immer den Typ des gerade in ihnen gespeicherten Werts. Sie sind zu **deklarieren**. Die Zeile

```
var( $MyVar, $MyVar2 )
```

deklariert beispielsweise die beiden Variablen `$MyVar` und `$MyVar2`. Bei selbstdefinierten Prozeduren und Funktionen kann die Deklaration auch einfach durch die Angabe als Parameter in der Definition erfolgen.

Das Gleichheitszeichen veranlasst eine Zuweisung. Die Zeile

```
$MyVar = ⟨expression⟩ # alternativ: set( $MyVar, ⟨expression⟩ )
```

weist der Variablen `$MyVar` den Wert des Ausdrucks `⟨expression⟩` zu. Ein Ausdruck ist ein Konstrukt, das einen Wert liefert. Die Variable hat dann den Typ dieses Wertes. In einem Skript deklarierte Variablen bleiben erhalten, solange das Skript läuft. Innerhalb eines selbst definierten Prozedurrumpfes deklarierte Variablen bleiben erhalten, solange die Prozedur läuft.

Das Konstrukt

```
var( $MyVar3 )
$MyVar3 = ⟨expression⟩
```

kann auch ersetzt werden durch folgende Funktion:

```
varset( $MyVar3, ⟨expression⟩ )
```

Hier wird die Variable $MyVar3 deklariert und ihr zugleich der Wert des Ausdrucks ⟨expression⟩ zugewiesen (**Definition**), dessen Wert die »**varset()**«-Funktion auch zurückgibt. Soll mehreren Variablen derselbe Ausdruck zugewiesen werden, so ist das möglich durch

```
varset( $MyVar4, $MyVar5, $MyVar6, ⟨expression⟩ )
```

Hier werden die drei Variablen $MyVar4 bis $MyVar6 deklariert und ihnen jeweils der Ausdruck ⟨expression⟩ zugewiesen.

Jeder **Variablenbezeichner** beginnt mit dem Dollarzeichen »$«, gefolgt von Groß- oder Kleinbuchstaben, Ziffern oder dem Unterstrich »_«. Dem Dollarzeichen muss ein Buchstabe folgen. Groß- und Kleinbuchstaben werden nicht unterschieden. Beispiele für **gültige Namen**:

```
$a
$i_____
$a_b
$Die_Antwort_ist_WIRKLICH_42
```

Beispiele für **ungültige Namen**:

```
a$
^^ »$« an der falschen Seite

$__i__
 ^ Kein Buchstabe als erstes Zeichen

$1st_cut
 ^ Kein Buchstabe als erstes Zeichen

$a.b
  ^ Punkt ist nicht zulässig

$Die_Antwort_ist_WIRKLICH_42!!!
                           ^^^ Ausrufezeichen sind unzulässig
```

Die Sichtbarkeit der Variablen kann durch die Anweisungen »`entercontext`« und »`leavecontext`« beeinflusst werden. Die Funktion »**entercontext**« eröffnet einen neuen Sichtbarkeitsbereich und die Funktion »**leavecontext**« schließt diesen wieder.

Beispiel:

```
#!hs2
var($a)
$a=42
entercontext("test")
var($a)
$a=999
print($a)
```

```
leavecontext
print($a)
quit
```

Konstanten

Konstanten sind während ihrer Lebensdauer fest an einen bestimmten Datentyp gebunden. Sie sind zu **deklarieren**. Die Zeile

```
const( $MyConst, ⟨expression⟩ )
```

deklariert beispielsweise die Konstante `$MyConst` und weist ihr den Wert ⟨expression⟩ zu. Die Zeile

```
ConstEnum( $MyConst1, $MyConst2 [ , $MyConst3 , ... , $MyConst100 ] )
```

deklariert Statuskonstanten und weist ihnen einen aufsteigenden Wert *beginnend mit 0* zu.

Beispiel:

```
ConstEnum( $Null, $Montag, $Dienstag, $Mittwoch, _
           $Donnerstag, $Freitag, $Samstag, $Sonntag )
# $Null erhält den Wert »0« und ist hier nur eingefügt,
# damit $Montag den Wert »1« erhält usw.
print ("Donnerstag ist der " + $Donnerstag + ". Tag in der Woche.")
```

Datentypen

Elementare Datentypen

Der Hamster kennt zwei elementare Datentypen: **vorzeichenbehaftete ganze Zahlen (Integer)** und **Zeichenketten (Strings)**. Die Funktionen »`isint()`« bzw. »`isstr()`« prüfen, ob der übergebene Ausdruck eine Zahl bzw. eine Zeichenkette liefert. In diesem Falle geben sie 1 zurück, sonst 0.

Beschreibung: isint / isstr

Strings können so lang werden, wie Arbeitsspeicher und Swapfile (Auslagerungsdatei) es zulassen. Sie enthalten beliebige Zeichen des **ASCII-Zeichensatzes**, das Zeichen **nul** hat hier keine terminierende Wirkung. Beim Größenvergleich etwa zum Zwecke der Sortierung sind die Ordinalwerte (siehe Funktion »`ord()`«) der Zeichen ausschlaggebend, die Windows-Ländereinstellungen werden nicht berücksichtigt. Die »`chr()`«-Funktion gibt zu einem Ordinalwert das zugehörige ASCII-Zeichen zurück.

ord / chr

Stringliterale werden in Anführungszeichen »"« eingeschlossen: »`"Hallo!"`«. Das Anführungszeichen selbst kann innerhalb von Strings durch doppelte Angabe dargestellt werden: »`"Hallo ""August"", wie geht's?"`« würde also als »Hallo "August", wie geht's?« ausgegeben. Das inhaltlose Stringliteral »""« ist der **Leerstring**.

Zahlentypen können Werte zwischen -2 147 483 648 (-2^{31}) und 2 147 483 647 ($2^{31} - 1$) darstellen.

Zahlenliterale sind **dezimal** ohne Tausenderpunkte oder **hexadezimal** anzugeben. Hexadezimale Literale sind durch »0x« (Ziffer Null, gefolgt vom Kleinbuchstaben x) einzuleiten: `0xFF` steht beispielsweise für den Dezimalwert 255. Dezimalen Literalen kann ein Vorzeichen + oder - vorangehen.

Wahrheitswerte (**Boolesche Werte**) werden durch ganze Zahlen dargestellt. Sie sind das Ergebnis von Ausdrücken oder Funktionen, die eine **Bedingung** prüfen (**Boolesche Ausdrücke**), wie etwa die schon erwähnten »**isint()**« oder »**isstr()**«. Die Funktion »`true`« liefert den Wert 1, der die Erfüllung einer Bedingung anzeigt; »`false`« den Wert 0 für die Nichterfüllung. Jeder von 0 verschiedene Wert wird als erfüllte Bedingung betrachtet.

Strukturierte Datentypen wie Arrays oder Verbünde (Records) kennt der Hamster nicht. Sie können aber durch **Listen** realisiert werden.

Typenkonversion: In Ausdrücken werden Werte unterschiedlicher Datentypen ggf. automatisch konvertiert. Zunächst geht der Interpreter beim Gesamtresultat des Ausdrucks von einer Zahl aus. String-Operanden werden vor der weiteren Berechnung in Zahlen konvertiert; falls diese Konversion fehlschlägt – nicht jeder String lässt sich als Zahl interpretieren –, ist auch das Gesamtresultat ein String. Es empfiehlt sich aus Gründen der Effizienz und der Klarheit, Typenkonversionen **explizit** vorzunehmen. Die »`int()`«-Funktion konvertiert einen String in eine Zahl, »`str()`« und »`hex()`« stellen eine Zahl als String dar.

Beschreibung: int / str / hex

Dynamische Datentypen

Zeiger: Auf dynamische Datentypen kann nicht direkt, sondern nur über ihre Zeiger zugegriffen werden. Die Zeiger referenzieren direkt den Arbeitsspeicher. Hierzu sind spezielle Funktionen zu verwenden, siehe den Abschnitt Zeiger-Funktionen. Zeiger müssen vor ihrer ersten Verwendung deklariert und initialisiert werden.

Ausdrücke und Operatoren

Ausdrücke

Der Interpreter des Hamsters erlaubt die Berechnung **arithmetischer** sowie **Boolescher Ausdrücke**. Ein Ausdruck liefert nach seiner Berechnung einen Wert, der sich **zuweisen**, beim Aufruf einer Prozedur oder Funktion als **Argument** übergeben oder in anderen Ausdrücken weiterverwenden lässt. Ausdrücke bestehen aus **Operatoren** und **Operanden**, deren Werte durch die Operatoren zu einem Gesamtergebnis verknüpft werden. Als Operand kann jeder Wert, also auch jeder Ausdruck eingesetzt werden.

Operatoren

Binäre Operatoren sind Operatoren mit zwei **Operanden**. Der Interpreter implementiert die für arithmetische Ausdrücke übliche Syntax, bei der die beiden Operanden jeweils vor und nach dem Operator anzugeben sind (**Infix-Notation**). Die Anweisung

```
$Summe = $Operand1 + $Operand2
```

addiert die Werte der Variablen $Operand1 und $Operand2 und weist das Ergebnis der Variable $Summe zu. Sofern beide Summanden Strings sind oder sich wenigstens einer nicht in eine Zahl konvertieren lässt, besteht $Summe aus $Operand1, verlängert um den Inhalt von $Operand2 (**Concatenation**, »String-Addition«).

Unäre Operatoren gehen ihrem einzigen Operanden voran (**Präfix-Notation**). Die Anweisung

```
$Neg = - $Operand
```

weist beispielsweise der Variablen $Neg den negativen Wert der Variablen $Operand zu.

Die Operatoren »**iif()**« und »**icase()**«, die wenigstens drei Operanden bei funktionaler Syntax verarbeiten, werden im Abschnitt über Verzweigungen behandelt. Folgende Operatoren kennt der Interpreter:

Operator	*Anz. Operanden*	*Bedeutung*
( )	1	Klammerung
!	1	logische Verneinung
~	1	binäre bitweise Verneinung (Zusatz)
+	1	positives Vorzeichen
-	1	negatives Vorzeichen
*	2	Multiplikation
/	2	Division
%	2	Modulo (»Rest« der Division)
+	2	Addition, Zeichenverkettung
-	2	Subtraktion
<<	2	binäres bitweises Schieben nach links
>>	2	binäres bitweises Schieben nach rechts
<	2	kleiner als
>	2	größer als
<=	2	kleiner gleich
>=	2	größer gleich
== oder =	2	gleich
!= oder <>	2	ungleich
&	2	binäres Und
^	2	binäres Xor
|	2	binäres Oder
&&	2	logisches Und
||	2	logisches Oder

Operatorpriorität und -Assoziativität

Die Operatorattribute **Priorität** und **Assoziativität** bestimmen die Reihenfolge der Berechnung, wenn sie nicht durch Klammern vorgegeben wird. Im Ausdruck

```
$Summe = func1 + $Operand2 * func3
```

würde zuerst das Produkt aus dem Wert in `$Operand2` und dem Resultat der Funktion `func3` berechnet und dann das Resultat der Funktion `func1` hinzuaddiert; `func3` würde also vor `func1` aufgerufen. Der Operator der Multiplikation hat eine größere **Priorität** (**Rang**) als derjenige der Addition, der Interpreter kann in arithmetischen Ausdrücken die Regel »Punktrechnung geht vor Strichrechnung« berücksichtigen. Im Ausdruck

```
$Summe = func1 + $Operand2 + func3
```

würde zuerst die Summe aus dem Resultat der Funktion `func1` und `$Operand2` berechnet und erst dann das Resultat der Funktion `func3` hinzuaddiert; `func1` würde also vor `func3` aufgerufen. Der zwischen den zwei Additionsoperatoren stehende `$Operand2` wandert sozusagen zuerst nach links, der Additionsoperator ist **linkssassoziativ**. Bei einem **rechtsassoziativen** Operator würde `$Operand2` demgegenüber in die Berechnung des rechten Operators einbezogen.

Der Interpreter des Hamsters berücksichtigt ohne Klammerung folgende Prioritäten und Assoziativitäten, gleichrangige Operatoren stehen in einer Zeile:

Operator	*Priorität*	*Assoziativität*	*Zuläss. Datentypen der Operanden*
( )	*(höchste)*	← links	Zahl oder String
+ - (unär) ! ~	…	→ rechts (Präfix)	Zahl
* / %	…	← links	Zahl
+ - (binär)	…	← links	Zahl (oder String bei +)
<< >>	…	← links	Zahl
<> >= <=	…	← links	Zahl oder String
== = != <>	…	← links	Zahl oder String
&	…	← links	Zahl
^	…	← links	Zahl
|	…	← links	Zahl
&&	…	← links	Zahl
||	*(niedrigste)*	← links	Zahl

Durch **Klammerung** mit runden Klammern »()« lässt sich die Reihenfolge der Berechnung vorgeben. Im Verlaufe der Berechnung von

```
$Summe = func1 * ( $Operand2 + func3 )
```

würde die Funktion `func3` vor der Funktion `func1` aufgerufen.

Boolesche Ausdrücke

Verzweigungen und Schleifen erlauben die **bedingte (wiederholte) Ausführung** einer oder mehrerer Anweisungen. Die zu prüfende Bedingung ist fast immer als **Boolescher Ausdruck** anzugeben. Da eine Bedingung entweder erfüllt oder nicht erfüllt sein kann, hat ein solcher Ausdruck nur zwei mögliche Ergebnisse; im Falle des Hamsters die ganzzahligen Werte 1 (Bedingung erfüllt, **True**) oder 0 (Bedingung nicht erfüllt, **False**). Innerhalb eines solchen Ausdrucks wird jede von 0 verschiedene Zahl wie der Wert True behandelt.

Beispielsweise hat der **Vergleichsausdruck** »$a < $b« den Wert 1, wenn die Variable $a einen kleineren Wert als $b enthält, ist er dagegen gleich dem Wert oder größer als derjenige von $b, hat er den Wert 0.

Boolesche Operatoren

In einem einzigen Booleschen Ausdruck können auch mehrere Bedingungen geprüft werden. Die Ergebnisse der einzelnen Prüfungen werden dabei durch **Boolesche Operatoren** (**logische Operatoren**) zu einem Gesamtergebnis zusammengefasst. Die Syntax gleicht derjenigen der arithmetischen Ausdrücke. So ist die Bedingung

```
( $a = $b ) && ( $c = $d )
```

nur dann erfüllt, wenn der Wert von $a mit dem von $b **und** der Wert von $c mit dem von $d übereinstimmt. Dagegen ist die Bedingung

```
( $a = $b ) || ( $c = $d )
```

bereits dann erfüllt, wenn der Wert von $a mit dem von $b **oder** der Wert von $c mit dem von $d übereinstimmt. Die Bedingung

```
!( $a = $b )
```

ist nur erfüllt, wenn der Wert von $a **nicht** mit dem von $b übereinstimmt; der Wert des vergleichenden Ausdrucks »($a = $b)« wird also **negiert**.

»&&«, »||« und »!« gehören zu den logischen Operatoren, »=« ist in diesem Zusammenhang einer von mehreren **Vergleichsoperatoren**. Eine Übersicht aller verfügbaren Operatoren findet sich im Abschnitt Operatoren.

Durch Klammerung kann die Auswertungsreihenfolge Boolescher Teilausdrücke wie in arithmetischen Ausdrücken vorgegeben werden. Der Interpreter des Hamsters berechnet Boolesche Ausdrücke immer vollständig (*keine* »**lazy Evaluation**«). Im Ausdruck

```
( $a = $b ) && Test( $c )
```

würde die Funktion »Test($c)« also in jedem Falle aufgerufen, auch wenn das Ergebnis des gesamten Ausdrucks bereits nach Auswertung des Teilausdrucks »$a = $b« feststehen sollte: Wenn $a und $b verschiedene Werte enthalten, ergibt sich für gesamten Ausdruck der Wert 0 unabhängig vom Resultat der »Test()«-Funktion.

Sprünge

Sprünge ermöglichen es, die Abarbeitung von Anweisungen an einer anderen Stelle im Skript fortzusetzen. Heutzutage gelten allerdings allgemein Sprünge als veraltet und obsolet, da es mittels Schleifen, Verzweigungen, Subroutinen usw. inzwischen wesentlich übersichtlichere und komfortablere Möglichkeiten der Ablaufsteuerung gibt. Trotzdem sollen die Anweisungen, mit denen Sprünge ermöglicht werden, hier dokumentiert werden:

Es gibt zwei Arten von Sprunganweisungen, »`goto()`« und »`gosub()`«, die sich in einem wesentlichen Punkt unterscheiden: Im Gegensatz zur »**Goto**«-Anweisung muss ein mit »**gosub**« angesprungener Anweisungsblock immer mit einer »**Return**«-Anweisung abgeschlossen werden, worauf die Abarbeitung des Skriptes mit der Anweisung hinter der »**Gosub**«-Anweisung fortfährt und ein möglicher »Return«-Rückgabewert ausgewertet werden kann. Bei »**goto**« gibt es keine solche Rückkehrmöglichkeit.

Bei beiden Anweisungen wird die Abarbeitung des Skriptes fortgeführt mit derjenigen Anweisung, die auf das durch eine »`Label()`«-Anweisung gekennzeichnete **Sprungziel** folgt. In den Klammern des Labels ist ein eindeutiger Bezeichner als **Sprungmarke** einzufügen. Dieser Bezeichner ist frei wählbar, er darf allerdings nicht mit dem Dollarzeichen »$« beginnen (das kennzeichnet Variablenbezeichner) und nicht in Anführungszeichen stehen (das wäre ein Stringliteral). Er muss mit einem Buchstaben beginnen und darf im Übrigen aus Buchstaben, Ziffern und dem Unterstrich »_« bestehen. Groß- und Kleinbuchstaben werden nicht unterschieden. Die Sprungmarke ist dann auch in den Klammern der Sprunganweisung anzugeben.

Beispiel für »Goto«:

```
print( "Letzte Anweisung vor dem Sprung" )
goto( Ziel ) # Sprunganweisung
warning( "Diese Warnung wird übersprungen und deshalb nie angezeigt." )
label( Ziel ) # Sprungziel
print( "Erste Anweisung nach dem Sprung" )
quit
```

Beispiel für »Gosub«:

```
gosub( Sprungziel1 )
print( gosub(Sprungziel2) ) # 2. Logeintrag, Ausgabe: "42"
print("Ende") # 3. und letzter Logeintrag
quit
label( Sprungziel2 )
return( 42 )
label( Sprungziel1 )
print("Gosub-Prozedur 1 ausgeführt") # 1. Logeintrag
return
```

Die Sichtbarkeit von Variablen kann mit »entercontext« und »leavecontext« beeinflusst werden.

Der »Gosub«-Mechanismus kann vollständig durch leichter verständliche selbst definierte Prozeduren und Funktionen (Subroutinen) ersetzt werden.

Die Kombination von »**Goto**«-Sprüngen und Verzweigungen stellt eine veraltete Form der Realisierung von **Schleifen** dar. Ein Beispiel:

```
# Anweisungen vor der Schleife
label( Schleifenkopf )
# Schleifenrumpf mit Anweisungen
if( ⟨boolean expression⟩ ) # Verzweigung
  goto( Schleifenende )
else
  goto( Schleifenkopf )
endif
label( Schleifenende )
# Anweisungen nach der Schleife
```

Das Beispiel lässt sich jedoch besser durch die Kontrollstruktur einer **postkonditionalen Schleife** formulieren:

```
# Anweisungen vor der Schleife
repeat
  # Schleifenrumpf mit Anweisungen
until( ⟨boolean expression⟩ )
# Anweisungen nach der Schleife
```

Verzweigungen

Verzweigung in Anweisungsfolgen

Eine **Verzweigung** ermöglicht die **alternative Ausführung** einer oder mehrerer Anweisungen. Ihr **Selektor** bestimmt die auszuführende Alternative, er ist als Boolescher Ausdruck anzugeben.

Die Verzweigung ist durch das Schlüsselwort »`if()`« einzuleiten, dem die zu prüfende Bedingung als Boolescher Ausdruck in runden Klammern folgt. In der nächsten Zeile folgen alle Anweisungen, die bei Erfüllung der Bedingung ausgeführt werden sollen. Nach der letzten Anweisungszeile wird die Verzweigung durch das Schlüsselwort »`endif`« abgeschlossen:

```
if( $a < $b )
  print( "Bedingung erfüllt." )
endif
```

Das Beispiel gibt den Text »Bedingung erfüllt.« durch Ausführung der »`print()`«-Anweisung nur dann aus, **wenn** der Boolesche Ausdruck »`$a < $b`« den Wert 1 (Bedingung erfüllt) liefert,

also die Variable $a einen kleineren Wert als $b enthält. Alle auf »**endif**« folgenden Anweisungen würden wieder unbedingt in jedem Falle ausgeführt.

Das Schlüsselwort »**else**« leitet eine Reihe von Anweisungen ein, die im Falle der Nichterfüllung auszuführen sind:

```
if( $a < $b )
  print( "Bedingung erfüllt." )
else
  print( "Bedingung leider nicht erfüllt!" )
endif
```

Dieses Beispiel verhält sich wie das vorhergehende; sofern die Variable $a allerdings *keinen* kleineren Wert als die Variable $b enthält (also »$a > $b« oder »$a = $b« gilt), wird jetzt der Text »Bedingung leider nicht erfüllt!« ausgegeben.

Das Schlüsselwort »**elseif**« leitet, gekoppelt an eine weitere Bedingung, eine Reihe von Anweisungen ein, die im Falle der Nichterfüllung der ersten »**if**«-Anweisung auszuführen sind:

```
if( $a < $b )
  print( "Bedingung_1 erfüllt." )
elseif( $a = $b )
  print( "Bedingung_1 nicht erfüllt, aber Bedingung_2 erfüllt." )
else
  print( "Keine der beiden Bedingungen erfüllt!" )
endif
```

Verzweigung innerhalb von Ausdrücken

Innerhalb eines Ausdrucks kann der »**iif()**«-Operator verwendet werden. Die Verzweigung

```
if( $a < $b )
  $x = $b - $a
else
  $x = $a - $b
endif
```

kann auch durch die Zeile

```
$x = iif( $a < $b, $b - $a, $a - $b )
```

ausgedrückt werden. Ist die Bedingung »$a < $b« erfüllt, wird $x der Wert des Ausdrucks »$b - $a« zugewiesen, anderenfalls der Wert des letzten Ausdrucks »$a - $b«. Es wird nur derjenige Ausdruck berechnet, dessen Wert der »**iif()**«-Operator zurückgibt. In der Zeile

```
$x = iif( $a < $b, $b - $a, calc( $c ) )
```

wird »calc($c)« also nicht aufgerufen, wenn die Bedingung »$a < $b« erfüllt ist und daher das Resultat von »$b - $a« zurückzugeben ist.

Auch der »`icase()`«-Operator kann innerhalb von Ausdrücken verwendet werden. Als Selektor dient ein Ausdruck, dessen Wert aus mehr als nur zwei Alternativen auswählen kann. Die Anweisung im folgenden Beispiel wurde der Übersichtlichkeit halber auf mehrere Textzeilen verteilt und durch die Unterstriche »_« zu einer Anweisungszeile zusammengefügt; das empfiehlt sich auch in eigenen Skripten. Die Zeile

```
$x = icase( $a + $b, 1, "Erster Ausdruck", _
                     2, "Zweiter Ausdruck", _
                     3, "Dritter Ausdruck", _
                     else, "Unbekannter Wert!" )
```

weist $x das Literal »Erster Ausdruck« zu, wenn der Ausdruck »$a + $b« den Wert 1 hat, das Literal »Zweiter Ausdruck«, wenn er den Wert 2 hat usw. Das Schlüsselwort »`else`« steht stellvertretend für alle anderen Werte; wenn der Ausdruck »$a + $b« im Beispiel also einen anderen Wert als 1, 2 oder 3 hat, wird $x das Literal »Unbekannter Wert!« zugewiesen. Der »else«-Zweig ist optional. Auch der »icase()«-Operator berechnet nur den Ausdruck, dessen Wert zurückgegeben wird. Allgemein hat der »icase()«-Operator folgende Syntax:

```
icase( ⟨exp_selector⟩, ⟨expression_alternative_1⟩, ⟨expression_result_1⟩, _
                       ⟨expression_alternative_2⟩, ⟨expression_result_2⟩, _
                       ...
                       ⟨expression_alternative_n⟩, ⟨expression_result_n⟩[, _
                       else, ⟨expression_result_else⟩] )
```

An jeder Stelle können also nicht nur Werte oder einzelne Variablen, sondern auch Ausdrücke verwendet werden. Der Operator muss nicht auf mehrere Textzeilen verteilt werden, wie dies hier aus Gründen der Übersicht geschieht.

Schleifen

Eine **Schleife** veranlasst die **wiederholte Ausführung** einer oder mehrerer Anweisungen. Die Wiederholung oder ihr Abbruch kann von einer Bedingung abhängig gemacht werden. Diese Bedingung ist fast immer als Boolescher Ausdruck anzugeben.

Jede Schleife beginnt mit dem **Schleifenkopf**, dem der **Schleifenrumpf** mit den zu wiederholenden (**iterierenden**) Anweisungen folgt und wird mit dem **Schleifenfuß** abgeschlossen:

```
do # Schleifenkopf
  # Schleifenrumpf mit Anweisungen
loop # Schleifenfuß
```

In dieser Form würde die Schleife alle Anweisungen in ihrem Rumpf endlos wiederholen. Schleifen können mit der »`break()`«-Anweisung bedingt oder unbedingt abgebrochen werden. In den Klammern ist ein Boolescher Ausdruck als **Abbruchbedingung** anzugeben. Die Schleife wird dann bei Erreichen der »**break()**«-Anweisung abgebrochen, wenn die Bedingung in den

Klammern erfüllt ist; »`break`« allein bricht unbedingt ab. Bei Abbruch der Schleife wird mit der ersten, auf den Schleifenfuß folgenden Anweisung fortgefahren, die der »**break()**«-Anweisung folgenden Anweisungen im Rumpf werden dann nicht mehr ausgeführt.

Die »`continue()`«-Anweisung verhindert im Schleifenrumpf ebenfalls die Ausführung der ihr folgenden Anweisungen, bricht aber die Schleife nicht ab, sondern veranlasst einen erneuten Durchlauf bei der ersten Anweisung des Rumpfes. In den Klammern lässt sich dafür eine Bedingung angeben, »`continue`« allein bewirkt in jedem Falle eine erneute Iteration.

Bedingte oder unbedingte »**break()**«- und »**continue()**«-Anweisungen dürfen in den Rümpfen aller Schleifenkonstrukte des Hamsters beliebig oft vorkommen. Schleifen können verschachtelt werden; eine »**break()**«-Anweisung bricht nur die Schleife ab, in deren Rumpf sie ausgeführt wird.

Präkonditionale Schleife

Mit einer dem Schleifenkopf unmittelbar folgenden »**break()**«-Anweisung erhält man eine **präkonditionale Schleife**. Dieser häufige Spezialfall wird mit einem Schleifenkonstrukt berücksichtigt, das im Schleifenkopf auch gleich die Bedingung aufnimmt, unter der die Anweisungen im Rumpf ausgeführt werden sollen:

```
while( ⟨boolean expression⟩ )
  # Schleifenrumpf mit Anweisungen
endwhile
```

Die Anweisungen im Rumpf werden ausgeführt, **solange** die durch den Booleschen Ausdruck gegebene Bedingung erfüllt ist. Die Bedingung wird *vor* jedem Durchlauf geprüft; bei Abbruch wird mit der auf »`endwhile`« folgenden Anweisung fortgefahren. Wenn die Bedingung schon vor dem ersten Durchlauf nicht erfüllt ist, werden die Anweisungen im Rumpf folglich überhaupt nicht ausgeführt.

Postkonditionale Schleife

Wenn die »**break()**«-Anweisung dem Schleifenfuß unmittelbar vorangeht, hat man eine **postkonditionale Schleife** formuliert. Auch für diesen Spezialfall gibt es ein Konstrukt, bei dem der Schleifenfuß die Abbruchbedingung aufnimmt:

```
repeat
  # Schleifenrumpf mit Anweisungen
until( ⟨boolean expression⟩ )
```

Die Anweisungen im Rumpf werden ausgeführt, **bis** die durch den Booleschen Ausdruck gegebene Bedingung erfüllt ist. Die Bedingung wird *nach* jedem Durchlauf geprüft; bei Abbruch wird mit der auf »`until`« folgenden Anweisung fortgefahren. Die Anweisungen im Rumpf werden also in jedem Falle einmal ausgeführt.

Es ist möglich, wenn auch aus Gründen der Übersicht und der Effizienz nicht empfehlenswert, den »**while()**«-Kopf mit dem »**until()**«-Fuß in einer Schleife zu kombinieren.

Zählschleife

Die **Zählschleife** dient dem häufig anzutreffenden Spezialfall, in dem sich die Zahl der Iterationen durch Ausdrücke angeben lässt. Die »**For**-Schleife«

```
for( ⟨loopvar⟩, ⟨expression_start⟩, ⟨expression_end⟩, ⟨expression_step⟩ )
  # Schleifenrumpf mit Anweisungen
endfor
```

wiederholt die Anweisungen in ihrem Rumpf solange, bis der in der **Schleifenvariablen** ⟨loopvar⟩ gespeicherte Wert den Wert des Ausdrucks ⟨expression_end⟩ erreicht oder überschreitet. Vor dem ersten Durchlauf wird der Schleifenvariablen der Wert des Ausdrucks ⟨expression_start⟩ (**Startwert**) zugewiesen. *Nach jeder* Iteration wird die Schleifenvariable um den Wert des Ausdrucks ⟨expression_step⟩ (**Schrittweite**) erhöht; ohne Angabe der Schrittweite wird nach jeder Iteration um 1 erhöht.

Eine derartige Schleife ließe sich auch mit den anderen Schleifen darstellen. Die For-Schleife ist aber in der Ausführung effizienter, weil der Schleifenkopf die Zählvariable auszeichnet und es dem Interpreter ermöglicht, sie für die Dauer der Iterationen für einen beschleunigten Zugriff vorzuhalten. Die Ausdrücke im Schleifenkopf werden nur vor dem ersten Durchlauf einmal berechnet.

Prozeduren und Funktionen

Wenn bestimmte Arbeitsgänge in einem Skript mehrfach benötigt werden, fasst man die erforderlichen Anweisungsfolgen in **Prozeduren** oder **Funktionen** unter einem möglichst aussagekräftigen Namen zusammen. Das erspart zum einen ineffiziente Wiederholungen gleichartigen Codes und erlaubt zum anderen eine übersichtliche Gliederung komplizierterer Skripte.

Eine **Prozedur** fasst eine Anweisungsfolge unter einem Namen als **Prozedurbezeichner** zusammen. Der Interpreter führt die Anweisungsfolge im **Prozedurrumpf** immer dann aus, wenn er den Prozedurbezeichner als Anweisung im Quelltext vorfindet (**Prozeduraufruf**). Danach fährt er mit der nächsten Anweisung nach dem Aufruf fort.

Funktionen sind **Prozeduren**, die ein **Resultat** zurückgeben. Sie können daher nicht nur wie eine Prozedur mit ihrem Namen als Anweisung aufgerufen werden, ihr Name kann auch als **Operand** in Ausdrücken verwendet werden. Die Anweisungen im Funktionsrumpf werden ausgeführt, bevor der Wert des Funktionsresultats wie jeder andere Operand im Ausdruck weiterverarbeitet wird.

Der Hamster bietet bereits viele eingebaute Prozeduren und Funktionen.

Eigene Prozeduren und Funktionen

Prozeduren allgemein

Selbst definierte Prozeduren müssen am Ende des Skriptes stehen, ihre Reihenfolge spielt keine Rolle. Sie werden durch das Schlüsselwort »**Sub**« eingeleitet und durch »**Endsub**« abgeschlossen:

```
sub ⟨identifier⟩ # Prozedurkopf
  # Prozedurrumpf mit Anweisungen
endsub
```

Eine so definierte Prozedur kann unter ihrem **Prozedurbezeichner** »**⟨identifier⟩**« im Quelltext **aufgerufen** werden. Der Interpreter führt bei jedem Aufruf alle Anweisungen im Prozedurrumpf aus, bis er eine »**Endsub**«- oder »**Return**«-Anweisung erreicht. Im Prozedurrumpf können beliebig viele »Return«-Anweisungen verwendet werden. Wegen der deutlich besseren Lesbarkeit ist zu empfehlen, *immer* »endsub« an das Ende einer Prozedurdefinition zu setzen.

Der Prozedurbezeichner **⟨identifier⟩** ist frei wählbar, er darf allerdings nicht mit dem Dollarzeichen »$« beginnen (das kennzeichnet Variablenbezeichner), nicht in Anführungszeichen stehen (das wäre ein Stringliteral) und kein Schlüsselwort (eine bereits definierte Prozedur) sein. Er muss mit einem Buchstaben beginnen und darf im Übrigen aus Buchstaben, Ziffern und dem Unterstrich »_« bestehen. Groß- und Kleinbuchstaben werden nicht unterschieden.

Prozeduren können nicht verschachtelt werden, eine Prozedur kann in ihrem Rumpf also keine weiteren Prozedurdefinitionen aufnehmen. Eine Prozedur kann jedoch mit ihren Anweisungen andere Prozeduren und auch sich selbst (**Rekursion**) aufrufen.

Eine veraltete Möglichkeit, mit dem Befehl »gosub« und Labeln das Gleiche ohne selbst definierte Prozeduren zu erreichen, wird im Abschnitt »Sprünge« beschrieben.

Globale und lokale Variablen: In einem Prozedurrumpf können Variablen **lokal** deklariert werden, außerhalb von Prozeduren auf der Skriptebene deklarierte Variablen sind **global**.

Lebensdauer

Lokale Variablen speichern ihre Werte, solange die Prozedur ausgeführt wird, in der sie deklariert sind. Sobald die Prozedur beendet wird, existieren sie nicht mehr und die in ihnen gespeicherten Werte sind verloren.

Globale Variablen speichern ihre Werte, solange das Skript läuft, in dem sie deklariert sind.

Sichtbarkeit

Auf **lokale** Variablen kann nur innerhalb der Prozedur zugegriffen werden, in der sie deklariert wurden. Im übrigen Skript oder anderen Prozeduren gelten sie als nicht deklariert, sie sind dort **unsichtbar**. Eine *aufgerufene* Prozedur kann daher auch die Variablen der sie *aufrufenden* Prozedur nicht sehen.

Globale Variablen sind in jeder Prozedur **sichtbar** *(siehe jedoch »**entercontext**« und »**leavecontext**«)*. Eine globale Variable kann allerdings durch eine *gleichnamige* lokale Variable **verdeckt** werden. Auf diese globale Variable ist dann in der Prozedur kein Zugriff möglich, obwohl sie weiterhin existiert und ihren Wert behält.

Beispiel:

```
#!hs2
var( $x {1}, $y ) # globale Variablen

# globaler Sichtbarkeitsbereich:
#  Hier sind $x und $y sichtbar.

Sub TestScope # Beginn der Prozedur TestScope

var( $a, $b, $x {2} ) # lokale Variablen

# lokaler Sichtbarkeitsbereich:
#   Hier sind $a, $b, das bei {2} deklarierte $x und das
#     global deklarierte $y sichtbar.
#   Die bei {1} global deklarierte Variable $x ist verdeckt,
#     kein Zugriff möglich!

endsub # Ende der Prozedur »TestScope«
```

Im Rumpf der Prozedur »TestScope« kann auf die globale, bei »{1}« deklarierte Variable $x nicht zugegriffen werden, nur das bei »{2}« deklarierte $x ist sichtbar. Außerhalb dieser Prozedur ist das globale $x sichtbar.

Funktionen

Damit Funktionen in der Lage sind, entsprechend ihrer Definition ein Resultat zurückzugeben, *muss* im **Funktionsrumpf** im Verlaufe der Ausführung eine »**Return()**«-Anweisung mit dem zurückzugebenden Wert erreicht werden. Diese »Return«-Anweisung beendet dann die Ausführung der Funktion und gibt den Wert des Ausdrucks in ihren Klammern als Resultat zurück. Fehlt eine Angabe, wird als Standard-Rückgabewert »0« verwendet. Im Funktionsrumpf können beliebig viele »Return«-Anweisungen verwendet werden.

In dem Skript

```
#!hs2
var( $a, $b, $x )

$a = 1
$b = 2
$x = Summe # vor der Zuweisung wird »Summe« ausgeführt, erst dann wird $x das
           # Resultat zugewiesen.
quit
```

```
sub Summe # Funktion »Summe«
  return( $a + $b )
endsub
```

liefert die Funktion »Summe« bei Aufruf die Summe der beiden globalen Variablen $a und $b mit »`return( $a + $b )`« zurück.

Parameter

Eine Prozedur oder Funktion arbeitet zwar immer die gleiche in ihrem Rumpf definierte Anweisungsfolge ab, aber man kann ihr durch **Parametrierung** bei jedem Aufruf andere Daten zur Verarbeitung übergeben.

Im Beispiel des vorhergehenden Abschnittes hat die Funktion »Summe« ihre Daten den globalen Variablen $a und $b entnommen. Falls mehr als nur ein Ergebnis berechnet wird, könnte jeder errechnete Wert im Rumpf weiteren globalen Variablen zugewiesen werden. Eine solche Parametrierung durch **Seiteneffekte** empfiehlt sich nur in kleinen Skripten und lässt vor allem bei der Berechnung von Ausdrücken mit mehreren Funktionsaufrufen schnell die Übersicht vermissen. Sie empfiehlt sich generell nur für solche Werte, die in mehreren Funktionen (etwa eines Moduls) gleichermaßen berücksichtigt werden und die sich nicht bei jedem Aufruf, sondern nur verhältnismäßig selten ändern.

Besser ist es, die jeweils zu verarbeitenden Daten in den Aufruf selbst aufzunehmen und sie als **Argumente** zu übergeben. Dazu muss die Prozedur oder Funktion in ihrer Definition eine **Parameterliste** vorgeben, die Anzahl und Verarbeitungsrichtung ihrer **Parameter** angibt. Beispielsweise definiert

```
sub Division( $x, $y, *$r ) # Funktionskopf mit Parameterliste
  $r = $x % $y
  return( $x / $y )
endsub
```

eine Funktion »Division«, die den Parameter $x ganzzahlig durch den Parameter $y dividiert. Die Parameter gelten im Rumpf als lokal deklariert und werden dort wie andere lokal deklarierte Variablen auch verwendet. Dem Parameter $r geht ein Stern (»*«) voran, um die Rückgabe eines Wertes in $r bei Funktionsende anzuzeigen. Der Aufruf

```
$div = Division( 10, 5, $divRest )
```

bewirkt dann die Zuweisung des Funktionsresultats (hier: 2) an die Variable $div und die Zuweisung des Divisionsrestes (hier: 0) an die Variable $divRest. Die Zuweisung an $divRest erfolgt erst beim Verlassen der Funktion.

Anstelle der beiden Literale 10 und 5 kann jeder Ausdruck als Argument übergeben werden; der Interpreter berechnet alle Argumente und kopiert ihre Werte in die als **Wertparameter** deklarierten lokalen Variablen (im Beispiel $x und $y), bevor er mit der Ausführung der Prozedur oder Funktion beginnt.

Der dritte, durch den vorangehenden Stern als **Rückgabeparameter** deklarierte Parameter setzt allerdings beim Aufruf *zwingend* eine Variable voraus, da einem Ausdruck kein Wert zugewiesen werden kann. Im Übrigen verhält sich ein Rückgabeparameter nicht anders als ein Wertparameter; er erlaubt also den Datenfluss in beide Richtungen, während ein Wertparameter nur Daten eingeben kann.

Selbst definierten Prozeduren und Funktionen müssen immer so viele Werte bzw. Variablen in genau der Reihenfolge übergeben werden, die die Parameterliste der Definition vorgibt. Lediglich einige eingebaute Prozeduren und Funktionen erlauben das Auslassen von Parametern, für die sie dann Standardwerte verwenden. Näheres ist im Einzelfall deren Dokumentation zu entnehmen.

Eingebaute Prozeduren

`quit( ⟨code⟩ )`
`quit`

Beendet ein Skript mit dem Rückgabewert ⟨code⟩.

`return( ⟨code⟩ )`
`return`

Beendet eine Subroutine mit dem Rückgabewert ⟨code⟩.

`beep( ⟨code⟩ )`
`beep`

Gibt Systemklänge aus. Sinnvolle Werte für den Parameter ⟨code⟩ sind im Modul »hwindows.hsm« als benannte Konstanten vordefiniert (`$SIGNAL_STOP`, `$SIGNAL_QUESTION`, `$SIGNAL_EXCLAMATION` und `$SIGNAL_ASTERISK`).

`error( ⟨errmsg⟩ )`
`error`

Beendet ein laufendes Skript und gibt dabei eine (rote) Fehlermeldung im Hamster-Protokoll aus, die durch ⟨errmsg⟩ vorgegeben werden kann.

Beispiel:

```
error( "Datei ", $Filename, " nicht gefunden!" )
```

`assert( ⟨assertion⟩, ⟨errmsg⟩ )`
`assert( ⟨assertion⟩ )`

Beendet ein laufendes Skript und gibt dabei eine Fehlermeldung im Hamster-Protokoll aus, die durch ⟨errmsg⟩ vorgegeben werden kann (Standard bei fehlender ⟨errmsg⟩: »Assertion failed!«), wenn der Fehlerparameter ⟨assertion⟩ erfüllt worden ist.

Beispiel:

```
assert( FileExists($Filename), "Datei ", $Filename, " nicht gefunden!" )
```

sleep(⟨time⟩)

Hält die Programmausführung für die in Millisekunden (= 1/1000 Sekunden) angegebene Zeitdauer ⟨time⟩ an.

Beispiel:

```
sleep( 60000 ) # Eine Minute Pause
```

trace(⟨onoff⟩)

Schaltet die Anzeige der auszuführenden Programmzeilen im Protokoll an oder aus.

⟨onoff⟩ = 1: Anzeige an,
⟨onoff⟩ = 0: Anzeige aus.

dump

Zeigt die Inhalte aller Variablen an.

debug(⟨level⟩, ⟨text⟩ [, ⟨text⟩ ...])
debug(⟨level⟩)

Ist nur der Level ⟨level⟩ gegeben, so wird dieser auf den gegebenen Wert gesetzt. Ist zusätzlich der Text ⟨text⟩ angegeben, wird der Text genau dann angezeigt, wenn der aktuelle Level größer oder gleich dem angegebenen Level ist.

Beispiel:

```
debug( 255 )
debug( 10, "Jetzt sieht man diesen Text! " )
debug( 5 )
debug( 10, "Jetzt nicht mehr! " )
```

runscript(⟨Skriptname⟩ , ⟨Parameter⟩ , ⟨warten⟩)
runscript(⟨Skriptname⟩ , ⟨Parameter⟩)
runscript(⟨Skriptname⟩)

Startet das Skript ⟨Scriptname⟩ mit den Parametern ⟨Parameter⟩. Wird der Parameter ⟨warten⟩ mit dem Wert »TRUE« angegeben, wartet das aufrufende Skript auf das Ende des aufgerufenen Skriptes, anderenfalls wird die Ausführung des aufrufenden Skriptes unmittelbar fortgesetzt (Multitasking), Standard ist »TRUE«. Das aufgerufene Skript kann die übergebenen Parameter mit den Funktionen paramstr und paramcount auswerten. Parameter können auch Anführungsstriche enthalten, indem man sie einfach verdoppelt, d. h. »"Hallo ""Paulchen""!"« wird zu »Hallo "Paulchen"!«.

(Beispiel auf der folgenden Seite)

```
# params.hsc:
#!hs2
trace(0)

var ($i1)
print ( "Params0-0:  Start")
print ("Params0-0:  Start params2.hsc/1")
RunScript ("params2.hsc", "A" + chr(13) + "5000" + chr(13) + "15000", true)
print ("Params0-0:  Ends  params2.hsc/1")
sleep (2000)
print ("Params0-0:  Start params2.hsc/2")
RunScript ("params2.hsc", "B" + chr(13) + "8000" + chr(13) + "5000", true)
print ("Params0-0:  Ends  params2.hsc/2")
sleep (6000)
print ("Params0-0:  Done.")
quit

# params2.hsc:
# Dieses Skript erwartet 3 Parameter:
#    1) eine ID (Buchstabe o.ä.)
#    2) Zeitdauer in Millisekunden für die 1. Pause
#    3) Zeitdauer in Millisekunden für die 2. Pause
#!hs2
var ($i1)
print ("params2-" + ParamStr (1) + " : Start ...")
print ("params2-" + ParamStr (1) + " : sleep (" + ParamStr (2) + ")")
sleep (ParamStr (2))
print ("params2-" + ParamStr (1) + " : ParamCount = " + ParamCount)
for ($i1, 1, ParamCount+2)
print ("params2-" + ParamStr (1) + " : " + $i1 + " : >" + ParamStr ($i1) + "<")
endfor
print ("params2-" + ParamStr (1) + " : sleep (" + ParamStr (3) + ")")
sleep (ParamStr (3))
print ("params2-" + ParamStr (1) + " : Done.")
quit (37)
```

Eingebaute Funktionen (allgemein)

- Allgemeine Funktionen zum Manipulieren von Variablen
- Numerische Funktionen
- Zeichenketten-/String-Funktionen
- Datums-/Zeit-Funktionen
- Fehlerbehandlung
- Dateien und Verzeichnisse
- Listen
- Ein- und Ausgabe
- Artikel bearbeiten
- Zeitgeber/Scheduler
- DFÜ-Netzwerk
- Zeiger-Funktionen
- Sonstige Funktionen

Allgemeine Funktionen zum Manipulieren von Variablen

`set( ⟨var⟩, ⟨expression⟩ )`
`⟨var⟩ = ⟨expression⟩`

Setzt die Variable `⟨var⟩` auf den Wert `⟨expression⟩`.
(Die Variable muss zuvor mit »`var(⟨var⟩)`« deklariert worden sein.)

Rückgabewert: Ergebnis von `⟨expression⟩`.

Beispiel:

```
var( $a, $b, $c )
$a = 42
set( $b, "Die Antwort lautet: " )
$c = $b + $a + "."
print( $c )  # Ausgabe: "Die Antwort lautet: 42."
```

`varset( ⟨var1⟩ [ , ⟨var2⟩ , ⟨var3⟩ , ... ] , ⟨expression⟩ )`

Deklariert die Variable `⟨var1⟩` sowie optional weitere Variablen `⟨varX⟩` und weist *allen* zugleich den Wert `⟨expression⟩` zu.

Rückgabewert: Ergebnis von `⟨expression⟩`.

Beispiel:

```
varset( $a, $b, 21 )
varset( $c , $a + $b )
var( $d )
$d = "Die Antwort lautet: " + $c + "."
print( $d )  # Ausgabe: "Die Antwort lautet: 42."
```

`inc( ⟨var⟩, ⟨count⟩ )`
`inc( ⟨var⟩ )`
`dec( ⟨var⟩, ⟨count⟩ )`
`dec( ⟨var⟩ )`

Erhöht (»`inc`«) bzw. vermindert (»`dec`«) die Variable `⟨var⟩` um den Betrag `⟨count⟩` bzw. um 1, wenn `⟨count⟩` nicht angegeben wird.

Rückgabewert: Erhöhte oder verminderte Variable `⟨var⟩`.

Beispiel:

```
inc( $a )
dec( $a, 2 )
```

Numerische Funktionen

`true`

Repräsentiert den logischen Wert Wahr.

Rückgabewert: 1

Beispiel:

```
$a = true
```

`false`

Repräsentiert den logischen Wert Falsch.

Rückgabewert: 0

Beispiel:

```
$a = false
```

`isint( ⟨expression⟩ )`

Testet, ob der Ausdruck `⟨expression⟩` vom Typ Integer ist.

Rückgabewert: True (1) = wahr oder False (0) = falsch.

Beispiel:

```
$result = iif( isint($a), $a, 0 )
```

`int( ⟨expression⟩, ⟨default⟩ )`
`int( ⟨expression⟩ )`

Konvertiert die Zeichenkette `⟨expression⟩` in einen numerischen Integerwert. Beginnt die Zeichenkette mit den Zeichen »0x«, wird sie als hexadezimal codiert betrachtet.

Rückgabewert: Integer-Variable oder im Fehlerfall Inhalt des Parameters `⟨default⟩`.

Beispiel:

```
$a = "1"
$b = 2
$c = $a + $b       # Ergebnis: "12"
$d = int($a) + $b  # Ergebnis: 3

print( int("0xFF") )  # Ausgabe: 255
```

`abs( ⟨number⟩ )`

Liefert den vorzeichenlosen Betrag von `⟨number⟩`.

Rückgabewert: Integer.

Beispiel:

```
print( abs( 42) )  # Ausgabe: 42
print( abs(-42) )  # Ausgabe: 42
```

`sgn( ⟨number⟩ )`

Liefert −1, wenn `⟨number⟩` negativ ist, +1, wenn `⟨number⟩` positiv ist, und 0, wenn `⟨number⟩` 0 ist.

Rückgabewert: Integer.

Beispiel:

```
print( sgn( 42) )  # Ergebnis: 1
print( sgn(-42) )  # Ergebnis: -1
```

Zeichenketten-/String-Funktionen

`isstr( ⟨expression⟩ )`

Testet, ob der Ausdruck vom Typ String ist.

Rückgabewert: True (1) = wahr oder False (0) = falsch.

Beispiel:

```
$result = iif( isstr($a), $a, "0x" + hex($a) )
```

`ord( ⟨string⟩ )`

Liefert den ASCII-Wert des ersten Zeichens einer Zeichenkette.

Rückgabewert: ASCII-Wert oder 0, wenn `⟨string⟩` leer ist.

Beispiel:

```
print( ord("*") )  # Ausgabe: 42
```

`chr( ⟨number⟩ )`

Liefert das ASCII-Zeichen an der Position `⟨number⟩` und ist somit die Gegenfunktion von »ord«.

Rückgabewert: String mit einem Zeichen.

Beispiel:

```
print( chr(42) )  # Ausgabe: "*"
```

`str( ⟨number⟩, ⟨length⟩, ⟨leadchar⟩ )`
`str( ⟨number⟩, ⟨length⟩ )`
`str( ⟨number⟩ )`

Konvertiert einen numerischen Wert `⟨number⟩` zu einer Zeichenkette. Ist die Länge `⟨length⟩` vorhanden, wird die Zeichenkette von links beginnend mit dem Zeichen `⟨leadchar⟩` auf die vorgegebene Länge aufgefüllt. (Standard für `⟨leadchar⟩` ist »0«.)

Rückgabewert: String.

Beispiel:

```
$a = 42
print( str( $a         ) )  # Ausgabe: "42"
print( str( $a, 5      ) )  # Ausgabe: "00042"
print( str( $a, 5, " " ) )  # Ausgabe: "   42"
```

hex(⟨number⟩, ⟨digits⟩)
hex(⟨number⟩)

Konvertiert einen numerischen Wert zu einer Zeichenkette in hexadezimaler Schreibweise. Ist der Wert ⟨digits⟩ vorhanden, wird die Zeichenkette diesem Wert entsprechend oft mit dem Zeichen »0« aufgefüllt.

Rückgabewert: String.

Beispiel:

```
$a = 42
print( hex( $a, 2 ) )          # Ausgabe: "002A"
print( int( "0x" + hex(42) ) )  # Ausgabe: 42
```

len(⟨string⟩)

Liefert die Länge einer Zeichenkette.

Rückgabewert: Integer.

Beispiel:

```
print( len("abc") )  # Ausgabe: 3
```

pos(⟨substr⟩, ⟨string⟩, ⟨startpos⟩)
pos(⟨substr⟩, ⟨string⟩)

Liefert die Position der Teilzeichenkette ⟨substr⟩ in der Zeichenkette ⟨string⟩. Ist eine Startposition ⟨startpos⟩ angegeben, wird ab dieser Position in der Zeichenkette gesucht.

Rückgabewert: Position oder 0, wenn Teilzeichenkette nicht gefunden.

Beispiel:

```
print( pos( "b", "abcde"    ) )  # Ausgabe: 2
print( pos( "b", "abcde", 3 ) )  # Ausgabe: 0
```

copy(⟨string⟩, ⟨startpos⟩, ⟨length⟩)
copy(⟨string⟩, ⟨startpos⟩)

Liefert eine Teilzeichenkette ⟨string⟩ ab Position ⟨startpos⟩ mit der Länge ⟨length⟩. Ist die Länge ⟨length⟩ nicht angegeben, wird der Rest der Zeichenkette geliefert.

Rückgabewert: String.

(Beispiel auf der folgenden Seite)

```
print( copy( "abcde", 3, 1 ) )  # Ausgabe: "c"
print( copy( "abcde", 3    ) )  # Ausgabe: "cde"
```

delete(⟨string⟩, ⟨startpos⟩, ⟨length⟩)
delete(⟨string⟩, ⟨startpos⟩)

Löscht in der Zeichenkette ⟨string⟩ ab Position ⟨startpos⟩. Ist die Länge ⟨length⟩ nicht angegeben, werden alle Zeichen bis zum Ende der Zeichenkette gelöscht, ansonsten so viele Zeichen wie angegeben.

Rückgabewert: String.

Beispiel:

```
print( delete( "abcde", 3, 1 ) )  # Ausgabe: "abde"
print( delete( "abcde", 3    ) )  # Ausgabe: "ab"
$s = delete( $s, 1, 1 )  # 1. Zeichen von »$s« löschen
                         # und wieder in »$s« speichern
```

trim(⟨string⟩, ⟨trimchars⟩)
trim(⟨string⟩)

Entfernt alle führenden und folgenden Zeichen in der Zeichenkette ⟨string⟩, soweit sie in ⟨trimchars⟩ enthalten sind. Fehlt ⟨trimchars⟩, werden alle führenden und folgenden Leerzeichen entfernt.

Rückgabewert: String.

Beispiel:

```
print( trim("  42  ") )  # Ausgabe: "42"

$WHITESPACE = " " + chr(9)
print( trim( $line, $WHITESPACE ) )
```

lowercase(⟨string⟩)

Konvertiert die Zeichenkette ⟨string⟩ in Kleinbuchstaben.

Rückgabewert: String.

Beispiel:

```
print( lowercase( "AbCd" ) )  # Ausgabe: "abcd"
```

uppercase(⟨string⟩)

Konvertiert die Zeichenkette ⟨string⟩ in Großbuchstaben.

Rückgabewert: String.

Beispiel:

```
print( uppercase( "AbCd" ) )  # Ausgabe: "ABCD"
```

`replace( ⟨string⟩, ⟨find⟩, ⟨replace⟩, ⟨all⟩, ⟨ignorecase⟩ )`
`replace( ⟨string⟩, ⟨find⟩, ⟨replace⟩, ⟨all⟩ )`
`replace( ⟨string⟩, ⟨find⟩, ⟨replace⟩ )`

Liefert als Rückgabewert einen String, in dem alle Vorkommen von `⟨find⟩` in `⟨string⟩` durch `⟨replace⟩` ersetzt wurden. Ist der Wert `⟨all⟩` TRUE = wahr, dann werden alle gefundenen Zeichenketten ersetzt, ansonsten nur das erste Vorkommen. Ist der Wert `⟨ignorecase⟩` TRUE = wahr, wird die Groß-/Kleinschreibweise nicht beachtet.

Rückgabewert: String.

Beispiel:

```
print( replace("abcABCabc", "b", "[X]" )           ) # Ausgabe: "a[X]cABCabc"
print( replace("abcABCabc", "b", "[X]", true )     ) # Ausgabe: "a[X]cABCa[X]c"
print( replace("abcABCabc", "b", "[X]", true, true )) # Ausgabe: "a[X]cA[X]Ca[X]c"
```

`DecodeMIMEHeaderString( ⟨string⟩, ⟨charset⟩)`
`DecodeMIMEHeaderString( ⟨string⟩)`

Glossar: Base64, Quoted-printable

Liefert die decodierte Version einer gemäß MIME base64- oder quoted-printable-codierten Zeichenkette zurück. Die Variable `⟨charset⟩` liefert, falls vorhanden, den Zeichensatz zurück, mit dem der String codiert war. Der Rückgabewert enthält nach erfolgreicher Decodierung den 8-Bit-Zeichencode an Stelle der ehemals MIME-codierten 7-Bit-Zeichenkette. Die Bedeutung der zurückgegebenen Zeichenkette ist von der Variable `⟨charset⟩` abhängig und muss nicht mit dem Windows-Zeichensatz übereinstimmen.

Beispiel:

```
Warning(Addlog(DecodeMimeHeaderString("=?ISO-8859-15?Q?0.02=80?="),3))
```

`DecodeToLocalCharset( ⟨Text⟩, ⟨Zeichensatz⟩ )`

Liefert den in `⟨Text⟩` enthaltenen codierten Text decodiert im lokalen Zeichensatz zurück. `⟨Zeichensatz⟩` muss den Zeichensatz des zu decodierenden Strings enthalten.

`UTF8toUCS32( ⟨string⟩ )`

Entfernt die UTF-Transportcodierung einer UTF-8-codierten Zeichenkette und liefert die 32 Bit (4 Byte) langen equivalenten UCS-Werte zurück.

`UTF7toUCS16( ⟨string⟩ )`

Entfernt die UTF-Transportcodierung einer UTF-7-codierten Zeichenkette und liefert die 16 Bit (2 Byte) langen equivalenten UCS-Werte zurück.

`DecodeBase64 ( ⟨string⟩ )`

Decodiert base64-codierte Texte.

`DecodeQP ( ⟨string⟩ )`

Decodiert quoted-printable-codierte Texte.

`eval( ⟨string⟩ )`

Verwendet den Inhalt der Zeichenkette `⟨string⟩` als Ausdruck und liefert das Ergebnis zurück.

Rückgabewert: Integer.

Beispiel:

```
print(       "4" + "+" + "2"    )  # Ausgabe: "4+2"
print( eval( "4" + "+" + "2" ) )  # Ausgabe: 6
```

`RE_Match( ⟨string⟩, ⟨regex⟩ )`

Liefert »TRUE«, wenn `⟨string⟩` zum Regulären Ausdruck `⟨regex⟩` passt (siehe auch die FAQ »Hamster und Reguläre Ausdrücke«).

Rückgabewert: TRUE (1) = Treffer oder FALSE (0) = kein Treffer.

Beispiel:

```
$answer = "abc4efg2hij"
if( RE_Match( $answer, "4.*2" ) )
  print( "OK." )
endif
```

`RE_Extract( ⟨string⟩, ⟨regex⟩ )`

Liefert den Teil von `⟨string⟩`, welcher zum Regulären Ausdruck `⟨regex⟩` passt (siehe auch die FAQ »Hamster und Reguläre Ausdrücke«).

Rückgabewert: String (ist leer, wenn nichts gefunden wurde).

Beispiel:

```
$answer = "abc4efg2hij"
print( RE_Extract( $answer, "4.*2"      ) )  # Ausgabe: "4efg2"
print( RE_Extract( $answer, "A.*C"      ) )  # Ausgabe: "abc"
print( RE_Extract( $answer, "(?-i)A.*C" ) )  # Ausgabe: ""
```

`RE_Parse( ⟨string⟩, ⟨regex⟩, ⟨var1⟩ [ , ⟨var2⟩ , ... ] )`

Teilt den `⟨string⟩` in Gruppen auf entsprechend dem gruppierten Regulären Ausdruck `⟨regex⟩`. Die Gruppen werden in den einzelnen Variablen `⟨varX⟩` gespeichert. Siehe auch die FAQ »Hamster und Reguläre Ausdrücke«.

Rückgabewert: TRUE (1) = OK oder FALSE (0) = Fehler.

Beispiel:

```
$line = "Subject: Das ist ein Test"
RE_Parse( $line, "(\S+:)\s+(.*)", $a, $b )  # Ausgabe: $a="Subject:",
                                            # $b="Das ist ein Test"
```

`RE_Split( ⟨string⟩, ⟨regex⟩, ⟨var1⟩, ⟨var2⟩ [ , ⟨var3⟩ , ... ] )`

Teilt eine Zeichenkette in einzelne Abschnitte auf, der Reguläre Ausdruck `⟨regex⟩` gibt dabei die erwünschte/gesuchte Trennstelle an. Die Teilabschnitte werden in den einzelnen Variablen `⟨varX⟩` gespeichert. Siehe auch die FAQ »Hamster und Reguläre Ausdrücke«.

Rückgabewert: TRUE (1) = OK oder FALSE (0) = Fehler.

Beispiel:

```
$line = "Das  ist    ein   Test" # Zuerst 2, dann 4 und zuletzt 3 Leerzeichen
RE_Split( $line, "\s+", $a, $b )   # Ausgabe: $a = "Das", $b = "ist    ein   Test"
RE_Split( $line, "\s+", $a, $b, $c )  # Ausgabe: $a = "Das", $b = "ist",
                                      # $c = "ein   Test"
RE_Split( $line, "\s+", $a, $b, $c, $d ) # Ausgabe: $a = "Das", $b = "ist",
                                         # $c = "ein", $d = "Test"
```

Zeit-Funktionen

`ticks`

Liefert die Zeit in Millisekunden (= 1/1000 Sekunden) seit dem letzten Start von Windows.

Vorsicht! *Diese Funktion unterliegt Beschränkungen im Wertebereich: Sie funktioniert nur die ersten 2 147 483 647 (= $2^{31} - 1$) ms korrekt, was einem Wert von ungefähr 24 Tagen, 20 Stunden, 31 Minuten und 23 Sekunden entspricht.*

Rückgabewert: Integer.

Beispiel:

```
$stop = ticks
print( "Zeit: " + $stop + " ms seit letztem Windows-Start" )
```

`time`

Liefert die aktuelle Zeit im Unix-Format (Sekunden seit dem 01.01.1970).

Rückgabewert: Integer.

Beispiel:

```
$now = time
# 1 Tag = 24 Std., 1 Std. = 60 min, 1 min = 60 s
# => 1 Tag = 24*60*60 Sekunden
$yesterday = $now - 24*60*60
```

`timegmt`

Liefert die aktuelle GMT (Greenwich Mean Time) im Unix-Format (Sekunden seit dem 01.01.1970).

Rückgabewert: Integer.

`RFCTimeZone`

Liefert die Zeitdifferenz zwischen GMT (Greenwich Mean Time) und Ortszeit unter Berücksichtigung der Sommerzeit zurück.

Für Deutschland wird +100 (Normalzeit) bzw. +0200 (Sommerzeit) zurückgeliefert.

Rückgabewert: String.

`decodetime( ⟨time⟩, ⟨yyyy⟩ [ , ⟨mm⟩, ⟨dd⟩, ⟨hh⟩, ⟨nn⟩, ⟨ss⟩, ⟨wd⟩ ] )`

Konvertiert das Unix-Zeitformat in Jahr, Monat, Tag, Stunde, Minute, Sekunde und Wochentag. Variablen: ⟨yyyy⟩ (Jahr), ⟨mm⟩ (Monat), ⟨dd⟩ (Tag), ⟨hh⟩ (Stunden), ⟨nn⟩ (Minute), ⟨ss⟩ (Sekunde), ⟨wd⟩ (Tag in der Woche: 1 = Sonntag, ... , 7 = Sonnabend).

Rückgabewert: ⟨time⟩.

Hinweis: Die Rückgabewerte werden dabei ***nicht*** automatisch formatiert. Den Variablen ⟨mm⟩, ⟨dd⟩, ⟨hh⟩, ⟨nn⟩ und ⟨ss⟩ wird bei einstelligem Ergebnis keine führende Null angefügt.

Beispiel:

```
varset( $time123, 1234567890 )
var( $jahr, $mon, $tag, $std, $min, $sec )
decodetime( $time123, $jahr, $mon, $tag, $std, $min, $sec )
print( "Die UNIX-Zeit ", $time123, " entspricht dem ", _
$tag, ".", $mon, ".", $jahr, ", ", $std, ":", $min, ":", $sec, "." )
decodetime( time, 0, 0, 0, $std, $min, $sec )
print( "Aktuelle Zeit: ", $std, ":", $min, ":", $sec )
```

`encodetime( ⟨yyyy⟩, ⟨mm⟩, ⟨dd⟩, ⟨hh⟩, ⟨nn⟩, ⟨ss⟩ )`

Liefert die Unix-Zeit zurück. Parameter: ⟨yyyy⟩: Jahr, ⟨mm⟩: Monat, ⟨dd⟩: Tag, ⟨hh⟩: Stunden, ⟨nn⟩: Minuten, ⟨ss⟩: Sekunden.

Rückgabewert: ⟨time⟩.

Beispiel:

```
encodetime( 2000, 12, 31, 23, 59, 59 )
```

`GetUptimeSecs( )`
`GetUptimeMins( )`
`GetUptimeHours( )`
`GetUptimeDays( )`

Liefert den jeweiligen Anteil an der Uptime des Hamsters.

Fehlerbehandlung

`ErrCatch( ⟨bool⟩ )`
`ErrCatch`

Mit ⟨bool⟩ gleich »TRUE« (d. h. <> 0) wird die interne Fehlerbehandlung abgeschaltet. Der Hamster beendet dann im Fehlerfall das Skript nicht. Für die Fehlerbehandlung müssen in diesem Fall im Skript selber Maßnahmen zu Fehlerbehandlung getroffen werden. Der Fehlerstatus kann vom Skript mit der Funktion »ErrNum« abgefragt werden, im Falle eines Fehlers wird dabei ein Wert ungleich Null zurückgeliefert. Die dazu gehörende Fehlermeldung kann mit »ErrMsg« angezeigt werden.

Rückgabewert:
FALSE (0): Abgeschaltet (Standard, stoppt Skript im Fehlerfall);
TRUE (1): Eingeschaltet.

Beispiel:

```
$OldErrCatch = ErrCatch
ErrCatch( True )
print( 42/0 )  # Division durch Null verursacht Fehler
if( ErrNum<>0 )
  print( "Error: ", ErrNum, " ", ErrMsg )
endif
ErrCatch( $OldErrCatch )
```

`SysErrorMessage( ⟨errnumber⟩ )`

Liefert den Fehlertext zu einer Windows-System-Fehlernummer zurück.

`ErrNum`

Liefert die letzte Fehlernummer zurück.

Achtung: Die Fehlernummern sind versionsabhängig.

Rückgabewert: Integer (= 0: Kein Fehler, <> 0: Fehler).

Beispiel:

```
if( ErrNum <> 0 )
  # weitere Anweisungen
endif
```

`ErrMsg`

Liefert die letzte Fehlermeldung zurück.

Rückgabewert: String.

Beispiel:

```
print( "Letzter Fehler: ", ErrMsg )
```

`ErrModule`

Liefert den Namen des Moduls, welches den Fehler verursacht hat.

Rückgabewert: String.

Beispiel:

```
print( "Letzter Fehler verursacht durch ", ErrModule )
```

`ErrLineNo`

Liefert die Nummer der Zeile, welche den Fehler verursacht hat.

Rückgabewert: Integer.

`ErrLine`

Liefert die Zeile, welche den Fehler verursacht hat.

Rückgabewert: String.

`ErrSender`

Liefert den Namen des Skript-Objektes, welches den Fehler verursacht hat.

Rückgabewert: String.

Beispiel:

```
print( "Letzter Fehler in Skript '", ErrSender, "'" )
```

Dateien und Verzeichnisse

Siehe auch: ListLoad, ListSave, ListFiles, ListDirs.

`IniRead( ⟨filename⟩, ⟨section⟩, ⟨ident⟩, ⟨default⟩ )`

Liest aus dem Abschnitt ⟨`section`⟩ der INI-Datei ⟨`filename`⟩ den Wert des Schlüssels ⟨`ident`⟩ aus. Ist kein Dateiname angegeben, wird die Datei »`HScripts.ini`« verwendet. Ist kein Abschnitt ⟨`section`⟩ angegeben, wird der Abschnitt »[All]« verwendet. Ist der gewünschte Eintrag nicht in der INI-Datei enthalten, wird der Wert ⟨`default`⟩ zurückgeliefert.

Rückgabewert: String.

Beispiel:

```
$answer = IniRead( "MyScript.ini", "Antwort", "Letzte", "42")
$LastConnected = int( IniRead( "", "", "LastConnected", 0 ) )
```

IniWrite(⟨filename⟩, ⟨section⟩, ⟨ident⟩, ⟨value⟩)

Setzt bzw. ändert im Abschnitt ⟨section⟩ der INI-Datei ⟨filename⟩ den Wert für den Schlüssel ⟨ident⟩ auf den neuen Wert ⟨value⟩. Ist kein Dateiname angegeben, wird die Datei »HScripts.ini« verwendet. Ist kein Abschnitt ⟨section⟩ angegeben, wird der Abschnitt »[All]« verwendet.

Rückgabewert: Integer (–1, wenn ein Fehler auftrat, 0 bei fehlerfreier Ausführung).

Beispiel:

```
IniWrite( "MyScript.ini", "Antwort", "Letzte", $answer )
IniWrite( "", "", "LastConnected", time )
```

IniEraseSection(⟨filename⟩, ⟨section⟩)

Löscht einen Abschnitt ⟨section⟩ der Ini-Datei ⟨filename⟩.

Rückgabewert: Integer (–1, wenn ein Fehler auftrat, 0 bei fehlerfreier Ausführung).

IniDelete (⟨filename⟩, ⟨section⟩ ⟨ident⟩)

Löscht in einem Abschnitt ⟨section⟩ der Ini-Datei ⟨filename⟩ den Schlüssel ⟨ident⟩.

Rückgabewert: Integer (–1, wenn ein Fehler auftrat, 0 bei fehlerfreier Ausführung).

FileExists(⟨filename⟩)

Testet, ob die Datei ⟨filename⟩ vorhanden ist. Die Joker »?« und »*« sind im Dateinamen zugelassen.

Rückgabewert: Integer (1 = Datei existiert, 0 = Datei nicht vorhanden).

Beispiel:

```
if( !fileexists("myfile.cfg" ) )
  error( "Falsche Konfiguration!" )
endif
```

DiskFreeKB(⟨Laufwerksbuchstabe⟩)

Liefert den freien Speicherplatz eines Laufwerkes in Kibibytes (1 KiB = 1024 Byte) zurück.

Rückgabewert: Integer (bei fehlerhaftem Parameter: –1).

Beispiel:

```
DiskFreeKB("C")
```

FileSize(⟨filename⟩)

Liefert die Länge der Datei ⟨filename⟩ in Bytes.

Rückgabewert: Integer (–1 im Fehlerfall).

Beispiel:

```
print( FileSize( HamPath + "Hamster.exe" ) )
```

FileTime(⟨filename⟩)

Liefert die Zeit des letzten Schreibzugriffes auf die Datei ⟨filename⟩ im Unix-Format (Sekunden seit dem 01.01.1970).

Rückgabewert: Integer (–1 im Fehlerfall).

Beispiel:

```
print( "Your Hamster's age: ", time - FileTime( HamPath + "Hamster.exe" ), _
       " seconds" )
```

FileDelete(⟨filename⟩)

Löscht die Datei ⟨filename⟩. Die Joker »?« und »*« sind im Dateinamen zugelassen.

Rückgabewert: 0: OK, <> 0: Fehler.

Beispiel:

```
FileDelete( "MyScript.tmp" )
```

FileRename(⟨oldname⟩, ⟨newname⟩)

Benennt die Datei ⟨oldname⟩ in ⟨newname⟩ um.

Rückgabewert: 0: OK, <> 0: Fehler.

Beispiel:

```
FileRename( "MyScript.dat", "MyScript.bak" )
```

FileCopy(⟨oldname⟩, ⟨newname⟩)

Kopiert die Datei ⟨oldname⟩ nach ⟨newname⟩. Für ⟨oldname⟩ dürfen die Joker »?« und »*« verwendet werden.

Anstelle von ⟨newname⟩ kann auch ein Verzeichnispfad angegeben werden, der unbedingt mit einem Backslash »\« abgeschlossen werden muss. Anderenfalls wird die Angabe als Name der neuen Datei interpretiert, solange ⟨oldname⟩ keine Joker enthält. Um Verwirrungen zu vermeiden, sollten Sie jedoch auch im letzteren Fall den Backslash einfügen (vgl. Beispiele).

Damit Sie sich sicher sind, welches Ihr Arbeitsverzeichnis ist, sollten Sie unmittelbar zuvor die Funktion DirChange ausführen oder besser mit absoluten Pfaden arbeiten.

Rückgabewert: 0: OK, <> 0: Fehler.

Beispiele:

```
FileCopy( "MyScript.dat", "MyScript.bak" )

# Legt eine Kopie von »MyScript.dat« unter Beibehaltung
# des Namen im Unterordner »backups« an:
FileCopy ("MyScript.dat", "backups\")
```

```
# Legt eine Kopie von »MyScript.dat« unter dem Namen »backups«
# im Rootverzeichnis auf demselben Laufwerk an:
FileCopy ("MyScript.dat", "\backups")

# Kopiert alle Dateien mit der Endung ».dat« unter gleichem
# Namen in den Ordner »\backups\« auf demselben Laufwerk:
FileCopy ("*.dat", "\backups")
FileCopy ("*.dat", "\backups\") # bessere Variante
```

FileMove(⟨source⟩, ⟨destination⟩)

Verschiebt die Datei ⟨source⟩ nach ⟨destination⟩. Für ⟨source⟩ dürfen die Joker »?« und »*« verwendet werden.

Soll für ⟨destination⟩ ein Verzeichnispfad angegeben werden, muss dieser unbedingt mit einem Backslash »\« abgeschlossen werden. Anderenfalls wird die Angabe als Name der neuen Datei interpretiert, solange ⟨source⟩ keine Joker enthält. Um Verwirrungen zu vermeiden, sollten Sie jedoch auch im letzteren Fall den Backslash einfügen (vgl. Beispiele).

Damit Sie sich sicher sind, welches Ihr Arbeitsverzeichnis ist, sollten Sie unmittelbar zuvor die Funktion DirChange ausführen oder besser mit absoluten Pfaden arbeiten.

Rückgabewert: 0: OK, <> 0: Fehler.

Beispiel:

```
FileMove( "MyScript.dat", "MyScript.bak" )

# Verschiebt die Datei »MyScript.dat« unter Beibehaltung
# des Namen in den Unterordner »backups«:
FileMove ("MyScript.dat", ".\backups\")
# Verschiebt die Datei »MyScript.dat« in das Rootverzeichnis
# auf demselben Laufwerk und benennt sie in »backups« um:
FileMove ("MyScript.dat", "\backups")
# Verschiebt alle Dateien mit der Endung ».dat« unter gleichem
# Namen in den Ordner »\backups\« auf demselben Laufwerk:
FileMove ("*.dat", "\backups")
FileMove ("*.dat", "\backups\") # bessere Variante
```

DirExists(⟨dirname⟩)

Testet, ob das Verzeichnis ⟨dirname⟩ existiert. Sofern kein absoluter Pfad angegeben ist, bezieht sich die relative Angabe auf das aktuelle Arbeitsverzeichnis.

Rückgabewert: 1 = Verzeichnis existiert, 0 = Verzeichnis nicht vorhanden.

Beispiel:

```
if( !direxists( "mydir" ) )
  # weitere Anweisungen
endif
```

DirMake(⟨dirname⟩)

Erstellt ein Verzeichnis mit dem Namen ⟨**dirname**⟩. Sofern kein absoluter Pfad angegeben ist, bezieht sich die relative Angabe auf das aktuelle Arbeitsverzeichnis.

Rückgabewert: 0: OK, <> 0: Fehler.

Beispiel:

```
DirMake( HamPath + "MyScript" )
```

DirRemove(⟨dirname⟩)

Löscht das Verzeichnis ⟨**dirname**⟩, wenn es leer ist. Sofern kein absoluter Pfad angegeben ist, bezieht sich die relative Angabe auf das aktuelle Arbeitsverzeichnis.

Rückgabewert: 0: OK, <> 0: Fehler.

Beispiel:

```
DirRemove( HamPath + "MyScript" )
```

DirChange(⟨dirname⟩)

Setzt das Verzeichnis ⟨**dirname**⟩ als aktuelles Arbeitsverzeichnis.

Hinweis: Es ist im Normalfall sinnvoller, stattdessen in den anderen Funktionen absolute Verzeichnisangaben zu verwenden.

Rückgabewert: 0: OK, <> 0: Fehler.

Beispiel:

```
DirChange( HamPath + "MyScript" )
```

DirCurrent

Liefert das aktuelle Arbeitsverzeichnis zurück.

Rückgabewert: String.

DirWindows

Liefert den Namen des Windows-Verzeichnisses.

Rückgabewert: String.

DirSystem

Liefert den Namen des Windows-System-Verzeichnisses.

Rückgabewert: String.

Listen

Siehe auch »Listbox«.

`ListAlloc( ⟨sorted⟩, ⟨duplicates⟩ )`
`ListAlloc( ⟨sorted⟩ )`
`ListAlloc`

Erstellt eine neue Liste und liefert das zugehörige Handle zurück. Dieses Handle wird als Parameter in allen anderen Listen-Funktionen erwartet. Vor dem Skriptende muss es wieder mittels »ListFree« freigegeben werden, um Speicherlöcher zu vermeiden.

Ist der Parameter `⟨sorted⟩` gleich »TRUE« (<> 0), wird die Liste nach dem Zufügen neuer Einträge automatisch sortiert. Falls `⟨duplicates⟩` gleich »TRUE« (<> 0) ist, sind doppelte Einträge erlaubt, anderenfalls werden Einträge nur einmal in die Liste aufgenommen.

Hinweis: Voraussetzung für die Duplikat-Erkennung ist eine sortierte Liste.

Rückgabewert: >= 0: Handle für die Liste, –1: Zu viele Listen offen.

Beispiel:

```
$list = ListAlloc
```

`ListFree( ⟨listhdl⟩ )`

Löscht die Liste und gibt deren Speicherplatz und das Handle wieder frei. Siehe auch »ListAlloc«.

Rückgabewert: 0: OK, –1: Fehlerhaftes Handle.

Beispiel:

```
listfree( $list )
```

`ListExists( ⟨listhdl⟩, ⟨index⟩ )`
`ListExists( ⟨listhdl⟩ )`

Testet, ob das List-Handle `⟨listhdl⟩` (siehe »ListAlloc«) gültig ist und – sofern angegeben – ob der Eintrag Nr. `⟨index⟩` in der Liste existiert.

Rückgabewert: TRUE (1): gültig; FALSE (0): ungültig.

Beispiel:

```
sub DoSomethingWithList( $list )
  if( !ListExists( $list ) )
    error( "Liste fehlt!" )
  endif
  # weitere Anweisungen
endsub
```

`ListClear( ⟨listhdl⟩ )`

Löscht alle Einträge einer Liste, deren Handle `⟨listhdl⟩` ist (siehe »ListAlloc«).

Rückgabewert: 0: OK; –1: Fehlerhaftes Handle.

Beispiel:

```
ListClear( $list )
```

`ListCount( ⟨listhdl⟩ )`

Liefert die Anzahl der Listeneinträge. Siehe auch »ListAlloc«.

Rückgabewert: Integer.

Beispiel:

```
print( ListCount( $list ) )
```

`ListGet( ⟨listhdl⟩, ⟨index⟩ )`

Liefert den Eintrag in einer Liste mit dem Handle `⟨listhdl⟩` (siehe »ListAlloc«) zum Index `⟨index⟩`. Die Zählung der Einträge beginnt bei 0 und geht bis »ListCount – 1«.

Rückgabewert: String (leer, wenn Eintrag Nr. `⟨index⟩` nicht existiert).

Beispiel:

```
$i = 0
while( $i < ListCount($list) )
  print( $i, ": ", ListGet($list,$i) )
  inc( $i )
endwhile
```

`ListSet( ⟨listhdl⟩, ⟨index⟩, ⟨value⟩ )`
`ListSet( ⟨listhdl⟩, ⟨index⟩ )`

Setzt in einer Liste mit dem Handle `⟨listhdl⟩` (siehe »ListAlloc«) den Listeneintrag mit dem Index `⟨index⟩` auf den Wert `⟨value⟩`. Sollte der entsprechende Listeneintrag noch nicht existieren, so wird er angelegt, was die Nutzung einer Liste als Array-Ersatz erlaubt. Sollte kein `⟨value⟩` übergeben werden, bekommt der Eintrag einen Leerstring zugewiesen.

Rückgabewert: `⟨value⟩`.

Beispiel:

```
ListSet( $list, 42, "The answer!" )
```

`ListGetTag( ⟨listhdl⟩, ⟨index⟩ )`

Liefert die »Tag«-Eigenschaft des Listeneintrages Nr. `⟨index⟩` in einer Liste mit dem Handle `⟨listhdl⟩` (siehe »ListAlloc«). Diese Eigenschaft existiert zusätzlich zum Zeichenketteninhalt jedes Listeneintrages und enthält eine Zahl. Falls bislang nicht gesetzt, wird 0 zurückgeliefert.

Rückgabewert: Integer.

Beispiel:

```
$entry_itself = ListGet( $list, 42 )
$tag_of_entry = ListGetTag( $list, 42 )
```

ListSetTag(⟨listhdl⟩, ⟨index⟩, ⟨tag⟩)

Setzt die »Tag«-Variable für den Listeneintrag mit dem Index Nr. ⟨index⟩ in einer Liste mit dem Handle ⟨listhdl⟩ (siehe »ListAlloc«).

Rückgabewert: ⟨tag⟩.

Beispiel:

```
$index = ListAdd( $list, "The answer!" )
ListSetTag( $list, $index, 42 )
```

ListGetKey(⟨listhdl⟩, ⟨key⟩)

Enthält die Liste mit dem Handle ⟨listhdl⟩ (siehe »ListAlloc«) Zeichenketten in Form von »Schlüssel=Wert«, so liefert diese Funktion den Wert zum Schlüssel ⟨key⟩.

Rückgabewert: String: Wert zu ⟨key⟩.

Beispiel siehe unter »ListSetKey«.

ListSetKey(⟨listhdl⟩, ⟨key⟩, ⟨value⟩)
ListSetKey(⟨listhdl⟩, ⟨key⟩)

Enthält die Liste mit dem Handle ⟨listhdl⟩ (siehe »ListAlloc«) Zeichenketten in Form von »Schlüssel=Wert«, setzt diese Funktion den Wert zum Schlüssel ⟨key⟩ auf ⟨value⟩ bzw. auf eine leere Zeichenkette.

Rückgabewert: ⟨value⟩.

Beispiel:

```
$list = ListAlloc
ListSetKey( $list, "RTFM", "Read the f... manual!" )
ListSetKey( $list, "SCNR", "Sorry, could not resist!" )
print( ListGetKey( $list, "SCNR" ) )
```

ListAdd(⟨listhdl⟩, ⟨value⟩)
ListAdd(⟨listhdl⟩)

Fügt am Ende der Liste mit dem Handle ⟨listhdl⟩ (siehe »ListAlloc«) einen Listeneintrag mit dem Wert ⟨value⟩ hinzu. Ohne dessen Angabe wird ein leerer Wert eingetragen.

Rückgabewert: Index des neuen Eintrages.

Beispiel:

```
$index = ListAdd( $list, $text )
```

`ListDelete( ⟨listhdl⟩, ⟨index⟩ )`

Löscht den Listeneintrag mit dem Index `⟨index⟩` in einer Liste mit dem Handle `⟨listhdl⟩` (siehe »ListAlloc«), die restlichen Einträge wandern entsprechend.

Rückgabewert: 0

`ListInsert( ⟨listhdl⟩, ⟨index⟩, ⟨value⟩ )`
`ListInsert( ⟨listhdl⟩, ⟨index⟩ )`

Fügt in einer Liste mit dem Handle `⟨listhdl⟩` (siehe »ListAlloc«) einen Listeneintrag mit dem Index `⟨index⟩` und dem Inhalt `⟨value⟩` ein, die folgenden Einträge verschieben sich entsprechend. Ohne Angabe von `⟨value⟩` wird ein leerer Wert eingetragen.

Rückgabewert: 0

Beispiel:

```
ListInsert( $list, 0, $text )
```

`ListSort( ⟨listhdl⟩ )`

Sortiert die Liste mit dem Handle `⟨listhdl⟩` (siehe »ListAlloc«).

Rückgabewert: 0

Beispiel:

```
ListSort( $list )
```

`ListSetText( ⟨listhdl⟩, ⟨text⟩ )`

Löscht den Inhalt einer Liste mit dem Handle `⟨listhdl⟩` (siehe »ListAlloc«) und fügt ihr den Text `⟨text⟩`, zeilenweise getrennt durch CR/LF, der Liste hinzu.

Rückgabewert: `⟨text⟩`.

Beispiel:

```
ListSetText( $list, $article )
print( ListGet( $list, 0 ) )  # Ausgabe: Erste Zeile von »$article«
```

`ListGetText( ⟨listhdl⟩ )`

Liefert den Inhalt der gesamten Liste mit dem Handle `⟨listhdl⟩` (siehe »ListAlloc«) als Zeichenkette zurück, die einzelnen Einträge (Zeilen) der Liste werden dabei nur mit CR/LF voneinander getrennt.

Rückgabewert: String.

Beispiel:

```
$article = ListGetText( $list )
```

ListIndexOf(⟨listhdl⟩, ⟨value⟩)

Sucht in der Liste mit dem Handle ⟨listhdl⟩ (siehe »ListAlloc«) den Eintrag ⟨value⟩ und liefert dessen Index zurück.

Rückgabewert: >= 0: Index, −1: nicht gefunden.

Beispiel:

```
ListSetText( $list, $article )
$EndOfHdr = ListIndexOf( $list, "" )
```

ListLoad(⟨listhdl⟩, ⟨textfile⟩)

Füllt die Liste mit dem Handle ⟨listhdl⟩ (siehe »ListAlloc«) mit dem Inhalt der Textdatei ⟨textfile⟩ zeilenweise.

Rückgabewert: 0 = OK, −1 = ungültiges List-Handle, −2 = fehlende Zugriffsrechte, −3 = Datei nicht gefunden.

Beispiel:

```
ListLoad( $list, "MyScript.cfg" )
```

ListSave(⟨listhdl⟩, ⟨textfile⟩)

Speichert die Liste mit dem Handle ⟨listhdl⟩ (siehe »ListAlloc«) in die Textdatei ⟨textfile⟩.

Rückgabewert: 0 = OK, −1 = ungültiges List-Handle, −2 = fehlende Zugriffsrechte.

Beispiel:

```
ListSave( $list, "MyScript.cfg" )
```

ListAppend(⟨listhdl⟩, ⟨textfile⟩)

Fügt die Liste mit dem Handle ⟨listhdl⟩ (siehe »ListAlloc«) an die Textdatei ⟨textfile⟩ an. Existiert die Datei nicht, wird sie erzeugt.

Rückgabewert: 0 = OK, −1 = ungültiges List-Handle, −2 = fehlende Zugriffsrechte.

Beispiel:

```
ListAppend( $list, "MyScript.cfg" )
```

ListFiles(⟨listhdl⟩, ⟨filemask⟩, ⟨fullpath⟩)
ListFiles(⟨listhdl⟩, ⟨filemask⟩)

Füllt die Liste, deren Handle ⟨listhdl⟩ ist (siehe »ListAlloc«), mit den Dateinamen aus dem Verzeichnis ⟨filemask⟩. Ist der Parameter ⟨fullpath⟩ gleich »TRUE« (<> 0), werden die Dateien inklusive Pfad aufgeführt, anderenfalls nur die Dateinamen.

Rückgabewert: Integer; >= 0: OK (Anzahl der Dateien), < 0: Datei(en) nicht gefunden.

Beispiel:

```
ListFiles( $list, HamPath + "Mails\Mail.Out\*.msg", true )
```

`ListDirs( ⟨listhdl⟩, ⟨dirmask⟩, ⟨fullpath⟩ )`
`ListDirs( ⟨listhdl⟩, ⟨dirmask⟩ )`

Füllt die Liste, deren Handle `⟨listhdl⟩` ist (siehe »ListAlloc«), mit den Verzeichnisnamen aus dem Verzeichnis `⟨dirmask⟩`. Ist der Parameter `⟨fullpath⟩` gleich »TRUE« (<> 0), werden die kompletten Pfade aufgeführt, anderenfalls nur die Verzeichnisnamen.

Rückgabewert: Integer; >= 0: OK (Anzahl der Verzeichnisse), < 0: Verzeichnis(se) nicht gefunden.

Beispiel:

```
ListDirs( $list, HamPath + "Mails\*." )
```

`ListRasEntries( ⟨listhdl⟩ )`

Füllt die Liste, deren Handle `⟨listhdl⟩` ist (siehe »ListAlloc«), mit DFÜ-Netzwerk-Einträgen (Telefonbuch des DFÜ-Netzwerkes).

Rückgabewert: >= 0: OK (Anzahl der Einträge), < 0: Liste fehlerhaft.

Beispiel:

```
var( $RasConnection )
$RasConnection = SelectRasConnection
if( $RasConnection <> "" )
  print( "Wähle ", $RasConnection )
  # weitere Anweisungen
endif
quit

sub SelectRasConnection
  var( $RasList, $RasSelIndex, $RasSelName )
  $RasSelName = ""
  $RasList = ListAlloc
  if( ListRasEntries( $RasList ) > 0 )
    $RasSelIndex = ListBox( $RasList, "Wählen Sie aus:" )
    if( $RasSelIndex >= 0 )
      $RasSelName = ListGet( $RasList, $RasSelIndex )
    endif
  endif
  ListFree( $RasList )
  return( $RasSelName )
endsub
```

Ein- und Ausgabe

`print( ⟨text⟩ [ , ⟨text⟩ , ...] )`

Gibt den enthaltenen Text im Hamsterprotokoll aus (auf Info-Niveau).

Rückgabewert: `⟨text⟩`.

Beispiel:

```
varset( $answer , 42 )
print( "Und die richtige Antwort lautet: ", $answer )
```

warning(⟨text⟩ [, ⟨text⟩ , ...])

Bewirkt das Gleiche wie »print«, der Text ist allerdings gelb unterlegt und wirkt dadurch auffälliger.

Rückgabewert: ⟨text⟩.

Beispiel:

```
warning( "Hirnfehlfunktion! Keine Ideen mehr! Beispiel terminiert!" )
```

Addlog(⟨text⟩ , ⟨x⟩ [, ⟨ZeigeSkriptname⟩])

Bewirkt das Gleiche wie »print«, nur kann die Art des Eintrages mit dem Parameter ⟨x⟩ beliebig festgelegt werden. Der Parameter ⟨ZeigeSkriptname⟩ (boolesch) legt fest, ob zusätzlich der Skriptname im Log erscheint.

Gültige Werte für ⟨x⟩:

1 = Debug,	5 = Warning,
2 = Detail,	6 = Error,
3 = Info,	7 = Status, d. h. Eintrag im Status-
4 = System,	fenster »Aufträge (Threads)«.

Wenn Sie das Modul »hamster.hsm« in Ihr Skript einbinden, können Sie die dort vordefinierten Konstanten verwenden.

Rückgabewert: ⟨text⟩.

Beispiel:

```
Addlog( "Hirnfehlfunktion! Keine Ideen mehr! Beispiel terminiert!" ,5)
```

HamRotatelog

Ein neues Logfile wird angelegt .

MsgBox(⟨text⟩, ⟨caption⟩, ⟨style⟩)
MsgBox(⟨text⟩, ⟨caption⟩)
MsgBox(⟨text⟩)

Zeigt eine Textbox mit dem Text ⟨text⟩ und der Überschrift ⟨caption⟩ an. ⟨style⟩ legt den Typ, die Schalter und eventuell die Vorauswahl der Schalter für die Textbox fest, getrennt durch senkrechte Striche »|« (Standard: »Information« und »OK«-Schalter).

Typen (inkl. passendem Symbol): 0x40 (Information), 0x30 (Warnung), 0x20 (Frage), 0x10 (Fehler).

Schalter: 0x0 (OK), 0x1 (OK+Abbrechen), 0x2 (Abbrechen+Wiederholen+Ignorieren), 0x3 (Ja+Nein+Abbrechen), 0x4 (Ja+Nein), 0x5 (Wiederholen+Abbrechen).

Für die Vorauswahl der Schalter: 0x100 für den zweiten Schalter, 0x200 für den dritten Schalter.

Wenn Sie das Modul »`hwindows.hsm`« in Ihr Skript einbinden, können Sie die dort vordefinierten Konstanten verwenden.

Rückgabewert: 0 = Fehler, 1 = OK, 2 = Abbruch, 3 = Abbruch, 4 = Wiederholen, 5 = Ignorieren, 6 = Ja, 7 = Nein.

Beispiel:

```
if( MsgBox( "Hast Du etwa eine Frage?", "Wer stört?", 0x20|0x4|0x100 ) = 6 )
  Error( "Falsche Antwort." )
else
  MsgBox( "Na siehst Du, geht doch ..." )
endif
```

PopupBox(⟨text⟩, ⟨caption⟩, ⟨style⟩, ⟨timeout⟩, ⟨default⟩)

Zeigt eine Textbox mit dem Text ⟨text⟩ und der Überschrift ⟨caption⟩ an. ⟨style⟩ legt den Typ und die Schalter der Textbox fest (getrennt durch einen senkrechten Strich »|«, Standard: »Information« und »OK«-Schalter). Falls kein Schalter gedrückt wird, schließt das Fenster nach ⟨timeout⟩ Sekunden (maximal 4194304 s [= 48 d, 13 h, 5 min, 4 s]) automatisch, und gibt ⟨default⟩ zurück (⟨default⟩ muss keinem Schalter zugeordnet sein).

Typen (inkl. passendem Symbol): 0x40 (Information), 0x30 (Warnung), 0x20 (Frage), 0x10 (Fehler).

Schalter: 0x0 (OK), 0x1 (OK+Abbrechen), 0x2 (Abbrechen+Wiederholen+Ignorieren), 0x3 (Ja+Nein+Abbrechen), 0x4 (Ja+Nein), 0x5 (Wiederholen+Abbrechen).

Wenn Sie das Modul »`hwindows.hsm`« in Ihr Skript einbinden, können Sie die dort vordefinierten Konstanten verwenden.

Rückgabewert: 0 = Fehler, 1 = OK, 2 = Abbruch, 3 = Default, 4 = Wiederholen, 5 = Ignorieren, 6 = Ja, 7 = Nein.

Beispiel:

```
var( $result )
$result = PopupBox( "Ist da jemand?", "Hallo!", 0x04, 5, 3 )
icase( $result, 3, print( "Niemand da." ), _
                6, print( "Ich bin nicht allein." ), _
                7, print( "Lügner!" ) )
```

InputBox(⟨text⟩ [, ⟨caption⟩, ⟨default⟩, ⟨code⟩])

Zeigt in einer Texteingabebox den Text ⟨text⟩ mit der Überschrift ⟨caption⟩ an. Die Eingabezeile wird mit dem Text ⟨default⟩ vorbelegt. Ist die Variable ⟨code⟩ nach Rückkehr 1 (TRUE), wurde der Schalter »OK« betätigt, anderenfalls der Schalter »Abbrechen«.

Rückgabewert: String (leer, wenn der Schalter »Abbrechen« betätigt wurde).

Beispiel:

```
$answer = InputBox( "What's the answer ...?", "", "42", $ok )
```

`InputPW( ⟨text⟩ [ , ⟨caption⟩, ⟨default⟩, ⟨code⟩ ] )`

Zeigt in einer Texteingabebox den Text ⟨text⟩ mit der Überschrift ⟨caption⟩ an. Die Eingabezeile wird mit dem Text ⟨default⟩ vorbelegt, aber mit Sternchen verschleiert dargestellt. Ist die Variable ⟨code⟩ nach Rückkehr 1 (TRUE), wurde der Schalter »OK« betätigt, anderenfalls der Schalter »Abbrechen«.

Rückgabewert: String (leer, wenn der Schalter »Abbrechen« betätigt wurde).

Beispiel:

```
$answer = InputPW( "What's the answer ...?", "", "42", $ok )
```

`ListBox( ⟨list⟩, ⟨text⟩ [ , ⟨caption⟩, ⟨default⟩, ⟨code⟩ ] )`

Zeigt eine Listbox mit dem Text ⟨text⟩ und der Überschrift ⟨caption⟩ an und erlaubt die Auswahl aus der Liste ⟨list⟩. Der Parameter ⟨default⟩ legt fest, welcher Listeneintrag vorausgewählt angezeigt wird (Standard ist –1, gleichbedeutend mit »Keine Vorauswahl«). Ist die Variable ⟨code⟩ nach Rückkehr 1 (TRUE), wurde der Schalter »OK« betätigt, anderenfalls der Schalter »Abbrechen«.

Rückgabewert: Index des ausgewählten Eintrages.

Beispiel:

```
var( $list, $selindex )
$list = ListAlloc
ListAdd( $list, "41" )
ListAdd( $list, "42" )
ListAdd( $list, "43" )
$selindex = ListBox( $list, "What's the answer ...?" )
print( iif( $selindex=1, "Correct!", "Wrong!" ) )
ListFree( $list )
```

Artikel verwalten

`ArtAlloc( ⟨Text⟩ )`
`ArtAlloc`

Erstellt einen neuen Message-Puffer und liefert ein Handle dafür zurück. Dieses Handle wird in den anderen Message-Funktionen mit *⟨msg⟩* bezeichnet. Es ist solange gültig, bis es mit »ArtFree« wieder freigegeben wird. Ist der Parameter ⟨text⟩ vorhanden, wird er als Message verwendet, anderenfalls wird eine leere Message angelegt.

Rückgabewert: Integer.

Beispiel:

```
$msg = ArtAlloc
```

`ArtFree( ⟨msg⟩ )`

Gibt ein Message-Handle wieder frei. Siehe auch »ArtAlloc«.

Rückgabewert: 0 = OK, −1 = Fehlerhaftes Handle.

Beispiel:

```
ArtFree( $msg )
```

ArtLoad(⟨msg⟩, ⟨filename⟩)

Lädt eine Textdatei ⟨**filename**⟩ in einen zuvor durch »ArtAlloc« angelegten Message-Puffer ⟨**msg**⟩.

Rückgabewert: 0 = OK, −1 = Datei nicht gefunden, −2 = Fehler.

Beispiel:

```
ArtLoad( $msg, HamMailsOutPath + "42.msg" )
```

ArtSave(⟨msg⟩, ⟨filename⟩)

Speichert einen Message-Puffer ⟨**msg**⟩ (siehe »ArtAlloc«) in eine Datei ⟨**filename**⟩.

Rückgabewert: 0 = OK, −2 = Fehler.

Beispiel:

```
ArtSave( $msg, HamMailsOutPath + "42.msg" )
```

ArtGetHeaders(⟨msg⟩)
ArtGetBody(⟨msg⟩)
ArtGetText(⟨msg⟩)

Liefert von einem Message-Puffer (siehe »ArtAlloc«) den Header, den Body oder beides zusammen als String zurück.

Rückgabewert: String.

Beispiel:

```
$Hdrs = ArtGetHeaders( $msg )
$Body = ArtGetBody( $msg )
$Text = ArtGetText( $msg )
```

Hiervon zu unterscheiden ist die Funktion »ArtGetHeader«.

ArtSetHeaders(⟨msg⟩, ⟨headers⟩)
ArtSetBody(⟨msg⟩, ⟨body⟩)
ArtSetText(⟨msg⟩, ⟨text⟩)

Speichert den Inhalt eines Strings für die Header(⟨**header**⟩), den Body (⟨**body**⟩) oder beides zusammen (⟨**text**⟩) komplett in einem zuvor durch »ArtAlloc« angelegten Message-Puffer ⟨**msg**⟩.

Rückgabewert: 0 = OK, −1 = Fehlerhaftes Handle.

(Beispiel auf der folgenden Seite)

```
ArtSetHeaders( $msg, $Headers )
ArtSetBody( $msg, $Body )
ArtSetText( $msg, $Text )
```

Hiervon zu unterscheiden ist die Funktion »ArtSetHeader«.

ArtHeaderExists(⟨msg⟩, ⟨hdrname⟩)

Testet die Existenz einer Headerzeile mit dem Namen ⟨hdrname⟩ in einem Message-Puffer ⟨msg⟩ (siehe »ArtAlloc«).

Rückgabewert: False (0) = nicht vorhanden, True (1) = vorhanden.

Beispiel:

```
if( ArtHeaderExists( $msg, "X-Old-Subject" ) )
  # weitere Anweisungen
endif
```

ArtGetHeader(⟨msg⟩, ⟨hdrname⟩, ⟨separate⟩)
ArtGetHeader(⟨msg⟩, ⟨hdrname⟩)

Liefert aus einem Message-Puffer ⟨msg⟩ (siehe »ArtAlloc«) den Inhalt der Header-Zeile mit dem Namen ⟨hdrname⟩ zurück. Wenn die Header-Zeile nicht gefunden wurde, liefert die Funktion eine leere Zeichenkette zurück. Hat die Header-Zeile eine oder mehrere Folgezeilen, so liefert die Funktion, abhängig vom Parameter ⟨separate⟩, die Zeilen entweder getrennt mit einem Leerzeichen (⟨separate⟩ = 0) oder im Originalzustand (⟨separate⟩ = 1) zurück.

Rückgabewert: String.

Beispiel:

```
$Subject = ArtGetHeader( $msg, "Subject" )
```

Hiervon zu unterscheiden ist die Funktion »ArtGetHeaders«.

ArtAddHeader(⟨msg⟩, ⟨hdrname⟩, ⟨hdrval⟩)

Erweitert im Message-Puffer ⟨msg⟩ (siehe »ArtAlloc«) den Header um eine Zeile mit dem Namen ⟨hdrname⟩ und dem Inhalt ⟨hdrval⟩.

Rückgabewert: 0 = OK.

Beispiel:

```
ArtAddHeader( $msg, "Subject", "This is the subject!" )
```

ArtSetHeader(⟨msg⟩, ⟨hdrname⟩, ⟨hdrval⟩, ⟨xhdrname⟩)
ArtSetHeader(⟨msg⟩, ⟨hdrname⟩, ⟨hdrval⟩)

Ersetzt im Message-Puffer ⟨msg⟩ (siehe »ArtAlloc«) den Inhalt einer Header-Zeile mit dem Namen ⟨hdrname⟩ durch den Inhalt ⟨hdrval⟩. Mit dem Parameter ⟨xhdrname⟩ kann ein optionaler Name angegeben werden, um die vorhandene Header-Zeile in eine andere Header-Zeile unter diesem neuen Namen zu kopieren.

Rückgabewert: 0 = OK.

Beispiel:

```
ArtSetHeader( $msg, "Subject", "This is the new subject!", "X-Old-Subject" )
```

Hiervon zu unterscheiden ist die Funktion »ArtSetHeaders«.

`ArtDelHeader( ⟨msg⟩, ⟨hdrname⟩ )`

Löscht eine Header-Zeile `⟨hdrname⟩` aus dem Header einer Message im Message-Puffer `⟨msg⟩` (siehe »ArtAlloc«).

Rückgabewert: 0 = OK.

Beispiel:

```
ArtDelHeader( $msg, "X-Old-Subject" )
```

Zeitgeber/Scheduler

`AtAdd( ⟨subroutine⟩, ⟨von⟩ [ , ⟨bis⟩, ⟨wtage⟩, ⟨repeat⟩, ⟨modus⟩ ] )`

Fügt dem Scheduler einen Eintrag hinzu.

Der Parameter `⟨subroutine⟩` muss den Namen der auszuführenden Funktion enthalten, welche zum gewünschten Zeitpunkt durch »AtExecute« ausgeführt werden soll. Für die Funktion selbst darf kein Parameter angegeben werden.

Die Parameter `⟨von⟩` und `⟨bis⟩` müssen den Zeitraum im »HH:MM«-Format enthalten, in welchem die Subroutine zyklisch ausgeführt werden soll. Führende Nullen dürfen *nicht* weggelassen werden, »"24:00"« ist nur für den »`⟨bis⟩`«-Parameter erlaubt. Wird der Parameter `⟨bis⟩` weggelassen oder als Leerstring angegeben, steht `⟨von⟩` für einen *Zeitpunkt*.

Die Steuerung der Subroutine erfolgt über den Parameter `⟨wtage⟩`. Er muss die Wochentage als eine sieben Zeichen lange Zeichenkette enthalten. Jedes Zeichen darf den Zustand 1 (= Ausführen) oder 0 (= nicht Ausführen) annehmen und repräsentiert – beginnend mit Montag – einen Wochentag. Der Standardwert ist »1111111«, d. h. die Subroutine wird jeden Tag ausgeführt.

Der Parameter `⟨repeat⟩` muss die Zeitspanne in Minuten enthalten, nach welcher die Ausführung der Subroutine wiederholt werden soll. Der Standardwert »0« steht für die *einmalige* Ausführung.

Der Parameter `⟨modus⟩` entscheidet, wann die Ausführung der Subroutine beginnt: 0 = Erste Ausführung nach Ablauf des ersten Intervalls; 1 = Erste Ausführung sofort (Standard); 2 = Ausführung nur exakt zu den Minuten, die sich aus Anfangszeit und Minutenabstand ergeben. Ein Intervall beginnt immer mit der aktuellen Minute des Skriptstarts. *Die Option »2« wurde erst in der Hamsterversion 2.0.3.0 eingeführt. Die Verwendung von »true« statt »1« bzw. »false« für »0« ist weiterhin möglich.*

Rückgabewert: Index für den Schedulereintrag. Gibt die Funktion einen Wert ungleich 0 zurück, beendet der Scheduler seine Funktion und das Skript wird mit der nächsten Anweisung nach dem »AtExecute«-Aufruf fortgesetzt, vgl. Beispiel unter »AtExecute«.

Beispiel:

```
AtClear
AtAdd( getmail, "00:00", "06:00", "1111100", 60, 1 )
AtAdd( getmail, "06:00", "18:00", "1111100", 15, 1 )
AtAdd( getmail, "18:00", "24:00", "1111100", 60, 1 )
AtAdd( getmail, "12:00",   ""   , "0000011" )
AtAdd( getnews, "03:15" )
AtAdd( getnews, "18:45" )
AtExecute
quit

sub getmail
  # weitere Anweisungen
endsub

sub getnews
  # weitere Anweisungen
endsub
```

AtClear

Löscht alle Einträge des Schedulers.

Rückgabewert: 0.

Siehe Beispiel unter AtAdd.

AtExecute(⟨timeout⟩)
AtExecute

Wartet, bis eine der Subroutinen, die mit »AtAdd« dem Scheduler zugefügt wurden, einen Wert ungleich 0 zurückliefert oder bis die im Parameter ⟨timeout⟩ festgelegte Zeit abgelaufen ist. Das Skript wird dann mit der nächsten Anweisung nach dem »AtExecute«-Aufruf fortgesetzt, vgl. Beispiel.

Rückgabewert: -1: Timeout, > 0: Rückgabewert der zuletzt ausgeführten Subroutine.

Beispiel:

```
# loop.hsc
#!hs2
Atadd( loop, "00:00", "24:00", "1111111", 1, 2 )
Atadd( stoploop, "00:00", "24:00", "1111111", 3, 0 )
AtExecute
FirstAfter
quit

sub loop
```

```
  print( "Loop: Wird jede Minute ausgeführt." )
  return(0)
endsub

sub stoploop
  print( "StopLoop: Stoppt den Scheduler nach 3 Minuten." )
  return(42)
endsub

sub FirstAfter
  print( "Erste Anweisung nach dem 'AtExecute'." )
  return(1)
endsub
```

Logausgabe für Beispieldatei:

```
2010.10.23 12:34:56 Sys {4d1} Starting script: C:\HAMSTER\\ loop.hsc, Wait=0
2010.10.23 12:34:56 Sys {bb1} {script  loop.hsc} Start
2010.10.23 12:34:56 u1  {bb1} > Loop: Wird jede Minute ausgeführt.
2010.10.23 12:35:56 u1  {bb1} > Loop: Wird jede Minute ausgeführt.
2010.10.23 12:36:56 u1  {bb1} > Loop: Wird jede Minute ausgeführt.
2010.10.23 12:37:56 u1  {bb1} > Loop: Wird jede Minute ausgeführt.
2010.10.23 12:37:56 u1  {bb1} > StopLoop: Stoppt den Scheduler nach 3 Minuten.
2010.10.23 12:37:56 u1  {bb1} > Erste Anweisung nach dem 'AtExecute'.
2010.10.23 12:37:56 Sys {bb1} {script  loop.hsc} End
```

AtCount

Liefert die Anzahl der Einträge im Scheduler zurück.

Rückgabewert: Integer.

AtSubFunction(⟨Index⟩)

Liefert die zu einem Scheduler-Eintrag gehörige Subfunktion zurück.

Rückgabewert: String.

AtFrom (⟨Index⟩)

Liefert die zu einem Scheduler-Eintrag gehörige Anfangszeit zurück.

Rückgabewert: String.

AtUntil (⟨Index⟩)

Liefert die zu einem Scheduler-Eintrag gehörige Endzeit zurück.

Rückgabewert: String.

AtOnDays (⟨Index⟩)

Liefert die zu einem Scheduler-Eintrag gehörigen Wochentagseinträge zurück.

Rückgabewert: String.

`AtEveryMins ( ⟨Index⟩ )`

Liefert die zu einem Scheduler-Eintrag gehörige Wartezeit zurück.

Rückgabewert: Integer.

DFÜ-Netzwerk

Siehe auch die Funktion ListRasEntries.

`RasGetConnection`

Liefert den Namen der aktuellen DFÜ-Netzwerkverbindung zurück.

Rückgabewert: String (leer, wenn nicht vorhanden).

`RasLastError`

Liefert den Fehlerstatus der letzten DFÜ-Netzwerkverbindung zurück.

Rückgabewert: Integer.

Hinweis: Nach Beendigung einer DFÜ-Netzwerkverbindung benötigen Modems und andere DFÜ-Hardware eine gewisse Erholungszeit. Dieses führt dazu, dass diese Funktion erst nach einer kleinen Pause den korrekten Zustand anzeigt. Der konkrete Wert für diese Verzögerung ist von der verwendeten Hardware abhängig.

`RasErrText`

Liefert den Fehlertext der letzten DFÜ-Netzwerkverbindung zurück.

Rückgabewert: String.

Hinweis: Nach Beendigung einer DFÜ-Netzwerkverbindung benötigen Modems und andere DFÜ-Hardware eine gewisse Erholungszeit. Dieses führt dazu, dass diese Funktion erst nach einer kleinen Pause den korrekten Zustand anzeigt. Der konkrete Wert für diese Verzögerung ist von der verwendeten Hardware abhängig.

`RasIsConnected`

Testet, ob eine DFÜ-Netzwerkverbindung besteht.

Rückgabewert: True (1), wenn ja; False (0), wenn nein.

Hinweis: Nach Beendigung einer DFÜ-Netzwerkverbindung benötigen Modems und andere DFÜ-Hardware eine gewisse Erholungszeit. Dieses führt dazu, dass diese Funktion erst nach einer kleinen Pause den korrekten Zustand anzeigt. Der konkrete Wert für diese Verzögerung ist von der verwendeten Hardware abhängig.

Beispiel:

```
Sleep(1000)
if( !RasIsConnected )
  # Verbindung aufbauen
endif
```

`RasDial( ⟨name⟩, ⟨user⟩, ⟨pass⟩, ⟨connid⟩ )`

Stellt eine DFÜ-Netzwerkverbindung mit dem Namen `⟨name⟩` unter Verwendung von Benutzername `⟨user⟩` und Passwort `⟨pass⟩` her. Die Variable `⟨connid⟩` enthält dann das Handle für die Verbindung. Der Name der DFÜ-Netzwerkverbindung muss bereits in Windows unter »Arbeitsplatz« → »DFÜ-Netzwerk« (Telefonbuch des DFÜ-Netzwerkes) definiert worden sein (Groß- und Kleinschreibweise beachten).

Hinweis: Diese Funktion wird nicht von den passwort-spezifischen Funktionen des Hamsters unterstützt! Hierzu müssten stattdessen die Funktionen »HamRasDial« und »HamRasHangup« verwendet werden.

Rückgabewert: 0: Verbindung aufgebaut; > 0: Fehler-Code des DFÜ-Netzwerkes; –1: DFÜ-Netzwerk nicht installiert.

Beispiel:

```
$try = 1
while( RasDial("MyProvider","MyUsername","MyPassword") != 0 )
  if( $try >= 3 )
    error( "Dialing failed!" )
  endif
  inc( $try )
  sleep( 10000 )
endwhile
```

`RasHangup( ⟨connid⟩ )`
`RasHangup`

Beendet DFÜ-Netzwerkverbindungen. Ist die Variable `⟨connid⟩` nicht angegeben, werden alle DFÜ-Verbindungen beendet.

Rückgabewert: 0: OK; > 0: Fehler-Code; –1: DFÜ-Netzwerk nicht installiert.

`RasGetIP`

Liefert die (in den meisten Fällen dynamische) IP-Adresse, die beim Verbinden über das DFÜ-Netzwerk vom Provider zugewiesen worden ist.

Rückgabewert: IP-Adresse als String (leer, wenn nicht vorhanden).

Zeiger-Funktionen

`MemAlloc( ⟨size⟩ )`

Reserviert den Speicher mit der Größe `⟨size⟩` und liefert einen Zeiger auf diesen Speicherplatz zurück, der in den folgenden Funktionen als *⟨ptr⟩* bezeichnet wird. Vor dem Skriptende muss er wieder mittels »MemFree« freigegeben werden, um Speicherlöcher zu vermeiden.

`MemSize( ⟨ptr⟩ )`

Liefert die Größe des reservierten Speicherplatzes zurück, auf den der Zeiger `⟨ptr⟩` verweist (siehe MemAlloc).

`MemFree( ⟨ptr⟩ )`

Gibt den reservierten Speicherplatz wieder frei, auf den der Zeiger `⟨ptr⟩` verweist (siehe MemAlloc).

`MemForget( ⟨ptr⟩ )`

Leert den Speicher, auf den der Zeiger `⟨ptr⟩` verweist (siehe MemAlloc), ohne den reservierten Speicherplatz wieder freizugeben.

`MemSetInt( ⟨ptr⟩ , ⟨int⟩ )`

Schreibt den Inhalt des Integerparameters `⟨int⟩` in den vom Zeiger `⟨ptr⟩` (siehe MemAlloc) referenzierten Speicherplatz.

`MemGetInt( ⟨ptr⟩ )`

Liefert aus dem vom Zeiger `⟨ptr⟩` (siehe MemAlloc) referenzierten Speicherplatz einen Integerwert zurück.

`MemSetStr( ⟨ptr⟩ , ⟨str⟩ , ⟨len⟩ )`
`MemSetStr( ⟨ptr⟩ , ⟨str⟩ )`

Schreibt den Inhalt des Stringparameters `⟨int⟩` in den vom Zeiger `⟨ptr⟩` (siehe MemAlloc) referenzierten Speicherplatz. Mit `⟨len⟩` kann optional die Länge des zu kopierenden Strings angegeben werden.

Achtung! Es wird von einem NULL-terminierten String ausgegangen, falls der Parameter `⟨len⟩` nicht angegeben wurde.

`MemGetStr( ⟨ptr⟩ , ⟨len⟩ )`
`MemGetStr( ⟨ptr⟩ )`

Liefert aus dem vom Zeiger `⟨ptr⟩` (siehe MemAlloc) referenzierten Speicherplatz einen String zurück. Mit `⟨len⟩` kann optional die Länge des zu kopierenden Strings angegeben werden.

Achtung! Es wird von einem NULL-terminierten String ausgegangen, falls der Parameter `⟨len⟩` nicht angegeben wurde.

`MemVarPtr( ⟨var⟩ )`

Liefert einen Zeiger auf den Speicherplatz der Variablen `⟨var⟩` zurück.

Sonstige Funktionen

`execute( ⟨cmdline⟩ [ , ⟨workdir⟩, ⟨show⟩, ⟨wait⟩, ⟨exitcode⟩ ] )`

Startet das Programm ⟨`cmdline`⟩, ggf. inkl. dazugehörigen angehängten Parametern, im Arbeitsverzeichnis ⟨`workdir`⟩.

Der Parameter ⟨`show`⟩ legt die Eigenschaften des Programmfensters fest (0 = unsichtbar, 1 = normal, 2 = minimiert, 3 = maximiert; wenn Sie das Modul »`hwindows.hsm`« in Ihr Skript einbinden, können Sie die dort vordefinierten Konstanten verwenden).

Ist der Parameter ⟨`wait`⟩ TRUE = wahr, wartet das Skript auf das Ende des aufgerufenen Programms, anderenfalls wird die Ausführung des Skriptes unmittelbar fortgesetzt, während das aufgerufene Programm im Hintergrund arbeitet.

Die Variable ⟨`exitcode`⟩ enthält den Beendigungscode des aufgerufenen Programms. Bitte beachten Sie, dass die Kommandozeilen-API von Windows benutzt wird, und der Programmaufruf daher analog zur Kommandozeile erfolgt. Der Rückgabewert der Funktion **execute()** bezieht sich immer auf das Funktionieren der Windows-API, nicht auf die Funktion des aufgerufenen Programms. Die Variable ⟨`exitcode`⟩ liefert den Wert –1 zurück, wenn die Windows-API nicht fehlerfrei funktioniert hat. Daher ist immer der Status beider Variablen (⟨`exitcode`⟩ und Funktionsergebnis von **execute()**) zu prüfen. Wird die Funktion durch einen userbedingten Skript-Stop beendet, liefert sie gleichfalls –1 als Funktionsergebnis zurück.

Rückgabewert: Integer (0: Kein Fehler, <> 0: Fehler).

Beispiel:

```
execute( "notepad.exe myfile.cfg" )
```

`Shell( ⟨filename⟩ [ , ⟨modus⟩, ⟨path⟩, ⟨operation⟩ ] )`

Erlaubt den Aufruf von Dokumenten über die in der Registry definierte Verknüpfung. Bis auf den Dateinamen ⟨`filename`⟩ sind alle Parameter optional. Der Parameter ⟨`modus`⟩ legt die Eigenschaften des Programmfensters fest (0 = unsichtbar, 1 = normal, 2 = minimiert, 3 = maximiert). Der Pfad ⟨`path`⟩ wird mit der Pfadangabe des Dateinamens vorbelegt. Mit dem Parameter ⟨`operation`⟩ sind Angaben wie »open«, »print« oder »explore« gemeint. Welche Operationen gültig sind, hängt vom jeweiligen Datei-Typ ab.

Rückgabewert: Integer (> 32: Kein Fehler, <= 32: Fehler [Wert entspricht Errorlevel]).

Beispiel:

```
shell( "readme.txt" , 1 , "" , "open" )
```

`GetEnvironment( ⟨Variablenname⟩ )`

Liefert den Inhalt einer Umgebungsvariablen des Betriebssystems zurück.

`ClipRead()`

Liefert den Inhalt der Zwischenablage als Text zurück.

`ClipWrite( ⟨text⟩ )`

Schreibt den Text `⟨text⟩` in die Zwischenablage.

Hinweis: Skripte für den allgemeinen Gebrauch sollten die ClipWrite-Funktion vermeiden oder deutlich auf die Verwendung hinweisen, da der Inhalt der Zwischenablage üblicherweise vom Anwender und nicht von Programmen verändert zu werden hat!

`GetProcessIdentifier`

Liefert einen String zur eindeutigen Identifizierung der Hamster-Instanz zurück.

`random( ⟨maximum⟩ )`

Liefert eine Pseudo-Zufallszahl zwischen 0 und `⟨maximum⟩`.

Achtung! Die Funktion ist nur bei Verwendung der OpenSSL-Bibliotheken kryptographisch sicher. *Siehe auch die SSL-FAQ.*

`ScriptPriority( ⟨priority⟩ )`

Ändert die Priorität des eigenen Skriptes. Zulässiger Wertebereich: 0–6.

Wenn Sie das Modul »`hwindows.hsm`« in Ihr Skript einbinden, können Sie die dort vordefinierten Konstanten verwenden.

Achtung! *Die Verwendung von 6 = $THREAD_PRIORITY_TIME_CRITICAL kann unter Umständen zum Blockieren des Systems führen. Testen Sie die Skripte immer vorher aus, bevor Sie diesen Parameter verwenden.*

`digest( ⟨typ⟩ , ⟨string⟩ , ⟨form⟩ )`

Liefert ein krypthographisches Digest (Fingerabdruck) bzw. die Checksumme des Strings `⟨string⟩` im Format `⟨form⟩` zurück. Mögliche Typen für `⟨typ⟩` sind:

0 = (CRC32, Integer),
1 = (MD5, Zeichenkette mit 16 Zeichen),
2 = (SHA1, Zeichenkette mit 20 Zeichen).

Mögliche Formate:

0 = String,
1 = String mit hexadezimalem Inhalt.

Rückgabewerte: Integer bzw. String für CRC32 und String für MD5 oder SHA1.

`paramcount`

Liefert die Anzahl der dem Skript beim Start übergebenen Parameter zurück.

`paramstr( ⟨index⟩ )`

Liefert den dem Skript beim Start übergebenen Parameter mit der laufenden Nummer ⟨`index`⟩ zurück. Der erste Parameter hat dabei den Index 1, durch den Index 0 wird der Skriptname zurückgeliefert.

`SetTaskLimiter( ⟨limit⟩ )`

Setzt die Anzahl der gleichzeitig zu bearbeitenden Tasks für Verbindungen zum Provider auf ⟨`limit`⟩ und liefert den zuvor verwendeten Wert zurück.

`GetTasksActive`

Liefert die Anzahl der aktiven Tasks zurück.

`GetTasksWait`

Liefert die Anzahl der wartenden Tasks zurück.

`GetTasksRun`

Liefert die Anzahl der laufenden Tasks zurück.

`StopThread ( ⟨ThreadID⟩ )`

Stoppt den Thread mit dem Identifier ⟨`ThreadID`⟩.

Rückgabewert: 1 = Thread gestoppt.

`SetHostsEntry_ByName( ⟨hostname⟩ , ⟨IP-address⟩ )`
`SetHostsEntry_ByName( ⟨hostname⟩ )`

Trägt zu einem Hostnamen ⟨`hostname`⟩ die IP-Adresse in die Datei »`hosts`« ein. Ist der Parameter ⟨`IP-address`⟩ leer, wird der Eintrag aus der Datei entfernt.

`SetHostsEntry_ByAddr( ⟨IP-address⟩ , ⟨hostname⟩)`
`SetHostsEntry_ByAddr( ⟨IP-address⟩ )`

Trägt zu einer IP-Adresse den Hostnamen ⟨`hostname`⟩ in die Datei »`hosts`« ein. Bestehende Einträge werden überschrieben. Ist der Parameter ⟨`hostname`⟩ leer, wird der Eintrag aus der Datei entfernt.

`DeleteHostsEntry( ⟨hostname⟩ , ⟨IP-address⟩)`

Löscht einen Eintrag mit den Parametern ⟨`hostname`⟩ und ⟨`IP-address`⟩ aus der Datei »`hosts`«.

`localhostname`

Liefert den eigenen Hostnamen von der Netzwerkschnittstelle zurück.

`localhostaddr`

Liefert die momentane eigene IP-Adresse von der Netzwerkschnittstelle zurück.

`lookuphostaddr( ⟨hostname⟩ )`

Liefert die IP-Adresse zu einem Hostnamen aus dem DNS.

`lookuphostname( ⟨IP-address⟩ )`

Liefert den Hostnamen aus dem DNS zu einer IP-Adresse.

`XCounter( ⟨Nummer⟩ )`

Liefert den Inhalt der Fußzeilenvariablen ⟨Nummer⟩ zurück.

`ClearXCounter( ⟨Nummer⟩ , ⟨bis_Nummer⟩ )`
`ClearXCounter( ⟨Nummer⟩ )`

Löscht den Inhalt der Fußzeilenvariablen von ⟨Nummer⟩ bis zu ⟨bis_Nummer⟩. Fehlt der zweite Wert, wird nur der Zähler ⟨Nummer⟩ gelöscht.

`SetXCounter( ⟨Nummer⟩ , ⟨Wert⟩ )`

Verändert den Inhalt der Fußzeilenvariablen ⟨Nummer⟩ zu ⟨Wert⟩.

`DecXCounter( ⟨Nummer⟩ , ⟨Wert⟩ )`
`DecXCounter( ⟨Nummer⟩ )`

Verringert den Inhalt der Fußzeilenvariablen ⟨Nummer⟩ um ⟨Wert⟩ oder bei fehlender Angabe um 1.

`IncXCounter( ⟨Nummer⟩ , ⟨Wert⟩ )`
`IncXCounter( ⟨Nummer⟩ )`

Erhöht den Inhalt der Fußzeilenvariablen ⟨Nummer⟩ um ⟨Wert⟩ oder bei fehlender Angabe um 1.

`DllLoad( ⟨Name⟩ )`

Lädt die DLL ⟨Name⟩ in den Speicher und liefert ein Handle auf die geladene DLL zurück. Diese Funktion ist für die Ausführung von »DLLCall« zwar nicht zwingend notwendig, oft aber sinnvoll, Näheres siehe dort. System-Bibliotheken (z. B. »`kernel32.dll`«) sollten nicht geladen werden, da sie bereits vom Betriebssystem permanent vorgehalten werden.

Eine mit **DllLoad** geladene Bibliothek muss vor dem Skriptende auch wieder mittels »DLLFree« aus dem Speicher entfernt werden, damit Speicherlöcher vermieden werden.

Rückgabewert: <> 0: Handle der DLL; = 0: Fehler (siehe »DllLastError«).

`DLLFree( ⟨Handle⟩ )`

Gibt eine zuvor mit »DllLoad« geladene DLL wieder frei, wobei das von dieser Funktion zurückgelieferte Handle anzugeben ist.

`DllLastError`

Liefert den letzten Fehlerstatus der DLL-Funktionen zurück.

Achtung! Aufgrund eines Fehlers im Hamster 2.1.0.11 wird der Rückgabewert von `DllLastError` nicht zuverlässig gesetzt. Dieser Fehler wurde in der Beta-Version 2.1.0.1517 behoben.

`DLLCall( ⟨Deklaration⟩ [ , ⟨par1⟩ , ... , ⟨par11⟩ ] )`

Ruft eine Funktion in einer DLL auf. Ist die DLL bereits geladen, wird diese Instanz verwendet. Ist die DLL noch nicht geladen, so wird sie für jeden Aufruf temporär geladen und nach Ausführung der Funktion sofort wieder freigegeben. Daher empfiehlt sich bei mehrfacher Nutzung einer oder mehrerer Funktionen einer DLL die Verwendung von »DllLoad«. Das Laden der System-Bibliotheken (z. B. »`kernel32.dll`«) ist jedoch nicht erforderlich, da sie bereits vom Betriebssystem permanent vorgehalten werden, und sollte daher unterbleiben.

Der Parameter `⟨Deklaration⟩` muss die für die Aktivierung notwendigen vier Felder, jeweils getrennt durch einen senkrechten Strich, in folgendem Format enthalten:

`"⟨DLL_Name⟩|⟨Funktions_Name⟩|⟨Ergebnis_Typ⟩|⟨Parameter_Typen⟩"`

`⟨DLL_Name⟩`:	gültiger Dateiname einer Bibliothek.
`⟨Funktions_Name⟩`:	Name einer darin vorhandenen Funktion.
`⟨Ergebnis_Typ⟩`:	Typ des Rückgabewertes der Funktion. Folgende Werte sind hierfür zulässig: v = kein Rückgabewert vorhanden (»void«), n = Integer oder Pointer. Integer steht für alle Integertypen bis zu 32 Bit (auch BOOL, CHAR usw.). Die Interpretation des Wertes (Integer, Pointer oder Sonstiges) ergibt sich aus der Beschreibung der DLL.
`⟨Parameter_Typen⟩`:	Liste der Typen der einzelnen Funktionsparameter. Die Anzahl der Einträge in dieser Liste muss der Anzahl der Parameter der Funktionsdeklaration entsprechen (außer wenn die Funktion keine Parameter besitzt). Folgende Werte sind möglich: v = Funktion hat keinen Parameter (»void«), n = Integer oder Pointer, 1 = Callback-Funktion mit 1 Parameter (s. Anmerkung), 2 = Callback-Funktion mit 2 Parametern (s. Anmerkung).

Anmerkung zu Callback-Funktionen: Die Skriptsprache des Hamster bietet eigentlich keine Möglichkeit, Callback-Funktionen zu erstellen. Anstelle eines Pointers auf eine Funktion erwartet der Hamster als Parameter einen Pointer auf einen 4 Byte großen Puffer. In diesen »Pointer«-Puffer legt die DLLCall-Funktion einen weiteren Pointer auf den eigentlichen

»Daten«-Puffer ab. In diesen »Daten«-Puffer legt eine intern realisierte Callback-Funktion die empfangenen Daten als 32-Bit-DWORD ab. Das erste DWORD enthält die Anzahl der bislang empfangenen Datensätze. Die folgenden DWORDs enthalten jeweils den Wert des ersten Parameters der Callback-Funktion, der dieser bei ihrem Aufruf übergeben wurde (siehe unten Beispiel zu »EnumWindows«). Treffen weitere Daten ein, kann sich die Größe des »Daten«-Puffers und damit auch der Wert des Pointers auf den »Daten«-Puffer ändern. Der »Daten«-Puffer wird zwar von der DLLCall-Funktion alloziert, muss aber bei Skriptende *zusätzlich* zum ursprünglich allozierten »Pointer«-Puffer freigegeben werden (siehe Beispiel zu »EnumWindows«). In »⟨Parameter_Typen⟩« wird für diesen Puffer (an der betreffenden Stelle in der Reihenfolge) die Anzahl der Parameter der Callback-Funktion eingetragen (max. 9).

Die optionalen Parameter ⟨par1⟩ bis ⟨par11⟩ stehen für eventuell von der DLL verarbeitete Werte oder Variablen, deren Parametertyp innerhalb von ⟨Deklaration⟩ festgelegt wird, vgl. auch die Beispiele.

Achtung! Die Angabe einer falschen Parameteranzahl kann zu Fehlern im Programmablauf des Hamsters 2.1.0.11 führen, die seine Programmstruktur zerstören. Dieser Fehler wurde in der Beta-Version 2.1.0.1517 behoben.

Beispiele:

```
# HWND GetForegroundWindow(VOID)
  sub ApiGetForegroundWindow
  var( $hwnd )
  $hwnd = DllCall( "user32.dll|GetForegroundWindow|n|v" )
  return( $hwnd )
endsub

# int GetWindowText( HWND hWnd, LPTSTR lpString, int nMaxCount )
sub ApiGetWindowText( $hwnd )
  varset( $buf, MemAlloc(256) )
  DllCall( "user32.dll|GetWindowTextA|n|nnn", $hwnd, $buf, 255 )
  varset( $Result, MemGetStr($buf) )
  MemFree( $buf )
  return( $Result )
endsub

# BOOL SetWindowText( HWND hwnd, LPCTSTR lpsz )
sub ApiSetWindowText( $hwnd, $Text )
  varset( $buf, MemAlloc( len($Text)+1 ) )
  MemSetStr( $buf, $Text )
  DllCall( "user32.dll|SetWindowTextA|n|nn", $hwnd, $buf )
  MemFree( $buf )
endsub

# Parameter $WindowList erwartet ein List-Objekt (ListAlloc).
sub ApiEnumWindowsList( $WindowList )

  # BOOL EnumWindows( WNDENUMPROC lpEnumFunc, LPARAM lParam )
```

```
  # [ BOOL CALLBACK EnumWindowsProc( HWND hwnd, LPARAM lParam ) ]
  # →  callback function type »2«
  varset( $ApiEnumWindows, "user32.dll|EnumWindows|n|2n" )

  var( $Count, $ptr, $i, $k, $wnd, $txt )
  assert( ListExists($WindowList), "ApiEnumWindowsList: list expected!" )

  $Count = 0
  ListClear( $WindowList )

  # Alloc Pointer-Buffer:
  varset( $pWnds, MemAlloc(4) )  # handle to buffer on call/on return
  MemSetInt( $pWnds, 0 )

  if( DllCall( $ApiEnumWindows, $pWnds, 0 ) )

    # Give Callback function time to work:
    sleep(100)

    # Get Pointer to Data-Buffer:
    $ptr = MemGetInt( $pWnds )

    # First entry in buffer is the number of reported results:
    $Count = MemGetInt( $ptr )

    # Additional entries are the values reported to callback function:
    for( $i, 1, $count )

      # get next entry
      inc( $ptr, 4 )
      $wnd = MemGetInt( $ptr )
      $txt = ApiGetWindowText( $wnd )

      # add entry to result list
      $k = ListAdd( $WindowList, $txt )
      ListSetTag( $WindowList, $k, $wnd )
    endfor
  endif

  # Free Data-Buffer:
  MemFree( MemGetInt($pWnds) )

  # Free Pointer-Buffer:
  MemFree( $pWnds )

  return( $Count )
endsub
```

Für ein Anwendungsbeispiel mit intensiver Verwendung von »DLLCall« siehe auch das Skript »`Online-Check.hsc`« im FAQ-Beitrag »Kann ich den Hamster über einen Router betreiben?«.

Eingebaute Funktionen (Server)

- Allgemeine Hamster-Steuerung
- Hamster-Statusabfragen
- Pfade und Verzeichnisse
- Remote-Access-Steuerung
- Senden und Empfangen von E-Mails
- Senden und Empfangen von News-Artikeln
- Liste der Newsjobs bearbeiten
- Newsgruppen verwalten
- News-Artikel verwalten
- Scoring

Allgemeine Hamster-Steuerung

`HamChangePassword( ⟨AccountName⟩, ⟨OldPW⟩ , ⟨NewPW⟩ )`

Ersetzt für einen lokalen Benutzer ⟨Accountname⟩ das alte Passwort ⟨OldPW⟩ durch das neue ⟨NewPW⟩.

Rückgabewerte: 0 = Fehler im alten Passwort, 1 = Passwort geändert, 2 = Fehler beim Ändern.

`HamReloadPassword`

Liest die Password-Datei neu ein.

`HamMessage( ⟨msg⟩, ⟨parameter⟩ )`
`HamMessage( ⟨msg⟩ )`

Löst verschiedene Ereignisse im Hamster aus, die alternativ auch per Menü steuerbar sind.

Für jede Message-Option ⟨msg⟩ gibt es auch eine eigene »sprechende« Funktion – siehe dazu die nachfolgend aufgeführten Funktionen.

⟨msg⟩	⟨parameter⟩	Wirkung
1	Zeit in Millisek. (= 1/1000 Sek.) bis Programmende	Hamster beenden
2	0 = Zähler+Log, 1 = Zähler, 2 = Log, 3 = Fehlerliste	Setzt Zähler und Log-Anzeige zurück
3	0 = Stop, 1 = Start, 2 = Neustart, 3 = Status	Start/Stop lokaler NNTP-Server (siehe Anmerkung 1)
4	0 = Stop, 1 = Start, 2 = Neustart, 3 = Status	Start/Stop lokaler POP3-Server (siehe Anmerkung 1)
5	0 = Stop, 1 = Start, 2 = Neustart, 3 = Status	Start/Stop lokaler SMTP-Server (siehe Anmerkung 1)

Fortsetzung nächste Seite

Fortsetzung

⟨msg⟩	⟨parameter⟩	**Wirkung**
6	0 = Verstecken, 1= Anzeigen, 2 = Status	Anzeigen/Verstecken des Hauptfensters
7	0 = Verstecken, 1= Anzeigen, 2 = Status	Anzeigen/Verstecken des Tray-Icons
8	*(ignoriert)*	Menü: Online/Alle Aufträge beenden
9	0 = Test, 1 = Laden	Konfiguration erneut laden (siehe Anmerkung 2)
10	*(ignoriert)*	»IPAccess.hst« erneut laden
11	0 = Stop, 1 = Start, 2 = Neustart, 3 = Status	Start/Stop lokaler Remote-Control-Server (siehe Anmerkung 1)
12	1 = Öffnen, 0 = Schließen, 2 = Status	Öffnet, schließt oder liefert den Status des Browsers für die News-/Mailverzeichnisse (siehe Anmerkung 3)
13	1 = Öffnen, 0 = Schließen, 2 = Status	Öffnet, schließt oder liefert den Status des Skriptdialogs (siehe Anmerkung 3)
14	0 = Stop, 1 = Start, 2 = Neustart, 3 = Status	Start/Stop lokaler IMAP-Server (siehe Anmerkung 1)
15	1 = Öffnen, 0 = Schließen, 2 = Status	Öffnet, schließt oder liefert den Status des News-Killfilelog-Dialogs (siehe Anmerkung 3)
16	1 = Öffnen, 0 = Schließen, 2 = Status	Öffnet, schließt oder liefert den Status des Ladeauftragsdialogs (siehe Anmerkung 3)
17	1 = Öffnen, 0 = Schließen, 2 = Status	Öffnet, schließt oder liefert den Status des Mail-Killfilelog-Dialogs (siehe Anmerkung 3)

Anmerkung 1 zu ⟨msg⟩ = 3–5, 11, 14: Die Rückgabewerte für den Parameter »3« sind 0 = aus und 1 = an.

Anmerkung 2 zu ⟨msg⟩ = 9: Der Parameter »1« führt nur dann zum Laden der Konfiguration, wenn alle Konfigurationsfenster geschlossen sind (Rückgabewert der Funktion ist 1 = TRUE). Anderenfalls wird nichts ausgeführt (Rückgabewert 0 = FALSE). Beim Parameter »0« (Test) wird grundsätzlich nichts ausgeführt, sondern nur die entsprechenden Rückgabewerte ermittelt. Damit kann getestet werden, ob das Laden gegebenenfalls möglich ist (d. h. die Konfiguration geschlossen ist).

Anmerkung 3 zu ⟨msg⟩ = 12, 13, 15–17: Der Parameter »2« ermöglicht die Abfrage, ob das jeweilige Fenster offen (Rückgabewert 1 = TRUE) oder geschlossen ist (Rückgabewert 0 = FALSE).

Übrige Rückgabewerte:
0 = OK; Befehl ausgeführt.
1 = Error; Befehl fehlerhaft.
2 = Unbekannter Parameter
3 = GPF, Fließkommafehler

Wenn Sie das Modul »hamster.hsm« in Ihr Skript einbinden, können Sie die dort vordefinierten Konstanten verwenden.

Beispiele:

```
HamMessage( 2, 0 ) # Alle Zähler zurücksetzen

# Anzeigeproblem mit dem Tray-Icon unter Windows XP beheben:
HamMessage( 7, 0 )  # Verbergen
Sleep( 500 )
HamMessage( 7, 1 )  # Anzeigen
```

HamShutdown(⟨delay⟩)

Den Hamster beenden. Die Zeit bis Programmende ⟨delay⟩ muss in Millisekunden (= 1/1000 Sekunden) angegeben werden.

Entspricht »HamMessage(1,⟨x⟩)«.

HamStopAllTasks

Stoppt alle Aufträge. Entspricht »HamMessage(8,⟨x⟩)«.

HamResetCounters(⟨x⟩)

Setzt die Zähler/Protokolle zurück: ⟨x⟩ = 0 für Protokolle und Zähler, ⟨x⟩ = 1 nur Zähler, ⟨x⟩ = 2 nur Protokolle, ⟨x⟩ = 3 nur Fehler/Warnungs-Protokoll; bei Weglassen des Parameters gilt ⟨x⟩ = 0.

Entspricht »HamMessage(2,⟨x⟩)«.

HamNNTPServer(⟨x⟩)
HamSMTPServer(⟨x⟩)
HamPOP3Server(⟨x⟩)
HamIMAPServer(⟨x⟩)
HamReCoServer(⟨x⟩)

Steuert einen lokalen Server: ⟨x⟩ = 0 stoppt den Server, ⟨x⟩ = 1 startet den Server, ⟨x⟩ = 2 startet den Server neu, ⟨x⟩ = 3 erlaubt die Abfrage, ob der Server läuft (*Rückgabewert:* 0 = aus, 1 = an).

Vgl. »HamMessage([3/4/5/11/14],⟨x⟩)«.

HamMainWindow(⟨x⟩)

Schließt mit ⟨x⟩ = 0 und öffnet mit ⟨x⟩ = 1 das Hauptfenster. ⟨x⟩ = 2 liefert den Status des Hauptfensters zurück.

Entspricht »HamMessage(6,⟨x⟩)«.

Rückgabewert: Integer, 1 = sichtbar, 0 = unsichtbar.

HamTrayIcon(⟨x⟩)

Verbirgt mit ⟨x⟩ = 0 das Tray-Icon und zeigt es mit ⟨x⟩ = 1 an. ⟨x⟩ = 2 liefert den Status des Tray-Icons zurück.

Entspricht »HamMessage(7,⟨x⟩)«.

Rückgabewert: Integer, 1 = sichtbar, 0 = unsichtbar.

Beispiel:

```
# Anzeigeproblem mit dem Tray-Icon unter Windows XP beheben:
HamTrayIcon ( 0 )  # Verbergen
Sleep( 500 )
HamTrayIcon ( 1 )  # Anzeigen
```

HamReloadConfig(⟨x⟩)

Mit ⟨x⟩ = 1 werden die Einstellungen neu geladen, sofern kein Einstellungsdialog offen ist. Mit ⟨x⟩ = 0 kann überprüft werden, ob ein Einstellungsdialog offen ist (*Rückgabewert:* 1 = ja, 0 = nein).

Entspricht »HamMessage(9,⟨x⟩)«.

HamReloadIPAccess

Lädt die Datei »IPAccess.hst« erneut. Entspricht »HamMessage(10,⟨x⟩)«.

HamDialogEditDirs(⟨x⟩)

Bezieht sich auf den Browser für die News-/Mailverzeichnisse. ⟨x⟩ = 1 zum Öffnen, ⟨x⟩ = 0 zum Schließen, ⟨x⟩ = 2 zur Statusabfrage (*Rückgabewert:* 1 = offen, 0 = geschlossen).

Entspricht »HamMessage(12,⟨x⟩)«.

HamDialogScripts(⟨x⟩)

Bezieht sich auf den Dialog zum Verwalten/Ändern von Skripten und Modulen. ⟨x⟩ = 1 zum Öffnen, ⟨x⟩ = 0 zum Schließen, ⟨x⟩ = 2 zur Statusabfrage (*Rückgabewert:* 1 = offen, 0 = geschlossen).

Entspricht »HamMessage(13,⟨x⟩)«.

`HamDialogAddPull( ⟨x⟩ )`

Bezieht sich auf den Dialog zum Erstellen von Ladeaufträgen. `⟨x⟩` = 1 zum Öffnen, `⟨x⟩` = 0 zum Schließen, `⟨x⟩` = 2 zur Statusabfrage (*Rückgabewert:* 1 = offen, 0 = geschlossen).

Entspricht »HamMessage(16,⟨x⟩)«.

`HamDialogKillFileLog( ⟨x⟩ )`

Bezieht sich auf den Dialog zum Bearbeiten des News-Killfile-Logs. `⟨x⟩` = 1 zum Öffnen, `⟨x⟩` = 0 zum Schließen, `⟨x⟩` = 2 zur Statusabfrage (*Rückgabewert:* 1 = offen, 0 = geschlossen).

Entspricht »HamMessage(15,⟨x⟩)«.

`HamDialogMailKillFileLog( ⟨x⟩ )`

Bezieht sich auf den Dialog zum Bearbeiten des E-Mail-Killfile-Logs. `⟨x⟩` = 1 zum Öffnen, `⟨x⟩` = 0 zum Schließen, `⟨x⟩` = 2 zur Statusabfrage (*Rückgabewert:* 1 = offen, 0 = geschlossen).

Entspricht »HamMessage(17,⟨x⟩)«.

`HamFlush`

Schreibcache entleeren – speichert alle noch ungesicherten Daten (z. B. History) auf die Festplatte. Dieses ist insbesondere nach dem Import von Artikeln zu empfehlen.

Rückgabewert: 0

`HamPurge( ⟨option⟩, ⟨group⟩ )`
`HamPurge( ⟨option⟩ )`

Startet einen Auftrag zum Aufräumen von Hamsterdaten. Der Parameter `⟨option⟩` legt fest, was aufgeräumt werden soll. Er ist bitorientiert und kann durch logisches UND gebildet werden. (Bit 0 = 1 = Artikel, Bit 1 = 2 = History-Einträge, Bit 2 = 4 = Killfile-Log, Bit 3 = 8 = Mail-History-Einträge). Bei der Option 0x01 (»HAM_PURGEOPT_DONEWS«) kann mit dem Parameter `⟨group⟩` das Aufräumen auf eine einzelne Gruppe beschränkt werden.

Wenn Sie das Modul »`hamster.hsm`« in Ihr Skript einbinden, können Sie die dort vordefinierten Konstanten verwenden.

Rückgabewert: 0: OK/gestartet, <> 0: nicht gestartet.

Beispiel:

```
HamWaitIdle
HamPurge( 4 )
HamPurge (1,"hamster.de.newuser")
HamWaitIdle
```

`HamCheckPurge`

Ermittelt, ob der Hamster im Moment die Daten aufräumt.

Rückgabewert: 0: Daten werden im Moment nicht aufgeräumt, <> 0: Hamster putzt im Moment die Datenbank.

`HamRebuildGlobalLists`

Startet das Erzeugen der globalen Listen.

Rückgabewert: 0: OK/gestartet, <> 0: nicht gestartet.

Beispiel:

```
HamWaitIdle
HamRebuildGlobalLists
HamWaitIdle
```

`HamRebuildHistory`

Startet das Erzeugen der History-Dateien (»`.\Groups\Hist*.dat`«) auf Basis des aktuellen Artikelbestandes. Diese Funktion sollte nur benutzt werden, wenn der Hamster »idle« ist (siehe Funktion »HamWaitIdle«).

Rückgabewert: 0: OK/gestartet, <> 0: nicht gestartet.

Beispiel:

```
HamWaitIdle
HamRebuildHistory
HamWaitIdle
```

Hamster-Statusabfragen

`HamVersion( ⟨info⟩ )`
`HamVersion`

Übermittelt die Versionsnummer des Hamsters, wenn der Parameter `⟨info⟩` 0 ist oder fehlt. Bei den folgenden Werten für den Parameter `⟨info⟩` wird folgendes Ergebnis zurückgeliefert:

1 = Produktname,	3 = Homepage des jeweiligen Hamsters,
2 = Maintainer inklusive E-Mail-Adresse,	4 = Status der Version (beta oder stable).

Rückgabewert: String (z. B. »1.3.17.0«).

Beispiel:

```
print( "Hamster Version " + HamVersion )
```

`HamGetStatus( ⟨status⟩, ⟨n⟩ )`

Liefert Auskünfte über den momentanen Zustand des Hamsters.

HamGetStatus(1,⟨n⟩) meldet, ob der entsprechende lokale Server gestartet ist.

Glossar:
NNTP
POP3
SMTP
IMAP
ReCo

Für ⟨n⟩ gilt:

1 = Lokaler NNTP-Server,	4 = Lokaler IMAP-Server,
2 = Lokaler POP3-Server,	5 = Lokaler ReCo-Server.
3 = Lokaler SMTP-Server,	

Rückgabewert: 1, wenn der Server aktiv ist, anderenfalls 0.

HamGetStatus(2,⟨n⟩) meldet die Anzahl der offenen Verbindungen am entsprechenden lokalen Server.

Für ⟨n⟩ gilt:

1 = Lokaler NNTP-Server,	4 = Lokaler IMAP-Server,
2 = Lokaler POP3-Server,	5 = Lokaler ReCo-Server.
3 = Lokaler SMTP-Server,	

Rückgabewert: Anzahl der offenen Verbindungen am lokalen Server.

HamGetStatus(3,⟨n⟩) gibt Auskunft über angekommene/wartende News & Mails.

Für ⟨n⟩ gilt:

1 = Neue Artikel,	6 = Neue E-Mails,
2 = Geladene Artikel,	7 = Zu versendende E-Mails,
3 = Bekannte (History-gefilterte) Artikel,	8 = Per Message-ID nachgeladene Artikel,
4 = Gelöschte Artikel,	9 = Anzahl der Fehlerzeilen im Log,
5 = Zu versendende Artikel,	10 = Anzahl der Warnungszeilen im Log.

HamGetStatus(4,⟨n⟩) gibt Auskunft über die momentan aktiven Loglevel.

Für ⟨n⟩ gilt:
1 = Loglevel für Anzeige,
2 = Loglevel für Speichern in Log-Datei.

Rückgabewert: Integer.

Beispiel:

```
Warning(HamGetStatus(3,1))
```

HamMainFQDN

Liefert den allgemeinen FQDN des Hamsters zurück.

FQDN

HamMIDFQDN

Liefert den FQDN für Message-ID des Hamsters zurück.

HamEnvelopeFrom

Liefert den vom Hamster verwendeten »Envelope-From:«-Header zurück.

Header

HamPostmaster

Liefert den vom Hamster verwendeten Postmaster-Account zurück.

`HamUsenetAcc`

Liefert den vom Hamster verwendeten Newsmaster-Account zurück.

Pfade und Verzeichnisse

`HamPath` `HamLogsPath` `HamNewsOutPath`
`HamExePath` `HamServerPath` `HamNewsErrPath`
`HamHscPath` `HamGroupsPath` `HamMailsOutPath`
`HamHsmPath` `HamMailPath` `HamRCPath`

Liefert interne Hamsterpfade zurück. Konkret geht es um das Hauptverzeichnis, das Verzeichnis, in dem die Hamster-Programmdatei liegt (unterscheidet sich von ersterem nur, wenn manuell geändert, s. »hamster.ini«, Sektion »[Directories]«, Eintrag Main; ***Achtung***, *liefert den Pfad bis einschließlich Hamster 2.1.0.1534 versehentlich in Großbuchstaben*), den Suchpfad für Skripte (»hsc«), den Suchpfad für Module (»hsm«), das Verzeichnis für Protokolldateien (Logs), das Serververzeichnis, das Newsgruppenverzeichnis, das Mailverzeichnis, das Verzeichnis für ausgehende Postings, das Verzeichnis für Postings, die wegen Fehlern nicht ausgeliefert wurden, das Verzeichnis für ausgehende Mails und das Skript-Verzeichnis für den lokalen ReCo-Server (»*RC*«).

Glossar: ReCo

Rückgabewert: String mit dem vollständigen Pfadnamen und anschließendem Backslash (»\«).

Beispiel:

```
if( FileExists( HamPath + "Hamster.exe" ) )
  # weitere Anweisungen
endif
```

Remote-Access-Steuerung

`HamSetLogin( ⟨identifier⟩, ⟨user⟩, ⟨pass⟩ )`

Setzt temporär den Namen und das Passwort für eine DFÜ-Netzwerkverbindung. Der Name `⟨identifier⟩` muss exakt (inkl. Groß-/Kleinschreibung) mit dem Namen im DFÜ-Netzwerk übereinstimmen. Wird für das Passwort `⟨pass⟩` ein Fragezeichen verwendet, fragt der Hamster das Passwort erst bei Bedarf ab. Außerdem können für Benutzernamen `⟨user⟩` und Passwort auch im Menü »Einstellungen« → »Benutzerverwaltung & Passworte« ⇒ »Passworte« definierte Variablen verwendet werden, das Passwort bleibt dann leer (vgl. unten Beispiel 3: Variable ist »$42«).

Rückgabewert: 0: OK, <> 0: Fehler.

Beispiele:

```
HamSetLogin( "MyProvider", "MyUsername", "MyPassword" )
HamSetLogin( "MyProvider", "MyUsername", "?" )
HamSetLogin( "MyProvider", "$42", "" )
```

`HamRasDial( ⟨name⟩, ⟨user⟩, ⟨pass⟩ )`
`HamRasDial( ⟨name⟩, ⟨user⟩ )`
`HamRasDial( ⟨name⟩ )`

Funktion, welche die mit `⟨name⟩` bezeichnete DFÜ-Verbindung aus dem Telefonbuch des DFÜ-Netzwerkes für die Einwahl verwendet und die Anmeldung am Server mit Benutzernamen `⟨user⟩` und Passwort `⟨pass⟩` versucht. Falls `⟨user⟩` und/oder `⟨pass⟩` fehlen oder leer sind (»""«), werden die per »HamSetLogin« eingestellten Angaben genutzt oder auf die im Hamster gespeicherten Werte zum Aufbau der Verbindung zugegriffen. Außerdem können für Benutzernamen `⟨user⟩` und Passwort auch im Menü »Einstellungen« → »Benutzerverwaltung & Passworte« ⇒ »Passworte« definierte Variablen verwendet werden (im Beispiel unten »$42«). Siehe auch den Befehl »RasDial«.

Hinweis: Wenn »HamRasDial« verwendet wird, sollte die Verbindung mit »HamRasHangup« beendet werden.

Rückgabewert: 0: OK/Verbindung aufgebaut, > 0: Fehler-Code; -1: DFÜ-Netzwerk nicht installiert.

Beispiel:

```
$try = 1
while( HamRasDial( "MyProvider","$42" ) != 0 )
  if( $try >= 3 )
    error( "Dialing failed!" )
  endif
  inc( $try )
  sleep( 10000 )
endwhile
```

`HamRasHangup`

Schließt die aktuelle, mit »HamRasDial« geöffnete DFÜ-Netzwerkverbindung.

Rückgabewert: 0: OK/Verbindung abgebaut; > 0: Fehler-Code; -1: DFÜ-Netzwerk nicht installiert.

`HamIsIdle`

Gibt »TRUE« (1) zurück, falls momentan keine aktiven Prozesse wie Purgen, Newsabruf, Mailversand etc. laufen.

Hinweis: Verbindungen von (Mail-/News-)Clients zum lokalen Server zählen nicht als aktive Tasks.

Rückgabewert: Integer.

Beispiel:

```
if( HamIsIdle )
  # weitere Anweisungen
endif
```

`HamWaitIdle( ⟨timeout⟩ )`
`HamWaitIdle`

Wartet, bis der Zeitpunkt `⟨timeout⟩` (in Millisekunden = 1/1000 Sekunden) erreicht ist oder keine Aufträge mehr im Hamster aktiv sind. Wird −1 (nach Einbindung des Moduls »`hamster.hsm`« funktioniert auch die Konstante »$HAM_WAITIDLE_INFINITE«) oder kein Parameter für das Timeout verwendet, kehrt die Funktion nur dann zurück, wenn keine Aufträge mehr aktiv sind.

Hinweis: Die Aufträge/Tasks für die lokalen Server werden hierbei nicht berücksichtigt.

Rückgabewert: 0 = OK, 1 = Timeout, 2 = Interrupted (unterbrochen).

`HamThreadCount`

Liefert die Anzahl der momentan aktiven Tasks/Aufträge zurück.

Hinweis: Verbindungen von (Mail-/News-)Clients zum lokalen Server zählen nicht als aktive Tasks.

Rückgabewert: Integer.

Senden und Empfangen von Emails

`HamMailExchange( ⟨server-list⟩, ⟨mode⟩)`
`HamMailExchange`

Mit dieser Funktion werden zuerst neue Mails vom Mailserver abgerufen und danach Mails (falls vorhanden) versendet. Wird die Serverauswahl `⟨server-list⟩` übergangen oder leer gelassen (»""«), werden alle im Hamster definierten POP3- und SMTP-Server verwendet. Anderenfalls ist der Transfer auf die eingetragenen Server eingeschränkt, die Server müssen in `⟨server-list⟩` dann in der Form »Servername1,Port1;Servername2,Port2; ... « angegeben werden. Mit dem zusätzlichen Parameter `⟨mode⟩` kann der Server-Typ eingeschränkt werden:

Glossar: POP3 SMTP Port

0: Es werden die POP3- und SMTP-Server verwendet (Standard),
1: Es werden nur die POP3-Server verwendet,
2: Es werden nur die SMTP-Server verwendet.

Anmerkung: Sinnvoller war lange Zeit die Verwendung von »HamFetchMail« und »HamSendMail«, da mit diesen Befehlen auch die Mailaccounts und Mailfilter vom Hamster einbezogen werden können. Inzwischen allerdings wird dafür bei »HamMailExchange« auf die Einstellungen der POP3-Server im Hamster zurückgegriffen, so dass der Befehl in der Mehrzahl der Fälle ausreicht.

Rückgabewert: 0: OK/gestartet, <> 0: nicht gestartet.

Beispiele:

```
HamMailExchange
HamMailExchange( "mailserver1.example.com,smtp;mailserver2.example.com,pop3" )
```

`HamFetchMail( ⟨server⟩ [ , ⟨port⟩, ⟨user⟩, ⟨pass⟩, ⟨destuser⟩, ⟨filter⟩, ⟨LeaveOnServer⟩, ⟨SSLMode⟩, ⟨SSLVerify⟩, ⟨SSLCaFile⟩ ] )`

Glossar: POP3

Startet einen Auftrag zum Abholen der E-Mails bei einem POP3-Server ⟨server⟩. Ohne Angabe weiterer Parameter wird auf die Einstellungen der POP3-Server im Hamster zurückgegriffen, die Schreibung des Namen muss dabei exakt übereinstimmen – inkl. Groß-/Kleinschreibung.

Port

Sonst gilt: Der Port muss in ⟨port⟩ nur angegeben werden, wenn er vom Standardport »110« bzw. »pop3« abweicht (vgl. die Beispiele unten). Für den Benutzernamen ⟨user⟩ und das Passwort ⟨pass⟩ kann auch eine im Menü »Einstellungen« → »Benutzerverwaltung & Passworte« ⇒ »Passworte« definierte Variable verwendet werden, das Passwort bleibt dann leer (in den Beispielen: »$7« und »$42«). Für ⟨destuser⟩ kann ein lokaler Benutzer angegeben werden, welcher die E-Mail erhalten soll. Bleibt der Parameter leer, wird der im Menü »Einstellungen« konfigurierte Benutzer verwendet. Zusätzlich zum lokalen Benutzer kann mit ⟨filter⟩ auch ein Abschnitt in der Mailfilter-Datei »Mailfilt.hst« angegeben werden; bleibt er leer, wird auch hier wieder auf das Menü »Einstellungen« zugegriffen. Falls der Parameter ⟨LeaveOnServer⟩ = 1 oder »True« ist, werden die Mails auf dem Server belassen, bei ⟨LeaveOnServer⟩ = 0 oder »False« werden sie vom Server gelöscht; bei jedem anderen Wert gelten die Einstellungen des Servers bzw. die Standardeinstellung.

SSL

Achtung! Um SSL in Verbindung mit dem Hamster nutzen zu können, werden zwei zusätzliche DLL-Dateien benötigt; mehr dazu siehe in der SSL-FAQ.

TLS

Der Parameter ⟨SSLMode⟩ ist ein Integer und gibt an, ob und wie SSL/TLS verwendet wird:

0 = SSL/TLS abgeschaltet,	2 = SSL/TLS auf Standardport, wenn möglich,
1 = SSL/TLS auf separatem Port (POP3S),	3 = SSL/TLS auf Standardport wird erzwungen.

Der Parameter ⟨SSLVerify⟩ ist ebenfalls Integer und regelt die Überprüfung der fremden Server-Zertifikate:

0 = Zertifikatsüberprüfung abgeschaltet,	2 = Zertifikatsüberprüfung immer,
1 = Zertifikatsüberprüfung, falls Zertifikat vorgelegt,	3 = Zertifikatsüberprüfung immer und Vergleich des Serverzertifikats mit lokaler Kopie.

Mit ⟨SSLCaFile⟩ kann eine Datei zur Zertifikatsüberprüfung angegeben werden. Dabei wird der Standard-Wert (Menü »Einstellungen« → »Grundeinstellungen« ⇒ »SSL« → »Datei mit Zertifikaten zwecks Überprüfung«) überschrieben.

Rückgabewert: 0: OK/gestartet, <> 0: nicht gestartet.

Beispiele:

```
HamFetchMail("pop3-server.example.com")
HamFetchMail("pop3-server.example.com", "pop3", "john.doe", "secret", "john", "")
HamFetchMail("pop3-server.example.com", "10110", "$7", "", "jane", "")
HamFetchMail("pop3-server.example.com", "", "$42", "", "admin", "myfilters")
```

`HamSendMail( ⟨server⟩ [ , ⟨port⟩, ⟨from-select⟩, ⟨to-select⟩, ⟨SSLMode⟩, ⟨SSLVerify⟩, ⟨SSLCaFile⟩ ] )`

Glossar: SMTP

Startet einen Auftrag zum Senden der E-Mails zu einem SMTP-Server ⟨server⟩. Ohne Angabe weiterer Parameter wird auf die Einstellungen der SMTP-Server im Hamster zurückgegriffen, die Schreibung des Namen muss dabei exakt übereinstimmen – inkl. Groß-/Kleinschreibung.

Port

Sonst gilt: Der Port muss in ⟨port⟩ nur angegeben werden, wenn er vom Standardport »25« bzw. »smtp« abweicht (vgl. die Beispiele unten). Ist der Parameter ⟨from-select⟩ oder ⟨to-select⟩ nicht leer, werden nur E-Mails versendet, deren »Envelope-From« bzw. »Rcpt-To« dem Regulären Ausdruck in diesem Parameter entspricht (siehe auch die FAQ »Hamster und Reguläre Ausdrücke«).

SSL

Achtung! Um SSL in Verbindung mit dem Hamster nutzen zu können, werden zwei zusätzliche DLL-Dateien benötigt; mehr dazu siehe in der SSL-FAQ.

TLS

Der Parameter ⟨SSLMode⟩ ist ein Integer und gibt an, ob und wie SSL/TLS verwendet wird:

0 = SSL/TLS abgeschaltet, 1 = SSL/TLS auf separatem Port (SMTPS),	2 = SSL/TLS auf Standardport, wenn möglich, 3 = SSL/TLS auf Standardport wird erzwungen.

Der Parameter ⟨SSLVerify⟩ ist ebenfalls Integer und regelt die Überprüfung der fremden Server-Zertifikate:

0 = Zertifikatsüberprüfung abgeschaltet, 1 = Zertifikatsüberprüfung, falls Zertifikat vorgelegt,	2 = Zertifikatsüberprüfung immer, 3 = Zertifikatsüberprüfung immer und Vergleich des Serverzertifikats mit lokaler Kopie.

Mit ⟨SSLCaFile⟩ kann eine Datei zur Zertifikatsüberprüfung angegeben werden. Dabei wird der Standard-Wert (Menü »Einstellungen« → »Grundeinstellungen« ⇒ »SSL« → »Datei mit Zertifikaten zwecks Überprüfung«) überschrieben.

Rückgabewert: 0: OK/gestartet, <> 0: nicht gestartet.

Beispiele:

```
HamSendMail( "smtp1.example.com", "smtp", "john\.doe@example\.com" )
HamSendMail( "smtp2.example.com" )
```

`HamSendMailAuth( ⟨server⟩ [ , ⟨port⟩, ⟨user⟩, ⟨pass⟩, ⟨from-select⟩, ⟨to-select⟩, ⟨SSLMode⟩, ⟨SSLVerify⟩, ⟨SSLCaFile⟩ ] )`

SMTP-AUTH

Startet einen Auftrag zum Senden der E-Mails zu einem SMTP-Server ⟨server⟩ unter Benutzung des SMTP-Auth-Verfahrens. Ohne Angabe weiterer Parameter wird auf die Einstellungen der SMTP-Server im Hamster zurückgegriffen, die Schreibung des Namen muss dabei exakt übereinstimmen – inkl. Groß-/Kleinschreibung.

Glossar: Port

Sonst gilt: Der Port muss in ⟨port⟩ nur angegeben werden, wenn er vom Standardport »25« bzw. »smtp« abweicht (vgl. die Beispiele unten). Für den Benutzernamen ⟨user⟩ und das Passwort ⟨pass⟩ kann auch eine im Menü »Einstellungen« → »Benutzerverwaltung & Passworte« ⇒ »Passworte« definierte Variable verwendet werden, das Passwort bleibt dann leer (in den Beispielen: »$7« und »$42«). Ist der Parameter ⟨from-select⟩ oder ⟨to-select⟩ nicht leer, werden nur E-Mails versendet, deren »Envelope-From« bzw. »Rcpt-To« dem Regulären Ausdruck in diesem Parameter entspricht (siehe auch die FAQ »Hamster und Reguläre Ausdrücke«).

SSL

Achtung! Um SSL in Verbindung mit dem Hamster nutzen zu können, werden zwei zusätzliche DLL-Dateien benötigt; mehr dazu siehe in der SSL-FAQ.

TLS

Der Parameter ⟨SSLMode⟩ ist ein Integer und gibt an, ob und wie SSL/TLS verwendet wird:

0 = SSL/TLS abgeschaltet,	2 = SSL/TLS auf Standardport, wenn möglich,
1 = SSL/TLS auf separatem Port (SMTPS),	3 = SSL/TLS auf Standardport wird erzwungen.

Der Parameter ⟨SSLVerify⟩ ist ebenfalls Integer und regelt die Überprüfung der fremden Server-Zertifikate:

0 = Zertifikatsüberprüfung abgeschaltet,	2 = Zertifikatsüberprüfung immer,
1 = Zertifikatsüberprüfung, falls Zertifikat vorgelegt,	3 = Zertifikatsüberprüfung immer und Vergleich des Serverzertifikats mit lokaler Kopie.

Mit ⟨SSLCaFile⟩ kann eine Datei zur Zertifikatsüberprüfung angegeben werden. Dabei wird der Standard-Wert (Menü »Einstellungen« → »Grundeinstellungen« ⇒ »SSL« → »Datei mit Zertifikaten zwecks Überprüfung«) überschrieben.

Rückgabewert: 0: OK/gestartet, <> 0: nicht gestartet.

Beispiele:

```
HamSendMailAuth( "smtp3.example.com", "587", "username", "passwort", _
                 "@mail\.example\." )
HamSendMailAuth( "smtp3.example.com", "", "$42", "", "john\.doe@example\.com", _
                 "jane\.doe@" )
```

HamNewMail(⟨EnvelopeFrom⟩, ⟨EnvelopeTo⟩, ⟨Mailtext⟩)

Mit dieser Funktion wird eine neue E-Mail erstellt. Als ⟨EnvelopeFrom⟩ muss die E-Mail-Adresse des Absenders bzw. der Name des Accounts angegeben werden. ⟨EnvelopeTo⟩ muss die E-Mail-Adresse bzw. den Accountnamen des Empfängers enthalten, es können mehrere Empfänger angegeben werden, jeder mit Semikolon (»;«) getrennt. ⟨Mailtext⟩ muss in einer Variablen die komplette E-Mail enthalten, also mit Mailheadern und Body entsprechend RFC 2881/RFC 2882.

Header

Achtung! Wer diesen Befehl nutzt, sollte sein Skript gründlich lokal austesten, bevor er eine E-Mail ins Internet versendet.

Rückgabewert: 0, wenn die E-Mail ordnungsgemäß gespeichert werden konnte, anderenfalls 1.

Beispiel:

```
# Beispiel für einen Empfänger:
varset( $crlf, chr(13) + chr(10) )
var( $Header, $Body )
$Header = "FROM: name@mail.example.org" + $crlf + _
          "TO: another_name@mail.example.org" + $crlf + _
          "Subject: HamNewMail" + $crlf + $crlf
$Body   = "Hallo," + $crlf + _
          "dieses Beispielskript koennte/sollte man noch" + $crlf + _
          "gegebenenfalls mit dem Date:-Header erweitern," + $crlf + _
          "denn der gehoert laut RFC auch dazu."
HamNewMail("name@mail.example.org", "another_name@example.org", $Header + $Body)

# Beispiel für mehrere Empfänger (einer davon als Blindkopie):
varset( $crlf, chr(13) + chr(10) )
var( $Header_and_Body )
$Header_and_Body = "From: absender@example.org" + $crlf + _
                   "To: you@example.com" + $crlf + _
                   "Subject: Schwierige Frage" + $crlf + _
                   "Mime-Version: 1.0" + $crlf + _
                   "Content-Type: text/plain; charset=ISO-8859-1" + $crlf + _
                   "Content-Transfer-Encoding: 8bit" + $crlf + $crlf + _
                   "Hallo Du da!" + $crlf + _
                   "Mich quält schon länger eine Frage:" + $crlf _
                   "Wer bin ich und, wenn ja, wie viele?"
HamNewMail ("absender@example.org", "you@example.com;me@example.net", _
            $Header_and_Body)
```

Senden und Empfangen von News-Artikeln

`HamNewsJobsClear`

Löscht die gesamte Jobliste.

Rückgabewert: 0

`HamNewsJobsStart( ⟨server-list⟩ )`
`HamNewsJobsStart`

Startet die Abarbeitung der Jobliste für die in der Auswahl ⟨`server-list`⟩ angegebenen Server. Ist die Auswahl leer oder fehlt, werden alle im Hamster definierten Newsserver verwendet, ansonsten müssen die Server in der Form »Servername1,Port1;Servername2,Port2;...« angegeben werden.

Rückgabewert: < 0: Fehler, >= 0: Anzahl der gestarteten Aufträge.

Beispiel:

```
HamNewsJobsStart( "newsserver1.example.com,nntp;newsserver2.example.org,119" )
```

`HamNewsJobsPullDef( ⟨server-list⟩ )`
`HamNewsJobsPullDef`

Fügt die Ladeaufträge für Artikel automatisch der Jobliste zu, welche mit den in der Auswahl ⟨`server-list`⟩ angegebenen Servern ausgeführt werden können. Ist die Auswahl leer oder fehlt, werden alle im Hamster definierten Newsserver verwendet, ansonsten müssen die Server in der Form »Servername1,Port1;Servername2,Port2; ... « angegeben werden.

Rückgabewert: < 0: Fehler, >= 0: Anzahl der hinzugefügten Newsjobs.

Beispiel:

```
HamNewsJobsPullDef( "news1.example.net,nntp;news2.example.com,nntp" )
```

`HamNewsJobsPull( ⟨server⟩, ⟨group-select⟩ )`
`HamNewsJobsPull( ⟨server⟩ )`

Fügt einzelne Ladeaufträge für Artikel der Jobliste zu, welche mit dem Server ⟨`server`⟩ und den in der Gruppenselektion angegebenen Gruppen realisierbar sind. Ist die Gruppenselektion leer, werden alle verfügbaren Gruppen verwendet, anderenfalls nur solche, die zum Regulären Ausdruck in ⟨`group-select`⟩ passen (siehe auch die FAQ »Hamster und Reguläre Ausdrücke«).

Rückgabewert: < 0: Fehler, > =0: Anzahl der hinzugefügten Newsjobs.

Beispiele:

```
HamNewsJobsPull( "news.one.example.com" )
HamNewsJobsPull( "news.two.example.net", "^hamster\.((de|en)\.|admin\b)" )
```

`HamNewsPull( ⟨server-list⟩ )`
`HamNewsPull`

Damit wird der Abruf neuer Artikel vom Newsserver begonnen. Ist die Auswahl ⟨`server-list`⟩ leer oder fehlt, werden alle im Hamster definierten Newsserver verwendet, ansonsten müssen die Server in der Form »Servername1,Port1;Servername2,Port2; ... « angegeben werden.

Achtung: Dieser Befehl ist nur noch aus Gründen der Kompatibilität vorhanden und sollte nicht mehr verwendet werden *(siehe Beispiel unten)*.

Rückgabewert: 0: OK/gestartet, <> 0: nicht gestartet.

Beispiele:

```
HamNewsPull( "news1.example.com,nntp;news2.example.net,119" )
HamNewsPull

# Ersatz für HamNewsPost und HamNewsPull:
HamNewsJobsClear
HamNewsJobsPostDef
HamNewsJobsPullDef
HamNewsJobsStart
```

`HamNewsJobsPostDef( ⟨server-list⟩ )`
`HamNewsJobsPostDef`

Fügt die Sendeaufträge für die Artikel in »News.out« automatisch der Jobliste zu. Ist die Auswahl `⟨server-list⟩` leer oder fehlt, werden alle im Hamster definierten Newsserver verwendet, ansonsten müssen die Server in der Form »Servername1,Port1;Servername2,Port2; ... « angegeben werden.

Rückgabewert: < 0: Fehler, >= 0: Anzahl der hinzugefügten Newsjobs.

Beispiele:

```
HamNewsJobsPostDef( "newsserver1.example.org,119;newsserver2.example.com,nntp" )
HamNewsJobsPostDef
```

`HamNewsJobsPost( ⟨server⟩, ⟨group-select⟩, ⟨header-select⟩ )`
`HamNewsJobsPost( ⟨server⟩, ⟨group-select⟩ )`
`HamNewsJobsPost( ⟨server⟩ )`

Fügt einzelne Sendeaufträge für Artikel der Jobliste zu, welche mit dem in `⟨server⟩` angegebenen Server und den zum Regulären Ausdruck `⟨group-select⟩` passenden Gruppen realisierbar sind. Ist die Gruppenselektion leer, werden alle verfügbaren Gruppen verwendet. Der Reguläre Ausdruck `⟨header-select⟩` (siehe auch die FAQ »Hamster und Reguläre Ausdrücke«) kann zur erweiterten Auswahl genutzt werden: Nur zu dieser Bedingung passende Artikel werden dann versandt.

Hinweis: Der Hamster sendet nur Artikel an Server, die auch Ladeaufträge (Newspulls) für die entsprechende Gruppe haben, in welche der Artikel gesendet werden soll.

Rückgabewert: < 0: Fehler, > =0: Anzahl der hinzugefügten Newsjobs.

Beispiele:

```
HamNewsJobsPost( "news.example.com", ".*", "^X-Post-To: news.example.com" )
HamNewsJobsPost( "news.example.com", "^alt\." )
HamNewsJobsPost( "news.example.com" )
```

`HamNewsPost( ⟨server-list⟩ )`
`HamNewsPost`

Startet einen Auftrag, der neue Artikel (falls vorhanden) postet. Ist die Auswahl `⟨server-list⟩` leer oder fehlt, werden alle im Hamster definierten Newsserver verwendet, ansonsten müssen die Server in der Form »Servername1,Port1;Servername2,Port2; ... « angegeben werden.

Achtung: Dieser Befehl ist nur noch aus Gründen der Kompatibilität vorhanden und sollte nicht mehr verwendet werden *(siehe Beispiel unter »HamNewsPull«).*

Rückgabewert: 0: OK/gestartet, <> 0: nicht gestartet.

Beispiele:

```
HamNewsPost( "news1.example.net,nntp;news2.example.org,119" )
HamNewsPost
```

`HamNewsJobsFeed( ⟨server⟩, ⟨group-select⟩ )`
`HamNewsJobsFeed( ⟨server⟩ )`

Fügt einzelne Aufträge zum Austausch von Artikeln mit anderen Server der Jobliste zu, welche mit dem Server `⟨server⟩` und der Gruppenselektion angegebenen Gruppen realisierbar sind. Ist die Gruppenselektion leer, werden alle verfügbaren Gruppen verwendet, anderenfalls nur solche, die zum Regulären Ausdruck in `⟨group-select⟩` passen (siehe auch die FAQ »Hamster und Reguläre Ausdrücke«).

Achtung! *Diese Funktion ist nur für den Artikelaustausch von Servern untereinander bestimmt und nicht für den gewöhnlichen Artikelaustausch mit den Providern geeignet. Es wird bei diesem Befehl die Funktionalität des »IHAVE«-Kommandos (RFC 977) verwendet.*

Rückgabewert: < 0: Fehler, >= 0: Anzahl der hinzugefügten Newsjobs.

Beispiele:

```
HamNewsJobsFeed( "news.example.com,nntp" )
HamNewsJobsFeed( "news.example.com,nntp", "^alt\." )
```

Liste der Newsjobs bearbeiten

`HamNewsJobsGetCounter`

Liefert die Anzahl in der Auftragsliste vorhandenen Einträge.

Rückgabewert: Integer.

Siehe Beispiel unter »HamNewsJobsSetPriority«.

`HamNewsJobsGetServer( ⟨index⟩ )`

Liefert den Servernamen des Eintrages `⟨index⟩` in der Auftragsliste.

Rückgabewert: String.

Siehe Beispiel unter »HamNewsJobsSetPriority«.

`HamNewsJobsGetParam( ⟨index⟩ )`

Liefert den Parameter des Eintrages `⟨index⟩` in der Auftragsliste. Der Parameter ist vom Typ des Eintrages abhängig.

Rückgabewert: String.

Siehe Beispiel unter »HamNewsJobsSetPriority«.

HamNewsJobsGetType(⟨index⟩)

Liefert den Typ des Eintrages ⟨index⟩ in der Auftragsliste.

Rückgabewert: Jobtyp Integer:

JOBTYPE_INVALID = 0	JOBTYPE_GETBYMID = 3
JOBTYPE_SRVINFOS = 1	JOBTYPE_NEWSPULL = 4
JOBTYPE_NEWSPOST = 2	JOBTYPE_NEWSFEED = 5

Siehe Beispiel unter »*HamNewsJobsSetPriority*«.

HamNewsJobsGetPriority(⟨index⟩)

Liefert die Priorität des Eintrages ⟨index⟩ in der Auftragsliste zurück.

Rückgabewert: Integer.

Standard für die jeweiligen Typen (vgl. »HamNewsJobsGetType«) ist:

JOBPRIO_SRVINFOS: MaxInt-1 (= 2 147 483 646)
JOBPRIO_NEWSPOST: MaxInt-2 (= 2 147 483 645)
JOBPRIO_GETBYMID: MaxInt-3 (= 2 147 483 644)
JOBTYPE_NEWSPULL: Priorität entspricht der Anzahl der Artikel in der Gruppe
JOBPRIO_NEWSFEED: MaxInt-4 (= 2 147 483 643)

Siehe Beispiel unter »*HamNewsJobsSetPriority*«.

HamNewsJobsSetPriority(⟨index⟩ , ⟨new_priority⟩)

Setzt die Priorität ⟨new_priority⟩ für den Eintrag ⟨index⟩ in der Auftragsliste.

Wenn Sie das Modul »hamster.hsm« in Ihr Skript einbinden, können Sie für ⟨new_priority⟩ auch die dort vordefinierten Konstanten verwenden.

Rückgabewert: Integer.

Beispiel:

```
warning("Es wurden: ",HamNewsJobsGetCounter, " NewsJobs nicht ausgeführt")

for ($i, 0, HamNewsJobsGetCounter-1)
  warning("Index: " + Str($i) + _
          " Server: " + HamNewsJobsGetServer ($i) + _
          " Parameter: " + HamNewsJobsGetParam($i) + _
          " Type: " + HamNewsJobsGetType($i) + _
          " Priorität " + HamNewsJobsGetPriority($i) + ".")
  HamNewsJobssetPriority($i,10*$i)
endfor

for ($i, 0, HamNewsJobsGetCounter-1)
  warning("Index: " + Str($i) + _
          " Server: " + HamNewsJobsGetServer($i) + _
```

```
            " Parameter: " + HamNewsJobsGetParam($i) + _
            " Type: " + HamNewsJobsGetType($i) + _
            " Priorität " + HamNewsJobsGetPriority($i) + ".")
    endfor

    HamNewsJobsStart
```

`HamNewsJobsAdd( ⟨server⟩ , ⟨type⟩ , ⟨parameter⟩, ⟨priority⟩ )`

Fügt der Auftragsliste einen Eintrag hinzu. Server ⟨`server`⟩ und Parameter ⟨`parameter`⟩ müssen Zeichenketten (Strings) sein, Typ ⟨`type`⟩ und Priorität ⟨`priority`⟩ dagegen numerische Werte.

Wenn Sie das Modul »`hamster.hsm`« in Ihr Skript einbinden, können Sie für ⟨`type`⟩ und ⟨`priority`⟩ die dort vordefinierten Konstanten verwenden.

Rückgabewert: Integer.

Beispiel:

```
NewsJobsAdd("news.wuff.de.example.org",4,"hamster.test",10000)
```

`HamNewsJobsDelete( ⟨index⟩ )`

Löscht ausschließlich den Job an der Position ⟨`index`⟩ aus der Auftragsliste.

Rückgabewert: 0

`HamNewsJobsCheckActive( ⟨server-list⟩ )`
`HamNewsJobsCheckActive`

Stellt einen Job zum Abholen der Serverinfos (»active«) in die Newsjobsqueue ein. Mit dem Parameter ⟨`server-list`⟩ kann gezielt ein einzelner Server ausgewählt werden.

Rückgabewert: Anzahl der in die Newsjobsqueue eingetragenen Jobs. Im Fehlerfall wird eine 0 zurückgeliefert.

Newsgruppen verwalten

`HamGroupCount`

Liefert die Anzahl der im Hamster verfügbaren Newsgruppen (inkl. lokale Gruppen) zurück.

Rückgabewert: >=0: OK/Anzahl der Gruppen, <0: Fehler.

Siehe Beispiel unter »*HamGroupName*«.

`HamGroupIndex( ⟨groupname⟩ )`

Liefert den Index der in ⟨`groupname`⟩ angegebenen Newsgruppe.

Rückgabewert: >=0: OK/Index-Nummer, −1: unbekannte Gruppe.

`HamGroupName( ⟨index⟩ )`

Liefert den Namen einer Newsgruppe zu ihrem Index. Achtung, die Zählung der Indizes beginnt bei Null und geht bis »HamGroupCount – 1«!

Rückgabewert: String.

Beispiel:

```
var( $grpidx )
$grpidx = 0
while( $grpidx < HamGroupCount )
  print( HamGroupName( $grpidx ) )
  inc( $grpidx )
endwhile
```

`HamAddGroup( ⟨groupname⟩ [ , ⟨description⟩, ⟨grouptype⟩, ⟨maintainer⟩, ⟨postserver⟩ ] )`

Legt eine neue lokale Gruppe mit dem Namen `⟨groupname⟩` an. Optional können die Parameter `⟨description⟩` für eine Kurzbeschreibung (»Tagline«), `⟨grouptype⟩`, `⟨maintainer⟩` und `⟨postserver⟩` angegeben werden; alle sind Strings.

Mögliche Werte für `⟨grouptype⟩`:

"y" = Lese-/Schreibzugriff (Standard)
"n" = kein Schreibzugriff (nur Lesen erlaubt)
"m" = moderierte Gruppe
"g" = News-to-Mail-Gateway

Bei moderierten Gruppen und News-to-Mail-Gateways ist in `⟨maintainer⟩` die E-Mail-Adresse des Moderators bzw. der Mailingliste anzugeben.

Rückgabewert: 0 = OK/angelegt, –1 = nicht angelegt.

`HamDelGroup( ⟨groupname⟩ )`

Löscht eine Gruppe mit dem Namen `⟨groupname⟩` und alle möglicherweise vorhandenen Pulls für diese Gruppe.

Rückgabewert: 0 = OK/gelöscht, –1 = nicht gelöscht.

`HamAddPull( ⟨server⟩ , ⟨groupname⟩ )`

Legt einen neuen Ladeauftrag für die Gruppe `⟨groupname⟩` vom Newsserver `⟨server⟩` an. Falls die Gruppe vorher lokal noch nicht existierte, wird sie automatisch angelegt.

Rückgabewert: 0 = OK/angelegt, –1 = nicht angelegt.

`HamDelPull( ⟨server⟩ , ⟨groupname⟩ )`

Löscht einen Ladeauftrag für die Gruppe `⟨groupname⟩` vom Newsserver `⟨server⟩`.

Rückgabewert: 0 = OK/gelöscht, –1 = nicht gelöscht.

`HamGroupOpen( ⟨groupname⟩ )`

Öffnet die Newsgruppe `⟨groupname⟩` und liefert ein Handle zurück, welches für die weiteren Zugriffe benötigt und in den folgenden Funktionen als *⟨grphdl⟩* bezeichnet wird. Trat ein Fehler beim Öffnen der Newsgruppe auf, wird ein negativer Wert zurückgeliefert.

Achtung: Newsgruppen müssen im Skript immer explizit geschlossen werden (mit »HamGroupClose«), ein Ende des Skriptes bzw. das Schließen des Hamsterobjektes schließt das Handle **nicht** automatisch!

Rückgabewert: >= 0: OK/Gruppen-Handle, < 0: Fehler.

`HamGroupClose( ⟨grphdl⟩ )`

Schließt die zu dem Handle `⟨grphdl⟩` (siehe »HamGroupOpen«) gehörige Newsgruppe.

Rückgabewert: 0

Beispiel:

```
$grphdl = HamGroupOpen( "internal.misc" )
  # weitere Anweisungen
HamGroupClose( $grphdl )
```

`HamGroupNameByHandle( ⟨grphdl⟩ )`

Liefert den Namen einer geöffneten Newsgruppe zu dem Handle `⟨grphdl⟩` (siehe »HamGroupOpen«) zurück.

Rückgabewert: String.

`HamGroupDescriptionByHandle( ⟨grphdl⟩ )`

Liefert die Gruppenbeschreibung der zu dem Handle `⟨grphdl⟩` (siehe »HamGroupOpen«) gehörenden Newsgruppe zurück.

Rückgabewert: String.

Beispiel:

```
var( $grphdl )
$grphdl = HamGroupOpen( "hamster.de.misc" )
print( HamgroupDescriptionByHandle( $grphdl ) )
HamGroupClose( $grphdl )
```

News-Artikel verwalten

`HamArtCount( ⟨grphdl⟩ )`

Liefert die Anzahl der Artikel der zum Handle `⟨grphdl⟩` (siehe »HamGroupOpen«) gehörenden Newsgruppe zurück.

Rückgabewert: >= 0: OK/Anzahl der Artikel, < 0: Fehler.

Siehe Beispiel unter »HamArtText«.

HamArtNoMin(⟨grphdl⟩)

Liefert die kleinste vorhandene Artikelnummer in der zum Handle ⟨grphdl⟩ (siehe »HamGroupOpen«) gehörenden Newsgruppe zurück.

Rückgabewert: >= 0: OK/Artikelnummer, < 0: Fehler.

Siehe Beispiel unter »HamArtText«.

HamArtNoMax(⟨grphdl⟩)

Liefert die größte vorhandene Artikelnummer in der zum Handle ⟨grphdl⟩ (siehe »HamGroupOpen«) gehörenden Newsgruppe zurück.

Rückgabewert: >= 0: OK/Artikelnummer, < 0: Fehler.

Siehe Beispiel unter »HamArtText«.

HamArtText(⟨grphdl⟩, ⟨artno⟩)

Liefert den Artikeltext inklusive Header zu dem Artikel mit der Nummer ⟨artno⟩ in der zum Handle ⟨grphdl⟩ (siehe »HamGroupOpen«) gehörenden Newsgruppe zurück.

Rückgabewert: String (Artikeltext oder Leerstring »""«, falls der Artikel nicht vorhanden ist).

Beispiel:

```
var( $grpnam, $grphdl, $artmin, $artmax, $artno, $arttxt )
$grpnam = "internal.misc"
$grphdl = HamGroupOpen( $grpnam )

if( $grphdl >= 0 )
  print( "Group ", $grpnam, ": ", HamArtCount($grphdl), " articles" )
  $artmin = HamArtNoMin( $grphdl )
  $artmax = HamArtNoMax( $grphdl )

  $artno = $artmin
  while( $artno <= $artmax )
    $arttxt = HamArtText( $grphdl, $artno )
    if( $arttxt <> "" )
      print( "Article ", $artno, ": ", len($arttxt), " byte" )
    endif
    inc( $artno )
  endwhile

  HamGroupClose( $grphdl )
endif
```

HamArtTextExport(⟨grphdl⟩, ⟨artno⟩)

Liefert den Artikeltext inklusive Header im Exportformat zu dem Artikel mit der Nummer ⟨artno⟩ in der zum Handle ⟨grphdl⟩ (siehe »HamGroupOpen«) gehörenden Newsgruppe zurück. Das Exportformat unterscheidet sich vom Artikelformat, wie man es mit »Ham-

ArtText« erhält, im Wesentlichen nur durch eine Zusatzzeile im Mbox-Format vor dem eigentlichen Posting.

Rückgabewert: String (Artikeltext oder Leerstring »""«, falls der Artikel nicht vorhanden ist).

Beispiel:

```
$arttxt = HamArtTextExport( $grphdl, $artno )
```

HamArtImport(⟨article⟩ [, ⟨goverride⟩, ⟨ignorehist⟩, ⟨marknoarch⟩])

Importiert den in einer String-Variablen ⟨**article**⟩ enthaltenen Artikel in die Datenbank des Hamsters und liefert im Erfolgsfall »TRUE« zurück. Der Artikeltext muss aus einem Headerblock, einer Leerzeile und dem Textbody bestehen. Im Headerblock müssen mindestens der »Date«-, der »From«- und der »Subject«-Header enthalten sein. Der Newsgroups-Header ist dann nötig, wenn der Parameter ⟨**goverride**⟩ leer ist, der das Speichern des Artikels in einer Gruppe unabhängig vom Newsgroups-Header erlaubt. Ist die Gruppe unbekannt, wird der Artikel in der entsprechenden internen Gruppe gespeichert (Standard »internal.misc«). Ist der Parameter ⟨**ignorehist**⟩ »TRUE«, wird der Artikel auch dann noch angenommen, wenn er bereits in der History bekannt ist. Ist der Parameter ⟨**marknoarch**⟩ »TRUE«, wird der Artikel im Header »X-Hamster-Info« mit »NoArchive=1« gekennzeichnet.

Rückgabewert: TRUE (1): OK, FALSE (0): Fehler.

Beispiel:

```
$article_text = ListGetText( $list)
HamArtImport( $article_text, "internal.misc" )
```

HamArtLocateMid(⟨msgid⟩, ⟨groupname⟩, ⟨artno⟩)
HamArtLocateMid(⟨msgid⟩)

Liefert 0 zurück, wenn ein Artikel mit der angegebenen Message-ID ⟨**msgid**⟩ in der History vorhanden ist und schreibt den Gruppennamen und die Artikelnummer innerhalb der Gruppe in die Variablen ⟨**groupname**⟩ und ⟨**artno**⟩.

Rückgabewert: = 0: OK/gefunden, <> 0: Message-ID nicht gefunden.

Beispiel:

```
if( HamArtLocateMid( "<12345-67890@abc.def.invalid>", $groupname, $artno ) )
  # weitere Anweisungen
endif
```

HamArtLocateMidInGroup(⟨msgid⟩, ⟨groupname⟩, ⟨artno⟩)

Liefert in ⟨**artno**⟩ die Artikelnummer des Artikels mit der Message-ID ⟨**msgid**⟩ innerhalb der Gruppe ⟨**groupname**⟩ zurück.

Rückgabewert: = 0: OK/gefunden, –1: Message-ID nicht gefunden.

`HamArtDeleteMID( ⟨message-id⟩)`

Löscht den Artikel mit der angegebenen Message-ID.

Rückgabewert: = 0: gelöscht, <> 0: Message-ID nicht gefunden/nicht gelöscht.

`HamArtDeleteNrInGroup( ⟨artno⟩ , ⟨groupname⟩ )`

Löscht den Artikel mit der Artikelnummer ⟨artno⟩ in der Gruppe ⟨groupname⟩. Die History bleibt dabei unberücksichtigt.

Rückgabewert: = 0: gelöscht, −1: Funktion nicht ausführbar, −2: Artikel existiert nicht.

Scoring

`HamScoreList( ⟨groupname⟩ )`

Liefert die Scoreliste für eine Gruppe zurück.

Rückgabewert: String.

Beispiel:

```
ListSetText( $ScoreList, HamScoreList( "hamster.de.talk" ) )
```

`HamScoreTest( ⟨groupname⟩, ⟨article⟩, ⟨matchlog⟩, ⟨scope⟩ )`
`HamScoreTest( ⟨groupname⟩, ⟨article⟩ )`

Liefert den Scorewert zurück, den der Artikel ⟨article⟩ in der Gruppe ⟨groupname⟩ erhalten würde, wobei die Variable ⟨matchlog⟩ anschließend alle relevanten Scorezeilen enthält. Die Variable ⟨scope⟩ legt fest, ob mit »1« nur der Overview-Score-Wert oder mit »2« der After-Load-Score-Wert oder die Summe aus beidem mit »0« zurückgeliefert wird.

Rückgabewert: Integer (Score-Wert).

Beispiel:

```
$ScoreValue = HamScoreTest( "hamster.de.talk", $Article, $MatchLog )
```

`TestMailfilterLine( ⟨filterline⟩ )`

Testet die Syntax einer Filterzeile auf Richtigkeit und zeigt Informationen zur Interpretation im Log an.

`HamMailScoreList( ⟨filtersection⟩ )`

Liefert alle Filterzeilen aus der »`Mailfilt.hst`«, die auf die ausgewählte Sektion ⟨filtersection⟩ in der Datei zutreffen.

Rückgabewert: String.

Beispiel auf der folgenden Seite

```
varset( $ScoreList, ListAlloc())
ListSetText( $ScoreList, HamMailScoreList( "test" ) )
         # auch möglich: HamMailScoreList( "[test]" )
varset( $i, 0 )
while( $i < ListCount( $ScoreList ) )
  print( $i, " : ", ListGet( $ScoreList, $i ) )
  inc( $i )
endwhile
ListFree( $ScoreList )
```

Hamsterskript: Nebenläufige Skripte

Nebenläufigkeit

Der Hamster kann mehrere Skripte nebeneinander ausführen. Jedes Skript läuft dabei völlig unabhängig von allen anderen, unabhängig auch von den übrigen Vorgängen im Hamster. Der Hamster sorgt mit Hilfe des Betriebssystems dafür, dass es beim Zugriff auf Netzwerkverbindungen, Dateien und anderen kontingentierten Ressourcen nicht zu **Kollisionen** kommt. Einen solcher Ablauf ist ein **Prozess**, jedes laufende Skript bildet zusammen mit seinen globalen Variablen einen **Hamsterprozess**. Selbst wenn zweimal das gleiche Skript gestartet wird, beginnen zwei verschiedene Hamsterprozesse mit eigenen Daten in voneinander unabhängig gespeicherten globalen Variablen.

Auf einem Rechner mit **einem Prozessor** oder unter *Windows 9x* und *Windows ME* würde jedes Skript für kurze Zeit fortgesetzt, um dann zugunsten eines anderen auszusetzen, so dass der Prozessor jedes Skript und die übrigen Vorgänge im Hamster immer nur für kurze Zeit vorantreibt. Dieser Wechsel wird jedoch schnell genug vollzogen, so dass es für den menschlichen Nutzer wie ein gleichzeitiger Ablauf aussieht.

Nur auf einem Rechner mit **mehreren Prozessoren** und unter *Windows NT* bzw. ab *Windows 2000* können Prozesse auch real gleichzeitig ablaufen. In der Regel hat allerdings auch hier das Betriebssystem eine Anzahl von Prozessen gestartet, so dass die Rechenzeit selbst in einem Mehrprozessorsystem nur alternierend zugeteilt wird.

Die Hamsterprozesse laufen letztlich als **Windows-Threads**, die zum **Windows-Prozess** des Hamsters gehören. Wenn der Hamster – genauer: sein Windows-Prozess – beendet wird (gleich aus welchem Grund), enden alle seine Threads und damit auch alle Hamsterprozesse. Im Folgenden bezeichnet der Begriff des Prozesses immer einen Hamsterprozess, also die Ausführung eines Skriptes oder eines anderen Vorganges im Hamster.

Synchronisation

Vor der Synchronisation steht oft der skriptgesteuerte Start anderer Skripte. Es macht beim Hamster allerdings keinen Unterschied, ob ein Skript per Kommandozeilenoption, über das

Skriptmenü oder skriptgesteuert gestartet wird. Die Prozedur `runscript()`[1] startet das Skript, dessen Namen im ersten Parameter ⟨`Skriptname`⟩ ohne Pfad, aber mit Extension »`.hsc`« anzugeben ist; es wird im Skripteverzeichnis des Hamsters gesucht. Diesem Skript wird im zweiten Parameter ⟨`Parameter`⟩ ein String übergeben, der die Kommandozeilenparameter für dieses Skript enthält. Der Boolesche Parameter ⟨`Wait`⟩ steuert die Nebenläufigkeit: Wird darin 0 übergeben, kehrt die Prozedur sofort nach dem Start des **aufgerufenen** Skriptes zurück, so dass aufgerufenes und **aufrufendes** Skript nun nebeneinander ausgeführt werden. Bei jedem anderen Wert kehrt die Prozedur erst zurück, wenn das aufgerufene Skript beendet wurde; das aufrufende Skript wird erst nach Beendigung des aufgerufenen Skriptes fortgesetzt.

Abgesehen von einer gewissen Zeitersparnis wäre die Möglichkeit der Nebenläufigkeit ohne Synchronisation kaum interessant: Man könnte nur solche Skripte nebenläufig ausführen lassen, deren Tätigkeit in keinerlei Beziehung zueinander steht. Die folgenden Prozeduren bzw. Funktionen ermöglichen in häufig anzutreffenden Spezialfällen eine recht einfache Synchronisation:

- `sleep( ⟨time⟩ )` bewirkt eine Pause im Skriptablauf mit der Zeitspanne von ⟨`time`⟩ (in Millisekunden = 1/1000 Sekunden).
- `HamWaitIdle` bzw. `HamWaitIdle( ⟨Timeout⟩ )` kehrt erst zurück, wenn der Hamster keine Jobs mehr ausführt oder das ⟨`Timeout`⟩ (in Millisekunden = 1/1000 Sekunden) vergangen ist. Ohne diesen Parameter oder mit dem Wert −1 wird unbegrenzt auf die Abarbeitung aller laufenden Jobs gewartet. Die Funktion gibt den Wert 0 zurück, wenn keine Jobs mehr laufen; den Wert 1, falls die Timeout-Zeit abgelaufen ist und den Wert 2, falls die laufenden Jobs abgebrochen wurden.

Synchronisation durch Events

Die im vorhergehenden Abschnitt erläuterten Aufrufe halten das aufrufende Skript an, bis der Hamster einen bestimmten **Zustand** einnimmt (`HamWaitIdle()`: Hamster führt keine Jobs aus) oder ein **Ereignis** eintritt (`sleep()`: Ablauf einer Zeitspanne). Generell bedeutet **Synchronisation**, den Ablauf von Prozessen zum Zwecke der **Koordination** vom Verhalten anderer Prozesse abhängig zu machen. Sie ist immer dann erforderlich, wenn **Kollisionsfreiheit** allein nicht ausreicht.

Windows und der Hamster garantieren beispielsweise, dass eine Datei zu jedem Zeitpunkt nur durch einen einzigen Prozess bearbeitet wird. Damit ist aber noch nicht sichergestellt, dass ein schreibender Prozess grundsätzlich vor einem lesenden Prozess zugreift, damit letzterer überhaupt Daten zur Verarbeitung vorfindet. Dazu müssten Windows und der Hamster die Skripte lesen und verstehen, die diese Prozesse beschreiben. Weil das nicht möglich ist, fügt man in die Skripte Anweisungen ein, die es bis zum Eintritt eines Ereignisses anhalten (**warten** lassen) oder seine Fortsetzung von einem Zustand abhängig machen.

[1] Vollständig: `runscript( ⟨Skriptname⟩, ⟨Parameter⟩, ⟨Wait⟩ )`

Seit der Version »*Hamster Classic 1.3.22.100 beta*« können Skripte auf beliebige Ereignisse (**Events**) warten, die von **nebenläufigen** Skripten oder bestimmten Vorgängen im Hamster **ausgelöst** bzw. als Zustand **gesetzt** werden. Sie werden zur Unterscheidung mit Namen versehen.

Weil auf einem Rechner mehrere Hamster arbeiten können und Ereignisnamen systemweit eindeutig sein müssen, ist dem Namen im Wege der Stringconkatenation eine Prozessidentifikation (»Mutex-String«) voranzustellen, die per Skript mit der eingebauten Funktion »GetProcessIdentifier« abgefragt werden kann (siehe auch die Erläuterungen zu den »Hamster-Mutex-Objekten«). Die Funktion

```
EventWait( ⟨Eventname⟩, ⟨Timeout⟩ )
```

kehrt zurück, wenn das Ereignis mit dem Namen `⟨Eventname⟩` eingetreten (gesetzt) ist oder die Zeit von `⟨Timeout⟩` (in Millisekunden = $\frac{1}{1000}$ Sekunden) abgelaufen ist. Ohne den Parameter `⟨Timeout⟩` oder mit dem Wert –1 wird unbegrenzt auf das Ereignis gewartet. Wenn und solange das Ereignis `⟨Eventname⟩` gesetzt ist, kehrt diese Prozedur bei jedem Aufruf sofort zurück, auch wenn zuvor schon Wartezeiten darauf abgelaufen sind.

Wenn das Ereignis unter dem Namen `⟨Eventname⟩` nicht gefunden werden konnte, gibt die Funktion dann einen Wert größer 1 als Fehlercode zurück, sonst den Wert 0 oder 1. Ein Rückgabewert von 1 bedeutet, dass der Hamster extern beendet werden soll und die Funktion daher zurückkehrt. Der Fehlercode 258 zeigt den Ablauf der maximalen Wartezeit `⟨Timeout⟩` an.

Diese Funktion wird also genau wie `HamWaitIdle()` gebraucht, nur dass auf ein beliebiges, durch seinen Namen gekennzeichnetes und im ersten Parameter zu übergebendes Ereignis gewartet wird.

Die folgenden beiden Funktionen dienen der Signalisierung von Zuständen:

```
EventSet( ⟨Eventname⟩ )
```

löst das Ereignis mit dem Namen `⟨Eventname⟩` aus. Das Ereignis bleibt dem Umlegen eines Schalters vergleichbar *dauernd ausgelöst*, bis es mit »EventReset()« oder automatisch nach Vorgabe in der Funktion »EventCreate()« zurückgesetzt wird. Daher kennzeichnet ein derartig ausgelöstes Ereignis auch einen Zustand. Wenn das Ereignis unter dem Namen `⟨Eventname⟩` nicht gefunden werden konnte, gibt die Funktion den Wert –1 zurück, sonst den Wert 0.

```
EventReset( ⟨Eventname⟩ )
```

setzt das Ereignis mit dem Namen `⟨Eventname⟩` zurück. Wenn das Ereignis unter dem Namen `⟨Eventname⟩` nicht gefunden werden konnte, gibt die Funktion den Wert –1 zurück, sonst den Wert 0.

Nachdem ein Ereignis durch »EventSet()« ausgelöst wurde, kehren auch alle *späteren* auf dieses Ereignis wartenden »EventWait()«-Aufrufe sofort zurück, bis ein »`EventReset()`«-Aufruf erfolgt. Die Funktion

`EventPulse( ⟨Eventname⟩ )`

löst das Ereignis mit dem Namen `⟨Eventname⟩` aus. Das Ereignis wird danach *sofort wieder zurückgesetzt*; das Verhalten der damit ausgelösten Ereignisse entspricht einem Taster, der nach Betätigung wieder in seine Ausgangsposition zurückfedert. Wenn das Ereignis unter dem Namen `⟨Eventname⟩` nicht gefunden werden konnte, gibt die Funktion den Wert –1 zurück, sonst den Wert 0.

Bei einem mit `EventPulse()` ausgelösten Ereignis kehren lediglich alle *vor* der Auslösung erfolgten »EventWait()«-Aufrufe für dieses Ereignis zurück. Spätere Aufrufe warten bis zur nächsten Auslösung.

Der Hamster kennt folgende Ereignisse (Ereignisnamen ohne Mutex-String aus »GetProcessIdentifier«):

Ereignisname	*auslösender Vorgang*	*Auslösung durch*
`evtHamster.IsIdle`	Keine Prozesse im Hamster aktiv und Hamster unbeschäftigt (idle), *siehe auch Hinweis*.	`EventSet()`
`evtHamster.StopScript`	Abbruch der Skriptbearbeitung gefordert.	`EventSet()`
`_main`	Hamster läuft.	`EventSet()`
`_mailin`	E-Mail-Empfang vom Remote-Server.	`EventPulse()`
`_mailout`	E-Mail-Empfang vom lokalen Server mit externem Ziel.	`EventPulse()`
`_mailinternal`	E-Mail-Empfang vom lokalen Server mit internem Ziel.	`EventPulse()`
`_newsout`	Artikel-Empfang vom lokalen Server.	`EventPulse()`
`_purge`	Der Hamster putzt sich.	`EventSet()`
`_rasconnected`	Aufbau einer DFÜ-Verbindung.	`EventSet()`
`_rashangup`	Trennung einer DFÜ-Verbindung.	`EventPulse()`

Hinweis: Verbindungen zu den lokalen Servern werden nicht als aktiver Auftrag betrachtet.

Wenn Sie das Modul »`hamster.hsm`« in Ihr Skript einbinden, können Sie die dort vordefinierten Konstanten verwenden.

Skripte können aber auch beliebig viele neue Ereignisse definieren. Sie sind mit der Funktion »EventCreate()« zu deklarieren. Dabei wird ein Name für sie festgelegt, unter dem sie ausgelöst werden können bzw. unter dem auf sie gewartet werden kann.

`EventCreate( ⟨Eventname⟩, ⟨Reset⟩, ⟨Initial⟩, ⟨Errorcode⟩ )`

deklariert ein neues Ereignis unter dem Namen `⟨Eventname⟩`. Wird im Parameter `⟨Reset⟩` 0 übergeben, wird das Ereignis nach der Abfrage nicht automatisch zurückgesetzt, Standardwert

ist 1. Wird in ⟨Initial⟩ der Wert 1 übergeben, wird das Ereignis sofort ausgelöst, bei dem Wert 0 nicht. Die Funktion gibt eine Zahl (das **Eventhandle**) zurück, die zum Löschen des Ereignisses mit »EventClose()« oder in »EventMultipleWait()« benötigt wird. Deklaration und Löschung müssen zumindest im ereignisauslösenden Skript erfolgen. Sie können zusätzlich auch im reagierenden Skript vorgenommen werden, um die Existenz eines Ereignisses sicherzustellen.

`EventClose( ⟨Eventhandle⟩ )`

löscht eine Ereignisdeklaration.

`EventMultipleWait( ⟨Timeout⟩, ⟨EventHandle1⟩ [ , ... , ⟨EventHandle10⟩ ] )`

wartet auf eines von maximal 10 Events, die durch ⟨EventHandleX⟩ anzugeben sind. Wird eines von mehreren Events aktiv oder tritt das ⟨Timeout⟩ (in Millisekunden = $\frac{1}{1000}$ Sekunden) auf, liefert diese Funktion folgende Ergebnisse:

- 1...10: Eintritt des jeweiligen Eventparameters
- 0: Stop-Script-Ereignis aufgetreten
- –1: Timeout
- –2: Fehler

Das folgende Beispiel realisiert eine Anwendung, die automatisch alle Mailaccounts abfragt und evtl. ausgehende Mails versendet, sobald eine DFÜ-Verbindung hergestellt wird. Sie besteht aus einem Skript »Einwahl.hsc«, das die DFÜ-Verbindung herstellt und den Erfolg durch die Auslösung eines Ereignisses signalisiert. Ein weiteres Skript »Mailexchange.hsc« wartet auf dieses Ereignis und führt, sobald es eintritt, den Mailexchange durch.

Beide folgenden Beispielskripte sind in je einer »*.hsc«-Textdatei im Skripteverzeichnis des Hamsters unterzubringen. Beide Skriptnamen sollten dann nach dem nächsten Start des Hamsters oder bei laufendem Hamster nach Selektion des Menüs »Datei« → »Aktualisiere Menüeinträge« im Skriptmenü erscheinen.

Das Skript »Einwahl.hsc« lautet wie folgt:

```
# Einwahl.hsc
# Stellt eine DFÜ-Verbindung her
#!hs2

var( $try, $err )

varset( $Tries, 3 )            # Anzahl der Einwahlversuche
varset( $Connection, "RAS-Provider" ) # DFÜ-Verbindung

# 1.) Ereignis deklarieren ...
# ... Prozess- und Ereignisnamen bestimmen
varset( $Proc, GetProcessIdentifier )
varset( $Event, $Proc + "_rasconnected" + "_" + $Connection )
```

```
# ... Ereignis erzeugen
varset( $hdlEvent, EventCreate( $Event, 0, 0, $err ) )

# 2.) Einwahlversuche
for( $try, 1, $Tries )

  # 2.a) Einwahlversuch
  $err = HamRasDial( $Connection )

  # 2.b) DFÜ-Verbindung brauchbar?
  if( $err = 0 )            # Kein DFÜ-Fehler aufgetreten?
    if( LocalhostAddr = 0 ) # Keine IP-Adresse zugeteilt?
      HamRasHangup          # →  Auflegen, da DFÜ-Verbindung ohne IP-Adresse
                            #    unbrauchbar
    else
      EventPulse( $Event )  # →  Ereignis auslösen, da Einwahl erfolgreich
      break                 # Einwahlschleife verlassen
    endif
  endif

endfor

# 3.) Ereignis löschen
EventClose( $hdlEvent )

quit
```

Das Skript deklariert im Abschnitt 1 ein Ereignis mit dem Namen »$Event«, das die Herstellung der DFÜ-Verbindung des Namens »$Connection« meldet. Im Abschnitt 2 wird die Einwahl versucht (2.a) und – falls das DFÜ-Netzwerk keine Fehler meldet – die Brauchbarkeit der DFÜ-Verbindung überprüft (2.b). Eine DFÜ-Verbindung ist brauchbar, wenn eine IP-Adresse zugeteilt wurde. Der Abschnitt 2 befindet sich in einer For-Schleife, damit bei fehlschlagender Einwahl mehrere Versuche unternommen werden.

Das Skript »`Mailexchange.hsc`« lautet wie folgt:

```
# Mailexchange.hsc
# Fragt POP3-Server ab und veranlasst den Versand ausgehender Mails
#!hs2

var( $err )

# A.) Ereignis deklarieren ...
# ... Prozess- und Ereignisnamen bestimmen
varset( $Proc, GetProcessIdentifier )
varset( $Connection, "_RAS-Provider" )
# (der Unterstrich »_« musste hier ergänzt werden, da er nicht in das folgende
# varset eingefügt werden soll, vgl. die Ausführungen unten im Text)
varset( $Event, $Proc + "_rasconnected" + $Connection )
```

```
# ... Ereignis erzeugen
varset( $hdlEvent, EventCreate( $Event, 0, 0, $err ) )

# B.) Ereignisschleife
do

  # B.1) Ereignis erwarten
  EventWait( $Event, -1 )

  # B.2) Mailexchange durchführen ...
  # ... Abfrage
  HamFetchMail( "pop3-server.one.example.com", "pop3", _
                "$1", "", "lokaler_Nutzer" )
  # ... Abfrage des lokalen Administrator-Accounts
  #     (wäre auch ohne DFÜ-Verbindung möglich)
  HamFetchMail( "localhost", "pop3", _
                "$99", "", "lokaler_Nutzer" )
  # ... Versand per SMTP-AUTH
  HamSendMailAuth( "smtp-server.one.example.com", "smtp", _
                   "$1", "", "user@one.example.com" )

loop

# C.) Ereignis löschen

EventClose( $hdlEvent )

quit
```

Im Abschnitt A wird das Ereignis deklariert, auf das hier mit dem Mailaustausch reagiert werden soll. Der Abschnitt B besteht aus der **Ereignisschleife**, die endlos – also bis zur Beendigung des Hamsters – läuft. Bei B.1 kehrt die Funktion **EventWait()** erst zurück, wenn das deklarierte Ereignis eingetreten ist; bis dahin wartet »`Mailexchange.hsc`«.

Das erwartete Ereignis kann erst eintreten, wenn der Nutzer im Skriptmenü auch den Eintrag »Einwahl« aufruft und damit das Skript »`Einwahl.hsc`« startet. Dieses versucht, eine DFÜ-Verbindung herzustellen und löst das Ereignis im Falle erfolgreicher Einwahl aus. Gelingt die Einwahl, »erwacht« auch das wartende Skript »`Mailexchange.hsc`«, weil jetzt dessen »**EventWait()**«-Aufruf unter B.1 zurückkehrt. Nun wird dort der Mailexchange durch die Anweisungen unter B.2 ausgeführt. »`Mailexchange.hsc`« muss vorher gestartet worden sein; ideal wäre, dies beim Start des Hamsters durch eine Action oder als Kommandozeilenoption zu erledigen.

Da »`Einwahl.hsc`« das Ereignis durch **EventPulse()** auslöst, wird es im Skript »`Mailexchange.hsc`« nach der Rückkehr des »**EventWait()**«-Aufrufes sofort wieder zurückgesetzt. Wenn die Ereignisschleife im Abschnitt B also nach Ausführung der Anweisungen unter B.2 erneut **EventWait()** aufruft, wartet es bis zur erneuten Herstellung der DFÜ-Verbindung. Anderenfalls würden die Anweisungen unter B.2 fortwährend wiederholt.

Im Skript »Mailexchange.hsc« kann die Zeile `varset( $Connection, "_RAS-Provider" )` auch durch die Zeile `varset( $Connection, "" )` ersetzt werden; das Skript würde damit das durch den Ausdruck »`GetProcessIdentifier + "_rasconnected"`« gegebene hamstereigene DFÜ-Einwahlereignis erwarten. Doch das durch »Einwahl.hsc« deklarierte Einwahlereignis hat den Vorzug, die DFÜ-Verbindung nicht nur herzustellen, sondern auch auf Zuteilung einer IP-Adresse zu überprüfen.

Die beiden Skripte dieses einfach gehaltenen Beispiels kommunizieren praktisch interprozessual durch das mit `EventCreate( $Event, 0, 0, $err )` deklarierte Ereignis. Dazu müssen beide Skripte vom gleichen Ereignisnamen ausgehen, also gemeinsame Informationen nutzen. In größeren Projekten empfiehlt es sich daher, die Ereignisnamen durch globale Variablen eines Moduls bereitzustellen. In dessen Initialisierungsteil veranlasst man auch gleich die erforderlichen »**EventCreate()**«-Aufrufe. Damit sind die selbstdefinierten Ereignisse bereits durch einfaches Laden dieses Moduls »gebrauchsfertig« und es ist sichergestellt, dass alle beteiligten Skripte auch nach Namensänderungen die richtigen Bezeichnungen verwenden. Das Modul sollte auch eine Prozedur bieten, die alle selbstdefinierten Ereignisse mit **EventClose()** wieder löscht (Nutzung eines Finalisierungsteils).

Mehr zu diesem Thema siehe auch im FAQ-Beitrag »Hamster und Events« und für ausführliche Beispiele unter »Sofortiges Versenden Von E-Mails«.

(Für die Nutzung von Events mit anderen Skriptsprachen siehe »Hamster-Event-Objekte«.)

Hamsterskript: Module

So wie eine Prozedur oder Funktion eine Anweisungsfolge mit einem Namen versieht, fasst ein **Modul** eine Gesamtheit von Variablen, Prozeduren und Funktionen unter einem Namen zusammen. Deren **Funktionalität** kann damit mehreren Skripten oder Modulen verfügbar gemacht werden, ohne die zugehörigen Variablendeklarationen und Prozedurdefinitionen in jedem einzelnen Quelltext wiederholen zu müssen. Skripte und Module, die auf solche **modular implementierte** Funktionalität zurückgreifen wollen, brauchen lediglich das **Laden** des Moduls zu veranlassen.

Module sind daher auch das Mittel der Wahl bei **arbeitsteiliger Entwicklung** eines Skriptprojektes, da sich damit anderen Entwicklern eine Funktionalität gebrauchsfertig bereitstellen lässt.

Definition von Modulen

Module sind wie Skripte in Textdateien zu speichern, wie sie beispielsweise der Windows-Editor »Notepad« erzeugen und bearbeiten kann. In einer Datei kann nur ein Modul definiert werden. Der Dateiname ist mit der Erweiterung »hsm« zu versehen. Moduldateien sind im

Moduleverzeichnis des Hamsters zu speichern. Dieses ist in der Voreinstellung das Hamster-Hauptverzeichnis. In der »hamster.ini«, Sektion »[Directories]« können Sie mit dem Schlüssel »Modules« ein anderes Verzeichnis festlegen.

Die Sprache der Hamsterskripte findet auch in Modulen unverändert Anwendung.

Anders als ein Skript sollte ein Modul einen durch die Präprozessor-Anweisung **#!initialize** einzuleitenden **Initialisierungsteil** besitzen (er ist dennoch optional). Dieser enthält Anweisungen, die nach dem ersten Laden des Moduls ausgeführt werden und wird durch eine »**return()**«-Anweisung abgeschlossen. In deren Klammern ist ein Ausdruck anzugeben, dessen Wert für den Erfolg der Initialisierung steht. Jeder Wert ungleich 0 bedeutet, dass die Initialisierung des Modules nicht erfolgreich war und die Ausführung des gesamten Skripts abgebrochen wird.

Optional kann beginnend mit **#!finalize** auch ein **Finalisierungsabschnitt** eingefügt werden, der Anweisungen enthält, die bei Beendigung des Skriptes ausgeführt werden. Sinnvoll ist dieser Abschnitt besonders dann, wenn im Initialisierungsteil Anweisungen verwendet wurden, die Speicher reservieren (siehe z. B. ArtAlloc, ListAlloc oder MemAlloc), der im Finalisierungsteil wieder freigegeben werden kann, so dass man nicht in den Skripten Sorge dafür tragen muss. Wird ein Finalisierungsabschnitt benutzt, muss auch ein Initialisierungsabschnitt vorhanden sein. Der Initialisierungsabschnitt muss sich vor dem Finalisierungsabschnitt befinden. Sollen aus irgendwelchen Gründen noch Anweisungen nach dem Finalisierungsabschnitt folgen, so muss auch dieser mit einer »**return()**«-Anweisung abgeschlossen werden.

Ein Modul ist prinzipiell wie folgt aufgebaut:

```
#!hs2

#!initialize
⟨Deklaration globaler Variablen⟩
⟨initialisierende Anweisungen⟩
return( ⟨InitResult⟩ )

⟨Prozedur- und Funktionsdefinitionen⟩

#!finalize
⟨abschließende Anweisungen⟩
[return( ⟨FinalResult⟩ )

⟨weitere Anweisungen⟩]
```

Es empfiehlt sich, alle in einem Modul definierten Bezeichner mit dessen Namen beginnen zu lassen, um Überschneidungen mit den Bezeichnern anderer Entwickler zu vermeiden. Wer Module veröffentlichen möchte, sollte dem Modulnamen ein (Namens-)Kürzel voranstellen, um auch insoweit Überschneidungen zu verhüten.

Gebrauch von Modulen

Module können durch Skripte und andere Module mit Hilfe der Präprozessor-Anweisung **#!load** genutzt werden. Jede »#!load«-Anweisung stellt das Laden eines Moduls sicher, dessen Dateiname ohne Pfad, aber mit der Erweiterung »hsm« anzugeben ist:

```
#!load MyModule.hsm
```

Die Datei »MyModule.hsm« wird im Moduleverzeichnis des Hamsters gesucht.

In einem Skript müssen »#!load«-Anweisungen vor der ersten Anweisung stehen, in einem Modul können sie auch noch im Initialisierungsteil vorkommen.

Da Module keinen eigenen **Sichtbarkeitsbereich** haben, sind Prozeduren und Funktionen eines Moduls direkt nach dem Laden **global** verfügbar, d. h. im Modul selbst und in allen Skripten und Modulen, in welchen dieses Modul geladen wird. Die im Initialisierungsteil deklarierten Variablen sind erst ab der Initialisierung des betreffendes Modules global verfügbar.

Die Initialisierungsteile der Module werden in umgekehrter Lade-Reihenfolge ausgeführt. Das zuletzt geladene Modul wird also zuerst initialisiert. Ein *Skript* wird erst ausgeführt, wenn alle geladenen Module auch initialisiert worden sind.

Auch wenn der Interpretierer auf mehrere »#!load«-Anweisungen für das gleiche Modul stößt, lädt er dennoch jedes Modul nur einmal und initialisiert es auch nur einmal.

Dabei sollten *genutzte* Module immer *nach* den sie *nutzenden* Modulen geladen werden. Sie werden dann so initialisiert, dass genutzte Module vor dem Zugriff anderer Module betriebsbereit sind. Wenn beispielsweise ein Modul »MyHighLib« Bezeichner aus einem Modul »MyLib« nutzt und dieses wiederum auf »MyBaseLib« aufsetzt, empfiehlt sich im Skript oder anderen Modulen folgende Ladereihenfolge:

```
#!load MyHighLib.hsm
#!load MyLib.hsm
#!load MyBaseLib.hsm
```

Sollte ein Skript oder Modul nur Bezeichner aus »MyHighLib« und »MyBaseLib« nutzen, kann die Ladeanweisung für »MyLib« allerdings entfallen.

Aufruf per Action

Unter Actions werden im Hamster die Möglichkeiten zusammengefasst, bei bestimmten Aktionen automatisch ein externes Programm und/oder ein Hamster-Skript (oder einzelne Hamsterskript-Zeilen) ausführen zu lassen.

Einstellen können Sie das im Menü-Dialog »Einstellungen« → »Automatische Abläufe« ⇒ »Aktionen« oder direkt in der »hamster.ini« im Abschnitt »[Actions]« (die Schlüssel für Skripte enden auf »*.script«, für einzelne Skriptzeilen gibt es »*.ScriptLines«).

Achtung: Bei der Verwendung von »hs2«-Befehlen in action-gesteuerten Skripten oder Skriptzeilen, welche auf Mail- oder News-Transport-Action reagieren, sollten keine News- oder Mailtransportbefehle verwendet werden, um Deadlocks zu vermeiden.

Hamster-Event-Objekte (Win32)

Der Hamster unterstützt »Event-Objekte«. Man kann diese Objekte z. B. für Programme und Skripte benutzen, welche Artikel und E-Mails nach dem Download nachbearbeiten.

Ein Event setzt sich jeweils aus dem Mutex und dem eigentlichen Eventnamen zusammen:

```
Mutex+"⟨Eventname⟩"
```

Siehe auch die Erläuterungen zu den Hamster-Mutex-Objekten.

Folgende Events werden unterstützt:

⟨Eventname⟩	***Bedeutung***
`evtHamster.IsIdle`	Signalisiert, dass keine Prozesse im Hamster aktiv sind und der Hamster unbeschäftigt (idle) ist, *siehe auch Hinweis*.
`evtHamster.StopScript`	Signalisiert, dass ein Abbruch der Skript-Bearbeitung gefordert wird.
`_main`	Globales Ereignis, das von der Existenz des Hamsters kündet.
`_mailin`	Eine E-Mail wurde vom Remote-Server empfangen.
`_mailout`	Eine E-Mail mit externem Ziel wurde vom lokalen Server empfangen.
`_mailinternal`	Eine E-Mail mit internem Ziel wurde vom lokalen Server empfangen.
`_newsout`	Ein Artikel wurde vom lokalen Server empfangen.
`_purge`	Der Hamster putzt sich.
`_rasconnected`	Eine DFÜ-Verbindung ist aufgebaut worden.
`_rashangup`	Eine DFÜ-Verbindung wurde getrennt.

Hinweis: Verbindungen zu den lokalen Servern werden nicht als aktiver Auftrag betrachtet.

Beispiele für die praktische Nutzung:

Perl:

```
use Win32::Event;
# Pfad und Name des Hamsters: C:\Programme\Online\Hamster\hamster.exe
$mutex = "mutexHamster.App.c__programme_online_hamster_hamster_exe";
$evtHamsterIsIdle = Win32::Event->open($mutex."evtHamster.IsIdle")
    or die "Zugriff auf Event fehlgeschlagen: $!\n";
# Start Hamster-Tasks
$evtHamsterIsIdle->wait();
```

Delphi:

```
uses windows;
var evtHamsterIsIdle: THandle;
// Pfad und Name des Hamsters: C:\Programme\Online\Hamster\hamster.exe
const Mutex = 'mutexHamster.App.c__programme_online_hamster_hamster_exe';
evtHamsterIsIdle := OpenEvent(SYNCHRONIZE, False, Mutex+'evtHamster.IsIdle');
// Start Hamster-Tasks
WaitForSingleObject(evtHamsterIsIdle, INFINITE);
CloseHandle(evtHamsterIsIdle);
```

Visual Basic Classic:

```
Private Const INFINITE = &HFFFF
Private Const SYNCHRONIZE = &H100000

Private Declare Function OpenEvent Lib "kernel32" Alias "OpenEventA" _
    (ByVal dwDesiredAccess As Long, ByVal bInheritHandle As Long, _
    ByVal lpName As String) As Long
Private Declare Function WaitForSingleObject Lib "kernel32" _
    (ByVal hHandle As Long, ByVal dwMilliseconds As Long) As Long
Private Declare Function CloseHandle Lib "kernel32" _
    (ByVal hObject As Long) As Long

' Pfad und Name des Hamsters: C:\Programme\Online\Hamster\hamster.exe
Private Const Mutex ="mutexHamster.App.c__programme_online_hamster_hamster_exe"

Dim evtHamsterIsIdle As Long, res As Long
evtHamsterIsIdle = OpenEvent(SYNCHRONIZE, False, Mutex+"evtHamster.IsIdle")
' Start Hamster-Tasks
res = WaitForSingleObject(evtHamsterIsIdle, INFINITE)
res = CloseHandle(evtHamsterIsIdle)
```

Hamster-Mutex-Objekte (Win32)

Das Mutex-Objekt »**mutexHamster.App.***« wird benutzt, um den gleichzeitigen Start von mehreren Instanzen des Hamsters aus ein- und demselben Verzeichnis heraus zu verhindern. Das anhand des aktuellen Hamster-Verzeichnisses benannte Mutex-Objekt wird gesperrt, sobald die erste Instanz des betreffenden Hamsters startet, und wird wieder freigegeben, wenn diese Instanz beendet wird.

Man kann anhand dieses Objektes auch von Programmen aus feststellen, ob der Hamster aktiv ist.

Den aktuellen Mutex kann man auch per COM-/DCOM-/OLE-Schnittstelle mit der Funktion »ControlGetProcessIdentifier« ermitteln.

Format:

`"mutexHamster.App." + ⟨Hamster.exe inkl. voller Pfadangabe⟩`

Die Buchstaben des Pfades und Programmnamens werden in Kleinbuchstaben konvertiert. Alle Sonderzeichen – auch Leerzeichen und Punkte – werden durch den Unterstrich »_« ersetzt.

Beispiele:

Programm: `C:\Program Files\News\Hamster.exe`
Mutex-Name: `mutexHamster.App.c__program_files_news_hamster_exe`

Programm: `\\Server\C$\News\Hamster.exe`
Mutex-Name: `mutexHamster.App.__server_c__news_hamster_exe`

OLE-Schnittstelle für andere Programme

- Allgemeine Funktionen
- Task-/Auftrags-Funktionen
- RAS-Funktionen
- News-Funktionen
- Sonstige Funktionen

Für die OLE-/COM-/DCOM-Schnittstelle stellt der Hamster ein Objekt mit dem Klassennamen »**Hamster.App**« bereit.

Beispiele für die Nutzung:

Perl:

```
use Win32::OLE;
my $Hamster = Win32::OLE->new("Hamster.App");
print $Hamster->ControlGetInfo;
undef $Hamster;
```

Delphi:

```
uses comobj;
var Hamster: Variant;
Hamster := CreateOleObject( 'Hamster.App' );
writeln( Hamster.ControlGetInfo );
Hamster := Unassigned;
```

Visual Basic Classic:

```
Dim Hamster as Object
Set Hamster = CreateObject( "Hamster.App" )
MsgBox Hamster.ControlGetInfo
Set Hamster = Nothing
```

VBScript (Windows Script Host):

```
Set Hamster = Wscript.CreateObject( "Hamster.App" )
Wscript.Echo Hamster.ControlGetInfo
Wscript.DisconnectObject Hamster
```

OLE-Schnittstelle bei Start des Hamsters als Service:

Wenn der Hamster als Service gestartet wurde, läuft er – und damit auch der OLE-Server – üblicherweise unter dem System-Account. Somit funktioniert der gewohnte Zugriff vom User-Account nicht mehr. Diese Einschränkung lässt sich leicht umgehen, indem man das OLE-Programm oder Skript durch den Hamster selbst starten lässt mit Hilfe der hs2-Funktion »execute«.

Außerdem sollte die Dateiendung »hsc« im Dateisystem für den Hamster reserviert werden, siehe unter dem Menü: »Einstellungen« → »Grundeinstellungen« ⇒ »Verschiedenes«. Anschließend kann das hs2-Skript durch Doppelklick auf die Datei oder beim Hamsterstart ausgeführt werden. Dadurch läuft das Programm oder Skript automatisch im selben Account wie der Hamster und kann ihn dann per OLE steuern wie gewohnt.

Beispiel für VBScript:

```
# Als Hamsterskript speichern, z. B. als »OLE-Start.hsc«:
execute("wscript.exe " + HamPath + "vbs\OLE-Script.vbs", , , false)
```

Allgemeine Funktionen

`ControlGetInfo: String`

Liefert die Hamster-Version und den Pfad zurück.

`ControlGetPath: String`

Liefert den Hauptpfad zurück.

`ControlGetHscPath: String`
`ControlGetHsmPath: String`
`ControlGetLogsPath: String`
`ControlGetServerPath: String`
`ControlGetGroupsPath: String`
`ControlGetNewsOutPath: String`
`ControlGetNewsErrPath: String`
`ControlGetMailsPath: String`
`ControlGetMailsOutPath: String`

Liefert den absoluten Pfad (d. h. inkl. Laufwerks- und Pfadangabe) für Skriptdateien (»hsc«), Moduldateien (»hsm«), Protokolldateien (Logs), Serververzeichnisse, Newsgruppenverzeichnisse, ausgehende Artikel, Newsfehlerverzeichnisse, Mailverzeichnisse bzw. ausgehende Mails zurück.

`ControlGetVersion: String`

Liefert die Version zurück (z. B. »2.1.0.11«).

`ControlSetLogin( ⟨Identifier⟩: String; ⟨Username⟩: String; ⟨Password⟩: String )`

Setzt temporär den Namen und das Passwort für eine DFÜ-Netzwerkverbindung. Der `⟨Identifier⟩` muss mit dem entsprechenden Namen des DFÜ-Netzwerkeintrages exakt übereinstimmen – inklusive Groß-/Kleinschreibung. Wird ein Fragezeichen für das Pass-

wort verwendet, fragt der Hamster das Passwort erst bei Bedarf ab. Außerdem kann anstelle des Benutzernamen auch eine im Passwort-Menü definierte Variable verwendet werden, das Passwort ist dann als Leerstring anzugeben.

`ControlMessage( ⟨Msg⟩: Integer; ⟨Param⟩: Integer ): Integer`

Löst verschiedene Ereignisse im Hamster aus, die alternativ auch per Menü steuerbar sind.

⟨Msg⟩	⟨Param⟩	Wirkung
1	Zeit in Millisek. (= 1/1000 Sek.) bis Programmende	Hamster beenden (siehe Anmerkung 1)
2	0 = Zähler+Log, 1 = Zähler, 2 = Log, 3 = Fehlerliste	Setzt Zähler und Log-Anzeige zurück
3	0 = Stop, 1 = Start, 2 = Neustart, 3 = Status	Start/Stop lokaler NNTP-Server (siehe Anmerkung 2)
4	0 = Stop, 1 = Start, 2 = Neustart, 3 = Status	Start/Stop lokaler POP3-Server (siehe Anmerkung 2)
5	0 = Stop, 1 = Start, 2 = Neustart, 3 = Status	Start/Stop lokaler SMTP-Server (siehe Anmerkung 2)
6	0 = Verstecken, 1= Anzeigen, 2 = Status	Anzeigen/Verstecken des Hauptfensters
7	0 = Verstecken, 1= Anzeigen, 2 = Status	Anzeigen/Verstecken des Tray-Icons
8	*(ignoriert)*	Menü: Online/Alle Aufträge beenden
9	0 = Test, 1 = Laden	Konfiguration erneut laden (siehe Anmerkung 3)
10	*(ignoriert)*	»IPAccess.hst« erneut laden
11	0 = Stop, 1 = Start, 2 = Neustart, 3 = Status	Start/Stop lokaler Remote-Control-Server (siehe Anmerkung 2)
12	1 = Öffnen, 0 = Schließen, 2 = Status	Öffnet, schließt oder liefert den Status des Browsers für die News-/Mailverzeichnisse (siehe Anmerkung 4)
13	1 = Öffnen, 0 = Schließen, 2 = Status	Öffnet, schließt oder liefert den Status des Skriptdialogs (siehe Anmerkung 4)
14	0 = Stop, 1 = Start, 2 = Neustart, 3 = Status	Start/Stop lokaler IMAP-Server (siehe Anmerkung 2)

Fortsetzung nächste Seite

Fortsetzung

⟨Msg⟩	⟨Param⟩	Wirkung
15	1 = Öffnen, 0 = Schließen, 2 = Status	Öffnet, schließt oder liefert den Status des News-Killfilelog-Dialogs (siehe Anmerkung 4)
16	1 = Öffnen, 0 = Schließen, 2 = Status	Öffnet, schließt oder liefert den Status des Ladeauftragsdialogs (siehe Anmerkung 4)
17	1 = Öffnen, 0 = Schließen, 2 = Status	Öffnet, schließt oder liefert den Status des Mail-Killfilelog-Dialogs (siehe Anmerkung 4)

Anmerkung 1: Die Funktion `ControlMessage(1, ⟨Param⟩)` ist ein Hack, da das COM-Modell eigentlich nicht vorsieht, dass ein Client einen Server beendet, welchen er nicht selbst aktiviert hat. `⟨Param⟩` hat den Gültigkeitsbereich 100–32 600 ms. Innerhalb dieser Verzögerungszeit muss der aufrufende Client seine Verbindung zum Server gelöst haben. Für Werte kleiner 100 ms werden automatisch 100 ms angenommen.

Anmerkung 2 zu ⟨Msg⟩ = 3–5, 11, 14: Die Rückgabewerte für ⟨Param⟩ = »3« sind 0 = aus und 1 = an.

Anmerkung 3 zu ⟨Msg⟩ = 9: ⟨Param⟩ = »1« führt nur dann zum Laden der Konfiguration, wenn alle Konfigurationsfenster geschlossen sind (Rückgabewert der Funktion ist 1 = TRUE). Anderenfalls wird nichts ausgeführt (Rückgabewert 0 = FALSE). Bei ⟨Param⟩ = »0« (Test) wird grundsätzlich nichts ausgeführt, sondern nur die entsprechenden Rückgabewerte ermittelt. Damit kann getestet werden, ob das Laden gegebenenfalls möglich ist (d. h. die Konfiguration geschlossen ist).

Anmerkung 4 zu ⟨Msg⟩ = 12, 13, 15–17: ⟨Param⟩ = »2« ermöglicht die Abfrage, ob das jeweilige Fenster offen (Rückgabewert 1 = TRUE) oder geschlossen ist (Rückgabewert 0 = FALSE). Bei ⟨Msg⟩ 15–17 und ⟨Param⟩ = »1« wartet »ControlMessage«, bis der Dialog wieder geschlossen wird.

Übrige Rückgabewerte:
0 = OK; Befehl ausgeführt.
1 = Error; Befehl fehlerhaft.
2 = Unbekannter Parameter
3 = GPF, Fließkommafehler

`ControlGetStatus( ⟨Status⟩: Integer; ⟨Parameter⟩: Integer ): Integer`

Liefert Auskünfte über den momentanen Zustand des Hamsters. Nachfolgend steht `⟨n⟩` für `⟨Parameter⟩`.

ControlGetStatus(1,⟨n⟩) meldet, ob der lokale Server »n« gestartet ist.

Für `⟨n⟩` gilt:

Glossar:
NNTP
POP3
SMTP
IMAP
ReCo

1 = Lokaler NNTP-Server,
2 = Lokaler POP3-Server,
3 = Lokaler SMTP-Server,
4 = Lokaler IMAP-Server,
5 = Lokaler ReCo-Server.

Rückgabewert: 1, wenn der Server aktiv ist, anderenfalls 0.

ControlGetStatus(2,⟨n⟩) meldet die Anzahl der offenen Verbindungen am entsprechenden lokalen Server »n«.

Für ⟨n⟩ gilt:
1 = Lokaler NNTP-Server,
2 = Lokaler POP3-Server,
3 = Lokaler SMTP-Server,
4 = Lokaler IMAP-Server,
5 = Lokaler ReCo-Server.

Rückgabewert ist die Anzahl der offenen Verbindungen.

ControlGetStatus(3,⟨n⟩) gibt Auskunft über angekommene/wartende News & Mails.

Für ⟨n⟩ gilt:
1 = Neue Artikel,
2 = Geladene Artikel,
3 = Bekannte (History-gefilterte) Artikel,
4 = Gelöschte Artikel,
5 = Zu versendende Artikel,
6 = Neue E-Mails,
7 = Zu versendende E-Mails,
8 = Per Message-ID nachgeladene Artikel,
9 = Anzahl der Fehlerzeilen im Log,
10 = Anzahl der Warnungszeilen im Log.

Der *Rückgabewert* enthält die entsprechenden Anzahl der Mails oder Artikel.

ControlGetStatus(4,⟨n⟩) gibt Auskunft über die momentan aktiven Loglevel.

Für ⟨n⟩ gilt:
1 = Loglevel für Anzeige,
2 = Loglevel für Speichern in Log-Datei.

Vordefinierte Werte:

```
LOGID_ERROR  = 0x8000
LOGID_WARN   = 0x4000
LOGID_SYSTEM = 0x0800
LOGID_INFO   = 0x0080
LOGID_DETAIL = 0x0040
LOGID_DEBUG  = 0x0008
LOGID_STATUS = 0x0004
```

Der *Rückgabewert* enthält »OR«-verknüpft das oder die aktiven Loglevel.

```
ControlRunScript( ⟨Skriptname⟩: String; ⟨Skriptparameter⟩: String;
                  ⟨Warten⟩: Bool ): Integer
```

Startet ein hs2-Skript. Der Parameter **⟨Warten⟩** legt fest, ob auf das Ende des auszuführenden Skriptes gewartet werden soll. Das aufgerufene Skript kann die übergebenen Parameter mit den Funktionen »paramstr« und »paramcount« auswerten. Der Rückgabewert ist noch unbenutzt. Siehe hierzu auch die Erläuterungen zu dem hs2-Befehl »RunScript«.

Hinweis: Aktivitäten des OLE-Servers wirken sich direkt auf die Reaktionsgeschwindigkeit des Hamsters (besonders des Hauptfensters) aus. Bei **⟨Warten⟩** = »TRUE« sind daher während der Skriptausführung keine Menüeingaben o. ä. möglich. Nach Möglichkeit sollte diese Option mit »FALSE« benutzt werden.

```
ControlAddLog( ⟨String⟩: String; ⟨X⟩: Integer )
```

Fügt einen Text in das Log ein. **⟨X⟩** kennzeichnet den Typ der Eintragung:

1 = Debug,	4 = System,	7 = Status, d. h. Eintrag im
2 = Detail,	5 = Warning,	Statusfenster »Aufträge
3 = Info,	6 = Error,	(Threads)«.

```
ControlChangePassword( ⟨AccountName⟩: String; ⟨OldPassWord⟩: String;
                       ⟨NewPassWord⟩: String ): Integer
```

Ändert das Passwort für einen lokalen Benutzer.

Rückgabewert: 0 = Fehler im alten Passwort, 1 = Passwort geändert, 2 = Fehler beim Ändern.

Task-/Auftrags-Funktionen:

```
ControlIsIdle: Bool
```

Liefert »TRUE« (= wahr) zurück, wenn der Hamster unbeschäftigt ist. Aktive Skripte und Verbindungen mit den lokalen Servern werden hierbei nicht berücksichtigt.

```
ControlWaitIdle( ⟨WaitTimeout⟩: Integer ): Bool
```

Die Funktion **ControlWaitIdle** wartet, bis entweder das Timeout erreicht ist, oder kein Auftrag mehr im Hamster aktiv ist. Wird 0 oder, sofern definiert, »HAM_WAITIDLE_INFINITE« für das Timeout verwendet, kehrt die Funktion nur dann zurück, wenn kein Auftrag im Hamster mehr aktiv ist. Die Verbindungen mit den lokalen Servern werden hierbei nicht berücksichtigt.

```
HAM_WAITIDLE_INFINITE = 0
```

```
ControlThreadCount: Integer
```

Liefert die Anzahl der aktiven Aufträge zurück. Die Aufträge für die lokalen Server werden hierbei nicht berücksichtigt.

`ControlFlush: Integer`

Speichert alle ungesicherten Daten (z. B. History) auf die Festplatte. Dies ist besonders nach dem Import von Artikeln zu empfehlen.

`ControlRunPurge( ⟨PurgeOptions⟩: Integer ): Integer`

Startet einen Auftrag zum Aufräumen der Hamsterdaten. Der Parameter `⟨PurgeOptions⟩` legt fest, was aufzuräumen ist. Er ist bit-orientiert und kann durch logisches ODER gebildet werden, so dass man mit einem einzigen Befehl eine beliebige Kombination von Aufräumaufträgen starten kann (Bit 0 = 1 = Artikel, Bit 1 = 2 = News-History-Einträge, Bit 2 = 4 = Killfile-Protokoll, Bit 3 = 8 = Mail-History-Einträge):

```
HAM_PURGEOPT_DONEWS      = 0x01
HAM_PURGEOPT_DOHISTORY   = 0x02
HAM_PURGEOPT_DOKILLS     = 0x04
HAM_PURGEOPT_DOMHISTORY  = 0x08
HAM_PURGEOPT_MAILINLOG   = 0x10
HAM_PURGEOPT_MAILOUTLOG  = 0x20
HAM_PURGEOPT_NEWSOUTLOG  = 0x40
HAM_PURGEOPT_RASDIALLOG  = 0x80
HAM_PURGEOPT_DOALL       = 0xFF
```

Rückgabewert: 1: OK/gestartet, <> 1: nicht gestartet.

`ControlRunPurgeGroup( ⟨GroupName⟩: String ): Integer`

Mit dieser Funktion kann man das Aufräumen auf eine einzelne Gruppe beschränken.

Rückgabewert: 1: OK/gestartet, <> 1: nicht gestartet.

`ControlRunRebuildHistory: Integer`

Startet das Erneuern der History-Dateien (»`.\Groups\Hist*.dat`«). Diese Funktion sollte aus Sicherheitsgründen nur benutzt werden, wenn der Hamster ansonsten unbeschäftigt ist (siehe Funktion »ControlWaitIdle«) und außerdem gerade keine Mails eingehen (lokale Server deaktivieren).

`ControlRunRebuildGlobalLists( ⟨Options⟩: Integer ): Integer`

Erneuert die globalen Listen (das heißt z. B. erneuert die Liste der Gruppenbeschreibungen). Der Parameter `⟨Options⟩` wird zur Zeit nicht benutzt, sollte aber sicherheitshalber mit Null belegt werden.

`ControlRunMail( ⟨ServerList⟩: String ): Integer`

Startet einen Auftrag zum Mailaustausch für die im Parameter `⟨ServerList⟩` angegebenen SMTP- und POP3-Server. Ist der Parameter leer, werden alle verfügbaren Server gestartet, ansonsten müssen die Server in der Form »Servername1,Port1;Servername2,Port2; ... « angegeben werden.

Glossar: SMTP POP3 Port

Rückgabewert: 1: OK/gestartet, <> 1: nicht gestartet.

```
ControlRunFetchMail( ⟨Server⟩: String; ⟨Port⟩: String; ⟨Benutzer⟩: String;
                     ⟨Passwort⟩: String; ⟨DestUser⟩: String ): Integer
```

Glossar: POP3 Port

Startet einen Auftrag zum Abholen der E-Mails bei einem POP3-Server. Für ⟨Benutzer⟩ kann auch eine im Passwort-Menü definierte Variable verwendet werden, ⟨Passwort⟩ bleibt dann leer. Für ⟨DestUser⟩ kann ein lokaler Benutzer angegeben werden, welcher die E-Mails erhalten soll, ansonsten wird der in den POP3-Server-Einstellungen festgelegte Benutzer genommen. Zusätzlich zum lokalen Benutzer kann, durch ein Komma getrennt, auch ein Abschnitt in der Mailfilter-Datei »Mailfilt.hst« angegeben werden, anderenfalls wird der in den POP3-Server-Einstellungen festgelegte Abschnitt genommen.

Rückgabewert: 1: OK/gestartet, <> 1: nicht gestartet.

```
ControlRunSendMail( ⟨Server⟩: String; ⟨Port⟩: String;
                    ⟨FromSelection⟩: String ): Integer
```

SMTP Header

Startet einen Auftrag zum Senden der E-Mails zu einem SMTP-Server. Enthält der Parameter ⟨FromSelection⟩ eine Auswahl, das heißt, ist er nicht leer, werden nur E-Mails versendet, deren »!MAIL FROM:«-Header dem Regulären Ausdruck in diesem Parameter entspricht (siehe auch die FAQ »Hamster und Reguläre Ausdrücke«).

Rückgabewert: 1: OK/gestartet, <> 1: nicht gestartet.

```
ControlRunSendMailAuth( ⟨Server⟩: String; ⟨Port⟩: String; ⟨Benutzer⟩: String;
                        ⟨Passwort⟩: String; ⟨FromSelection⟩: String ): Integer
```

SMTP-AUTH

Startet einen Auftrag zum Senden der E-Mails zu einem SMTP-Server unter Benutzung des SMTP-AUTH-Verfahrens. Für ⟨Benutzer⟩ kann auch eine im Passwort-Menü definierte Variable verwendet werden, ⟨Passwort⟩ bleibt dann leer. Enthält der Parameter ⟨FromSelection⟩ eine Auswahl, das heißt, ist er nicht leer, werden nur E-Mails versendet, deren »!MAIL FROM:«-Header dem Regulären Ausdruck in diesem Parameter entspricht (siehe auch die FAQ »Hamster und Reguläre Ausdrücke«).

Rückgabewert: 1: OK/gestartet, <> 1: nicht gestartet.

```
ControlRunFetchMailTLS( ⟨Server⟩: String; ⟨Port⟩: String; ⟨Benutzer⟩: String;
                        ⟨Passwort⟩: String; ⟨Destuser⟩: String;
                        ⟨SSLMode⟩: Integer; ⟨SSLVerify⟩: Integer;
                        ⟨SSLCaFile⟩: String ): Integer
```

Startet einen Auftrag zum Abholen der E-Mails bei einem POP3-Server. Für ⟨Benutzer⟩ kann auch eine im Passwort-Menü definierte Variable verwendet werden, ⟨Passwort⟩ bleibt dann leer. Für ⟨Destuser⟩ kann ein lokaler Benutzer angegeben werden, welcher die E-Mail erhalten soll, ansonsten wird der im Menü »Einstellungen« konfigurierte Benutzer verwendet. Zusätzlich zum lokalen Benutzer kann, durch ein Komma getrennt, auch ein Abschnitt in der Mailfilter-Datei »Mailfilt.hst« angegeben werden, bei fehlender Angabe wird auch hier wieder auf das Menü »Einstellungen« zugegriffen.

Glossar: SSL

Achtung! Um SSL in Verbindung mit dem Hamster nutzen zu können, werden zwei zusätzliche DLL-Dateien benötigt; mehr dazu siehe in der SSL-FAQ.

TLS

Der Parameter ⟨SSLMode⟩ gibt an, ob und wie SSL/TLS verwendet wird:

0 = SSL/TLS abgeschaltet, 1 = SSL/TLS auf separatem Port (POP3S),	2 = SSL/TLS auf Standardport, wenn möglich, 3 = SSL/TLS auf Standardport wird erzwungen.

Der Parameter ⟨SSLVerify⟩ regelt die Überprüfung der fremden Server-Zertifikate:

0 = Zertifikatsüberprüfung abgeschaltet, 1 = Zertifikatsüberprüfung, falls Zertifikat vorgelegt,	2 = Zertifikatsüberprüfung immer, 3 = Zertifikatsüberprüfung immer und Vergleich des Serverzertifikats mit lokaler Kopie.

Mit ⟨SSLCaFile⟩ kann eine Datei zur Zertifikatsüberprüfung angegeben werden. Dabei wird der Standard-Wert (Menü »Einstellungen« → »Grundeinstellungen« ⇒ »SSL« → »Datei mit Zertifikaten zwecks Überprüfung«) überschrieben.

Rückgabewert: 1: OK/gestartet, <> 1: nicht gestartet.

```
ControlRunSendMailTLS( ⟨Server⟩: String; ⟨Port⟩: String;
                       ⟨FromSelection⟩: String; ⟨SSLMode⟩: Integer;
                       ⟨SSLVerify⟩: Integer; ⟨SSLCaFile⟩: String ): Integer
```

SMTP Header Port

Startet einen Auftrag zum Senden der E-Mails zu einem SMTP-Server. Ist der Parameter ⟨FromSelection⟩ nicht leer, werden nur E-Mails versendet, deren »!MAIL FROM:«-Header dem Regulären Ausdruck in diesem Parameter entspricht (siehe auch die FAQ »Hamster und Reguläre Ausdrücke«).

Achtung! Um SSL in Verbindung mit dem Hamster nutzen zu können, werden zwei zusätzliche DLL-Dateien benötigt; mehr dazu siehe in der SSL-FAQ.

Der Parameter ⟨SSLMode⟩ gibt an, ob und wie SSL/TLS verwendet wird:

0 = SSL/TLS abgeschaltet, 1 = SSL/TLS auf separatem Port (SMTPS),	2 = SSL/TLS auf Standardport, wenn möglich, 3 = SSL/TLS auf Standardport wird erzwungen.

Der Parameter ⟨SSLVerify⟩ regelt die Überprüfung der fremden Server-Zertifikate:

0 = Zertifikatsüberprüfung abgeschaltet, 1 = Zertifikatsüberprüfung, falls Zertifikat vorgelegt,	2 = Zertifikatsüberprüfung immer, 3 = Zertifikatsüberprüfung immer und Vergleich des Serverzertifikats mit lokaler Kopie.

Mit ⟨SSLCaFile⟩ kann eine Datei zur Zertifikatsüberprüfung angegeben werden. Dabei wird der Standard-Wert (Menü »Einstellungen« → »Grundeinstellungen« ⇒ »SSL« → »Datei mit Zertifikaten zwecks Überprüfung«) überschrieben.

Rückgabewert: 1: OK/gestartet, <> 1: nicht gestartet.

```
ControlRunSendMailAuthTLS( ⟨Server⟩: String; ⟨Port⟩: String;
                           ⟨Benutzer⟩: String; ⟨Passwort⟩: String;
                           ⟨FromSelection⟩: String; ⟨SSLMode⟩: Integer;
                           ⟨SSLVerify⟩: Integer; ⟨SSLCaFile⟩: String
                          ): Integer
```

Glossar: SMTP, SMTP-AUTH, Port, Header

Startet einen Auftrag zum Senden der E-Mails zu einem SMTP-Server unter Benutzung des SMTP-AUTH-Verfahrens. Für ⟨Benutzer⟩ kann auch eine im Passwort-Menü definierte Variable verwendet werden, ⟨Passwort⟩ bleibt dann leer. Ist der Parameter ⟨FromSelection⟩ nicht leer, werden nur E-Mails versendet, deren »!MAIL FROM:«-Header dem Regulären Ausdruck in diesem Parameter entspricht (siehe auch die FAQ »Hamster und Reguläre Ausdrücke«).

SSL

Achtung! Um SSL in Verbindung mit dem Hamster nutzen zu können, werden zwei zusätzliche DLL-Dateien benötigt; mehr dazu siehe in der SSL-FAQ.

TLS

Der Parameter ⟨SSLMode⟩ gibt an, ob und wie SSL/TLS verwendet wird:

0 = SSL/TLS abgeschaltet, 1 = SSL/TLS auf separatem Port (SMTPS),	2 = SSL/TLS auf Standardport, wenn möglich, 3 = SSL/TLS auf Standardport wird erzwungen.

Der Parameter ⟨SSLVerify⟩ regelt die Überprüfung der fremden Server-Zertifikate:

0 = Zertifikatsüberprüfung abgeschaltet, 1 = Zertifikatsüberprüfung, falls Zertifikat vorgelegt,	2 = Zertifikatsüberprüfung immer, 3 = Zertifikatsüberprüfung immer und Vergleich des Serverzertifikats mit lokaler Kopie.

Mit ⟨SSLCaFile⟩ kann eine Datei zur Zertifikatsüberprüfung angegeben werden. Dabei wird der Standard-Wert (Menü »Einstellungen« → »Grundeinstellungen« ⇒ »SSL« → »Datei mit Zertifikaten zwecks Überprüfung«) überschrieben.

Rückgabewert: 1: OK/gestartet, <> 1: nicht gestartet.

```
ControlNewMail ( ⟨EnvelopeFrom⟩: String; ⟨EnvelopeTo⟩: String;
                 ⟨Mailtext⟩: String ): Integer
```

Mit dieser Funktion wird eine neue E-Mail erstellt. Als ⟨EnvelopeFrom⟩ muss die E-Mail-Adresse des Absenders bzw. der Name des Accounts angegeben werden. ⟨EnvelopeTo⟩ muss die E-Mail-Adresse bzw. den Accountnamen des Empfängers enthalten, es können mehrere Empfänger, jeder mit Semikolon getrennt, angegeben werden. ⟨Mailtext⟩ muss die komplette E-Mail in einer Variablen enthalten, also mit Mailheadern und Body entsprechend RFC 2881/RFC 2882.

Achtung! Wer diesen Befehl nutzt, sollte sein Skript gründlich lokal austesten, bevor er eine E-Mail ins Internet versendet.

Rückgabewert: 1, wenn die E-Mail ordnungsgemäß gespeichert werden konnte, anderenfalls 0.

`ControlRunNewsPost( ⟨ServerList⟩: String ): Integer`

Startet einen Auftrag zum Versenden von Artikeln an einen oder mehrere Newsserver. Ist die Serverliste leer, werden alle verfügbaren Newsserver verwendet, ansonsten müssen die Server in der Form »Servername1,Port1;Servername2,Port2; ... « angegeben werden. Ab der Hamster-Version 1.3.19.0 sollte diese Funktion nicht mehr verwendet werden. Sie wurde durch die nachfolgend aufgeführte Funktionsgruppe »*Newsjobs**« ersetzt.

`ControlRunNewsPull( ⟨ServerList⟩: String ): Integer`

Startet einen Auftrag zum Abholen von Artikeln bei einem oder mehreren Newsservern. Ist die Serverliste leer, werden alle verfügbaren Newsserver verwendet, ansonsten müssen die Server in der Form »Servername1,Port1;Servername2,Port2; ... « angegeben werden. Ab der Hamster-Version 1.3.19.0 sollte diese Funktion nicht mehr verwendet werden. Sie wurde durch die nachfolgend aufgeführte Funktionsgruppe »*Newsjobs**« ersetzt.

Rückgabewert: 1: OK/gestartet, <> 1: nicht gestartet.

`NewsJobsClear: Integer`

Löscht die Jobliste.

`NewsJobsPullDef( ⟨ServerList⟩: String ): Integer`

Fügt die Ladeaufträge für Artikel automatisch der Jobliste zu, welche mit den in der Serverliste angegebenen Servern verknüpft sind. Ist die Serverliste leer, werden alle verfügbaren Newsserver verwendet, ansonsten müssen die Server in der Form »Servername1,Port1;Servername2,Port2; ... « angegeben werden.

Rückgabewert: Anzahl der hinzugefügten Pull-Jobs.

`NewsJobsPostDef( ⟨ServerList⟩: String ): Integer`

Fügt die Sendeaufträge für Artikel automatisch der Jobliste zu, welche mit den in der Serverliste angegebenen Servern verknüpft sind. Ist die Serverliste leer, werden alle verfügbaren Newsserver verwendet, ansonsten müssen die Server in der Form »Servername1,Port1;Servername2,Port2; ... « angegeben werden.

Rückgabewert: Anzahl der hinzugefügten Post-Jobs.

`NewsJobsPull( ⟨Servername⟩: String; ⟨REGroupSelect⟩: String ): Integer`

Fügt einzelne Ladeaufträge für Artikel der Jobliste zu, welche mit dem angegebenen Server und den zu der Gruppenselektion passenden Gruppen verknüpft sind. Ist die Gruppenselektion leer, werden alle für den angegebenen Server verfügbaren Gruppen verwendet. (Siehe auch die FAQ »Hamster und Reguläre Ausdrücke«.)

Rückgabewert: Anzahl der hinzugefügten Pull-Jobs.

`NewsJobsPost( ⟨Servername⟩: String; ⟨REGroupSelect⟩: String; ⟨REMessageSelect⟩: String ): Integer`

Fügt einzelne Sendeaufträge für Artikel der Jobliste zu, welche mit dem angegebenen Server und den zu der Gruppenselektion passenden Gruppen verknüpft sind. Ist die Gruppenselektion leer, werden alle verfügbaren Newsgruppen verwendet. Der Reguläre Ausdruck in `⟨REMessageSelect⟩` kann zur erweiterten Auswahl verwendet werden: Nur wenn die dort angegebene RegExp auf den Header des Artikels zutrifft, wird er auch versandt (siehe auch die FAQ »Hamster und Reguläre Ausdrücke«).

Rückgabewert: Anzahl der hinzugefügten Post-Jobs.

`NewsJobsFeed( ⟨Server⟩: String; ⟨REGroupSelect⟩: String ): Integer`

Fügt einzelne Aufträge zum Austausch von Artikeln mit anderen Servern der Jobliste zu, welche mit dem Server `⟨Server⟩` und den in `⟨REGroupSelect⟩` angegebenen Gruppen realisierbar sind. Ist die Gruppenselektion leer, werden alle verfügbaren Gruppen verwendet, anderenfalls nur solche, die zum Regulären Ausdruck in `⟨REGroupSelect⟩` passen (siehe auch die FAQ »Hamster und Reguläre Ausdrücke«).

Achtung! *Diese Funktion ist nur für den Artikelaustausch von Servern untereinander bestimmt und nicht für den gewöhnlichen Artikelaustausch mit den Providern geeignet. Es wird bei diesem Befehl die Funktionalität des »IHAVE«-Kommando (RFC 977) verwendet.*

Rückgabewert: < 0: Fehler, >= 0: Anzahl der hinzugefügten Newsjobs.

`NewsJobsStart( ⟨ServerList⟩: String ): Integer`

Startet die Abarbeitung der Jobliste für die in der Serverliste angegebenen Server. Wenn die Serverliste leer ist, werden alle in der Serverliste eingetragenen Server für die Abarbeitung der Jobliste herangezogen, ansonsten müssen die Server in der Form »Servername1,Port1;Servername2,Port2; . . . « angegeben werden.

Rückgabewert: Anzahl der gestarteten Threads.

`ControlNewsJobsGetCounter: Integer`

Liefert die Anzahl in der Auftragsliste vorhandenen Einträge.

`ControlNewsJobsGetServer( ⟨Index⟩: Integer ): String`

Liefert den Servernamen eines Eintrages der Auftragsliste.

`ControlNewsJobsGetParam( ⟨Index⟩ : Integer ): String`

Liefert den Parameter eines Eintrages der Auftragsliste. Der Parameter ist vom Typ des Eintrages abhängig.

`ControlNewsJobsGetType( ⟨Index⟩ : Integer ): Integer`

Liefert den Typ des Eintrages.

Rückgabewert: Jobtyp:

```
JOBTYPE_INVALID  = 0
JOBTYPE_SRVINFOS = 1
JOBTYPE_NEWSPOST = 2
JOBTYPE_GETBYMID = 3
JOBTYPE_NEWSPULL = 4
JOBTYPE_NEWSFEED = 5
```

`ControlNewsJobsGetPriority( ⟨Index⟩ :Integer ): Integer`

Liefert die Priorität eines Eintrages der Auftragsliste zurück.

Standardrückgabewert für die jeweiligen Typen (vgl. »ControlNewsJobsGetType«):

JOBPRIO_SRVINFOS: MaxInt-1 (= 2 147 483 646)
JOBPRIO_NEWSPOST: MaxInt-2 (= 2 147 483 645)
JOBPRIO_GETBYMID: MaxInt-3 (= 2 147 483 644)
JOBTYPE_NEWSPULL: Priorität entspricht der Anzahl der Artikel in der Gruppe
JOBPRIO_NEWSFEED: MaxInt-4 (= 2 147 483 643)

`ControlNewsJobsSetPriority( ⟨Index⟩: Integer; ⟨Priority⟩: Integer ): Integer`

Setzt die Priorität für einen Eintrag in der Auftragsliste.

Vorgesehene Werte für `⟨Priority⟩` *(vgl. »ControlNewsJobsGetType«):*

JOBPRIO_SRVINFOS: MaxInt-1 (= 2 147 483 646)
JOBPRIO_NEWSPOST: MaxInt-2 (= 2 147 483 645)
JOBPRIO_GETBYMID: MaxInt-3 (= 2 147 483 644)
JOBTYPE_NEWSPULL: Priorität entspricht der Anzahl der Artikel in der Gruppe
JOBPRIO_NEWSFEED: MaxInt-4 (= 2 147 483 643)

`ControlNewsJobsAdd( ⟨Server⟩: String; ⟨Type⟩: Integer; ⟨Parameter⟩: String; ⟨Priority⟩: Integer ): Integer`

Fügt der Auftragsliste einen Eintrag hinzu. Zu `⟨Type⟩` siehe »ControlNewsJobsGetType«.

Vorgesehene Werte für `⟨Priority⟩`:

JOBPRIO_SRVINFOS: MaxInt-1 (= 2 147 483 646)
JOBPRIO_NEWSPOST: MaxInt-2 (= 2 147 483 645)
JOBPRIO_GETBYMID: MaxInt-3 (= 2 147 483 644)
JOBTYPE_NEWSPULL: Priorität entspricht der Anzahl der Artikel in der Gruppe
JOBPRIO_NEWSFEED: MaxInt-4 (= 2 147 483 643)

`ControlNewsJobsDelete( ⟨Index⟩ )`

Löscht einen einzelnen Job an der Position `⟨Index⟩` aus der Jobliste.

Rückgabewert: 0

RAS-Funktionen:

`RasDial( ⟨ConnectionID⟩: String; ⟨Username⟩: String; ⟨Password⟩: String ): Bool`

Stellt eine DFÜ-Netzwerkverbindung mit dem DFÜ-Namen `⟨ConnectionID⟩` her. Der Name muss einem Eintrag im Telefonbuch des DFÜ-Netzwerkes exakt entsprechen. Sind

`⟨Username⟩` und `⟨Password⟩` leer, werden die Voreinstellungen aus den lokalen Einstellungen des Hamsters bzw. die zuletzt per »ControlSetLogin« gesetzten Einstellungen verwendet.

Wurde die Verbindung erfolgreich hergestellt, wird »TRUE«, anderenfalls »FALSE« zurückgeliefert.

`RasHangup`

Schließt die aktuelle DFÜ-Netzwerkverbindung.

`RasIsConnected: Bool`

Falls eine DFÜ-Netzwerkverbindung besteht, wird »TRUE«, anderenfalls »FALSE« zurückgeliefert.

`RasLastError: Integer`

Liefert den (letzten) Fehlerstatus des DFÜ-Netzwerkes zurück.

`RasGetConnection: String`

Liefert den Namen der DFÜ-Verbindung zurück.

`RasGetIP: String`

Liefert die IP-Adresse der DFÜ-Verbindung zurück.

`RasListEntries: String`

Liefert die Namen aller zur Verfügung stehenden DFÜ-Verbindungen zurück.

News-Funktionen:

`ControlAddGroup( ⟨GroupName⟩: String ): Integer`

Legt eine neue lokale Gruppe mit dem Namen `⟨GroupName⟩` an.

Rückgabewert: 0 = OK/angelegt, –1 = nicht angelegt.

`ControlDelGroup( ⟨GroupName⟩: String )`

Löscht eine Gruppe mit dem Namen ⟨**GroupName**⟩ und alle möglicherweise vorhandenen Pulls für diese Gruppe.

Rückgabewert: 0 = OK/gelöscht, –1 = nicht gelöscht.

`ControlAddPull( ⟨Server⟩: String; ⟨GroupName⟩: String ): Integer`

Legt einen neuen Ladeauftrag an. Falls die Gruppe vorher noch nicht existierte, wird sie automatisch angelegt.

Rückgabewert: 0 = OK/angelegt, –1 = nicht angelegt.

`ControlDelPull( ⟨Server⟩: String; ⟨GroupName⟩: String ): Integer`

Löscht einen Ladeauftrag. Die Gruppe bleibt dabei erhalten.

Rückgabewert: 0 = OK/gelöscht, –1 = nicht gelöscht.

`NewsGrpCount: Integer`

Liefert die Anzahl der im Hamster verfügbaren Newsgruppen zurück.

`NewsGrpName( ⟨GroupIndex⟩: Integer ): String`

Liefert den Namen einer Newsgruppe zu Ihrem Index zurück.

Hinweis: Die Zählung der Indizes beginnt bei Null und endet mit »NewsGrpCount – 1«.

`NewsGrpIndex( ⟨GroupName⟩: String ): Integer`

Liefert den Index einer Newsgruppe oder –1 bei einer unbekannten Newsgruppe zurück.

Hinweis: Die Zählung der Indizes beginnt bei Null und endet mit »NewsGrpCount – 1«.

`NewsGrpOpen( ⟨GroupName⟩: String ): Integer`

Öffnet eine Newsgruppe. Das zurückgelieferte Handle dient dann dem weiteren Zugriff auf die geöffnete Gruppe und wird in den nachfolgend aufgeführten Funktionen als *⟨GroupHandle⟩* bezeichnet. Trat ein Fehler beim Öffnen der Newsgruppe auf, wird ein negativer Wert zurückgeliefert.

Hinweis: Newsgruppen müssen im Skript immer explizit geschlossen werden (mit »NewsGrpClose«), da das Skriptende oder das Schließen des Hamsterobjektes die Gruppe nicht automatisch schließen und somit ein Speicherleck entsteht.

`NewsGrpClose( ⟨GroupHandle⟩: Integer )`

Schließt eine Newsgruppe. ⟨**GroupHandle**⟩ ist der vorher durch »NewsGrpOpen« zurückgelieferte Wert.

Hinweis: Newsgruppen müssen im Skript immer explizit geschlossen werden, da das Skriptende oder das Schließen des Hamsterobjektes die Gruppe nicht automatisch schließen und somit ein Speicherleck entsteht.

`NewsGrpXOver( ⟨GroupHandle⟩: Integer; ⟨Anfang⟩: Integer; ⟨Ende⟩: Integer ): String`

Der Befehl gibt aus einer Gruppe den XOver der Artikel von ⟨Anfang⟩ bis ⟨Ende⟩ zurück.

Die Gruppe ist die dem ⟨GroupHandle⟩ zugeordnete. Dieses ⟨GroupHandle⟩ ist der vorher durch »NewsGrpOpen« zurückgelieferte Wert.

Der XOver beinhaltet (in der gegebenen Reihenfolge):

ArtikelNummer	Message-ID	X-Ref
Subject	References	X-Hamster-Info
From	Artsize	
Datum	Lines	

Der Trenner zwischen diesen ist #9 (Tab). CR (#10), LF (#13) und bereits vorhandene Tabs (#9) werden jeweils durch ein Leerzeichen (#32) ersetzt.

`NewsGrpNameByHandle( ⟨GroupHandle⟩: Integer ): String`

Gibt den Namen der angegebene Newsgruppe, die dem ⟨GroupHandle⟩ entspricht. (Das ⟨GroupHandle⟩ ist der vorher durch »NewsGrpOpen« zurückgelieferte Wert.)

`NewsGrpInformation( ⟨GroupHandle⟩: Integer ): String`

Diese Funktion gibt Informationen über die durch das ⟨GroupHandle⟩ selektierte Newsgruppe zurück. Dieses ⟨GroupHandle⟩ ist der vorher durch »NewsGrpOpen« zurückgelieferte Wert.

Die Ausgabe ist wie ein »XOver« codiert (CR [#10], LF [#13] und Tabs [#9] werden jeweils durch ein Leerzeichen (#32) ersetzt; jeder Wert ist mit #9 [Tab] getrennt).

Diese Informationen werden (in der aufgeführten Reihenfolge) zurückgegeben:

Description	type
pull.limit	pullserver
purge.articles.keepdays	GroupName
postserver	

`NewsArtCount( ⟨GroupHandle⟩: Integer ): Integer`

Liefert die Anzahl der Artikel in einer Gruppe zurück. ⟨GroupHandle⟩ ist der vorher durch »NewsGrpOpen« zurückgelieferte Wert.

`NewsArtNoMin( ⟨GroupHandle⟩: Integer ): Integer`

Liefert die niedrigste noch vorhandene Artikelnummer zurück. ⟨GroupHandle⟩ ist der vorher durch »NewsGrpOpen« zurückgelieferte Wert.

`NewsArtNoMax( ⟨GroupHandle⟩: Integer ): Integer`

Liefert die höchste bislang vorhandene Artikelnummer zurück. `⟨GroupHandle⟩` ist der vorher durch »NewsGrpOpen« zurückgelieferte Wert.

`NewsArtText( ⟨GroupHandle⟩: Integer; ⟨Artikelnummer⟩: Integer ): String`

Liefert den Artikeltext inklusive Header zu der gewünschten Artikelnummer zurück. Ist der Rückgabe-String leer, so ist der Artikel nicht vorhanden. `⟨GroupHandle⟩` ist der vorher durch »NewsGrpOpen« zurückgelieferte Wert.

Glossar: Header

`NewsArtTextExport( ⟨GroupHandle⟩: Integer; ⟨Artikelnummer⟩: Integer ): String`

Liefert den Artikeltext inklusive Header zu der gewünschten Artikelnummer im Exportformat zurück. Das Exportformat bewirkt, dass noch eine Zusatzzeile mit einigen wenigen Informationen aus dem normalen Header dem Artikel vorangestellt wird. Das entspricht dem MBox-Format. Ist der Rückgabe-String leer, so ist der Artikel nicht vorhanden. `⟨GroupHandle⟩` ist der vorher durch »NewsGrpOpen« zurückgelieferte Wert.

`NewsArtXOver( ⟨GroupHandle⟩: Integer; ⟨Artikelnummer⟩: Integer ): String`

Der Befehl gibt den XOver des Artikels mit der `⟨Artikelnummer⟩` zurück aus der Gruppe, die dem `⟨GroupHandle⟩` zugeordnet ist. Dieses `⟨GroupHandle⟩` ist der vorher durch »NewsGrpOpen« zurückgelieferte Wert.

Der XOver beinhaltet (in der gegebenen Reihenfolge):

ArtikelNummer	Message-ID	X-Ref
Subject	References	X-Hamster-Info
From	Artsize	
Datum	Lines	

Der Trenner zwischen diesen ist #9 (Tab). CR (#10), LF (#13) und bereits vorhandene Tabs (#9) werden jeweils durch ein Leerzeichen (#32) ersetzt.

`NewsScoreListFor( ⟨GroupName⟩: String ): String`

Liefert die Scoreliste für eine Gruppe zurück.

`NewsScoreTest( ⟨GroupName⟩: String; ⟨ArtText⟩: String; var ⟨MatchLog⟩: String ): Integer`

Liefert den Scorewert für einen Artikel zurück, den er in der bezeichneten Gruppe mit den aktuellen Einstellungen im Scorefile erhalten würde. Die Variable `⟨MatchLog⟩` enthält anschließend alle auf den Artikel zutreffenden Zeilen aus dem Scorefile.

```
NewsLocateMID( ⟨Message-ID⟩: String; var ⟨GroupName⟩: String;
               var ⟨ArticleNo⟩: Integer ): Bool
```

Liefert »TRUE« zurück, wenn ein Artikel mit der angegebenen Message-ID in der History vorhanden ist. Der Gruppenname und die Artikelnummer innerhalb der Gruppe werden in die Variablen ⟨GroupName⟩ und ⟨ArticleNo⟩ geschrieben.

```
NewsLocateMID2( ⟨Message-ID⟩: String; var ⟨GroupName⟩: Variant;
                var ⟨ArticleNo⟩: Variant ): Bool
```

Liefert »TRUE« zurück, wenn ein Artikel mit der angegebenen Message-ID in der History vorhanden ist. Der Gruppenname und die Artikelnummer innerhalb der Gruppe werden in die Variablen ⟨GroupName⟩ und ⟨ArticleNo⟩ geschrieben.

Spezialausfertigung zur Verwendung in VBS/WSH!

```
NewsDeleteByMID( ⟨Message-ID⟩: String ): Bool
```

Löscht den Artikel mit der angegebenen Message-ID. Liefert bei erfolgreichem Löschen »TRUE« zurück.

```
NewsImport( ⟨ArticleText⟩: String; ⟨OverrideGroups⟩: String;
            ⟨IgnoreHistory⟩: Bool; ⟨MarkNoArchive⟩: Bool ): Bool
```

Importiert den Artikel in den Hamster und liefert im Erfolgsfalle »TRUE« zurück. Der Artikeltext im Parameter ⟨ArticleText⟩ muss aus einem Headerblock, einer Leerzeile und dem Textbody bestehen. Im Headerblock müssen mindestens der »Date«-, der »From«- und der »Subject«-Header enthalten sein.

Glossar: Header

Der Newsgroups-Header ist dann nötig, wenn der Parameter ⟨OverrideGroups⟩ leer ist. Dieser Parameter erlaubt es, statt der eigentlichen Zielgruppe(n) eine beliebige Gruppe als Ziel für den Artikel festzulegen. Ist die Gruppe unbekannt, wird der Artikel in der dafür zuständigen internen Gruppe gespeichert. Ist der Parameter ⟨IgnoreHistory⟩ »TRUE«, wird der Artikel auch dann importiert, wenn er bereits in der History bekannt ist.

Ist der Parameter ⟨MarkNoArchive⟩ »TRUE«, wird der Artikel im Header »X-Hamster-Info« mit »NoArchive=1« gekennzeichnet.

Nach dem Importieren der Artikel empfiehlt es sich, die Funktion »ControlFlush« zum Sichern der Daten auf die Festplatte aufzurufen.

Sonstige Funktionen

```
ControlGetProcessIdentifier: String
```

Liefert den Mutexstring des aktuellen Hamsters zurück.

```
XCounter( ⟨Nummer⟩: Integer ): Integer
```

Liefert den Inhalt der Fußzeilenvariablen ⟨Nummer⟩ zurück.

`ClearXCounter( ⟨von_Nummer⟩: Integer; ⟨bis_Nummer⟩: Integer ): Integer`

Löscht den Inhalt der Fußzeilenvariablen von ⟨von_Nummer⟩ bis zu ⟨bis_Nummer⟩.

`SetXCounter( ⟨Nummer⟩: Integer; ⟨Wert⟩: Integer ): Integer`

Verändert den Inhalt der Fußzeilenvariablen ⟨Nummer⟩ zu ⟨Wert⟩.

`DecXCounter( ⟨Nummer⟩: Integer; ⟨Wert⟩: Integer ): Integer`

Verringert den Inhalt der Fußzeilenvariablen ⟨Nummer⟩ um ⟨Wert⟩.

`IncXCounter( ⟨Nummer⟩: Integer; ⟨Wert⟩: Integer ): Integer`

Erhöht den Inhalt der Fußzeilenvariablen ⟨Nummer⟩ um ⟨Wert⟩.

`ControlRotatelog`

Legt ein neues Logfile an.

`DecodeMIMEHeaderString( ⟨Text⟩: String; ⟨Charset⟩: String ): String`

Glossar: Base64 Quoted-printable

Liefert die decodierte Version einer gemäß MIME base64- oder quoted-printable-codierten Zeichenkette zurück. Die Variable ⟨Charset⟩ liefert, falls vorhanden, den Zeichensatz zurück, mit dem der String codiert war. Der Rückgabewert enthält nach erfolgreicher Decodierung den 8-Bit-Zeichencode an Stelle der ehemals MIME-codierten 7-Bit-Zeichenkette. Die Bedeutung der zurückgegebenen Zeichenkette ist von ⟨Charset⟩ abhängig und muss nicht mit dem Windows-Zeichensatz übereinstimmen.

`DecodeBase64( ⟨Text⟩: String ): String`

Decodiert base64-codierte Texte.

`DecodeQP( ⟨Text⟩: String ): String`

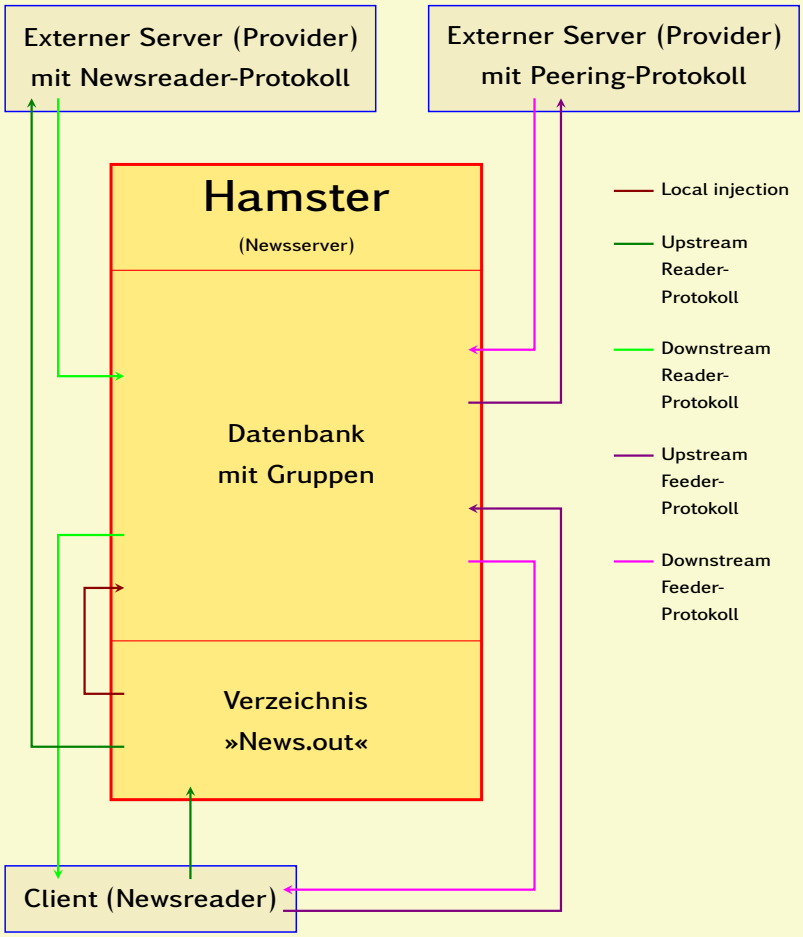
Decodiert quoted-printable-codierte Texte.

`hs2AsyncCommand( ⟨Command⟩: String )`

Aufruf von Skriptbefehlen, die nicht als OLE-Befehl verfügbar sind. Ein direkter Rückwert der aufgerufenen hs2-Funktion ist nicht möglich.

Sonstiges

Übersicht über den Datenfluss der Newsartikel im Hamster



(*Vgl. auch* Wie werden E-Mails im Hamster geroutet?)

Fehlermeldungen und Statuscodes

Häufiger vorkommende Fehler – Hamster-Logdatei, Hamster-Meldungen, WinSock-Fehler

Windows-Sockets-Fehler (Netzwerk-Fehler) – Nummern: 10004–11004

Remote-Access-Service-Fehler (DFÜ-Netzwerk-Fehler) – Nummern: 600–752

Win32-System-Fehler – Nummern: 1–6118

Win32-Netzwerk-Fehler – Nummern: 2102–2693

SSL-Fehler

Servermeldungen bzw. Protokoll-Antwortcodes

Häufiger vorkommende Fehler

Hamster-Protokoll

Start des lokalen XXXX-Servers misslungen

Der bezeichnete lokale Server kann nicht gestartet werden. Häufigste Ursache: Ein anderes Programm benutzt bereits den benötigten TCP-Port des Servers, z. B. Virus-Scanner für E-Mails – wie beispielsweise »Norton AntiVirus« (NAV) – welche selber als POP3-Server beim Betriebssystem angemeldet sind, oder auch ein zweiter Hamster (aus einem anderem Verzeichnis als der schon laufende Hamster). Abhilfe schafft hier die Verwendung einer anderen Port-Nummer für den lokalen Server. Diese Änderung muss dann natürlich auch im News- bzw. E-Mail-Client/Reader eingetragen werden, da ansonsten nicht auf den Hamster, sondern eben auf das andere Programm zugegriffen würde.

Glossar: TCP Port POP3

Hamster-Meldungen

500 Permission denied - closing connection. [local NNTP/SMTP]
-ERR Permission denied - closing connection. [local POP3]

Die für den Zugriff auf den Hamster verwendete IP-Adresse ist nicht freigegeben. Details sind hierzu der Beschreibung zur Datei »`IPAccess.hst`« zu entnehmen. Mit entsprechenden Änderungen in dieser Datei ist der Zugriff einschaltbar.

NNTP SMTP

500 Permission denied (not authorized by POP3). [local SMTP]

Der Zugriff auf den lokalen SMTP-Server ist durch das »SMTP after POP3«-Verfahren gesperrt, d. h. Mails können nur dann verschickt werden, wenn kurz vorher eine Abfrage nach neuen Mails erfolgt ist. Siehe hierzu die Ausführungen zum Menü »Einstellungen« → »Lokale Server« ⇒ »SMTP«.

501 Syntax error, missing brackets. [local SMTP]

Beim Mailversand gehören um den Absender/Empfänger im »MAIL FROM:/RCPT TO:« spitze Klammern, und wenn das einliefernde Programm diese nicht benutzt, beschwert sich der Hamster. Um ihn etwas toleranter zu machen, kann man in der »`hamster.ini`«, Sektion »[Setup]«, folgenden Eintrag vornehmen:

```
mail.smtpneedbrackets=0
```

WinSock-Fehler

10048 WSAEADDRINUSE *(Address already in use.)*

Mögliche Ursache: Eine Firewall verhindert wahrscheinlich den Start des lokalen Servers.

10049 WSAEADDRNOTAVAIL *(Cannot assign requested address.)*

Der Server ist nicht erreichbar.

Mögliche Ursachen:

- Keine Online-Verbindung ins Internet.
- Die Internetverbindung wurde unterbrochen.
- Der Server ist vorübergehend abgeschaltet worden.

10053 WSAECONNABORTED *(Software caused connection abort.)*

Mögliche Ursache: Internetverbindung oder Verbindung zum Server wurde unterbrochen.

10055 WSAENOBUFS *(No buffer space available.)*

Bei Auftreten eines WinSock-Fehlers »WSAENOBUFS (10055)« bei NT/2000/XP liegt ein internes Problem des TCP/IP-Systems vor. Dieses ist normalerweise nur darauf ausgelegt, für ausgehende TCP/IP-Verbindungen die Ports von 1024 bis 5000 zu verwenden. Beim gleichzeitigen Einsatz von Filesharing-Programmen kann dies zu wenig sein. Um den Fehler zu umgehen, muss die maximale Anzahl der verfügbaren Portnummern heraufgesetzt werden. Hierzu wird folgender Wert in der Registry verändert:

Glossar: TCP Port

Schlüssel:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters

Name: MaxUserPort
Werttyp: DWORD
Wert: 65534
Zulässiger Bereich: 5000–65534 (dezimal)
Standardwert: 0x1388 (= 5000 dezimal)

Die Erklärung zu diesem Verhalten findet sich in folgendem Artikel in der Microsoft Knowledge Base: http://support.microsoft.com/?kbid=196271.

10061 WSAECONNREFUSED *(Connection refused.)*

Mögliche Ursachen:

- Der lokale Server des Hamster wurde nicht gestartet.
- Fehlerhafte Porteinstellung für den lokalen Server oder bei der Client-Software.
- Der Port wird durch eine Firewall blockiert.

10065 WSAEHOSTUNREACH *(No route to host.)*

Ähnlich wie Fehler 10049, nur dass eine vorher korrekte Verbindung nicht mehr funktioniert.

10071 WSAEREMOTE *(Too many levels of remote in path.)*

Mögliche Ursache: Verbindung wurde vom Remote-Server z. B. wegen fehlender Benutzerrechte wieder beendet.

11001 WSAHOST_NOT_FOUND *(Host not found.)*

Mögliche Ursache: Servername wurde nicht gefunden, eventuell ist dieses ein simpler Schreibfehler.

WinSock-Fehler

Windows Sockets (WinSock) Error Codes

WSAEINTR	10004	Interrupted system call.
WSAEBADF	10009	Bad file number.
WSEACCES	10013	Permission denied.
WSAEFAULT	10014	Bad address.
WSAEINVAL	10022	Invalid argument.
WSAEMFILE	10024	Too many open files/sockets.
WSAEWOULDBLOCK	10035	Operation would block.
WSAEINPROGRESS	10036	Operation now in progress. This error is returned if any Windows Sockets API function is called while a blocking function is in progress.
WSAEALREADY	10037	Operation already in progress.
WSAENOTSOCK	10038	Socket operation on nonsocket.
WSAEDESTADDRREQ	10039	Destination address required.
WSAEMSGSIZE	10040	Message too long.
WSAEPROTOTYPE	10041	Protocol wrong type for socket.
WSAENOPROTOOPT	10042	Protocol not available/bad protocol option.
WSAEPROTONOSUPPORT	10043	Protocol not supported.
WSAESOCKTNOSUPPORT	10044	Socket type not supported.
WSAEOPNOTSUPP	10045	Operation not supported on socket.
WSAEPFNOSUPPORT	10046	Protocol family not supported.
WSAEAFNOSUPPORT	10047	Address family not supported by protocol family.
WSAEADDRINUSE	10048	Address already in use.
WSAEADDRNOTAVAIL	10049	Cannot assign requested address.
WSAENETDOWN	10050	Network is down. This error may be reported at any time if the Windows Sockets implementation detects an underlying failure.
WSAENETUNREACH	10051	Network is unreachable.
WSAENETRESET	10052	Network dropped connection on reset.
WSAECONNABORTED	10053	Software caused connection abort.
WSAECONNRESET	10054	Connection reset by peer.
WSAENOBUFS	10055	No buffer space available.
WSAEISCONN	10056	Socket is already connected.
WSAENOTCONN	10057	Socket is not connected.
WSAESHUTDOWN	10058	Cannot send after socket shutdown.
WSAETOOMANYREFS	10059	Too many references: cannot splice.
WSAETIMEDOUT	10060	Connection timed out.
WSAECONNREFUSED	10061	Connection refused.
WSAELOOP	10062	Too many levels of symbolic links.
WSAENAMETOOLONG	10063	File name too long.
WSAEHOSTDOWN	10064	Host is down.

WSAEHOSTUNREACH	10065	No route to host.
WSAENOTEMPTY	10066	Directory not empty.
WSAEPROCLIM	10067	Too many processes.
WSAEUSERS	10068	Too many users.
WSAEDQUOT	10069	Disc quota exceeded.
WSAESTALE	10070	Stale NFS file handle
WSAEREMOTE	10071	Too many levels of remote in path.
WSASYSNOTREADY	10091	Network subsystem is unavailable.
WSAVERNOTSUPPORTED	10092	Winsock version not supported.
WSANOTINITIALISED	10093	Winsock not yet initialized.
WSAEDISCON	10101	Graceful disconnect in progress.
WSAENOMORE	10102	- (Winsock2)
WSAECANCELLED	10103	- (Winsock2)
WSAEINVALIDPROCTABLE	10104	- (Winsock2)
WSAEINVALIDPROVIDER	10105	- (Winsock2)
WSAEPROVIDERFAILEDINIT	10106	- (Winsock2)
WSASYSCALLFAILURE	10107	System call failure. (WS2)
WSASERVICE_NOT_FOUND	10108	- (Winsock2)
WSATYPE_NOT_FOUND	10109	- (Winsock2)
WSA_E_NO_MORE	10110	- (Winsock2)
WSA_E_CANCELLED	10111	- (Winsock2)
WSAEREFUSED	10112	- (Winsock2)
WSAHOST_NOT_FOUND	11001	Host not found. This message indicates that the key (name, address, and so on) was not found.
WSATRY_AGAIN	11002	Non-authoritative host not found. This error may suggest that the name service itself is not functioning.
WSANO_RECOVERY	11003	Non-recoverable error. This error may suggest that the name service itself is not functioning.
WSANO_DATA	11004	Valid name, no data record of requested type. This error indicates that the key (name, address, and so on) was not found.
WSA_NOT_ENOUGH_MEMORY		Insufficient memory available
WSA_OPERATION_ABORTED		Overlapped operation aborted
WSA_IO_INCOMPLETE		Overlapped I/O object not signalled
WSA_IO_PENDING		Overlapped I/O will complete later
WSA_INVALID_PARAMETER		One or more parameters are invalid
WSA_INVALID_HANDLE		Event object handle not valid

Fehler des RAS (DFÜ)

Remote Access Service (RAS) Error Codes

600 An operation is pending.
601 The port handle is invalid.
602 The port is already open.
603 Caller's buffer is too small.
604 Wrong information specified.
605 Cannot set port information.
606 The port is not connected.
607 The event is invalid.
608 The device does not exist.
609 The device type does not exist.
610 The buffer is invalid.
611 The route is not available.

612 The route is not allocated.
613 Invalid compression specified.
614 Out of buffers.
615 The port was not found.
616 An asynchronous request is pending.
617 The port or device is already disconnecting.
618 The port is not open.
619 The port is disconnected.
620 There are no endpoints.
621 Cannot open the phone book file.
622 Cannot load the phone book file.
623 Cannot find the phone book entry.
624 Cannot write the phone book file.
625 Invalid information found in the phone book.
626 Cannot load a string.
627 Cannot find key.
628 The port was disconnected.
629 The port was disconnected by the remote machine.
630 The port was disconnected due to hardware failure.
631 The port was disconnected by the user.
632 The structure size is incorrect.
633 The port is already in use or is not configured for Remote Access dialout.
634 Cannot register your computer on the remote network.
635 Unknown error.
636 The wrong device is attached to the port.
637 The string could not be converted.
638 The request has timed out.
639 No asynchronous net available.
640 A NetBIOS error has occurred.
641 The server cannot allocate NetBIOS resources needed to support the client.
642 One of your NetBIOS names is already registered on the remote network.
643 A network adapter at the server failed.
644 You will not receive network message popups.
645 Internal authentication error.
646 The account is not permitted to log on at this time of day.
647 The account is disabled.
648 The password has expired.
649 The account does not have Remote Access permission.
650 The Remote Access server is not responding.
651 Your modem (or other connecting device) has reported an error.
652 Unrecognized response from the device.
653 A macro required by the device was not found in the device .INF file section.
654 A command or response in the device .INF file section refers to an undefined macro.
655 The <message> macro was not found in the device .INF file section.
656 The <defaultoff> macro in the device .INF file section contains an undefined macro.
657 The device .INF file could not be opened.
658 The device name in the device .INF or media .INI file is too long.
659 The media .INI file refers to an unknown device name.
660 The device .INF file contains no responses for the command.
661 The device .INF file is missing a command.
662 Attempted to set a macro not listed in device .INF file section.
663 The media .INI file refers to an unknown device type.
664 Cannot allocate memory.
665 The port is not configured for Remote Access.

666 Your modem (or other connecting device) is not functioning.
667 Cannot read the media .INI file.
668 The connection dropped.
669 The usage parameter in the media .INI file is invalid.
670 Cannot read the section name from the media .INI file.
671 Cannot read the device type from the media .INI file.
672 Cannot read the device name from the media .INI file.
673 Cannot read the usage from the media .INI file.
674 Cannot read the maximum connection BPS rate from the media .INI file.
675 Cannot read the maximum carrier BPS rate from the media .INI file.
676 The line is busy.
677 A person answered instead of a modem.
678 There is no answer.
679 Cannot detect carrier.
680 There is no dial tone.
681 General error reported by device.
682 ERROR WRITING SECTIONNAME
683 ERROR WRITING DEVICETYPE
684 ERROR WRITING DEVICENAME
685 ERROR WRITING MAXCONNECTBPS
686 ERROR WRITING MAXCARRIERBPS
687 ERROR WRITING USAGE
688 ERROR WRITING DEFAULTOFF
689 ERROR READING DEFAULTOFF
690 ERROR EMPTY INI FILE
691 Access denied because username and/or password is invalid on the domain.
692 Hardware failure in port or attached device.
693 ERROR NOT BINARY MACRO
694 ERROR DCB NOT FOUND
695 ERROR STATE MACHINES NOT STARTED
696 ERROR STATE MACHINES ALREADY STARTED
697 ERROR PARTIAL RESPONSE LOOPING
698 A response keyname in the device .INF file is not in the expected format.
699 The device response caused buffer overflow.
700 The expanded command in the device .INF file is too long.
701 The device moved to a BPS rate not supported by the COM driver.
702 Device response received when none expected.
703 ERROR INTERACTIVE MODE
704 ERROR BAD CALLBACK NUMBER
705 ERROR INVALID AUTH STATE
706 ERROR WRITING INITBPS
707 X.25 diagnostic indication.
708 The account has expired.
709 Error changing password on domain.
710 Serial overrun errors were detected while communicating with your modem.
711 RasMan initialization failure. Check the event log.
712 Biplex port is initializing. Wait a few seconds and redial.
713 No active ISDN lines are available.
714 Not enough ISDN channels are available to make the call.
715 Too many errors occurred because of poor phone line quality.
716 The Remote Access IP configuration is unusable.
717 No IP addresses are available in the static pool of Remote Access IP addresses.
718 PPP timeout.
719 PPP terminated by remote machine.

720 No PPP control protocols configured.
721 Remote PPP peer is not responding.
722 The PPP packet is invalid.
723 The phone number, including prefix and suffix, is too long.
724 The IPX protocol cannot dial-out on the port because the computer is an IPX router.
725 The IPX protocol cannot dial-in on the port because the IPX router is not installed.
726 The IPX protocol cannot be used for dial-out on more than one port at a time.
727 Cannot access TCPCFG.DLL.
728 Cannot find an IP adapter bound to Remote Access.
729 SLIP cannot be used unless the IP protocol is installed.
730 Computer registration is not complete.
731 The protocol is not configured.
732 The PPP negotiation is not converging.
733 The PPP control protocol for this network protocol is not available on the server.
734 The PPP link control protocol terminated..
735 The requested address was rejected by the server..
736 The remote computer terminated the control protocol.
737 Loopback detected.
738 The server did not assign an address.
739 The remote server cannot use the Windows NT encrypted password.
740 The TAPI devices configured for Remote Access failed to initialize or were not installed correctly.
741 The local computer does not support encryption.
742 The remote server does not support encryption.
743 The remote server requires encryption.
744 Cannot use the IPX net number assigned by the remote server. Check the event log.
752 A syntax error was encountered while processing a script.

Win32-System-Fehler

Win32 System Error Codes

0 The operation completed successfully. ERROR_SUCCESS
1 Incorrect function. ERROR_INVALID_FUNCTION
2 The system cannot find the file specified. ERROR_FILE_NOT_FOUND
3 The system cannot find the path specified. ERROR_PATH_NOT_FOUND
4 The system cannot open the file. ERROR_TOO_MANY_OPEN_FILES
5 Access is denied. ERROR_ACCESS_DENIED
6 The handle is invalid. ERROR_INVALID_HANDLE
7 The storage control blocks were destroyed. ERROR_ARENA_TRASHED
8 Not enough storage is available to process this command. ERROR_NOT_ENOUGH_MEMORY
9 The storage control block address is invalid. ERROR_INVALID_BLOCK
10 The environment is incorrect. ERROR_BAD_ENVIRONMENT
11 An attempt was made to load a program with an incorrect format. ERROR_BAD_FORMAT
12 The access code is invalid. ERROR_INVALID_ACCESS
13 The data is invalid. ERROR_INVALID_DATA
14 Not enough storage is available to complete this operation. ERROR_OUTOFMEMORY
15 The system cannot find the drive specified. ERROR_INVALID_DRIVE
16 The directory cannot be removed. ERROR_CURRENT_DIRECTORY
17 The system cannot move the file to a different disk drive. ERROR_NOT_SAME_DEVICE
18 There are no more files. ERROR_NO_MORE_FILES
19 The media is write protected. ERROR_WRITE_PROTECT
20 The system cannot find the device specified. ERROR_BAD_UNIT

21 The device is not ready. ERROR_NOT_READY
22 The device does not recognize the command. ERROR_BAD_COMMAND
23 Data error (cyclic redundancy check). ERROR_CRC
24 The program issued a command but the command length is incorrect. ERROR_BAD_LENGTH
25 The drive cannot locate a specific area or track on the disk. ERROR_SEEK
26 The specified disk or diskette cannot be accessed. ERROR_NOT_DOS_DISK
27 The drive cannot find the sector requested. ERROR_SECTOR_NOT_FOUND
28 The printer is out of paper. ERROR_OUT_OF_PAPER
29 The system cannot write to the specified device. ERROR_WRITE_FAULT
30 The system cannot read from the specified device. ERROR_READ_FAULT
31 A device attached to the system is not functioning. ERROR_GEN_FAILURE
32 The process cannot access the file because it is being used by another process. ERROR_SHARING_VIOLATION
33 The process cannot access the file because another process has locked a portion of the file. ERROR_LOCK_VIOLATION
34 The wrong diskette is in the drive. Insert %2 (Volume Serial Number: %3) into drive %1. ERROR_WRONG_DISK
36 Too many files opened for sharing. ERROR_SHARING_BUFFER_EXCEEDED
38 Reached the end of the file. ERROR_HANDLE_EOF
39 The disk is full. ERROR_HANDLE_DISK_FULL
50 The network request is not supported. ERROR_NOT_SUPPORTED
51 The remote computer is not available. ERROR_REM_NOT_LIST
52 A duplicate name exists on the network. ERROR_DUP_NAME
53 The network path was not found. ERROR_BAD_NETPATH
54 The network is busy. ERROR_NETWORK_BUSY
55 The specified network resource or device is no longer available. ERROR_DEV_NOT_EXIST
56 The network BIOS command limit has been reached. ERROR_TOO_MANY_CMDS
57 A network adapter hardware error occurred. ERROR_ADAP_HDW_ERR
58 The specified server cannot perform the requested operation. ERROR_BAD_NET_RESP
59 An unexpected network error occurred. ERROR_UNEXP_NET_ERR
60 The remote adapter is not compatible. ERROR_BAD_REM_ADAP
61 The printer queue is full. ERROR_PRINTQ_FULL
62 Space to store the file waiting to be printed is not available on the server. ERROR_NO_SPOOL_SPACE
63 Your file waiting to be printed was deleted. ERROR_PRINT_CANCELLED
64 The specified network name is no longer available. ERROR_NETNAME_DELETED
65 Network access is denied. ERROR_NETWORK_ACCESS_DENIED
66 The network resource type is not correct. ERROR_BAD_DEV_TYPE
67 The network name cannot be found. ERROR_BAD_NET_NAME
68 The name limit for the local computer network adapter card was exceeded. ERROR_TOO_MANY_NAMES
69 The network BIOS session limit was exceeded. ERROR_TOO_MANY_SESS
70 The remote server has been paused or is in the process of being started. ERROR_SHARING_PAUSED
71 No more connections can be made to this remote computer at this time because there are already as many connections as the computer can accept. ERROR_REQ_NOT_ACCEP
72 The specified printer or disk device has been paused. ERROR_REDIR_PAUSED
80 The file exists. ERROR_FILE_EXISTS
82 The directory or file cannot be created. ERROR_CANNOT_MAKE
83 Fail on INT 24. ERROR_FAIL_I24
84 Storage to process this request is not available. ERROR_OUT_OF_STRUCTURES
85 The local device name is already in use. ERROR_ALREADY_ASSIGNED
86 The specified network password is not correct. ERROR_INVALID_PASSWORD
87 The parameter is incorrect. ERROR_INVALID_PARAMETER

88 A write fault occurred on the network. ERROR_NET_WRITE_FAULT
89 The system cannot start another process at this time. ERROR_NO_PROC_SLOTS
100 Cannot create another system semaphore. ERROR_TOO_MANY_SEMAPHORES
101 The exclusive semaphore is owned by another process. ERROR_EXCL_SEM_ALREADY_OWNED
102 The semaphore is set and cannot be closed. ERROR_SEM_IS_SET
103 The semaphore cannot be set again. ERROR_TOO_MANY_SEM_REQUESTS
104 Cannot request exclusive semaphores at interrupt time. ERROR_INVALID_AT_INTERRUPT_TIME
105 The previous ownership of this semaphore has ended. ERROR_SEM_OWNER_DIED
106 Insert the diskette for drive %1. ERROR_SEM_USER_LIMIT
107 The program stopped because an alternate diskette was not inserted. ERROR_DISK_CHANGE
108 The disk is in use or locked by another process. ERROR_DRIVE_LOCKED
109 The pipe has been ended. ERROR_BROKEN_PIPE
110 The system cannot open the device or file specified. ERROR_OPEN_FAILED
111 The file name is too long. ERROR_BUFFER_OVERFLOW
112 There is not enough space on the disk. ERROR_DISK_FULL
113 No more internal file identifiers available. ERROR_NO_MORE_SEARCH_HANDLES
114 The target internal file identifier is incorrect. ERROR_INVALID_TARGET_HANDLE
117 The IOCTL call made by the application program is not correct. ERROR_INVALID_CATEGORY
118 The verify-on-write switch parameter value is not correct. ERROR_INVALID_VERIFY_SWITCH
119 The system does not support the command requested. ERROR_BAD_DRIVER_LEVEL
120 This function is not supported on this system. ERROR_CALL_NOT_IMPLEMENTED
121 The semaphore timeout period has expired. ERROR_SEM_TIMEOUT
122 The data area passed to a system call is too small. ERROR_INSUFFICIENT_BUFFER
123 The filename, directory name, or volume label syntax is incorrect. ERROR_INVALID_NAME
124 The system call level is not correct. ERROR_INVALID_LEVEL
125 The disk has no volume label. ERROR_NO_VOLUME_LABEL
126 The specified module could not be found. ERROR_MOD_NOT_FOUND
127 The specified procedure could not be found. ERROR_PROC_NOT_FOUND
128 There are no child processes to wait for. ERROR_WAIT_NO_CHILDREN
129 The %1 application cannot be run in Win32 mode. ERROR_CHILD_NOT_COMPLETE
130 Attempt to use a file handle to an open disk partition for an operation other than raw disk I/O. ERROR_DIRECT_ACCESS_HANDLE
131 An attempt was made to move the file pointer before the beginning of the file. ERROR_NEGATIVE_SEEK
132 The file pointer cannot be set on the specified device or file. ERROR_SEEK_ON_DEVICE
133 A JOIN or SUBST command cannot be used for a drive that contains previously joined drives. ERROR_IS_JOIN_TARGET
134 An attempt was made to use a JOIN or SUBST command on a drive that has already been joined. ERROR_IS_JOINED
135 An attempt was made to use a JOIN or SUBST command on a drive that has already been substituted. ERROR_IS_SUBSTED
136 The system tried to delete the JOIN of a drive that is not joined. ERROR_NOT_JOINED
137 The system tried to delete the substitution of a drive that is not substituted. ERROR_NOT_SUBSTED
138 The system tried to join a drive to a directory on a joined drive. ERROR_JOIN_TO_JOIN
139 The system tried to substitute a drive to a directory on a substituted drive. ERROR_SUBST_TO_SUBST
140 The system tried to join a drive to a directory on a substituted drive. ERROR_JOIN_TO_SUBST
141 The system tried to SUBST a drive to a directory on a joined drive. ERROR_SUBST_TO_JOIN
142 The system cannot perform a JOIN or SUBST at this time. ERROR_BUSY_DRIVE
143 The system cannot join or substitute a drive to or for a directory on the same drive. ERROR_SAME_DRIVE
144 The directory is not a subdirectory of the root directory. ERROR_DIR_NOT_ROOT

145 The directory is not empty. ERROR_DIR_NOT_EMPTY
146 The path specified is being used in a substitute. ERROR_IS_SUBST_PATH
147 Not enough resources are available to process this command. ERROR_IS_JOIN_PATH
148 The path specified cannot be used at this time. ERROR_PATH_BUSY
149 An attempt was made to join or substitute a drive for which a directory on the drive is the target of a previous substitute. ERROR_IS_SUBST_TARGET
150 System trace information was not specified in your CONFIG.SYS file, or tracing is disallowed. ERROR_SYSTEM_TRACE
151 The number of specified semaphore events for DosMuxSemWait is not correct. ERROR_INVALID_EVENT_COUNT
152 DosMuxSemWait did not execute; too many semaphores are already set. ERROR_TOO_MANY_MUXWAITERS
153 The DosMuxSemWait list is not correct. ERROR_INVALID_LIST_FORMAT
154 The volume label you entered exceeds the label character limit of the target file system. ERROR_LABEL_TOO_LONG
155 Cannot create another thread. ERROR_TOO_MANY_TCBS
156 The recipient process has refused the signal. ERROR_SIGNAL_REFUSED
157 The segment is already discarded and cannot be locked. ERROR_DISCARDED
158 The segment is already unlocked. ERROR_NOT_LOCKED
159 The address for the thread ID is not correct. ERROR_BAD_THREADID_ADDR
160 The argument string passed to DosExecPgm is not correct. ERROR_BAD_ARGUMENTS
161 The specified path is invalid. ERROR_BAD_PATHNAME
162 A signal is already pending. ERROR_SIGNAL_PENDING
164 No more threads can be created in the system. ERROR_MAX_THRDS_REACHED
167 Unable to lock a region of a file. ERROR_LOCK_FAILED
170 The requested resource is in use. ERROR_BUSY
173 A lock request was not outstanding for the supplied cancel region. ERROR_CANCEL_VIOLATION
174 The file system does not support atomic changes to the lock type. ERROR_ATOMIC_LOCKS_NOT_SUPPORTED
180 The system detected a segment number that was not correct. ERROR_INVALID_SEGMENT_NUMBER
182 The operating system cannot run %1. ERROR_INVALID_ORDINAL
183 Cannot create a file when that file already exists. ERROR_ALREADY_EXISTS
186 The flag passed is not correct. ERROR_INVALID_FLAG_NUMBER
187 The specified system semaphore name was not found. ERROR_SEM_NOT_FOUND
188 The operating system cannot run %1. ERROR_INVALID_STARTING_CODESEG
189 The operating system cannot run %1. ERROR_INVALID_STACKSEG
190 The operating system cannot run %1. ERROR_INVALID_MODULETYPE
191 Cannot run %1 in Win32 mode. ERROR_INVALID_EXE_SIGNATURE
192 The operating system cannot run %1. ERROR_EXE_MARKED_INVALID
193 is not a valid Win32 application. ERROR_BAD_EXE_FORMAT
194 The operating system cannot run %1. ERROR_ITERATED_DATA_EXCEEDS_64k
195 The operating system cannot run %1. ERROR_INVALID_MINALLOCSIZE
196 The operating system cannot run this application program. ERROR_DYNLINK_FROM_INVALID_RING
197 The operating system is not presently configured to run this application. ERROR_IOPL_NOT_ENABLED
198 The operating system cannot run %1. ERROR_INVALID_SEGDPL
199 The operating system cannot run this application program. ERROR_AUTODATASEG_EXCEEDS_64k
200 The code segment cannot be greater than or equal to 64K. ERROR_RING2SEG_MUST_BE_MOVABLE
201 The operating system cannot run %1. ERROR_RELOC_CHAIN_XEEDS_SEGLIM
202 The operating system cannot run %1. ERROR_INFLOOP_IN_RELOC_CHAIN
203 The system could not find the environment option that was entered. ERROR_ENVVAR_NOT_FOUND
205 No process in the command subtree has a signal handler. ERROR_NO_SIGNAL_SENT
206 The filename or extension is too long. ERROR_FILENAME_EXCED_RANGE
207 The ring 2 stack is in use. ERROR_RING2_STACK_IN_USE

208 The global filename characters, * or ?, are entered incorrectly or too many global filename characters are specified. ERROR_META_EXPANSION_TOO_LONG
209 The signal being posted is not correct. ERROR_INVALID_SIGNAL_NUMBER
210 The signal handler cannot be set. ERROR_THREAD_1_INACTIVE
212 The segment is locked and cannot be reallocated. ERROR_LOCKED
214 Too many dynamic-link modules are attached to this program or dynamic-link module. ERROR_TOO_MANY_MODULES
215 Can't nest calls to LoadModule. ERROR_NESTING_NOT_ALLOWED
216 The image file %1 is valid, but is for a machine type other than the current machine. ERROR_EXE_MACHINE_TYPE_MISMATCH
230 The pipe state is invalid. ERROR_BAD_PIPE
231 All pipe instances are busy. ERROR_PIPE_BUSY
232 The pipe is being closed. ERROR_NO_DATA
233 No process is on the other end of the pipe. ERROR_PIPE_NOT_CONNECTED
234 More data is available. ERROR_MORE_DATA
240 The session was canceled. ERROR_VC_DISCONNECTED
254 The specified extended attribute name was invalid. ERROR_INVALID_EA_NAME
255 The extended attributes are inconsistent. ERROR_EA_LIST_INCONSISTENT
259 No more data is available. ERROR_NO_MORE_ITEMS
266 The copy functions cannot be used. ERROR_CANNOT_COPY
267 The directory name is invalid. ERROR_DIRECTORY
275 The extended attributes did not fit in the buffer. ERROR_EAS_DIDNT_FIT
276 The extended attribute file on the mounted file system is corrupt. ERROR_EA_FILE_CORRUPT
277 The extended attribute table file is full. ERROR_EA_TABLE_FULL
278 The specified extended attribute handle is invalid. ERROR_INVALID_EA_HANDLE
282 The mounted file system does not support extended attributes. ERROR_EAS_NOT_SUPPORTED
288 Attempt to release mutex not owned by caller. ERROR_NOT_OWNER
298 Too many posts were made to a semaphore. ERROR_TOO_MANY_POSTS
299 Only part of a ReadProcessMemoty or WriteProcessMemory request was completed. ERROR_PARTIAL_COPY
300 The oplock request is denied. ERROR_OPLOCK_NOT_GRANTED
301 An invalid oplock acknowledgment was received by the system. ERROR_INVALID_OPLOCK_PROTOCOL
317 The system cannot find message text for message number 0x%1 in the message file for %2. ERROR_MR_MID_NOT_FOUND
487 Attempt to access invalid address. ERROR_INVALID_ADDRESS
534 Arithmetic result exceeded 32 bits. ERROR_ARITHMETIC_OVERFLOW
535 There is a process on other end of the pipe. ERROR_PIPE_CONNECTED
536 Waiting for a process to open the other end of the pipe. ERROR_PIPE_LISTENING
994 Access to the extended attribute was denied. ERROR_EA_ACCESS_DENIED
995 The I/O operation has been aborted because of either a thread exit or an application request. ERROR_OPERATION_ABORTED
996 Overlapped I/O event is not in a signaled state. ERROR_IO_INCOMPLETE
997 Overlapped I/O operation is in progress. ERROR_IO_PENDING
998 Invalid access to memory location. ERROR_NOACCESS
999 Error performing inpage operation. ERROR_SWAPERROR
1001 Recursion too deep; the stack overflowed. ERROR_STACK_OVERFLOW
1002 The window cannot act on the sent message. ERROR_INVALID_MESSAGE
1003 Cannot complete this function. ERROR_CAN_NOT_COMPLETE
1004 Invalid flags. ERROR_INVALID_FLAGS
1005 The volume does not contain a recognized file system. Please make sure that all required file system drivers are loaded and that the volume is not corrupted. ERROR_UNRECOGNIZED_VOLUME
1006 The volume for a file has been externally altered so that the opened file is no longer

valid. ERROR_FILE_INVALID
1007 The requested operation cannot be performed in full-screen mode. ERROR_FULLSCREEN_MODE
1008 An attempt was made to reference a token that does not exist. ERROR_NO_TOKEN
1009 The configuration registry database is corrupt. ERROR_BADDB
1010 The configuration registry key is invalid. ERROR_BADKEY
1011 The configuration registry key could not be opened. ERROR_CANTOPEN
1012 The configuration registry key could not be read. ERROR_CANTREAD
1013 The configuration registry key could not be written. ERROR_CANTWRITE
1014 One of the files in the registry database had to be recovered by use of a log or alternate copy. The recovery was successful. ERROR_REGISTRY_RECOVERED
1015 The registry is corrupted. The structure of one of the files that contains registry data is corrupted, or the system's image of the file in memory is corrupted, or the file could not be recovered because the alternate copy or log was absent or corrupted. ERROR_REGISTRY_CORRUPT
1016 An I/O operation initiated by the registry failed unrecoverably. The registry could not read in, or write out, or flush, one of the files that contain the system's image of the registry. ERROR_REGISTRY_IO_FAILED
1017 The system has attempted to load or restore a file into the registry, but the specified file is not in a registry file format. ERROR_NOT_REGISTRY_FILE
1018 Illegal operation attempted on a registry key that has been marked for deletion. ERROR_KEY_DELETED
1019 System could not allocate the required space in a registry log. ERROR_NO_LOG_SPACE
1020 Cannot create a symbolic link in a registry key that already has subkeys or values. ERROR_KEY_HAS_CHILDREN
1021 Cannot create a stable subkey under a volatile parent key. ERROR_CHILD_MUST_BE_VOLATILE
1022 A notify change request is being completed and the information is not being returned in the caller's buffer. The caller now needs to enumerate the files to find the changes. ERROR_NOTIFY_ENUM_DIR
1051 A stop control has been sent to a service that other running services are dependent on. ERROR_DEPENDENT_SERVICES_RUNNING
1052 The requested control is not valid for this service. ERROR_INVALID_SERVICE_CONTROL
1053 The service did not respond to the start or control request in a timely fashion. ERROR_SERVICE_REQUEST_TIMEOUT
1054 A thread could not be created for the service. ERROR_SERVICE_NO_THREAD
1055 The service database is locked. ERROR_SERVICE_DATABASE_LOCKED
1056 An instance of the service is already running. ERROR_SERVICE_ALREADY_RUNNING
1057 The account name is invalid or does not exist. ERROR_INVALID_SERVICE_ACCOUNT
1058 The service cannot be started, either because it is disabled or because it has no enabled devices associated with it. ERROR_SERVICE_DISABLED
1059 Circular service dependency was specified. ERROR_CIRCULAR_DEPENDENCY
1060 The specified service does not exist as an installed service. ERROR_SERVICE_DOES_NOT_EXIST
1061 The service cannot accept control messages at this time. ERROR_SERVICE_CANNOT_ACCEPT_CTRL
1062 The service has not been started. ERROR_SERVICE_NOT_ACTIVE
1063 The service process could not connect to the service controller. ERROR_FAILED_SERVICE_CONTROLLER_CONNECT
1064 An exception occurred in the service when handling the control request. ERROR_EXCEPTION_IN_SERVICE
1065 The database specified does not exist. ERROR_DATABASE_DOES_NOT_EXIST
1066 The service has returned a service-specific error code. ERROR_SERVICE_SPECIFIC_ERROR
1067 The process terminated unexpectedly. ERROR_PROCESS_ABORTED
1068 The dependency service or group failed to start. ERROR_SERVICE_DEPENDENCY_FAIL
1069 The service did not start due to a logon failure. ERROR_SERVICE_LOGON_FAILED
1070 After starting, the service hung in a start-pending state. ERROR_SERVICE_START_HANG

1071 The specified service database lock is invalid. ERROR_INVALID_SERVICE_LOCK
1072 The specified service has been marked for deletion. ERROR_SERVICE_MARKED_FOR_DELETE
1073 The specified service already exists. ERROR_SERVICE_EXISTS
1074 The system is currently running with the last-known-good configuration. ERROR_ALREADY_RUNNING_LKG
1075 The dependency service does not exist or has been marked for deletion. ERROR_SERVICE_DEPENDENCY_DELETED
1076 The current boot has already been accepted for use as the last-known-good control set. ERROR_BOOT_ALREADY_ACCEPTED
1077 No attempts to start the service have been made since the last boot. ERROR_SERVICE_NEVER_STARTED
1078 The name is already in use as either a service name or a service display name. ERROR_DUPLICATE_SERVICE_NAME
1079 The account specified for this service is different from the account specified for other services running in the same process. ERROR_DIFFERENT_SERVICE_ACCOUNT
1080 Failure actions can only be set for Win32 services, not for drivers. ERROR_CANNOT_DETECT_DRIVER_FAILURE
1081 This service runs in the same process as the service control manager. Therefore, the service control manager cannot take action if this service's process terminates unexpectedly. ERROR_CANNOT_DETECT_PROCESS_ABORT
1082 No recovery program has been configured for this service. ERROR_NO_RECOVERY_PROGRAM
1100 The physical end of the tape has been reached. ERROR_END_OF_MEDIA
1101 A tape access reached a filemark. ERROR_FILEMARK_DETECTED
1102 The beginning of the tape or a partition was encountered. ERROR_BEGINNING_OF_MEDIA
1103 A tape access reached the end of a set of files. ERROR_SETMARK_DETECTED
1104 No more data is on the tape. ERROR_NO_DATA_DETECTED
1105 Tape could not be partitioned. ERROR_PARTITION_FAILURE
1106 When accessing a new tape of a multivolume partition, the current blocksize is incorrect. ERROR_INVALID_BLOCK_LENGTH
1107 Tape partition information could not be found when loading a tape. ERROR_DEVICE_NOT_PARTITIONED
1108 Unable to lock the media eject mechanism. ERROR_UNABLE_TO_LOCK_MEDIA
1109 Unable to unload the media. ERROR_UNABLE_TO_UNLOAD_MEDIA
1110 The media in the drive may have changed. ERROR_MEDIA_CHANGED
1111 The I/O bus was reset. ERROR_BUS_RESET
1112 No media in drive. ERROR_NO_MEDIA_IN_DRIVE
1113 No mapping for the Unicode character exists in the target multi-byte code page. ERROR_NO_UNICODE_TRANSLATION
1114 A dynamic link library (DLL) initialization routine failed. ERROR_DLL_INIT_FAILED
1115 A system shutdown is in progress. ERROR_SHUTDOWN_IN_PROGRESS
1116 Unable to abort the system shutdown because no shutdown was in progress. ERROR_NO_SHUTDOWN_IN_PROGRESS
1117 The request could not be performed because of an I/O device error. ERROR_IO_DEVICE
1118 No serial device was successfully initialized. The serial driver will unload. ERROR_SERIAL_NO_DEVICE
1119 Unable to open a device that was sharing an interrupt request (IRQ) with other devices. At least one other device that uses that IRQ was already opened. ERROR_IRQ_BUSY
1120 A serial I/O operation was completed by another write to the serial port. The IOCTL_SERIAL_XOFF_COUNTER reached zero.) ERROR_MORE_WRITES
1121 A serial I/O operation completed because the timeout period expired. The IOCTL_SERIAL_XOFF_COUNTER did not reach zero.) ERROR_COUNTER_TIMEOUT
1122 No ID address mark was found on the floppy disk. ERROR_FLOPPY_ID_MARK_NOT_FOUND
1123 Mismatch between the floppy disk sector ID field and the floppy disk controller track address. ERROR_FLOPPY_WRONG_CYLINDER

1124 The floppy disk controller reported an error that is not recognized by the floppy disk driver. ERROR_FLOPPY_UNKNOWN_ERROR
1125 The floppy disk controller returned inconsistent results in its registers. ERROR_FLOPPY_BAD_REGISTERS
1126 While accessing the hard disk, a recalibrate operation failed, even after retries. ERROR_DISK_RECALIBRATE_FAILED
1127 While accessing the hard disk, a disk operation failed even after retries. ERROR_DISK_OPERATION_FAILED
1128 While accessing the hard disk, a disk controller reset was needed, but even that failed. ERROR_DISK_RESET_FAILED
1129 Physical end of tape encountered. ERROR_EOM_OVERFLOW
1130 Not enough server storage is available to process this command. ERROR_NOT_ENOUGH_SERVER_MEMORY
1131 A potential deadlock condition has been detected. ERROR_POSSIBLE_DEADLOCK
1132 The base address or the file offset specified does not have the proper alignment. ERROR_MAPPED_ALIGNMENT
1140 An attempt to change the system power state was vetoed by another application or driver. ERROR_SET_POWER_STATE_VETOED
1141 The system BIOS failed an attempt to change the system power state. ERROR_SET_POWER_STATE_FAILED
1142 An attempt was made to create more links on a file than the file system supports. ERROR_TOO_MANY_LINKS
1150 The specified program requires a newer version of Windows. ERROR_OLD_WIN_VERSION
1151 The specified program is not a Windows or MS-DOS program. ERROR_APP_WRONG_OS
1152 Cannot start more than one instance of the specified program. ERROR_SINGLE_INSTANCE_APP
1153 The specified program was written for an earlier version of Windows. ERROR_RMODE_APP
1154 One of the library files needed to run this application is damaged. ERROR_INVALID_DLL
1155 No application is associated with the specified file for this operation. ERROR_NO_ASSOCIATION
1156 An error occurred in sending the command to the application. ERROR_DDE_FAIL
1157 One of the library files needed to run this application cannot be found. ERROR_DLL_NOT_FOUND
1158 The current process has used all of its system allowance of handles for Window Manager objects. ERROR_NO_MORE_USER_HANDLES
1159 The message can be used only with synchronous operations. ERROR_MESSAGE_SYNC_ONLY
1160 The indicated source element has no media. ERROR_SOURCE_ELEMENT_EMPTY
1161 The indicated destination element already contains media. ERROR_DESTINATION_ELEMENT_FULL
1162 The indicated element does not exist. ERROR_ILLEGAL_ELEMENT_ADDRESS
1163 The indicated element is part of a magazine that is not present. ERROR_MAGAZINE_NOT_PRESENT
1164 The indicated device requires reinitialization due to hardware errors. ERROR_DEVICE_REINITIALIZATION_NEEDED
1165 The device has indicated that cleaning is required before further operations are attempted. ERROR_DEVICE_REQUIRES_CLEANING
1166 The device has indicated that its door is open. ERROR_DEVICE_DOOR_OPEN
1167 The device is not connected. ERROR_DEVICE_NOT_CONNECTED
1168 Element not found. ERROR_NOT_FOUND
1169 There was no match for the specified key in the index. ERROR_NO_MATCH
1170 The property set specified does not exist on the object. ERROR_SET_NOT_FOUND
1171 The point passed to GetMouseMovePoints is not in the buffer. ERROR_POINT_NOT_FOUND
1172 The tracking (workstation) service is not running. ERROR_NO_TRACKING_SERVICE
1173 The Volume ID could not be found. ERROR_NO_VOLUME_ID
1200 The specified device name is invalid. ERROR_BAD_DEVICE
1201 The device is not currently connected but it is a remembered connection.

ERROR_CONNECTION_UNAVAIL
1202 An attempt was made to remember a device that had previously been remembered. ERROR_DEVICE_ALREADY_REMEMBERED
1203 No network provider accepted the given network path. ERROR_NO_NET_OR_BAD_PATH
1204 The specified network provider name is invalid. ERROR_BAD_PROVIDER
1205 Unable to open the network connection profile. ERROR_CANNOT_OPEN_PROFILE
1206 The network connection profile is corrupted. ERROR_BAD_PROFILE
1207 Cannot enumerate a noncontainer. ERROR_NOT_CONTAINER
1208 An extended error has occurred. ERROR_EXTENDED_ERROR
1209 The format of the specified group name is invalid. ERROR_INVALID_GROUPNAME
1210 The format of the specified computer name is invalid. ERROR_INVALID_COMPUTERNAME
1211 The format of the specified event name is invalid. ERROR_INVALID_EVENTNAME
1212 The format of the specified domain name is invalid. ERROR_INVALID_DOMAINNAME
1213 The format of the specified service name is invalid. ERROR_INVALID_SERVICENAME
1214 The format of the specified network name is invalid. ERROR_INVALID_NETNAME
1215 The format of the specified share name is invalid. ERROR_INVALID_SHARENAME
1216 The format of the specified password is invalid. ERROR_INVALID_PASSWORDNAME
1217 The format of the specified message name is invalid. ERROR_INVALID_MESSAGENAME
1218 The format of the specified message destination is invalid. ERROR_INVALID_MESSAGEDEST
1219 The credentials supplied conflict with an existing set of credentials. ERROR_SESSION_CREDENTIAL_CONFLICT
1220 An attempt was made to establish a session to a network server, but there are already too many sessions established to that server. ERROR_REMOTE_SESSION_LIMIT_EXCEEDED
1221 The workgroup or domain name is already in use by another computer on the network. ERROR_DUP_DOMAINNAME
1222 The network is not present or not started. ERROR_NO_NETWORK
1223 The operation was canceled by the user. ERROR_CANCELLED
1224 The requested operation cannot be performed on a file with a user-mapped section open. ERROR_USER_MAPPED_FILE
1225 The remote system refused the network connection. ERROR_CONNECTION_REFUSED
1226 The network connection was gracefully closed. ERROR_GRACEFUL_DISCONNECT
1227 The network transport endpoint already has an address associated with it. ERROR_ADDRESS_ALREADY_ASSOCIATED
1228 An address has not yet been associated with the network endpoint. ERROR_ADDRESS_NOT_ASSOCIATED
1229 An operation was attempted on a nonexistent network connection. ERROR_CONNECTION_INVALID
1230 An invalid operation was attempted on an active network connection. ERROR_CONNECTION_ACTIVE
1231 The remote network is not reachable by the transport. ERROR_NETWORK_UNREACHABLE
1232 The remote system is not reachable by the transport. ERROR_HOST_UNREACHABLE
1233 The remote system does not support the transport protocol. ERROR_PROTOCOL_UNREACHABLE
1234 No service is operating at the destination network endpoint on the remote system. ERROR_PORT_UNREACHABLE
1235 The request was aborted. ERROR_REQUEST_ABORTED
1236 The network connection was aborted by the local system. ERROR_CONNECTION_ABORTED
1237 The operation could not be completed. A retry should be performed. ERROR_RETRY
1238 A connection to the server could not be made because the limit on the number of concurrent connections for this account has been reached. ERROR_CONNECTION_COUNT_LIMIT
1239 Attempting to log in during an unauthorized time of day for this account. ERROR_LOGIN_TIME_RESTRICTION
1240 The account is not authorized to log in from this station. ERROR_LOGIN_WKSTA_RESTRICTION
1241 The network address could not be used for the operation requested. ERROR_INCORRECT_ADDRESS
1242 The service is already registered. ERROR_ALREADY_REGISTERED

1243 The specified service does not exist. ERROR_SERVICE_NOT_FOUND
1244 The operation being requested was not performed because the user has not been authenticated. ERROR_NOT_AUTHENTICATED
1245 The operation being requested was not performed because the user has not logged on to the network. The specified service does not exist. ERROR_NOT_LOGGED_ON
1246 Continue with work in progress. ERROR_CONTINUE
1247 An attempt was made to perform an initialization operation when initialization has already been completed. ERROR_ALREADY_INITIALIZED
1248 No more local devices. ERROR_NO_MORE_DEVICES
1249 The specified site does not exist. ERROR_NO_SUCH_SITE
1250 A domain controller with the specified name already exists. ERROR_DOMAIN_CONTROLLER_EXISTS
1251 An error occurred while installing the Windows NT directory service. Please view the event log for more information. ERROR_DS_NOT_INSTALLED
1300 Not all privileges referenced are assigned to the caller. ERROR_NOT_ALL_ASSIGNED
1301 Some mapping between account names and security IDs was not done. ERROR_SOME_NOT_MAPPED
1302 No system quota limits are specifically set for this account. ERROR_NO_QUOTAS_FOR_ACCOUNT
1303 No encryption key is available. A well-known encryption key was returned. ERROR_LOCAL_USER_SESSION_KEY
1304 The Windows NT password is too complex to be converted to a LAN Manager password. The LAN Manager password returned is a NULL string. ERROR_NULL_LM_PASSWORD
1305 The revision level is unknown. ERROR_UNKNOWN_REVISION
1306 Indicates two revision levels are incompatible. ERROR_REVISION_MISMATCH
1307 This security ID may not be assigned as the owner of this object. ERROR_INVALID_OWNER
1308 This security ID may not be assigned as the primary group of an object. ERROR_INVALID_PRIMARY_GROUP
1309 An attempt has been made to operate on an impersonation token by a thread that is not currently impersonating a client. ERROR_NO_IMPERSONATION_TOKEN
1310 The group may not be disabled. ERROR_CANT_DISABLE_MANDATORY
1311 There are currently no logon servers available to service the logon request. ERROR_NO_LOGON_SERVERS
1312 A specified logon session does not exist. It may already have been terminated. ERROR_NO_SUCH_LOGON_SESSION
1313 A specified privilege does not exist. ERROR_NO_SUCH_PRIVILEGE
1314 A required privilege is not held by the client. ERROR_PRIVILEGE_NOT_HELD
1315 The name provided is not a properly formed account name. ERROR_INVALID_ACCOUNT_NAME
1316 The specified user already exists. ERROR_USER_EXISTS
1317 The specified user does not exist. ERROR_NO_SUCH_USER
1318 The specified group already exists. ERROR_GROUP_EXISTS
1319 The specified group does not exist. ERROR_NO_SUCH_GROUP
1320 Either the specified user account is already a member of the specified group, or the specified group cannot be deleted because it contains a member. ERROR_MEMBER_IN_GROUP
1321 The specified user account is not a member of the specified group account. ERROR_MEMBER_NOT_IN_GROUP
1322 The last remaining administration account cannot be disabled or deleted. ERROR_LAST_ADMIN
1323 Unable to update the password. The value provided as the current password is incorrect. ERROR_WRONG_PASSWORD
1324 Unable to update the password. The value provided for the new password contains values that are not allowed in passwords. ERROR_ILL_FORMED_PASSWORD
1325 Unable to update the password because a password update rule has been violated. ERROR_PASSWORD_RESTRICTION
1326 Logon failure: unknown user name or bad password. ERROR_LOGON_FAILURE
1327 Logon failure: user account restriction. ERROR_ACCOUNT_RESTRICTION
1328 Logon failure: account logon time restriction violation. ERROR_INVALID_LOGON_HOURS

1329 Logon failure: user not allowed to log on to this computer. ERROR_INVALID_WORKSTATION
1330 Logon failure: the specified account password has expired. ERROR_PASSWORD_EXPIRED
1331 Logon failure: account currently disabled. ERROR_ACCOUNT_DISABLED
1332 No mapping between account names and security IDs was done. ERROR_NONE_MAPPED
1333 Too many local user identifiers (LUIDs) were requested at one time. ERROR_TOO_MANY_LUIDS_REQUESTED
1334 No more local user identifiers (LUIDs) are available. ERROR_LUIDS_EXHAUSTED
1335 The subauthority part of a security ID is invalid for this particular use. ERROR_INVALID_SUB_AUTHORITY
1336 The access control list (ACL) structure is invalid. ERROR_INVALID_ACL
1337 The security ID structure is invalid. ERROR_INVALID_SID
1338 The security descriptor structure is invalid. ERROR_INVALID_SECURITY_DESCR
1340 The inherited access control list (ACL) or access control entry (ACE) could not be built. ERROR_BAD_INHERITANCE_ACL
1341 The server is currently disabled. ERROR_SERVER_DISABLED
1342 The server is currently enabled. ERROR_SERVER_NOT_DISABLED
1343 The value provided was an invalid value for an identifier authority. ERROR_INVALID_ID_AUTHORITY
1344 No more memory is available for security information updates. ERROR_ALLOTTED_SPACE_EXCEEDED
1345 The specified attributes are invalid, or incompatible with the attributes for the group as a whole. ERROR_INVALID_GROUP_ATTRIBUTES
1346 Either a required impersonation level was not provided, or the provided impersonation level is invalid. ERROR_BAD_IMPERSONATION_LEVEL
1347 Cannot open an anonymous level security token. ERROR_CANT_OPEN_ANONYMOUS
1348 The validation information class requested was invalid. ERROR_BAD_VALIDATION_CLASS
1349 The type of the token is inappropriate for its attempted use. ERROR_BAD_TOKEN_TYPE
1350 Unable to perform a security operation on an object that has no associated security. ERROR_NO_SECURITY_ON_OBJECT
1351 Indicates a Windows NT Server could not be contacted or that objects within the domain are protected such that necessary information could not be retrieved. ERROR_CANT_ACCESS_DOMAIN_INFO
1352 The security account manager (SAM) or local security authority (LSA) server was in the wrong state to perform the security operation. ERROR_INVALID_SERVER_STATE
1353 The domain was in the wrong state to perform the security operation. ERROR_INVALID_DOMAIN_STATE
1354 This operation is only allowed for the Primary Domain Controller of the domain. ERROR_INVALID_DOMAIN_ROLE
1355 The specified domain did not exist. ERROR_NO_SUCH_DOMAIN
1356 The specified domain already exists. ERROR_DOMAIN_EXISTS
1357 An attempt was made to exceed the limit on the number of domains per server. ERROR_DOMAIN_LIMIT_EXCEEDED
1358 Unable to complete the requested operation because of either a catastrophic media failure or a data structure corruption on the disk. ERROR_INTERNAL_DB_CORRUPTION
1359 The security account database contains an internal inconsistency. ERROR_INTERNAL_ERROR
1360 Generic access types were contained in an access mask which should already be mapped to nongeneric types. ERROR_GENERIC_NOT_MAPPED
1361 A security descriptor is not in the right format (absolute or self-relative). ERROR_BAD_DESCRIPTOR_FORMAT
1362 The requested action is restricted for use by logon processes only. The calling process has not registered as a logon process. ERROR_NOT_LOGON_PROCESS
1363 Cannot start a new logon session with an ID that is already in use. ERROR_LOGON_SESSION_EXISTS
1364 A specified authentication package is unknown. ERROR_NO_SUCH_PACKAGE

1365 The logon session is not in a state that is consistent with the requested operation. ERROR_BAD_LOGON_SESSION_STATE
1366 The logon session ID is already in use. ERROR_LOGON_SESSION_COLLISION
1367 A logon request contained an invalid logon type value. ERROR_INVALID_LOGON_TYPE
1368 Unable to impersonate using a named pipe until data has been read from that pipe. ERROR_CANNOT_IMPERSONATE
1369 The transaction state of a registry subtree is incompatible with the requested operation. ERROR_RXACT_INVALID_STATE
1370 An internal security database corruption has been encountered. ERROR_RXACT_COMMIT_FAILURE
1371 Cannot perform this operation on built-in accounts. ERROR_SPECIAL_ACCOUNT
1372 Cannot perform this operation on this built-in special group. ERROR_SPECIAL_GROUP
1373 Cannot perform this operation on this built-in special user. ERROR_SPECIAL_USER
1374 The user cannot be removed from a group because the group is currently the user's primary group. ERROR_MEMBERS_PRIMARY_GROUP
1375 The token is already in use as a primary token. ERROR_TOKEN_ALREADY_IN_USE
1376 The specified local group does not exist. ERROR_NO_SUCH_ALIAS
1377 The specified account name is not a member of the local group. ERROR_MEMBER_NOT_IN_ALIAS
1378 The specified account name is already a member of the local group. ERROR_MEMBER_IN_ALIAS
1379 The specified local group already exists. ERROR_ALIAS_EXISTS
1380 Logon failure: the user has not been granted the requested logon type at this computer. ERROR_LOGON_NOT_GRANTED
1381 The maximum number of secrets that may be stored in a single system has been exceeded. ERROR_TOO_MANY_SECRETS
1382 The length of a secret exceeds the maximum length allowed. ERROR_SECRET_TOO_LONG
1383 The local security authority database contains an internal inconsistency. ERROR_INTERNAL_DB_ERROR
1384 During a logon attempt, the user's security context accumulated too many security IDs. ERROR_TOO_MANY_CONTEXT_IDS
1385 Logon failure: the user has not been granted the requested logon type at this computer. ERROR_LOGON_TYPE_NOT_GRANTED
1386 A cross-encrypted password is necessary to change a user password. ERROR_NT_CROSS_ENCRYPTION_REQUIRED
1387 A new member could not be added to a local group because the member does not exist. ERROR_NO_SUCH_MEMBER
1388 A new member could not be added to a local group because the member has the wrong account type. ERROR_INVALID_MEMBER
1389 Too many security IDs have been specified. ERROR_TOO_MANY_SIDS
1390 A cross-encrypted password is necessary to change this user password. ERROR_LM_CROSS_ENCRYPTION_REQUIRED
1391 Indicates an ACL contains no inheritable components. ERROR_NO_INHERITANCE
1392 The file or directory is corrupted and unreadable. ERROR_FILE_CORRUPT
1393 The disk structure is corrupted and unreadable. ERROR_DISK_CORRUPT
1394 There is no user session key for the specified logon session. ERROR_NO_USER_SESSION_KEY
1395 The service being accessed is licensed for a particular number of connections. No more connections can be made to the service at this time because there are already as many connections as the service can accept. ERROR_LICENSE_QUOTA_EXCEEDED
1400 Invalid window handle. ERROR_INVALID_WINDOW_HANDLE
1401 Invalid menu handle. ERROR_INVALID_MENU_HANDLE
1402 Invalid cursor handle. ERROR_INVALID_CURSOR_HANDLE
1403 Invalid accelerator table handle. ERROR_INVALID_ACCEL_HANDLE
1404 Invalid hook handle. ERROR_INVALID_HOOK_HANDLE
1405 Invalid handle to a multiple-window position structure. ERROR_INVALID_DWP_HANDLE
1406 Cannot create a top-level child window. ERROR_TLW_WITH_WSCHILD
1407 Cannot find window class. ERROR_CANNOT_FIND_WND_CLASS

1408 Invalid window; it belongs to other thread. ERROR_WINDOW_OF_OTHER_THREAD
1409 Hot key is already registered. ERROR_HOTKEY_ALREADY_REGISTERED
1410 Class already exists. ERROR_CLASS_ALREADY_EXISTS
1411 Class does not exist. ERROR_CLASS_DOES_NOT_EXIST
1412 Class still has open windows. ERROR_CLASS_HAS_WINDOWS
1413 Invalid index. ERROR_INVALID_INDEX
1414 Invalid icon handle. ERROR_INVALID_ICON_HANDLE
1415 Using private DIALOG window words. ERROR_PRIVATE_DIALOG_INDEX
1416 The list box identifier was not found. ERROR_LISTBOX_ID_NOT_FOUND
1417 No wildcards were found. ERROR_NO_WILDCARD_CHARACTERS
1418 Thread does not have a clipboard open. ERROR_CLIPBOARD_NOT_OPEN
1419 Hot key is not registered. ERROR_HOTKEY_NOT_REGISTERED
1420 The window is not a valid dialog window. ERROR_WINDOW_NOT_DIALOG
1421 Control ID not found. ERROR_CONTROL_ID_NOT_FOUND
1422 Invalid message for a combo box because it does not have an edit control. ERROR_INVALID_COMBOBOX_MESSAGE
1423 The window is not a combo box. ERROR_WINDOW_NOT_COMBOBOX
1424 Height must be less than 256. ERROR_INVALID_EDIT_HEIGHT
1425 Invalid device context (DC) handle. ERROR_DC_NOT_FOUND
1426 Invalid hook procedure type. ERROR_INVALID_HOOK_FILTER
1427 Invalid hook procedure. ERROR_INVALID_FILTER_PROC
1428 Cannot set nonlocal hook without a module handle. ERROR_HOOK_NEEDS_HMOD
1429 This hook procedure can only be set globally. ERROR_GLOBAL_ONLY_HOOK
1430 The journal hook procedure is already installed. ERROR_JOURNAL_HOOK_SET
1431 The hook procedure is not installed. ERROR_HOOK_NOT_INSTALLED
1432 Invalid message for single-selection list box. ERROR_INVALID_LB_MESSAGE
1433 LB_SETCOUNT sent to non-lazy list box. ERROR_SETCOUNT_ON_BAD_LB
1434 This list box does not support tab stops. ERROR_LB_WITHOUT_TABSTOPS
1435 Cannot destroy object created by another thread. ERROR_DESTROY_OBJECT_OF_OTHER_THREAD
1436 Child windows cannot have menus. ERROR_CHILD_WINDOW_MENU
1437 The window does not have a system menu. ERROR_NO_SYSTEM_MENU
1438 Invalid message box style. ERROR_INVALID_MSGBOX_STYLE
1439 Invalid system-wide (SPI_*) parameter. ERROR_INVALID_SPI_VALUE
1440 Screen already locked. ERROR_SCREEN_ALREADY_LOCKED
1441 All handles to windows in a multiple-window position structure must have the same parent. ERROR_HWNDS_HAVE_DIFF_PARENT
1442 The window is not a child window. ERROR_NOT_CHILD_WINDOW
1443 Invalid GW_* command. ERROR_INVALID_GW_COMMAND
1444 Invalid thread identifier. ERROR_INVALID_THREAD_ID
1445 Cannot process a message from a window that is not a multiple document interface (MDI) window. ERROR_NON_MDICHILD_WINDOW
1446 Popup menu already active. ERROR_POPUP_ALREADY_ACTIVE
1447 The window does not have scroll bars. ERROR_NO_SCROLLBARS
1448 Scroll bar range cannot be greater than 0x7FFF. ERROR_INVALID_SCROLLBAR_RANGE
1449 Cannot show or remove the window in the way specified. ERROR_INVALID_SHOWWIN_COMMAND
1450 Insufficient system resources exist to complete the requested service. ERROR_NO_SYSTEM_RESOURCES
1451 Insufficient system resources exist to complete the requested service. ERROR_NONPAGED_SYSTEM_RESOURCES
1452 Insufficient system resources exist to complete the requested service. ERROR_PAGED_SYSTEM_RESOURCES
1453 Insufficient quota to complete the requested service. ERROR_WORKING_SET_QUOTA
1454 Insufficient quota to complete the requested service. ERROR_PAGEFILE_QUOTA
1455 The paging file is too small for this operation to complete. ERROR_COMMITMENT_LIMIT

1456 A menu item was not found. ERROR_MENU_ITEM_NOT_FOUND
1457 Invalid keyboard layout handle. ERROR_INVALID_KEYBOARD_HANDLE
1458 Hook type not allowed. ERROR_HOOK_TYPE_NOT_ALLOWED
1459 This operation requires an interactive window station. ERROR_REQUIRES_INTERACTIVE_WINDOWSTATION
1460 This operation returned because the timeout period expired. ERROR_TIMEOUT
1461 Invalid monitor handle. ERROR_INVALID_MONITOR_HANDLE
1500 The event log file is corrupted. ERROR_EVENTLOG_FILE_CORRUPT
1501 No event log file could be opened, so the event logging service did not start. ERROR_EVENTLOG_CANT_START
1502 The event log file is full. ERROR_LOG_FILE_FULL
1503 The event log file has changed between read operations. ERROR_EVENTLOG_FILE_CHANGED
1601 Failure accessing install service. ERROR_INSTALL_SERVICE
1602 The user canceled the installation. ERROR_INSTALL_USEREXIT
1603 Fatal error during installation. ERROR_INSTALL_FAILURE
1604 Installation suspended, incomplete. ERROR_INSTALL_SUSPEND
1605 Product code not registered. ERROR_UNKNOWN_PRODUCT
1606 Feature ID not registered. ERROR_UNKNOWN_FEATURE
1607 Component ID not registered. ERROR_UNKNOWN_COMPONENT
1608 Unknown property. ERROR_UNKNOWN_PROPERTY
1609 Handle is in an invalid state. ERROR_INVALID_HANDLE_STATE
1610 Configuration data corrupt. ERROR_BAD_CONFIGURATION
1611 Language not available. ERROR_INDEX_ABSENT
1612 Install source unavailable. ERROR_INSTALL_SOURCE_ABSENT
1613 Database version unsupported. ERROR_BAD_DATABASE_VERSION
1614 Product is uninstalled. ERROR_PRODUCT_UNINSTALLED
1615 SQL query syntax invalid or unsupported. ERROR_BAD_QUERY_SYNTAX
1616 Record field does not exist. ERROR_INVALID_FIELD
1700 The string binding is invalid. RPC_S_INVALID_STRING_BINDING
1701 The binding handle is not the correct type. RPC_S_WRONG_KIND_OF_BINDING
1702 The binding handle is invalid. RPC_S_INVALID_BINDING
1703 The RPC protocol sequence is not supported. RPC_S_PROTSEQ_NOT_SUPPORTED
1704 The RPC protocol sequence is invalid. RPC_S_INVALID_RPC_PROTSEQ
1705 The string universal unique identifier (UUID) is invalid. RPC_S_INVALID_STRING_UUID
1706 The endpoint format is invalid. RPC_S_INVALID_ENDPOINT_FORMAT
1707 The network address is invalid. RPC_S_INVALID_NET_ADDR
1708 No endpoint was found. RPC_S_NO_ENDPOINT_FOUND
1709 The timeout value is invalid. RPC_S_INVALID_TIMEOUT
1710 The object universal unique identifier (UUID) was not found. RPC_S_OBJECT_NOT_FOUND
1711 The object universal unique identifier (UUID) has already been registered. RPC_S_ALREADY_REGISTERED
1712 The type universal unique identifier (UUID) has already been registered. RPC_S_TYPE_ALREADY_REGISTERED
1713 The RPC server is already listening. RPC_S_ALREADY_LISTENING
1714 No protocol sequences have been registered. RPC_S_NO_PROTSEQS_REGISTERED
1715 The RPC server is not listening. RPC_S_NOT_LISTENING
1716 The manager type is unknown. RPC_S_UNKNOWN_MGR_TYPE
1717 The interface is unknown. RPC_S_UNKNOWN_IF
1718 There are no bindings. RPC_S_NO_BINDINGS
1719 There are no protocol sequences. RPC_S_NO_PROTSEQS
1720 The endpoint cannot be created. RPC_S_CANT_CREATE_ENDPOINT
1721 Not enough resources are available to complete this operation. RPC_S_OUT_OF_RESOURCES
1722 The RPC server is unavailable. RPC_S_SERVER_UNAVAILABLE
1723 The RPC server is too busy to complete this operation. RPC_S_SERVER_TOO_BUSY

1724 The network options are invalid. RPC_S_INVALID_NETWORK_OPTIONS
1725 There are no remote procedure calls active on this thread. RPC_S_NO_CALL_ACTIVE
1726 The remote procedure call failed. RPC_S_CALL_FAILED
1727 The remote procedure call failed and did not execute. RPC_S_CALL_FAILED_DNE
1728 A remote procedure call (RPC) protocol error occurred. RPC_S_PROTOCOL_ERROR
1730 The transfer syntax is not supported by the RPC server. RPC_S_UNSUPPORTED_TRANS_SYN
1732 The universal unique identifier (UUID) type is not supported. RPC_S_UNSUPPORTED_TYPE
1733 The tag is invalid. RPC_S_INVALID_TAG
1734 The array bounds are invalid. RPC_S_INVALID_BOUND
1735 The binding does not contain an entry name. RPC_S_NO_ENTRY_NAME
1736 The name syntax is invalid. RPC_S_INVALID_NAME_SYNTAX
1737 The name syntax is not supported. RPC_S_UNSUPPORTED_NAME_SYNTAX
1739 No network address is available to use to construct a universal unique identifier (UUID). RPC_S_UUID_NO_ADDRESS
1740 The endpoint is a duplicate. RPC_S_DUPLICATE_ENDPOINT
1741 The authentication type is unknown. RPC_S_UNKNOWN_AUTHN_TYPE
1742 The maximum number of calls is too small. RPC_S_MAX_CALLS_TOO_SMALL
1743 The string is too long. RPC_S_STRING_TOO_LONG
1744 The RPC protocol sequence was not found. RPC_S_PROTSEQ_NOT_FOUND
1745 The procedure number is out of range. RPC_S_PROCNUM_OUT_OF_RANGE
1746 The binding does not contain any authentication information. RPC_S_BINDING_HAS_NO_AUTH
1747 The authentication service is unknown. RPC_S_UNKNOWN_AUTHN_SERVICE
1748 The authentication level is unknown. RPC_S_UNKNOWN_AUTHN_LEVEL
1749 The security context is invalid. RPC_S_INVALID_AUTH_IDENTITY
1750 The authorization service is unknown. RPC_S_UNKNOWN_AUTHZ_SERVICE
1751 The entry is invalid. EPT_S_INVALID_ENTRY
1752 The server endpoint cannot perform the operation. EPT_S_CANT_PERFORM_OP
1753 There are no more endpoints available from the endpoint mapper. EPT_S_NOT_REGISTERED
1754 No interfaces have been exported. RPC_S_NOTHING_TO_EXPORT
1755 The entry name is incomplete. RPC_S_INCOMPLETE_NAME
1756 The version option is invalid. RPC_S_INVALID_VERS_OPTION
1757 There are no more members. RPC_S_NO_MORE_MEMBERS
1758 There is nothing to unexport. RPC_S_NOT_ALL_OBJS_UNEXPORTED
1759 The interface was not found. RPC_S_INTERFACE_NOT_FOUND
1760 The entry already exists. RPC_S_ENTRY_ALREADY_EXISTS
1761 The entry is not found. RPC_S_ENTRY_NOT_FOUND
1762 The name service is unavailable. RPC_S_NAME_SERVICE_UNAVAILABLE
1763 The network address family is invalid. RPC_S_INVALID_NAF_ID
1764 The requested operation is not supported. RPC_S_CANNOT_SUPPORT
1765 No security context is available to allow impersonation. RPC_S_NO_CONTEXT_AVAILABLE
1766 An internal error occurred in a remote procedure call (RPC). RPC_S_INTERNAL_ERROR
1767 The RPC server attempted an integer division by zero. RPC_S_ZERO_DIVIDE
1768 An addressing error occurred in the RPC server. RPC_S_ADDRESS_ERROR
1769 A floating-point operation at the RPC server caused a division by zero. RPC_S_FP_DIV_ZERO
1770 A floating-point underflow occurred at the RPC server. RPC_S_FP_UNDERFLOW
1771 A floating-point overflow occurred at the RPC server. RPC_S_FP_OVERFLOW
1772 The list of RPC servers available for the binding of auto handles has been exhausted. RPC_X_NO_MORE_ENTRIES
1773 Unable to open the character translation table file. RPC_X_SS_CHAR_TRANS_OPEN_FAIL
1774 The file containing the character translation table has fewer than bytes. RPC_X_SS_CHAR_TRANS_SHORT_FILE
1775 A null context handle was passed from the client to the host during a remote procedure call. RPC_X_SS_IN_NULL_CONTEXT
1777 The context handle changed during a remote procedure call. RPC_X_SS_CONTEXT_DAMAGED

1778 The binding handles passed to a remote procedure call do not match. RPC_X_SS_HANDLES_MISMATCH
1779 The stub is unable to get the remote procedure call handle. RPC_X_SS_CANNOT_GET_CALL_HANDLE
1780 A null reference pointer was passed to the stub. RPC_X_NULL_REF_POINTER
1781 The enumeration value is out of range. RPC_X_ENUM_VALUE_OUT_OF_RANGE
1782 The byte count is too small. RPC_X_BYTE_COUNT_TOO_SMALL
1783 The stub received bad data. RPC_X_BAD_STUB_DATA
1784 The supplied user buffer is not valid for the requested operation. ERROR_INVALID_USER_BUFFER
1785 The disk media is not recognized. It may not be formatted. ERROR_UNRECOGNIZED_MEDIA
1786 The workstation does not have a trust secret. ERROR_NO_TRUST_LSA_SECRET
1787 The SAM database on the Windows NT Server does not have a computer account for this workstation trust relationship. ERROR_NO_TRUST_SAM_ACCOUNT
1788 The trust relationship between the primary domain and the trusted domain failed. ERROR_TRUSTED_DOMAIN_FAILURE
1789 The trust relationship between this workstation and the primary domain failed. ERROR_TRUSTED_RELATIONSHIP_FAILURE
1790 The network logon failed. ERROR_TRUST_FAILURE
1791 A remote procedure call is already in progress for this thread. RPC_S_CALL_IN_PROGRESS
1792 An attempt was made to logon, but the network logon service was not started. ERROR_NETLOGON_NOT_STARTED
1793 The user's account has expired. ERROR_ACCOUNT_EXPIRED
1794 The redirector is in use and cannot be unloaded. ERROR_REDIRECTOR_HAS_OPEN_HANDLES
1795 The specified printer driver is already installed. ERROR_PRINTER_DRIVER_ALREADY_INSTALLED
1796 The specified port is unknown. ERROR_UNKNOWN_PORT
1797 The printer driver is unknown. ERROR_UNKNOWN_PRINTER_DRIVER
1798 The print processor is unknown. ERROR_UNKNOWN_PRINTPROCESSOR
1799 The specified separator file is invalid. ERROR_INVALID_SEPARATOR_FILE
1800 The specified priority is invalid. ERROR_INVALID_PRIORITY
1801 The printer name is invalid. ERROR_INVALID_PRINTER_NAME
1802 The printer already exists. ERROR_PRINTER_ALREADY_EXISTS
1803 The printer command is invalid. ERROR_INVALID_PRINTER_COMMAND
1804 The specified datatype is invalid. ERROR_INVALID_DATATYPE
1805 The environment specified is invalid. ERROR_INVALID_ENVIRONMENT
1806 There are no more bindings. RPC_S_NO_MORE_BINDINGS
1807 The account used is an interdomain trust account. Use your global user account or local user account to access this server. ERROR_NOLOGON_INTERDOMAIN_TRUST_ACCOUNT
1808 The account used is a computer account. Use your global user account or local user account to access this server. ERROR_NOLOGON_WORKSTATION_TRUST_ACCOUNT
1809 The account used is a server trust account. Use your global user account or local user account to access this server. ERROR_NOLOGON_SERVER_TRUST_ACCOUNT
1810 The name or security ID (SID) of the domain specified is inconsistent with the trust information for that domain. ERROR_DOMAIN_TRUST_INCONSISTENT
1811 The server is in use and cannot be unloaded. ERROR_SERVER_HAS_OPEN_HANDLES
1812 The specified image file did not contain a resource section. ERROR_RESOURCE_DATA_NOT_FOUND
1813 The specified resource type cannot be found in the image file. ERROR_RESOURCE_TYPE_NOT_FOUND
1814 The specified resource name cannot be found in the image file. ERROR_RESOURCE_NAME_NOT_FOUND
1815 The specified resource language ID cannot be found in the image file. ERROR_RESOURCE_LANG_NOT_FOUND
1816 Not enough quota is available to process this command. ERROR_NOT_ENOUGH_QUOTA

1817 No interfaces have been registered. RPC_S_NO_INTERFACES
1818 The remote procedure call was cancelled. RPC_S_CALL_CANCELLED
1819 The binding handle does not contain all required information. RPC_S_BINDING_INCOMPLETE
1820 A communications failure occurred during a remote procedure call. RPC_S_COMM_FAILURE
1821 The requested authentication level is not supported. RPC_S_UNSUPPORTED_AUTHN_LEVEL
1822 No principal name registered. RPC_S_NO_PRINC_NAME
1823 The error specified is not a valid Windows RPC error code. RPC_S_NOT_RPC_ERROR
1824 A UUID that is valid only on this computer has been allocated. RPC_S_UUID_LOCAL_ONLY
1825 A security package specific error occurred. RPC_S_SEC_PKG_ERROR
1826 Thread is not canceled. RPC_S_NOT_CANCELLED
1827 Invalid operation on the encoding/decoding handle. RPC_X_INVALID_ES_ACTION
1828 Incompatible version of the serializing package. RPC_X_WRONG_ES_VERSION
1829 Incompatible version of the RPC stub. RPC_X_WRONG_STUB_VERSION
1830 The RPC pipe object is invalid or corrupted. RPC_X_INVALID_PIPE_OBJECT
1831 An invalid operation was attempted on an RPC pipe object. RPC_X_WRONG_PIPE_ORDER
1832 Unsupported RPC pipe version. RPC_X_WRONG_PIPE_VERSION
1898 The group member was not found. RPC_S_GROUP_MEMBER_NOT_FOUND
1899 The endpoint mapper database entry could not be created. EPT_S_CANT_CREATE
1900 The object universal unique identifier (UUID) is the nil UUID. RPC_S_INVALID_OBJECT
1901 The specified time is invalid. ERROR_INVALID_TIME
1902 The specified form name is invalid. ERROR_INVALID_FORM_NAME
1903 The specified form size is invalid. ERROR_INVALID_FORM_SIZE
1904 The specified printer handle is already being waited on ERROR_ALREADY_WAITING
1905 The specified printer has been deleted. ERROR_PRINTER_DELETED
1906 The state of the printer is invalid. ERROR_INVALID_PRINTER_STATE
1907 The user must change his password before he logs on the first time. ERROR_PASSWORD_MUST_CHANGE
1908 Could not find the domain controller for this domain. ERROR_DOMAIN_CONTROLLER_NOT_FOUND
1909 The referenced account is currently locked out and may not be logged on to. ERROR_ACCOUNT_LOCKED_OUT
1910 The object exporter specified was not found. OR_INVALID_OXID
1911 The object specified was not found. OR_INVALID_OID
1912 The object resolver set specified was not found. OR_INVALID_SET
1913 Some data remains to be sent in the request buffer. RPC_S_SEND_INCOMPLETE
1914 Invalid asynchronous remote procedure call handle. RPC_S_INVALID_ASYNC_HANDLE
1915 Invalid asynchronous RPC call handle for this operation. RPC_S_INVALID_ASYNC_CALL
1916 The RPC pipe object has already been closed. RPC_X_PIPE_CLOSED
1917 The RPC call completed before all pipes were processed. RPC_X_PIPE_DISCIPLINE_ERROR
1918 No more data is available from the RPC pipe. RPC_X_PIPE_EMPTY
1919 No site name is available for this machine. ERROR_NO_SITENAME
1920 The file can not be accessed by the system. ERROR_CANT_ACCESS_FILE
1921 The name of the file cannot be resolved by the system. ERROR_CANT_RESOLVE_FILENAME
1922 The directory service evaluated group memberships locally. ERROR_DS_MEMBERSHIP_EVALUATED_LOCALLY
1923 The specified directory service attribute or value does not exist. ERROR_DS_NO_ATTRIBUTE_OR_VALUE
1924 The attribute syntax specified to the directory service is invalid. ERROR_DS_INVALID_ATTRIBUTE_SYNTAX
1925 The attribute type specified to the directory service is not defined. ERROR_DS_ATTRIBUTE_TYPE_UNDEFINED
1926 The specified directory service attribute or value already exists. ERROR_DS_ATTRIBUTE_OR_VALUE_EXISTS
1927 The directory service is busy. ERROR_DS_BUSY
1928 The directory service is unavailable. ERROR_DS_UNAVAILABLE

1929 The directory service was unable to allocate a relative identifier. ERROR_DS_NO_RIDS_ALLOCATED
1930 The directory service has exhausted the pool of relative identifiers. ERROR_DS_NO_MORE_RIDS
1931 The requested operation could not be performed because the directory service is not the master for that type of operation. ERROR_DS_INCORRECT_ROLE_OWNER
1932 The directory service was unable to initialize the subsystem that allocates relative identifiers. ERROR_DS_RIDMGR_INIT_ERROR
1933 The requested operation did not satisfy one or more constraints associated with the class of the object. ERROR_DS_OBJ_CLASS_VIOLATION
1934 The directory service can perform the requested operation only on a leaf object. ERROR_DS_CANT_ON_NON_LEAF
1935 The directory service cannot perform the requested operation on the RDN attribute of an object. ERROR_DS_CANT_ON_RDN
1936 The directory service detected an attempt to modify the object class of an object. ERROR_DS_CANT_MOD_OBJ_CLASS
1937 The requested cross domain move operation could not be performed. ERROR_DS_CROSS_DOM_MOVE_ERROR
1938 Unable to contact the global catalog server. ERROR_DS_GC_NOT_AVAILABLE
2000 The pixel format is invalid. ERROR_INVALID_PIXEL_FORMAT
2001 The specified driver is invalid. ERROR_BAD_DRIVER
2002 The window style or class attribute is invalid for this operation. ERROR_INVALID_WINDOW_STYLE
2003 The requested metafile operation is not supported. ERROR_METAFILE_NOT_SUPPORTED
2004 The requested transformation operation is not supported. ERROR_TRANSFORM_NOT_SUPPORTED
2005 The requested clipping operation is not supported. ERROR_CLIPPING_NOT_SUPPORTED
2108 The network connection was made successfully, but the user had to be prompted for a password other than the one originally specified. ERROR_CONNECTED_OTHER_PASSWORD
2202 The specified username is invalid. ERROR_BAD_USERNAME
2250 This network connection does not exist. ERROR_NOT_CONNECTED
2300 The specified color management module is invalid. ERROR_INVALID_CMM
2301 The specified color profile is invalid. ERROR_INVALID_PROFILE
2302 The specified tag was not found. ERROR_TAG_NOT_FOUND
2303 A required tag is not present. ERROR_TAG_NOT_PRESENT
2304 The specified tag is already present. ERROR_DUPLICATE_TAG
2305 The specified color profile is not associated with any device. ERROR_PROFILE_NOT_ASSOCIATED_WITH_DEVICE
2306 The specified color profile was not found. ERROR_PROFILE_NOT_FOUND
2307 The specified color space is invalid. ERROR_INVALID_COLORSPACE
2308 Image Color Management is not enabled. ERROR_ICM_NOT_ENABLED
2309 There was an error while deleting the color transform. ERROR_DELETING_ICM_XFORM
2310 The specified color transform is invalid. ERROR_INVALID_TRANSFORM
2401 This network connection has files open or requests pending. ERROR_OPEN_FILES
2402 Active connections still exist. ERROR_ACTIVE_CONNECTIONS
2404 The device is in use by an active process and cannot be disconnected. ERROR_DEVICE_IN_USE
3000 The specified print monitor is unknown. ERROR_UNKNOWN_PRINT_MONITOR
3001 The specified printer driver is currently in use. ERROR_PRINTER_DRIVER_IN_USE
3002 The spool file was not found. ERROR_SPOOL_FILE_NOT_FOUND
3003 A StartDocPrinter call was not issued. ERROR_SPL_NO_STARTDOC
3004 An AddJob call was not issued. ERROR_SPL_NO_ADDJOB
3005 The specified print processor has already been installed. ERROR_PRINT_PROCESSOR_ALREADY_INSTALLED
3006 The specified print monitor has already been installed. ERROR_PRINT_MONITOR_ALREADY_INSTALLED

3007 The specified print monitor does not have the required functions. ERROR_INVALID_PRINT_MONITOR
3008 The specified print monitor is currently in use. ERROR_PRINT_MONITOR_IN_USE
3009 The requested operation is not allowed when there are jobs queued to the printer. ERROR_PRINTER_HAS_JOBS_QUEUED
3010 The requested operation is successful. Changes will not be effective until the system is rebooted. ERROR_SUCCESS_REBOOT_REQUIRED
3011 The requested operation is successful. Changes will not be effective until the service is restarted. ERROR_SUCCESS_RESTART_REQUIRED
4000 WINS encountered an error while processing the command. ERROR_WINS_INTERNAL
4001 The local WINS can not be deleted. ERROR_CAN_NOT_DEL_LOCAL_WINS
4002 The importation from the file failed. ERROR_STATIC_INIT
4003 The backup failed. Was a full backup done before? ERROR_INC_BACKUP
4004 The backup failed. Check the directory to which you are backing the database. ERROR_FULL_BACKUP
4005 The name does not exist in the WINS database. ERROR_REC_NON_EXISTENT
4006 Replication with a nonconfigured partner is not allowed. ERROR_RPL_NOT_ALLOWED
4100 The DHCP client has obtained an IP address that is already in use on the network. The local interface will be disabled until the DHCP client can obtain a new address. ERROR_DHCP_ADDRESS_CONFLICT
4200 The GUID passed was not recognized as valid by a WMI data provider. ERROR_WMI_GUID_NOT_FOUND
4201 The instance name passed was not recognized as valid by a WMI data provider. ERROR_WMI_INSTANCE_NOT_FOUND
4202 The data item ID passed was not recognized as valid by a WMI data provider. ERROR_WMI_ITEMID_NOT_FOUND
4203 The WMI request could not be completed and should be retried. ERROR_WMI_TRY_AGAIN
4204 The WMI data provider could not be located. ERROR_WMI_DP_NOT_FOUND
4205 The WMI data provider references an instance set that has not been registered. ERROR_WMI_UNRESOLVED_INSTANCE_REF
4206 The WMI data block or event notification has already been enabled. ERROR_WMI_ALREADY_ENABLED
4207 The WMI data block is no longer available. ERROR_WMI_GUID_DISCONNECTED
4208 The WMI data service is not available. ERROR_WMI_SERVER_UNAVAILABLE
4209 The WMI data provider failed to carry out the request. ERROR_WMI_DP_FAILED
4210 The WMI MOF information is not valid. ERROR_WMI_INVALID_MOF
4211 The WMI registration information is not valid. ERROR_WMI_INVALID_REGINFO
4300 The media identifier does not represent a valid medium. ERROR_INVALID_MEDIA
4301 The library identifier does not represent a valid library. ERROR_INVALID_LIBRARY
4302 The media pool identifier does not represent a valid media pool. ERROR_INVALID_MEDIA_POOL
4303 The drive and medium are not compatible or exist in different libraries. ERROR_DRIVE_MEDIA_MISMATCH
4304 The medium currently exists in an offline library and must be online to perform this operation. ERROR_MEDIA_OFFLINE
4305 The operation cannot be performed on an offline library. ERROR_LIBRARY_OFFLINE
4306 The library, drive, or media pool is empty. ERROR_EMPTY
4307 The library, drive, or media pool must be empty to perform this operation. ERROR_NOT_EMPTY
4308 No media is currently available in this media pool or library. ERROR_MEDIA_UNAVAILABLE
4309 A resource required for this operation is disabled. ERROR_RESOURCE_DISABLED
4310 The media identifier does not represent a valid cleaner. ERROR_INVALID_CLEANER
4311 The drive cannot be cleaned or does not support cleaning. ERROR_UNABLE_TO_CLEAN
4312 The object identifier does not represent a valid object. ERROR_OBJECT_NOT_FOUND
4313 Unable to read from or write to the database. ERROR_DATABASE_FAILURE

4314 The database is full. ERROR_DATABASE_FULL
4315 The medium is not compatible with the device or media pool. ERROR_MEDIA_INCOMPATIBLE
4316 The resource required for this operation does not exist. ERROR_RESOURCE_NOT_PRESENT
4317 The operation identifier is not valid. ERROR_INVALID_OPERATION
4318 The media is not mounted or ready for use. ERROR_MEDIA_NOT_AVAILABLE
4319 The device is not ready for use. ERROR_DEVICE_NOT_AVAILABLE
4320 The operator or administrator has refused the request. ERROR_REQUEST_REFUSED
4350 The remote storage service was not able to recall the file. ERROR_FILE_OFFLINE
4351 The remote storage service is not operational at this time. ERROR_REMOTE_STORAGE_NOT_ACTIVE
4352 The remote storage service encountered a media error. ERROR_REMOTE_STORAGE_MEDIA_ERROR
4390 The file or directory is not a reparse point. ERROR_NOT_A_REPARSE_POINT
4391 The reparse point attribute cannot be set because it conflicts with an existing attribute. ERROR_REPARSE_ATTRIBUTE_CONFLICT
5001 The cluster resource cannot be moved to another group because other resources are dependent on it. ERROR_DEPENDENT_RESOURCE_EXISTS
5002 The cluster resource dependency cannot be found. ERROR_DEPENDENCY_NOT_FOUND
5003 The cluster resource cannot be made dependent on the specified resource because it is already dependent. ERROR_DEPENDENCY_ALREADY_EXISTS
5004 The cluster resource is not online. ERROR_RESOURCE_NOT_ONLINE
5005 A cluster node is not available for this operation. ERROR_HOST_NODE_NOT_AVAILABLE
5006 The cluster resource is not available. ERROR_RESOURCE_NOT_AVAILABLE
5007 The cluster resource could not be found. ERROR_RESOURCE_NOT_FOUND
5008 The cluster is being shut down. ERROR_SHUTDOWN_CLUSTER
5009 A cluster node cannot be evicted from the cluster while it is online. ERROR_CANT_EVICT_ACTIVE_NODE
5010 The object already exists. ERROR_OBJECT_ALREADY_EXISTS
5011 The object is already in the list. ERROR_OBJECT_IN_LIST
5012 The cluster group is not available for any new requests. ERROR_GROUP_NOT_AVAILABLE
5013 The cluster group could not be found. ERROR_GROUP_NOT_FOUND
5014 The operation could not be completed because the cluster group is not online. ERROR_GROUP_NOT_ONLINE
5015 The cluster node is not the owner of the resource. ERROR_HOST_NODE_NOT_RESOURCE_OWNER
5016 The cluster node is not the owner of the group. ERROR_HOST_NODE_NOT_GROUP_OWNER
5017 The cluster resource could not be created in the specified resource monitor. ERROR_RESMON_CREATE_FAILED
5018 The cluster resource could not be brought online by the resource monitor. ERROR_RESMON_ONLINE_FAILED
5019 The operation could not be completed because the cluster resource is online. ERROR_RESOURCE_ONLINE
5020 The cluster resource could not be deleted or brought offline because it is the quorum resource. ERROR_QUORUM_RESOURCE
5021 The cluster could not make the specified resource a quorum resource because it is not capable of being a quorum resource. ERROR_NOT_QUORUM_CAPABLE
5022 The cluster software is shutting down. ERROR_CLUSTER_SHUTTING_DOWN
5023 The group or resource is not in the correct state to perform the requested operation. ERROR_INVALID_STATE
5024 The properties were stored but not all changes will take effect until the next time the resource is brought online. ERROR_RESOURCE_PROPERTIES_STORED
5025 The cluster could not make the specified resource a quorum resource because it does not belong to a shared storage class. ERROR_NOT_QUORUM_CLASS
5026 The cluster resource could not be deleted since it is a core resource. ERROR_CORE_RESOURCE
5027 The quorum resource failed to come online. ERROR_QUORUM_RESOURCE_ONLINE_FAILED

5028 The quorum log could not be created or mounted successfully. ERROR_QUORUMLOG_OPEN_FAILED
5029 The cluster log is corrupt. ERROR_CLUSTERLOG_CORRUPT
5030 The record could not be written to the cluster log since it exceeds the maximum size. ERROR_CLUSTERLOG_RECORD_EXCEEDS_MAXSIZE
5031 The cluster log exceeds its maximum size. ERROR_CLUSTERLOG_EXCEEDS_MAXSIZE
5032 No checkpoint record was found in the cluster log. ERROR_CLUSTERLOG_CHKPOINT_NOT_FOUND
5033 The minimum required disk space needed for logging is not available. ERROR_CLUSTERLOG_NOT_ENOUGH_SPACE
6000 The specified file could not be encrypted. ERROR_ENCRYPTION_FAILED
6001 The specified file could not be decrypted. ERROR_DECRYPTION_FAILED
6002 The specified file is encrypted and the user does not have the ability to decrypt it. ERROR_FILE_ENCRYPTED
6003 There is no encryption recovery policy configured for this system. ERROR_NO_RECOVERY_POLICY
6004 The required encryption driver is not loaded for this system. ERROR_NO_EFS
6005 The file was encrypted with a different encryption driver than is currently loaded. ERROR_WRONG_EFS
6006 There are no EFS keys defined for the user. ERROR_NO_USER_KEYS
6007 The specified file is not encrypted. ERROR_FILE_NOT_ENCRYPTED
6008 The specified file is not in the defined EFS export format. ERROR_NOT_EXPORT_FORMAT
6118 The list of servers for this workgroup is not currently available ERROR_NO_BROWSER_SERVERS_FOUND

Win32-Netzwerk-Fehler

Win32 Net Error Codes

2102 The workstation driver is not installed. NETERR_INCLUDED
2103 The server could not be located. NERR_UnknownServer
2104 An internal error occurred. The network cannot access a shared memory segment. NERR_ShareMem
2105 A network resource shortage occurred . NERR_NoNetworkResource
2106 This operation is not supported on workstations. NERR_RemoteOnly
2107 The device is not connected. NERR_DevNotRedirected
2114 The Server service is not started. NERR_ServerNotStarted
2115 The queue is empty. NERR_ItemNotFound
2116 The device or directory does not exist. NERR_UnknownDevDir
2117 The operation is invalid on a redirected resource. NERR_RedirectedPath
2118 The name has already been shared. NERR_DuplicateShare
2119 The server is currently out of the requested resource. NERR_NoRoom
2121 Requested addition of items exceeds the maximum allowed. NERR_TooManyItems
2122 The Peer service supports only two simultaneous users. NERR_InvalidMaxUsers
2123 The API return buffer is too small. NERR_BufTooSmall
2127 A remote API error occurred. NERR_RemoteErr
2131 An error occurred when opening or reading the configuration file. NERR_LanmanIniError
2136 A general network error occurred. NERR_NetworkError
2137 The Workstation service is in an inconsistent state. Restart the computer before restarting the Workstation service. NERR_WkstaInconsistentState
2138 The Workstation service has not been started. NERR_WkstaNotStarted
2139 The requested information is not available. NERR_BrowserNotStarted
2140 An internal Windows NT error occurred. NERR_InternalError
2141 The server is not configured for transactions. NERR_BadTransactConfig
2142 The requested API is not supported on the remote server. NERR_InvalidAPI

2143 The event name is invalid. NERR_BadEventName
2144 The computer name already exists on the network. Change it and restart the computer. NERR_DupNameReboot
2146 The specified component could not be found in the configuration information. NERR_CfgCompNotFound
2147 The specified parameter could not be found in the configuration information. NERR_CfgParamNotFound
2149 A line in the configuration file is too long. NERR_LineTooLong
2150 The printer does not exist. NERR_QNotFound
2151 The print job does not exist. NERR_JobNotFound
2152 The printer destination cannot be found. NERR_DestNotFound
2153 The printer destination already exists. NERR_DestExists
2154 The printer queue already exists. NERR_QExists
2155 No more printers can be added. NERR_QNoRoom
2156 No more print jobs can be added. NERR_JobNoRoom
2157 No more printer destinations can be added. NERR_DestNoRoom
2158 This printer destination is idle and cannot accept control operations. NERR_DestIdle
2159 This printer destination request contains an invalid control function. NERR_DestInvalidOp
2160 The print processor is not responding. NERR_ProcNoRespond
2161 The spooler is not running. NERR_SpoolerNotLoaded
2162 This operation cannot be performed on the print destination in its current state. NERR_DestInvalidState
2163 This operation cannot be performed on the printer queue in its current state. NERR_QInvalidState
2164 This operation cannot be performed on the print job in its current state. NERR_JobInvalidState
2165 A spooler memory allocation failure occurred. NERR_SpoolNoMemory
2166 The device driver does not exist. NERR_DriverNotFound
2167 The data type is not supported by the print processor. NERR_DataTypeInvalid
2168 The print processor is not installed. NERR_ProcNotFound
2180 The service database is locked. NERR_ServiceTableLocked
2181 The service table is full. NERR_ServiceTableFull
2182 The requested service has already been started. NERR_ServiceInstalled
2183 The service does not respond to control actions. NERR_ServiceEntryLocked
2184 The service has not been started. NERR_ServiceNotInstalled
2185 The service name is invalid. NERR_BadServiceName
2186 The service is not responding to the control function. NERR_ServiceCtlTimeout
2187 The service control is busy. NERR_ServiceCtlBusy
2188 The configuration file contains an invalid service program name. NERR_BadServiceProgName
2189 The service could not be controlled in its present state. NERR_ServiceNotCtrl
2190 The service ended abnormally. NERR_ServiceKillProc
2191 The requested pause or stop is not valid for this service. NERR_ServiceCtlNotValid
2192 The service control dispatcher could not find the service name in the dispatch table. NERR_NotInDispatchTbl
2193 The service control dispatcher pipe read failed. NERR_BadControlRecv
2194 A thread for the new service could not be created. NERR_ServiceNotStarting
2200 This workstation is already logged on to the local-area network. NERR_AlreadyLoggedOn
2201 The workstation is not logged on to the local-area network. NERR_NotLoggedOn
2202 The user name or group name parameter is invalid. NERR_BadUsername
2203 The password parameter is invalid. NERR_BadPassword
2204 @W The logon processor did not add the message alias. NERR_UnableToAddName_W
2205 The logon processor did not add the message alias. NERR_UnableToAddName_F
2206 @W The logoff processor did not delete the message alias. NERR_UnableToDelName_W
2207 The logoff processor did not delete the message alias. NERR_UnableToDelName_F

2209 Network logons are paused. NERR_LogonsPaused
2210 A centralized logon-server conflict occurred. NERR_LogonServerConflict
2211 The server is configured without a valid user path. NERR_LogonNoUserPath
2212 An error occurred while loading or running the logon script. NERR_LogonScriptError
2214 The logon server was not specified. Your computer will be logged on as STANDALONE. NERR_StandaloneLogon
2215 The logon server could not be found. NERR_LogonServerNotFound
2216 There is already a logon domain for this computer. NERR_LogonDomainExists
2217 The logon server could not validate the logon. NERR_NonValidatedLogon
2219 The security database could not be found. NERR_ACFNotFound
2220 The group name could not be found. NERR_GroupNotFound
2221 The user name could not be found. NERR_UserNotFound
2222 The resource name could not be found. NERR_ResourceNotFound
2223 The group already exists. NERR_GroupExists
2224 The user account already exists. NERR_UserExists
2225 The resource permission list already exists. NERR_ResourceExists
2226 This operation is only allowed on the primary domain controller of the domain. NERR_NotPrimary
2227 The security database has not been started. NERR_ACFNotLoaded
2228 There are too many names in the user accounts database. NERR_ACFNoRoom
2229 A disk I/O failure occurred. NERR_ACFFileIOFail
2230 The limit of 64 entries per resource was exceeded. NERR_ACFTooManyLists
2231 Deleting a user with a session is not allowed. NERR_UserLogon
2232 The parent directory could not be located. NERR_ACFNoParent
2233 Unable to add to the security database session cache segment. NERR_CanNotGrowSegment
2234 This operation is not allowed on this special group. NERR_SpeGroupOp
2235 This user is not cached in user accounts database session cache. NERR_NotInCache
2236 The user already belongs to this group. NERR_UserInGroup
2237 The user does not belong to this group. NERR_UserNotInGroup
2238 This user account is undefined. NERR_AccountUndefined
2239 This user account has expired. NERR_AccountExpired
2240 The user is not allowed to log on from this workstation. NERR_InvalidWorkstation
2241 The user is not allowed to log on at this time. NERR_InvalidLogonHours
2242 The password of this user has expired. NERR_PasswordExpired
2243 The password of this user cannot change. NERR_PasswordCantChange
2244 This password cannot be used now. NERR_PasswordHistConflict
2245 The password is shorter than required. NERR_PasswordTooShort
2246 The password of this user is too recent to change. NERR_PasswordTooRecent
2247 The security database is corrupted. NERR_InvalidDatabase
2248 No updates are necessary to this replicant network/local security database. NERR_DatabaseUpToDate
2249 This replicant database is outdated; synchronization is required. NERR_SyncRequired
2250 The network connection could not be found. NERR_UseNotFound
2251 This asg_type is invalid. NERR_BadAsgType
2252 This device is currently being shared. NERR_DeviceIsShared
2270 The computer name could not be added as a message alias. The name may already exist on the network. NERR_NoComputerName
2271 The Messenger service is already started. NERR_MsgAlreadyStarted
2272 The Messenger service failed to start. NERR_MsgInitFailed
2273 The message alias could not be found on the network. NERR_NameNotFound
2274 This message alias has already been forwarded. NERR_AlreadyForwarded
2275 This message alias has been added but is still forwarded. NERR_AddForwarded
2276 This message alias already exists locally. NERR_AlreadyExists
2277 The maximum number of added message aliases has been exceeded. NERR_TooManyNames

2278 The computer name could not be deleted. NERR_DelComputerName
2279 Messages cannot be forwarded back to the same workstation. NERR_LocalForward
2280 An error occurred in the domain message processor. NERR_GrpMsgProcessor
2281 The message was sent, but the recipient has paused the Messenger service. NERR_PausedRemote
2282 The message was sent but not received. NERR_BadReceive
2283 The message alias is currently in use. Try again later. NERR_NameInUse
2284 The Messenger service has not been started. NERR_MsgNotStarted
2285 The name is not on the local computer. NERR_NotLocalName
2286 The forwarded message alias could not be found on the network. NERR_NoForwardName
2287 The message alias table on the remote station is full. NERR_RemoteFull
2288 Messages for this alias are not currently being forwarded. NERR_NameNotForwarded
2289 The broadcast message was truncated. NERR_TruncatedBroadcast
2294 This is an invalid device name. NERR_InvalidDevice
2295 A write fault occurred. NERR_WriteFault
2297 A duplicate message alias exists on the network. NERR_DuplicateName
2298 @W This message alias will be deleted later. NERR_DeleteLater
2299 The message alias was not successfully deleted from all networks. NERR_IncompleteDel
2300 This operation is not supported on computers with multiple networks. NERR_MultipleNets
2310 This shared resource does not exist. NERR_NetNameNotFound
2311 This device is not shared. NERR_DeviceNotShared
2312 A session does not exist with that computer name. NERR_ClientNameNotFound
2314 There is not an open file with that identification number. NERR_FileIdNotFound
2315 A failure occurred when executing a remote administration command. NERR_ExecFailure
2316 A failure occurred when opening a remote temporary file. NERR_TmpFile
2317 The data returned from a remote administration command has been truncated to 64K. NERR_TooMuchData
2318 This device cannot be shared as both a spooled and a non-spooled resource. NERR_DeviceShareConflict
2319 The information in the list of servers may be incorrect. NERR_BrowserTableIncomplete
2320 The computer is not active in this domain. NERR_NotLocalDomain
2321 The share must be removed from the Distributed File System before it can be deleted. NERR_IsDfsShare
2331 The operation is invalid for this device. NERR_DevInvalidOpCode
2332 This device cannot be shared. NERR_DevNotFound
2333 This device was not open. NERR_DevNotOpen
2334 This device name list is invalid. NERR_BadQueueDevString
2335 The queue priority is invalid. NERR_BadQueuePriority
2337 There are no shared communication devices. NERR_NoCommDevs
2338 The queue you specified does not exist. NERR_QueueNotFound
2340 This list of devices is invalid. NERR_BadDevString
2341 The requested device is invalid. NERR_BadDev
2342 This device is already in use by the spooler. NERR_InUseBySpooler
2343 This device is already in use as a communication device. NERR_CommDevInUse
2351 This computer name is invalid. NERR_InvalidComputer
2354 The string and prefix specified are too long. NERR_MaxLenExceeded
2356 This path component is invalid. NERR_BadComponent
2357 Could not determine the type of input. NERR_CantType
2362 The buffer for types is not big enough. NERR_TooManyEntries
2370 Profile files cannot exceed 64K. NERR_ProfileFileTooBig
2371 The start offset is out of range. NERR_ProfileOffset
2372 The system cannot delete current connections to network resources. NERR_ProfileCleanup
2373 The system was unable to parse the command line in this file. NERR_ProfileUnknownCmd
2374 An error occurred while loading the profile file. NERR_ProfileLoadErr

2375 @W Errors occurred while saving the profile file. The profile was partially saved. NERR_ProfileSaveErr
2377 Log file %1 is full. NERR_LogOverflow
2378 This log file has changed between reads. NERR_LogFileChanged
2379 Log file %1 is corrupt. NERR_LogFileCorrupt
2380 The source path cannot be a directory. NERR_SourceIsDir
2381 The source path is illegal. NERR_BadSource
2382 The destination path is illegal. NERR_BadDest
2383 The source and destination paths are on different servers. NERR_DifferentServers
2385 The Run server you requested is paused. NERR_RunSrvPaused
2389 An error occurred when communicating with a Run server. NERR_ErrCommRunSrv
2391 An error occurred when starting a background process. NERR_ErrorExecingGhost
2392 The shared resource you are connected to could not be found. NERR_ShareNotFound
2400 The LAN adapter number is invalid. NERR_InvalidLana
2401 There are open files on the connection. NERR_OpenFiles
2402 Active connections still exist. NERR_ActiveConns
2403 This share name or password is invalid. NERR_BadPasswordCore
2404 The device is being accessed by an active process. NERR_DevInUse
2405 The drive letter is in use locally. NERR_LocalDrive
2430 The specified client is already registered for the specified event. NERR_AlertExists
2431 The alert table is full. NERR_TooManyAlerts
2432 An invalid or nonexistent alert name was raised. NERR_NoSuchAlert
2433 The alert recipient is invalid. NERR_BadRecipient
2434 A user's session with this server has been deleted NERR_AcctLimitExceeded
2440 The log file does not contain the requested record number. NERR_InvalidLogSeek
2450 The user accounts database is not configured correctly. NERR_BadUasConfig
2451 This operation is not permitted when the Netlogon service is running. NERR_InvalidUASOp
2452 This operation is not allowed on the last administrative account. NERR_LastAdmin
2453 Could not find domain controller for this domain. NERR_DCNotFound
2454 Could not set logon information for this user. NERR_LogonTrackingError
2455 The Netlogon service has not been started. NERR_NetlogonNotStarted
2456 Unable to add to the user accounts database. NERR_CanNotGrowUASFile
2457 This server's clock is not synchronized with the primary domain controller's clock. NERR_TimeDiffAtDC
2458 A password mismatch has been detected. NERR_PasswordMismatch
2460 The server identification does not specify a valid server. NERR_NoSuchServer
2461 The session identification does not specify a valid session. NERR_NoSuchSession
2462 The connection identification does not specify a valid connection. NERR_NoSuchConnection
2463 There is no space for another entry in the table of available servers. NERR_TooManyServers
2464 The server has reached the maximum number of sessions it supports. NERR_TooManySessions
2465 The server has reached the maximum number of connections it supports. NERR_TooManyConnections
2466 The server cannot open more files because it has reached its maximum number. NERR_TooManyFiles
2467 There are no alternate servers registered on this server. NERR_NoAlternateServers
2470 Try down-level (remote admin protocol) version of API instead. NERR_TryDownLevel
2480 The UPS driver could not be accessed by the UPS service. NERR_UPSDriverNotStarted
2481 The UPS service is not configured correctly. NERR_UPSInvalidConfig
2482 The UPS service could not access the specified Comm Port. NERR_UPSInvalidCommPort
2483 The UPS indicated a line fail or low battery situation. Service not started. NERR_UPSSignalAsserted
2484 The UPS service failed to perform a system shut down. NERR_UPSShutdownFailed
2500 The program below returned an MS-DOS error code: NERR_BadDosRetCode

2501 The program below needs more memory: NERR_ProgNeedsExtraMem
2502 The program below called an unsupported MS-DOS function: NERR_BadDosFunction
2503 The workstation failed to boot. NERR_RemoteBootFailed
2504 The file below is corrupt. NERR_BadFileCheckSum
2505 No loader is specified in the boot-block definition file. NERR_NoRplBootSystem
2506 NetBIOS returned an error: The NCB and SMB are dumped above. NERR_RplLoadrNetBiosErr
2507 A disk I/O error occurred. NERR_RplLoadrDiskErr
2508 Image parameter substitution failed. NERR_ImageParamErr
2509 Too many image parameters cross disk sector boundaries. NERR_TooManyImageParams
2510 The image was not generated from an MS-DOS diskette formatted with /S. NERR_NonDosFloppyUsed
2511 Remote boot will be restarted later. NERR_RplBootRestart
2512 The call to the Remoteboot server failed. NERR_RplSrvrCallFailed
2513 Cannot connect to the Remoteboot server. NERR_CantConnectRplSrvr
2514 Cannot open image file on the Remoteboot server. NERR_CantOpenImageFile
2515 Connecting to the Remoteboot server... NERR_CallingRplSrvr
2516 Connecting to the Remoteboot server... NERR_StartingRplBoot
2517 Remote boot service was stopped; check the error log for the cause of the problem. NERR_RplBootServiceTerm
2518 Remote boot startup failed; check the error log for the cause of the problem. NERR_RplBootStartFailed
2519 A second connection to a Remoteboot resource is not allowed. NERR_RPL_CONNECTED
2550 The browser service was configured with MaintainServerList=No. NERR_BrowserConfiguredToNotRun
2610 Service failed to start since none of the network adapters started with this service. NERR_RplNoAdaptersStarted
2611 Service failed to start due to bad startup information in the registry. NERR_RplBadRegistry
2612 Service failed to start because its database is absent or corrupt. NERR_RplBadDatabase
2613 Service failed to start because RPLFILES share is absent. NERR_RplRplfilesShare
2614 Service failed to start because RPLUSER group is absent. NERR_RplNotRplServer
2615 Cannot enumerate service records. NERR_RplCannotEnum
2616 Workstation record information has been corrupted. NERR_RplWkstaInfoCorrupted
2617 Workstation record was not found. NERR_RplWkstaNotFound
2618 Workstation name is in use by some other workstation. NERR_RplWkstaNameUnavailable
2619 Profile record information has been corrupted. NERR_RplProfileInfoCorrupted
2620 Profile record was not found. NERR_RplProfileNotFound
2621 Profile name is in use by some other profile. NERR_RplProfileNameUnavailable
2622 There are workstations using this profile. NERR_RplProfileNotEmpty
2623 Configuration record information has been corrupted. NERR_RplConfigInfoCorrupted
2624 Configuration record was not found. NERR_RplConfigNotFound
2625 Adapter id record information has been corrupted. NERR_RplAdapterInfoCorrupted
2626 An internal service error has occurred. NERR_RplInternal
2627 Vendor id record information has been corrupted. NERR_RplVendorInfoCorrupted
2628 Boot block record information has been corrupted. NERR_RplBootInfoCorrupted
2629 The user account for this workstation record is missing. NERR_RplWkstaNeedsUserAcct
2630 The RPLUSER local group could not be found. NERR_RplNeedsRPLUSERAcct
2631 Boot block record was not found. NERR_RplBootNotFound
2632 Chosen profile is incompatible with this workstation. NERR_RplIncompatibleProfile
2633 Chosen network adapter id is in use by some other workstation. NERR_RplAdapterNameUnavailable
2634 There are profiles using this configuration. NERR_RplConfigNotEmpty
2635 There are workstations, profiles, or configurations using this boot block. NERR_RplBootInUse

2636 Service failed to backup Remoteboot database. NERR_RplBackupDatabase
2637 Adapter record was not found. NERR_RplAdapterNotFound
2638 Vendor record was not found. NERR_RplVendorNotFound
2639 Vendor name is in use by some other vendor record. NERR_RplVendorNameUnavailable
2640 (boot name, vendor id) is in use by some other boot block record. NERR_RplBootNameUnavailable
2641 Configuration name is in use by some other configuration. NERR_RplConfigNameUnavailable
2660 The internal database maintained by the Dfs service is corrupt NERR_DfsInternalCorruption
2661 One of the records in the internal Dfs database is corrupt NERR_DfsVolumeDataCorrupt
2662 There is no volume whose entry path matches the input Entry Path NERR_DfsNoSuchVolume
2663 A volume with the given name already exists NERR_DfsVolumeAlreadyExists
2664 The server share specified is already shared in the Dfs NERR_DfsAlreadyShared
2665 The indicated server share does not support the indicated Dfs volume NERR_DfsNoSuchShare
2666 The operation is not valid on a non-leaf volume NERR_DfsNotALeafVolume
2667 The operation is not valid on a leaf volume NERR_DfsLeafVolume
2668 The operation is ambiguous because the volume has multiple servers NERR_DfsVolumeHasMultipleServers
2669 Unable to create a junction point NERR_DfsCantCreateJunctionPoint
2670 The server is not Dfs Aware NERR_DfsServerNotDfsAware
2671 The specified rename target path is invalid NERR_DfsBadRenamePath
2672 The specified Dfs volume is offline NERR_DfsVolumeIsOffline
2673 The specified server is not a server for this volume NERR_DfsNoSuchServer
2674 A cycle in the Dfs name was detected NERR_DfsCyclicalName
2675 The operation is not supported on a server-based Dfs NERR_DfsNotSupportedInServerDfs
2690 Dfs internal error NERR_DfsInternalError
2691 This machine is already joined to a domain. NERR_SetupAlreadyJoined
2692 This machine is not currently joined to a domain. NERR_SetupNotJoined
2693 This machine is a domain controller and cannot be unjoined from a domain. NERR_SetupDomainController

SSL-Fehler

Glossar: SSL

SSL-Fehler haben das Format:

`error:[error code]:[library name]:[function name]:[reason string]`

`error code`: ist ein 8-stelliger hexadezimaler Code,
`library name`: bezeichnet die Bibliothek
`function name`: bezeichnet die jeweilige Funktion innerhalb der Bibliothek
`reason string`: verbale Fehlerbeschreibung

Angegeben sind im Folgenden die Reason-Strings und die Fehlerbeschreibung in einer besser lesbaren Form.

SSL_R_APP_DATA_IN_HANDSHAKE	100	app data in handshake
SSL_R_ATTEMPT_TO_REUSE_SESSION_IN_DIFFERENT_CONTEXT	272	attempt to reuse session in different context
SSL_R_BAD_ALERT_RECORD	101	bad alert record
SSL_R_BAD_AUTHENTICATION_TYPE	102	bad authentication type
SSL_R_BAD_CHANGE_CIPHER_SPEC	103	bad change cipher spec
SSL_R_BAD_CHECKSUM	104	bad checksum
SSL_R_BAD_DATA_RETURNED_BY_CALLBACK	106	bad data returned by callback

SSL_R_BAD_DECOMPRESSION	107	bad decompression
SSL_R_BAD_DH_G_LENGTH	108	bad dh g length
SSL_R_BAD_DH_PUB_KEY_LENGTH	109	bad dh pub key length
SSL_R_BAD_DH_P_LENGTH	110	bad dh p length
SSL_R_BAD_DIGEST_LENGTH	111	bad digest length
SSL_R_BAD_DSA_SIGNATURE	112	bad dsa signature
SSL_R_BAD_HELLO_REQUEST	105	bad hello request
SSL_R_BAD_LENGTH	271	bad length
SSL_R_BAD_MAC_DECODE	113	bad mac decode
SSL_R_BAD_MESSAGE_TYPE	114	bad message type
SSL_R_BAD_PACKET_LENGTH	115	bad packet length
SSL_R_BAD_PROTOCOL_VERSION_NUMBER	116	bad protocol version number
SSL_R_BAD_RESPONSE_ARGUMENT	117	bad response argument
SSL_R_BAD_RSA_DECRYPT	118	bad rsa decrypt
SSL_R_BAD_RSA_ENCRYPT	119	bad rsa encrypt
SSL_R_BAD_RSA_E_LENGTH	120	bad rsa e length
SSL_R_BAD_RSA_MODULUS_LENGTH	121	bad rsa modulus length
SSL_R_BAD_RSA_SIGNATURE	122	bad rsa signature
SSL_R_BAD_SIGNATURE	123	bad signature
SSL_R_BAD_SSL_FILETYPE	124	bad ssl filetype
SSL_R_BAD_SSL_SESSION_ID_LENGTH	125	bad ssl session id length
SSL_R_BAD_STATE	126	bad state
SSL_R_BAD_WRITE_RETRY	127	bad write retry
SSL_R_BIO_NOT_SET	128	bio not set
SSL_R_BLOCK_CIPHER_PAD_IS_WRONG	129	block cipher pad is wrong
SSL_R_BN_LIB	130	bn lib
SSL_R_CA_DN_LENGTH_MISMATCH	131	ca dn length mismatch
SSL_R_CA_DN_TOO_LONG	132	ca dn too long
SSL_R_CCS_RECEIVED_EARLY	133	ccs received early
SSL_R_CERTIFICATE_VERIFY_FAILED	134	certificate verify failed
SSL_R_CERT_LENGTH_MISMATCH	135	cert length mismatch
SSL_R_CHALLENGE_IS_DIFFERENT	136	challenge is different
SSL_R_CIPHER_CODE_WRONG_LENGTH	137	cipher code wrong length
SSL_R_CIPHER_OR_HASH_UNAVAILABLE	138	cipher or hash unavailable
SSL_R_CIPHER_TABLE_SRC_ERROR	139	cipher table src error
SSL_R_COMPRESSED_LENGTH_TOO_LONG	140	compressed length too long
SSL_R_COMPRESSION_FAILURE	141	compression failure
SSL_R_COMPRESSION_LIBRARY_ERROR	142	compression library error
SSL_R_CONNECTION_ID_IS_DIFFERENT	143	connection id is different
SSL_R_CONNECTION_TYPE_NOT_SET	144	connection type not set
SSL_R_DATA_BETWEEN_CCS_AND_FINISHED	145	data between ccs and finished
SSL_R_DATA_LENGTH_TOO_LONG	146	data length too long
SSL_R_DECRYPTION_FAILED	147	decryption failed
SSL_R_DECRYPTION_FAILED_OR_BAD_RECORD_MAC	1109	decryption failed or bad record mac
SSL_R_DH_PUBLIC_VALUE_LENGTH_IS_WRONG	148	dh public value length is wrong
SSL_R_DIGEST_CHECK_FAILED	149	digest check failed
SSL_R_ENCRYPTED_LENGTH_TOO_LONG	150	encrypted length too long
SSL_R_ERROR_GENERATING_TMP_RSA_KEY	1092	error generating tmp rsa key
SSL_R_ERROR_IN_RECEIVED_CIPHER_LIST	151	error in received cipher list
SSL_R_EXCESSIVE_MESSAGE_SIZE	152	excessive message size
SSL_R_EXTRA_DATA_IN_MESSAGE	153	extra data in message
SSL_R_GOT_A_FIN_BEFORE_A_CCS	154	got a fin before a ccs

SSL_R_HTTPS_PROXY_REQUEST	155	https proxy request
SSL_R_HTTP_REQUEST	156	http request
SSL_R_ILLEGAL_PADDING	1110	illegal padding
SSL_R_INTERNAL_ERROR	157	internal error
SSL_R_INVALID_CHALLENGE_LENGTH	158	invalid challenge length
SSL_R_INVALID_COMMAND	280	invalid command
SSL_R_INVALID_PURPOSE	278	invalid purpose
SSL_R_INVALID_TRUST	279	invalid trust
SSL_R_LENGTH_MISMATCH	159	length mismatch
SSL_R_LENGTH_TOO_SHORT	160	length too short
SSL_R_LIBRARY_BUG	274	library bug
SSL_R_LIBRARY_HAS_NO_CIPHERS	161	library has no ciphers
SSL_R_MESSAGE_TOO_LONG	1111	message too long
SSL_R_MISSING_DH_DSA_CERT	162	missing dh dsa cert
SSL_R_MISSING_DH_KEY	163	missing dh key
SSL_R_MISSING_DH_RSA_CERT	164	missing dh rsa cert
SSL_R_MISSING_DSA_SIGNING_CERT	165	missing dsa signing cert
SSL_R_MISSING_EXPORT_TMP_DH_KEY	166	missing export tmp dh key
SSL_R_MISSING_EXPORT_TMP_RSA_KEY	167	missing export tmp rsa key
SSL_R_MISSING_RSA_CERTIFICATE	168	missing rsa certificate
SSL_R_MISSING_RSA_ENCRYPTING_CERT	169	missing rsa encrypting cert
SSL_R_MISSING_RSA_SIGNING_CERT	170	missing rsa signing cert
SSL_R_MISSING_TMP_DH_KEY	171	missing tmp dh key
SSL_R_MISSING_TMP_RSA_KEY	172	missing tmp rsa key
SSL_R_MISSING_TMP_RSA_PKEY	173	missing tmp rsa pkey
SSL_R_MISSING_VERIFY_MESSAGE	174	missing verify message
SSL_R_NON_SSLV2_INITIAL_PACKET	175	non sslv2 initial packet
SSL_R_NO_CERTIFICATES_RETURNED	176	no certificates returned
SSL_R_NO_CERTIFICATE_ASSIGNED	177	no certificate assigned
SSL_R_NO_CERTIFICATE_RETURNED	178	no certificate returned
SSL_R_NO_CERTIFICATE_SET	179	no certificate set
SSL_R_NO_CERTIFICATE_SPECIFIED	180	no certificate specified
SSL_R_NO_CIPHERS_AVAILABLE	181	no ciphers available
SSL_R_NO_CIPHERS_PASSED	182	no ciphers passed
SSL_R_NO_CIPHERS_SPECIFIED	183	no ciphers specified
SSL_R_NO_CIPHER_LIST	184	no cipher list
SSL_R_NO_CIPHER_MATCH	185	no cipher match
SSL_R_NO_CLIENT_CERT_RECEIVED	186	no client cert received
SSL_R_NO_COMPRESSION_SPECIFIED	187	no compression specified
SSL_R_NO_METHOD_SPECIFIED	188	no method specified
SSL_R_NO_PRIVATEKEY	189	no privatekey
SSL_R_NO_PRIVATE_KEY_ASSIGNED	190	no private key assigned
SSL_R_NO_PROTOCOLS_AVAILABLE	191	no protocols available
SSL_R_NO_PUBLICKEY	192	no publickey
SSL_R_NO_SHARED_CIPHER	193	no shared cipher
SSL_R_NO_VERIFY_CALLBACK	194	no verify callback
SSL_R_NULL_SSL_CTX	195	null ssl ctx
SSL_R_NULL_SSL_METHOD_PASSED	196	null ssl method passed
SSL_R_OLD_SESSION_CIPHER_NOT_RETURNED	197	old session cipher not returned
SSL_R_PACKET_LENGTH_TOO_LONG	198	packet length too long
SSL_R_PATH_TOO_LONG	270	path too long
SSL_R_PEER_DID_NOT_RETURN_A_CERTIFICATE	199	peer did not return a certificate

SSL_R_PEER_ERROR	200	peer error
SSL_R_PEER_ERROR_CERTIFICATE	201	peer error certificate
SSL_R_PEER_ERROR_NO_CERTIFICATE	202	peer error no certificate
SSL_R_PEER_ERROR_NO_CIPHER	203	peer error no cipher
SSL_R_PEER_ERROR_UNSUPPORTED_CERTIFICATE_TYPE	204	peer error unsupported certificate type
SSL_R_PRE_MAC_LENGTH_TOO_LONG	205	pre mac length too long
SSL_R_PROBLEMS_MAPPING_CIPHER_FUNCTIONS	206	problems mapping cipher functions
SSL_R_PROTOCOL_IS_SHUTDOWN	207	protocol is shutdown
SSL_R_PUBLIC_KEY_ENCRYPT_ERROR	208	public key encrypt error
SSL_R_PUBLIC_KEY_IS_NOT_RSA	209	public key is not rsa
SSL_R_PUBLIC_KEY_NOT_RSA	210	public key not rsa
SSL_R_READ_BIO_NOT_SET	211	read bio not set
SSL_R_READ_WRONG_PACKET_TYPE	212	read wrong packet type
SSL_R_RECORD_LENGTH_MISMATCH	213	record length mismatch
SSL_R_RECORD_TOO_LARGE	214	record too large
SSL_R_RECORD_TOO_SMALL	1093	record too small
SSL_R_REQUIRED_CIPHER_MISSING	215	required cipher missing
SSL_R_REUSE_CERT_LENGTH_NOT_ZERO	216	reuse cert length not zero
SSL_R_REUSE_CERT_TYPE_NOT_ZERO	217	reuse cert type not zero
SSL_R_REUSE_CIPHER_LIST_NOT_ZERO	218	reuse cipher list not zero
SSL_R_SESSION_ID_CONTEXT_UNINITIALIZED	277	session id context uninitialized
SSL_R_SHORT_READ	219	short read
SSL_R_SIGNATURE_FOR_NON_SIGNING_CERTIFICATE	220	signature for non signing certificate
SSL_R_SSL23_DOING_SESSION_ID_REUSE	221	ssl23 doing session id reuse
SSL_R_SSL3_SESSION_ID_TOO_SHORT	222	ssl3 session id too short
SSL_R_SSLV3_ALERT_BAD_CERTIFICATE	1042	sslv3 alert bad certificate
SSL_R_SSLV3_ALERT_BAD_RECORD_MAC	1020	sslv3 alert bad record mac
SSL_R_SSLV3_ALERT_CERTIFICATE_EXPIRED	1045	sslv3 alert certificate expired
SSL_R_SSLV3_ALERT_CERTIFICATE_REVOKED	1044	sslv3 alert certificate revoked
SSL_R_SSLV3_ALERT_CERTIFICATE_UNKNOWN	1046	sslv3 alert certificate unknown
SSL_R_SSLV3_ALERT_DECOMPRESSION_FAILURE	1030	sslv3 alert decompression failure
SSL_R_SSLV3_ALERT_HANDSHAKE_FAILURE	1040	sslv3 alert handshake failure
SSL_R_SSLV3_ALERT_ILLEGAL_PARAMETER	1047	sslv3 alert illegal parameter
SSL_R_SSLV3_ALERT_NO_CERTIFICATE	1041	sslv3 alert no certificate
SSL_R_SSLV3_ALERT_PEER_ERROR_CERTIFICATE	223	sslv3 alert peer error certificate
SSL_R_SSLV3_ALERT_PEER_ERROR_NO_CERTIFICATE	224	sslv3 alert peer error no certificate
SSL_R_SSLV3_ALERT_PEER_ERROR_NO_CIPHER	225	sslv3 alert peer error no cipher
SSL_R_SSLV3_ALERT_PEER_ERROR_UNSUPPORTED_CERTIFICATE_TYPE	226	sslv3 alert peer error unsupported certificate type
SSL_R_SSLV3_ALERT_UNEXPECTED_MESSAGE	1010	sslv3 alert unexpected message
SSL_R_SSLV3_ALERT_UNKNOWN_REMOTE_ERROR_TYPE	227	sslv3 alert unknown remote error type
SSL_R_SSLV3_ALERT_UNSUPPORTED_CERTIFICATE	1043	sslv3 alert unsupported

		certificate
SSL_R_SSL_CTX_HAS_NO_DEFAULT_SSL_VERSION	228	ssl ctx has nodefault ssl version
SSL_R_SSL_HANDSHAKE_FAILURE	229	ssl handshake failure
SSL_R_SSL_LIBRARY_HAS_NO_CIPHERS	230	ssl library has no ciphers
SSL_R_SSL_SESSION_ID_CONTEXT_TOO_LONG	273	ssl session id context too long
SSL_R_SSL_SESSION_ID_IS_DIFFERENT	231	ssl session id is different
SSL_R_TLSV1_ALERT_ACCESS_DENIED	1049	tlsv1 alert access denied
SSL_R_TLSV1_ALERT_DECODE_ERROR	1050	tlsv1 alert decode error
SSL_R_TLSV1_ALERT_DECRYPTION_FAILED	1021	tlsv1 alert decryption failed
SSL_R_TLSV1_ALERT_DECRYPT_ERROR	1051	tlsv1 alert decrypt error
SSL_R_TLSV1_ALERT_EXPORT_RESTRICTION	1060	tlsv1 alert export restriction
SSL_R_TLSV1_ALERT_INSUFFICIENT_SECURITY	1071	tlsv1 alert insufficient security
SSL_R_TLSV1_ALERT_INTERNAL_ERROR	1080	tlsv1 alert internal error
SSL_R_TLSV1_ALERT_NO_RENEGOTIATION	1100	tlsv1 alert no renegotiation
SSL_R_TLSV1_ALERT_PROTOCOL_VERSION	1070	tlsv1 alert protocol version
SSL_R_TLSV1_ALERT_RECORD_OVERFLOW	1022	tlsv1 alert record overflow
SSL_R_TLSV1_ALERT_UNKNOWN_CA	1048	tlsv1 alert unknown ca
SSL_R_TLSV1_ALERT_USER_CANCELLED	1090	tlsv1 alert user cancelled
SSL_R_TLS_CLIENT_CERT_REQ_WITH_ANON_CIPHER	232	tls client cert req with anon cipher
SSL_R_TLS_PEER_DID_NOT_RESPOND_WITH_CERTIFICATE_LIST	233	tls peer did not respond with certificate list
SSL_R_TLS_RSA_ENCRYPTED_VALUE_LENGTH_IS_WRONG	234	tls rsa encrypted value length is wrong
SSL_R_TRIED_TO_USE_UNSUPPORTED_CIPHER	235	tried to use unsupported cipher
SSL_R_UNABLE_TO_DECODE_DH_CERTS	236	unable to decode dh certs
SSL_R_UNABLE_TO_EXTRACT_PUBLIC_KEY	237	unable to extract public key
SSL_R_UNABLE_TO_FIND_DH_PARAMETERS	238	unable to find dh parameters
SSL_R_UNABLE_TO_FIND_PUBLIC_KEY_PARAMETERS	239	unable to find public key parameters
SSL_R_UNABLE_TO_FIND_SSL_METHOD	240	unable to find ssl method
SSL_R_UNABLE_TO_LOAD_SSL2_MD5_ROUTINES	241	unable to load ssl2 md5 routines
SSL_R_UNABLE_TO_LOAD_SSL3_MD5_ROUTINES	242	unable to load ssl3 md5 routines
SSL_R_UNABLE_TO_LOAD_SSL3_SHA1_ROUTINES	243	unable to load ssl3 sha1 routines
SSL_R_UNEXPECTED_MESSAGE	244	unexpected message
SSL_R_UNEXPECTED_RECORD	245	unexpected record
SSL_R_UNINITIALIZED	276	uninitialized
SSL_R_UNKNOWN_ALERT_TYPE	246	unknown alert type
SSL_R_UNKNOWN_CERTIFICATE_TYPE	247	unknown certificate type
SSL_R_UNKNOWN_CIPHER_RETURNED	248	unknown cipher returned
SSL_R_UNKNOWN_CIPHER_TYPE	249	unknown cipher type
SSL_R_UNKNOWN_KEY_EXCHANGE_TYPE	250	unknown key exchange type
SSL_R_UNKNOWN_PKEY_TYPE	251	unknown pkey type
SSL_R_UNKNOWN_PROTOCOL	252	unknown protocol
SSL_R_UNKNOWN_REMOTE_ERROR_TYPE	253	unknown remote error type
SSL_R_UNKNOWN_SSL_VERSION	254	unknown ssl version
SSL_R_UNKNOWN_STATE	255	unknown state

SSL_R_UNSUPPORTED_CIPHER	256 unsupported cipher
SSL_R_UNSUPPORTED_COMPRESSION_ALGORITHM	257 unsupported compression algorithm
SSL_R_UNSUPPORTED_OPTION	1091 unsupported option
SSL_R_UNSUPPORTED_PROTOCOL	258 unsupported protocol
SSL_R_UNSUPPORTED_SSL_VERSION	259 unsupported ssl version
SSL_R_WRITE_BIO_NOT_SET	260 write bio not set
SSL_R_WRONG_CIPHER_RETURNED	261 wrong cipher returned
SSL_R_WRONG_MESSAGE_TYPE	262 wrong message type
SSL_R_WRONG_NUMBER_OF_KEY_BITS	263 wrong number of key bits
SSL_R_WRONG_SIGNATURE_LENGTH	264 wrong signature length
SSL_R_WRONG_SIGNATURE_SIZE	265 wrong signature size
SSL_R_WRONG_SSL_VERSION	266 wrong ssl version
SSL_R_WRONG_VERSION_NUMBER	267 wrong version number
SSL_R_X509_LIB	268 x509 lib
SSL_R_X509_VERIFICATION_SETUP_PROBLEMS	269 x509 verification setup problems

Servermeldungen bzw. Protokoll-Antwortcodes

Auszug (leicht geändert) aus Posting von Wolfgang Jäth in Newsgroup `hamster.de.misc` am 26.01.2010, vgl. Message-ID `<news:hjmc46.58c.1@ID-4636.user.dfncis.de>`[1]

Glossar:
POP3
SMTP
NNTP

Antwortcodes für POP3, SMTP und NNTP

Bei POP3 gibt es nur »+OK« und »-ERR«.

Für SMTP (RFC 2821) gelten:

```
211  System status, or system help reply
214  Help message
220  <domain> Service ready
221  <domain> Service closing transmission channel
250  Requested mail action okay, completed
251  User not local; will forward to <forward-path>
252  Cannot VRFY user, but will accept message and attempt delivery

354  Start mail input; end with <CRLF>.<CRLF>

421  <domain> Service not available, closing transmission channel (This may be a reply to any
     command if the service knows it must shut down)
450  Requested mail action not taken: mailbox unavailable (e.g., mailbox busy)
451  Requested action aborted: local error in processing
452  Requested action not taken: insufficient system storage

500  Syntax error, command unrecognized (This may include errors such as command line too long)
501  Syntax error in parameters or arguments
502  Command not implemented (see section 4.2.4)
503  Bad sequence of commands
504  Command parameter not implemented
550  Requested action not taken: mailbox unavailable (e.g., mailbox not found, no access, or
```

[1] Siehe beispielsweise Fassung bei Google Groups.

```
     command rejected for policy reasons)
551  User not local; please try <forward-path>
552  Requested mail action aborted: exceeded storage allocation
553  Requested action not taken: mailbox name not allowed (e.g., mailbox syntax incorrect)
554  Transaction failed (Or, in the case of a connection-opening response, "No SMTP service
     here")
```

Glossar: NNTP

Gemäß RFC 977 gelten für NNTP folgende Codes (wahrscheinlich gibt es noch ein paar mehr in Erweiterungen zu diesem RFC; aber das wäre eine Heidenarbeit, die alle zu durchsuchen):

```
100  help text follows
199  debug output

200  server ready - posting allowed
201  server ready - no posting allowed
202  slave status noted
205  closing connection - goodbye!
211  n f l s group selected
215  list of newsgroups follows
220  n <a> article retrieved - head and body follow 221 n <a> article retrieved - head follows
222  n <a> article retrieved - body follows
223  n <a> article retrieved - request text separately 230 list of new articles by message-id
     follows
231  list of new newsgroups follows
235  article transferred ok
240  article posted ok

335  send article to be transferred.  End with <CR-LF>.<CR-LF>
340  send article to be posted. End with <CR-LF>.<CR-LF>

400  service discontinued
411  no such news group
412  no newsgroup has been selected
420  no current article has been selected
421  no next article in this group
422  no previous article in this group
423  no such article number in this group
430  no such article found
435  article not wanted - do not send it
436  transfer failed - try again later
437  article rejected - do not try again.
440  posting not allowed
441  posting failed

500  command not recognized
501  command syntax error
502  access restriction or permission denied
503  program fault - command not performed
```

Fernsteuerung des Hamsters

Allgemeine Grundlagen und Installation

Glossar: TCP

Zum Zwecke der Fernsteuerung über TCP/IP-Netze besitzt der Hamster einen Remote-Control-Server, kurz ReCo-Server. Dieser ReCo-Server basiert auf dem Telnet-Protokoll und verwendet in der Voreinstellung den TCP-Port 23. Über das Menü »Einstellungen« → »Lokale Server« ⇒ »Fernzugriff« kann jedoch ein anderer Port konfiguriert werden. (Wenn weder »Allgemeines« noch »Fernzugriff« zu sehen sind, muss zuvor bei »Temp. erweiterte Einstellungen« ein Haken gesetzt werden.)

Es kann jeder Telnet-Client für den Verbindungsaufbau mittels Telnet verwendet werden. Jedoch sind Telnet-Clienten, welche ausschließlich das Zeichen CR für den Zeilenumbruch verwenden, ungeeignet, da der ReCo-Server des Hamsters für den Zeilenumbruch nur CRLF verwendet.

Der ReCo-Server erzeugt kein selbständiges Echo, weshalb man ein eventuell beim Telnet-Clienten vorhandenes lokales Echo einschalten sollte.

Um den ReCo-Server nutzen zu können, muss er aktiviert und für die Nutzer freigeschaltet werden. Dazu sind vier Dinge erforderlich:

1. ReCo-Server im Menü »Lokale Server« starten (»Start/Stop Telnet-Server«).
2. Die Bindung des ReCo-Servers an das gewünschte Netzwerkinterface im Menü »Einstellungen« → »Lokale Server« ⇒ »Allgemeines« → »IP-Access.hst bearbeiten« festlegen.
3. Die gewünschte IP-Adresse für den Zugriff auf den ReCo-Server im Menü »Einstellungen« → »Lokale Server« ⇒ »Fernzugriff« bei »An folgende IP binden« einstellen.
4. Der gewünschte Benutzer muss im Menü »Einstellungen« → »Benutzerverwaltung & Passworte« ⇒ »Lokale Benutzer« für die Benutzung des ReCo-Servers freigeschaltet werden (siehe den Reiter »Telnet-Server« bei Benutzern oder Benutzergruppen).

Bitte beachten Sie, dass ein potentieller Angreifer bei einem erfolgreichen Einbruch in den Remote-Control-Server vollen Zugriff auf den Hamster und je nach vorhandenen Skripten eventuell auch auf Ihren Rechner bzw. Ihr Rechnernetz haben könnte.

Glossar: SASL SSL

Stellen Sie daher die Zugriffsberechtigungen für den ReCo-Server so restriktiv wie möglich ein. Benutzen Sie nach Möglichkeit eine gesicherte Übertragung des Passwortes mittels der SASL-Mechanismen DIGEST-MD5 oder CRAM-MD5. Betreiben Sie Verbindungen mit dem ReCo-Server über das Internet nur bei Verwendung eines sicheren SSL-Tunnels (»Stunnel«) oder eines SSL-fähigen Clienten – oder direkt mit OpenSSL:

```
openssl s_client -connect ⟨Hamster-Servername_oder_-IP-Adresse⟩:⟨Hamster-ReCo-Port⟩
```

Syntax des Remote-Control-Servers

```
AUTH ([-SIMPLE] ⟨Username⟩ ⟨Password⟩) |
     (-SASL ⟨Mechanism⟩ [⟨InitialParameters⟩])
```

AUTH [-SIMPLE] ⟨Username⟩ ⟨Password⟩

Einfache Authentifizierung des Benutzers. Das Passwort wird hierbei als Klartext über die Leitung gesendet. Diese Form der Authentifizierung ist ohne SSL-Verschlüsselung nur in kleinen privaten Netzen geeignet, da sie selbst keinerlei Schutz des Passwortes gegen das Protokollieren der Verbindung bietet.

Glossar: SSL

AUTH -SASL ⟨Mechanism⟩ [⟨InitialParameters⟩]

Verbessertes SASL-Authentifizierungsverfahren nach RFC 2222.

SASL

Dieses Authentifizierungsverfahren unterstützt mehrere einzelne Mechanismen. Die Sicherheit dieser Mechanismen ist unterschiedlich. Folgende Mechanismen werden unterstützt:

CRAM-MD5 (RFC 1321, 2104 2195, 2202)
: Dieser Mechanismus bietet einen sehr guten Schutz des Passwortes bei der Übertragung vom Clienten zum Server. Es sind zwar theoretische Angriffe gegen MD5 bekannt, diese liegen aber dennoch weit außerhalb des praktisch Machbaren.

DIGEST-MD5 (RFC 1321, 2104 2195, 2202)
: Dieser Mechanismus bietet einen sehr guten Schutz des Passwortes bei der Übertragung vom Clienten zum Server. Es sind zwar theoretische Angriffe gegen MD5 bekannt, diese liegen aber dennoch weit außerhalb des praktisch Machbaren. Dieser Mechanismus ist CRAM-MD5 überlegen, da er den Server ebenfalls gegenüber dem User authentifiziert.

LOGIN** und **PLAIN
: Diese Algorithmen verschleiern das Passwort nur gering. Der Klartext des Passwortes kann an Hand von Tabellen sehr schnell ermittelt werden. Das Passwort wird hierbei als base64-codierter Klartext über die Leitung gesendet. Diese Form der Authentifizierung ist ohne SSL-Verschlüsselung nur in kleinen privaten Netzen geeignet, da sie selbst keinerlei Schutz des Passwortes gegen das Protokollieren der Verbindung bietet.

Base64

LOGIN und PLAIN unterscheiden sich für den Telnet-Nutzer nur in der Implementation, wie sie die Eingabe von Nutzername und Passwort erwarten. PLAIN ist im Vergleich komplizierter.

Der Vollständigkeit halber seien noch folgende Mechanismen erwähnt, die zusätzlich von den anderen Servern im Hamster, nicht aber durch den Remote-Control-Server unterstützt werden:

EXTERNAL (RFC 2487)
: Authentifizierung durch externe Software, z. B. SSL in Verbindung mit TLS-Handshake.

TLS

CRAM-SHA1 (RFC 2104, 2195, 2202, 3174)

Dieser kryptografisch sichere Mechanismus bietet den besten Schutz des Passwortes bei der Übertragung vom Clienten zum Server. Er ist beweisbar sicher gegen kryptografische Angriffe. Brute-Force-Angriffe liegen weit außerhalb des technisch Machbaren.

`HELP [⟨Command⟩]`

Listet die Befehlssyntax allgemein oder zu einem speziellen Befehl auf.

`LOG (LIST [-S ⟨Size⟩] [-P ⟨Pattern⟩]) | ROTATE`

LOG LIST [-S ⟨Size⟩] [-P ⟨Pattern⟩]

Listet die Log-Datei des Hamsters auf. Mit der Option **-S** kann die Anzeige der Datei auf eine bestimmte Anzahl von Kibibytes begrenzt werden. Der Standardwert ist 16. Die Option **-P** begrenzt die Anzeige der Log-Datei auf die Zeilen, welche dem Regulären Ausdruck **⟨Pattern⟩** entsprechen (siehe auch die FAQ »Hamster und Reguläre Ausdrücke«).

LOG ROTATE

Schließt die aktuelle Log-Datei und eröffnet eine neue Log-Datei. Dabei rotiert der Dateiname der Log-Datei entsprechend dem in den Grundeinstellungen angegebenen Wert.

```
NEWS (ADD|DEL GROUP ⟨Groupname⟩) |
     (ADD|DEL PULL ⟨Groupname⟩ ⟨Servername⟩ ) |
     (LIST GROUP|PULL [-P ⟨Pattern⟩])
```

NEWS ADD GROUP ⟨Groupname⟩
NEWS DEL GROUP ⟨Groupname⟩

ADD fügt eine Newsgruppe hinzu, und **DEL** entfernt eine Newsgruppe.

NEWS ADD PULL ⟨Groupname⟩ ⟨Servername⟩
NEWS DEL PULL ⟨Groupname⟩ ⟨Servername⟩

ADD fügt eine Newsgruppe der Liste der zu pullenden Newsgruppen für einen Server hinzu, **DEL** entfernt eine Newsgruppe aus der Liste der zu pullenden Newsgruppen.

NEWS LIST GROUP [-P ⟨Pattern⟩]
NEWS LIST PULL [-P ⟨Pattern⟩]

Listet alle Newsgruppen oder alle Newspull-Einträge auf. Die Auflistung kann durch den Regulären Ausdruck **⟨Pattern⟩** eingeschränkt werden (siehe auch die FAQ »Hamster und Reguläre Ausdrücke«).

`QUIT`

Beendet die Verbindung mit dem Remote-Control-Server.

`SCRIPT (LIST [-P ⟨Pattern⟩]) |`
`(START [-W] ⟨Scriptname⟩ [⟨Scriptparameter⟩]) |`
`STOP`

SCRIPT LIST [-P ⟨Pattern⟩]

Listet alle am Remote-Control-Server verfügbaren Skripte auf. Die Auflistung kann durch den Regulären Ausdruck **⟨Pattern⟩** eingeschränkt werden (siehe auch die FAQ »Hamster und Reguläre Ausdrücke«).

Das Verzeichnis, aus welchem diese Skripte gelesen werden, kann in der Datei »`hamster.ini`« im Abschnitt »[Directories]« mit dem Schlüssel »`ScriptsRC`« eingestellt werden.

SCRIPT START [-W] ⟨Scriptname⟩ [⟨Scriptparameter⟩]

Startet ein Skript unter optionaler Verwendung von Parametern. Der optionale Parameter **-W** legt fest, ob der Remote-Control-Server auf das Skript warten soll, bevor er seine Befehlsabarbeitung fortsetzt.

SCRIPT STOP

Stoppt alle Skripte.

`SERVER (STATUS|START|STOP|RESTART) (RECO|SMTP|POP3|NNTP)`

Startet oder stoppt einen lokalen Server oder liefert dessen momentanen Zustand zurück.

»`SERVER RESTART RECO`« lässt den Remote-Control-Server neu starten. Sie müssten sich danach aber wieder authentifizieren.

`TASK [-P ⟨Pattern⟩]`

Liefert eine Liste der aktiven Tasks zurück. Diese Liste kann durch den Regulären Ausdruck **⟨Pattern⟩** eingeschränkt werden (siehe auch die FAQ »Hamster und Reguläre Ausdrücke«).

`TASK STOP ⟨ThreadID⟩`

Stoppt einen einzelnen Task mit dem Identifier **⟨ThreadID⟩**.

Kommandozeilenoptionen des Hamsters

Der Hamster kann auch von der Kommandozeile aus und per Windows-Verknüpfung gestartet werden. Die Befehlszeile bzw. das Verknüpfungsziel sehen so aus:

```
hamster.exe [/w] [/svc] [/rw] [/ro] [/unregserver] [/unregserver]
            [[⟨Verzeichnis⟩!]⟨Skriptname⟩ [⟨Parameter1⟩] ... ]
```

Bedeutung der Parameter:

/w sorgt ggf. dafür, dass das aufrufende Batch- oder VB-Skript erst fortgesetzt wird, wenn der Hamster wieder beendet wird. Anderenfalls werden solche Skripte weiter ausgeführt, ohne das Ende des aufgerufenen Programms abzuwarten.

/svc verhindert, dass der als Dienst eingerichtete Hamster (nur Windows NT und Nachfolger) beim Abmelden des Windows-Benutzers beendet wird. Für die Einrichtung als Dienst benötigt man ein separates Programm wie Microsofts »srvany« oder »HamSvc« von Jürgen Haible (siehe in der Linkliste).

/ro ermöglicht eine Installation des Hamsters auf einem schreibgeschützten Datenträger.

/rw unterdrückt den Schreibschutz beim Start des Hamsters.

/regserver registriert den OLE/COM/DCOM-Server des Hamsters beim Betriebssystem. Hierfür sind Administratorrechte erforderlich *(nicht unter Windows 9x)*.

/unregserver meldet den OLE/COM/DCOM-Server des Hamsters beim Betriebssystem ab und entfernt die Einträge aus der Registry. Hierfür sind Administratorrechte erforderlich *(nicht unter Windows 9x)*.

`⟨Skriptname⟩` startet das Skript und übergibt diesem die folgenden Parameter (`⟨Parameter1⟩` usw.), sofern angegeben. In Verzeichnispfaden ist der Backslash »\« durch das Ausrufezeichen »!« zu ersetzen. Fehlt die Pfadangabe, muss das Skript sich im Skripteverzeichnis des Hamsters befinden.

Wenn Sie die Dateiendung »`.hsc`« für den Start mit dem Hamster registrieren, können Sie Skripte auch durch Doppelklick auf das Skriptdatei-Symbol oder durch die direkte Verknüpfung mit einem Skript starten. Dazu klicken Sie ganz einfach im Menü-Dialog »Einstellungen« → »Grundeinstellungen« ⇒ »Verschiedenes« auf den Button »*Endung hsc registrieren, um per Doppelklick starten zu können*«.

Links für Informationen, weitere Hilfe und Zusatztools zum Hamster

`http://tglsoft.de/freeware.html`
Homepage des Classic-Hamsters und einiger Zusatztools wie Korrnews und CopyIf.

`http://www.arcorhome.de/newshamster/micha/archive.php`
Viele Infos zu Hamster, Zusatztools wie Texteditoren mit Syntaxhighlighting und ein, *nein*, **das** Skriptarchiv von Michael Gebert.

`http://www.elbiah.de/hamster/`
Die Hamster-Seiten von Jürgen Haible, dem Urhamster-Entwickler; dort erhältlich eine andere Hamstervariante (Hamster Playground) und einige Zusatztools wie HamFind, HamRC und HamSvc.

`http://www.ximera.de/hamster.html`
: Infos zu Hamster und SSL von Martin Germann, Zusatztools.

  Achtung! Die dort angebotenen OpenSSL-Bibliotheken sind veraltet, bitte nicht mehr verwenden!

Glossar: SSL

`http://code.google.com/p/anytools/`
: Das »anytools«-Projekt von Remo Müller bietet unter anderem aktuelle SSL-Sicherheitszertifikate als Einzeldateien in der Form, wie sie der Hamster benötigt.

`http://home.pages.de/~heikor/`
: Zusatzprogramme und nützliche WSH-Erweiterungen von Heiko Rost.

`news:hamster.de.newuser`
: Hamster-Newsgruppe für Neulinge. In der Newshierarchie »hamster.*« gibt es weitere Gruppen – außer in deutsch auch in anderen Sprachen.

Hamster-Mailinglisten

deutsch: `http://www.yahoogroups.com/subscribe/usehamsternet/`
englisch: `http://www.yahoogroups.com/subscribe/ehamster/`
französisch: `http://www.yahoogroups.com/subscribe/fhamster/`
italienisch: `http://www.yahoogroups.com/subscribe/ihamster/`
polnisch: `http://www.yahoogroups.com/subscribe/plhamster/`

Versionsinformationen

Ab Version 2.0 wurde der SSL-Teil von Martin Germann komplett auf eigenen Code umgestellt. Daher ist ab dieser Version das Zusatzprogramm Stunnel nicht mehr erforderlich. Durch diese Umstellung musste die Syntax einiger hs2-Befehle geändert werden. Siehe hierzu die Updatehinweise.

Die alte Skriptsprache hs1 wird nicht mehr unterstützt.

Die Action »news.nntp« wurde in »news.in.nntp« umbenannt.

Ab Version 1.3.23 verursachen hs1-Skripte ohne Deklaration mit `#!hs1` eine Fehlermeldung. `HamWaitIdle` ohne Parameter verzögert mindestens 100 ms und 5 Schleifendurchläufe zur Abfrage des Idle-Zustandes. Per Parameter kann dieser Wert bis auf 50 ms reduziert werden. Bei Parameterwerten unter 50 ms wird ein Zwangs-Timeout von 100 ms eingestellt. Diese Vorgabe ist notwendig, damit langsam startende asynchrone Befehle unmittelbar vor dem HamWaitIdle vom HamWaitIdle sicher als aktiv erkannt werden.

Beim Update von einer Beta-Version muss eventuell der FQDN im Menü angepasst werden. Der jetzt eigenständige FQDN des Hamsters wird für einige zusätzliche Header und für Hamster-interne Info-Mails verwendet.

FQDN Header

Sollte in einer vorhergehenden Version der Benutzernamen in Form einer E-Mail-Adresse verwendet worden sein, so ist es angeraten, diesen auf ein RFC-2822-konformes Format für Mailboxnamen umzustellen. Folgende Zeichen sind für Benutzernamen zulässig:

[»!« | »#« | »$« | »&« | »'« | »+« | »-« | »_« |
»=« | ».« | »0«...»9« | »A«...»Z« | »a«...»z«]

Die Bindung der lokalen Server wurde aus Sicherheitsgründen von 0.0.0.0 als Standardwert auf 127.0.0.1 geändert. Gegebenfalls müssen die Einstellungen im Menü »Lokale Server« → »IP-Access« angepasst werden.

Ab Version 1.3.20 ist der Quelltext des Hamsters frei verfügbar (Open Source).

Ab Version 1.3.19 wurden die hamster-internen Gruppen `hamster.misc`, `hamster.posted` und `hamster.errors` durch die gemeinsame Gruppe `internal.misc` als Standardziel für interne Meldungen ersetzt. Diese Maßnahme war wegen der Gründung einer eigenen »hamster«-Top-Level-Hierarchie im Usenet notwendig geworden. Um dies ggf. wieder umzustellen, kann man entweder in der `hamster.ini` (Sektion [Setup]) die folgenden Änderungen durchführen:

```
internalgroup.default=hamster.misc
internalgroup.postok=hamster.posted
internalgroup.posterrors=hamster.errors
```

oder einfach den Menüpunkt »Einstellungen« → »Grundeinstellungen« ⇒ Dialog-Tab »Interne Gruppen« nutzen.

Ab Version 1.3.5 wird ein neues Format für die History-Datei verwendet. Eine Anpassung des Datenformats erfolgt automatisch beim Start einer höheren Version. Ein Wechsel auf eine niedrigere Version ist danach nicht mehr ohne weiteres möglich.

Glossar: POP3 NNTP

Ab Version 1.3.3 ist es nicht mehr möglich, sich ohne Name und Passwort an den lokalen POP3- und NNTP-Server des Hamsters anzumelden. Beim Wechsel von niedrigeren Versionen ist es erforderlich, ein Passwort für den Benutzer »admin« zu vergeben. Dieses muss dann natürlich auch beim verwendeten News- und E-Mail-Reader für den Zugriff auf den Hamster benutzt werden.

Update von Hamster < 1.3.23.130 auf ≧ 1.3.23.140

Ab dem Hamster 1.3.13.140 (Beta-Version im Vorfeld von Version 2.0.0.0) wurde die SSL-Funktionalität direkt in den Hamster integriert. Hierdurch wurde eine Änderung der HamFetchMail- bzw. HamSendMail-Skriptbefehle nötig.

Was ist zu tun?

1. **Für alle Nutzer, also auch solche, die bisher kein SSL benutzt haben:**

   Alle Skripte, die HamFetchMail- bzw. HamSendMail-Befehle nutzen, sollten daraufhin überprüft werden, ob bei den FetchMail-Parametern hinter dem ⟨LeaveOnServer⟩-Parameter bzw. den HamSendMail-Parametern hinter dem ⟨from-select⟩-Parameter Einträge existieren, die angepasst werden müssen.

   Die neue Syntax der Befehle lautet:

   ```
   HamFetchMail ( ⟨server⟩, ⟨port⟩, ⟨user⟩, ⟨pass⟩, ⟨destuser⟩, ⟨filter⟩,
                  ⟨LeaveOnServer⟩, ⟨SSLMode⟩, ⟨SSLVerify⟩, ⟨SSLCaFile⟩ )
   ```

   bzw.

   ```
   HamSendMail ( ⟨server⟩, ⟨port⟩, ⟨from-select⟩, ⟨to-select⟩,
                 ⟨SSLMode⟩, ⟨SSLVerify⟩, ⟨SSLCaFile⟩ )
   HamSendMailAuth ( ⟨server⟩, ⟨port⟩, ⟨user⟩, ⟨pass⟩, ⟨from-select⟩,
                     ⟨to-select⟩, ⟨SSLMode⟩, ⟨SSLVerify⟩, ⟨SSLCaFile⟩ )
   ```

   Wichtig ist nicht nur die neue Anzahl der Parameter, sondern auch das neue Parameterformat!

   ⟨SSLMode⟩: Integer [0...3]
   ⟨SSLVerify⟩: Integer [0...3]
   ⟨SSLCaFile⟩: String

   D. h. ein Befehl mit vollständig ausgefüllten Parametern sieht ohne Nutzung von SSL z. B. folgendermaßen aus:

   ```
   HamFetchMail("pop3-server", "110", "user", "pass", "admin", "", 0, 0, 0, "")
   ```

   bzw.

   ```
   HamSendMail ( "smtp-server", "25", ".*", ".*", 0, 0, "" )
   ```

   oder

   ```
   HamSendMailAuth( "smtp-server", "25", "", "", ".*", ".*", 0, 0, "" )
   ```

2. **Für Nutzer, die bisher SSL über Stunnel benutzt haben oder jetzt SSL nutzen wollen:**

   2.1. **Hamster als Client (Zugriff auf externe Server):**

   Hier sei wärmstens die Lektüre der SSL-FAQ empfohlen: Hamster und SSL.

   Soviel an dieser Stelle:

   Um SSL überhaupt nutzen zu können, werden zwei zusätzliche DLL-Dateien benötigt: »libssl32.dll« und »libeay32.dll« in aktueller Ausführung, d. h. minimal erforderlich ist OpenSSL Version 0.9.6b, mehr dazu siehe in der SSL-FAQ.

Die beiden DLLs müssen in das Hamster-Hauptverzeichnis oder ein Verzeichnis, das in der Umgebungsvariable `%PATH%` enthalten ist, kopiert werden. Der Hamster findet sie beim Start automatisch.

Unter »Einstellungen« → »Grundeinstellungen« ⇒ »SSL« stehen die Grundeinstellungen für SSL. Es gibt dort zwei Möglichkeiten von Zertifikatsspeichern:

Unter »Path to certificates for verification« kann ein Verzeichnis angegeben werden, in dem alle Zertifikate in Einzeldateien vorliegen. Die Namen der Dateien müssen dann dem Format »⟨Hash_des_Zertifikates⟩.0« entsprechen (s. SSL-FAQ).

Glossar: PEM

Eine alternative Möglichkeit besteht in der Angabe einer PEM-Datei unter »File with certificates for verification«, die alle Zertifikate gesammelt enthält.

Beide Möglichkeiten können additiv verwendet werden, nötig ist nur die Angabe einer von beiden.

Hier ist zu beachten, dass die eingetragenen Zertifikatsspeicher zur Überprüfung aller SSL-Zertifikate verwendet werden – unabhängig davon, wie die SSL-Funktionen aufgerufen wurden (Menü oder Skript). Individuell abweichende Einstellungen für einzelne Server können jeweils in deren Einzelkonfiguration vorgenommen bzw. über die SSL-Parameter der Skriptbefehle übergeben werden.

Auch müssen alle SSL-Einstellungen für die Einzelserver erneut aktiviert werden, da die alten Einstellungen nicht übernommen werden.

SSL-Settings:

- Choose Usage:

  ⟨SSLMode⟩ Wie soll SSL verwendet werden?
  0 – SSL abgeschaltet (default)
  1 – SSL auf separatem Port (z. B. 465, 995, 563)
  2 – TLS, wenn möglich (Fallback auf unverschlüsselte Verbindung, wenn TLS nicht möglich ist)
  3 – TLS wird erzwungen (auf Standardport, z. B. 25, 110, 119)

- Choose Verification:

  ⟨SSLVerify⟩ Steuert die Zertifikatsüberprüfung (entspricht der Option »-v« von Stunnel).
  0 – Zertifikatsüberprüfung abgeschaltet
  1 – Zertifikatsüberprüfung, falls Zertifikat vorgelegt
  2 – Zertifikatsüberprüfung immer
  3 – Zertifikatsüberprüfung immer und Vergleich des Serverzertifikats mit lokaler Kopie

- Verify with File:

  ⟨SSLCaFile⟩ Der Pfad zum Zertifikat für die Überprüfung. Wenn leer, wird der in den Grundeinstellungen eingetragene Standard genommen.

2.2. **Hamster als Server (lokale Server):** Um SSL mit den lokalen Servern von Hamster verwenden zu können, wird ein Zertifikat mit geheimem Schlüssel benötigt. Beim Versuch, die lokalen Server ohne ein solches Schlüsselpaar in Betrieb zu nehmen, beschwert sich der Hamster mit einer Fehlermeldung und nimmt die lokalen Server ohne SSL-Unterstützung in Betrieb (!).

   Näheres zur Erzeugung und Installation steht in der SSL-FAQ.

3. **Aufräumen alter SSL-Einträge:** Wenn sicher ist, dass die alte Methode über Stunnel nicht mehr benötigt wird, können folgende Einträge entfernt werden:

   aus der »hamster.ini«:

   SSLWrapperNotClose
   SSLProg
   SSLStartTimeout
   SSLlocalStartTimeout
   SSLKeyFile
   StunnelOption

   aus allen »server.ini«-Dateien:

   UseSSL
   SSL-Port
   StunnelOption

   und natürlich die »stunnelw.exe« und zugehörige Dateien.

Danksagungen

Die im Hamster verwendeten »Regular Expression«-Funktionen nutzen eine Win32-Umsetzung von »Perl Compatible Regular Expression« (PCRE), geschrieben von Philip Hazel, Copyright © 1997–2000 University of Cambridge (Näheres unter http://www.pcre.org/).

Obwohl die für PCRE benutzte Wrapper-Class ein wenig anders aufgebaut ist, sind zumindest Teile von den Konzepten inspiriert, welche in »uperlre« zum Tragen kommen. »uperlre« ist eine Wrapper-Class für PCRE, die von Luu Tran geschrieben wurde, der auch der Autor des »XNews«-Newsreaders ist (siehe auch http://xnews.newsguy.com/).

Besonderen Dank an Jörn Weber für die lange Pflege und massive Erweiterung dieser Hamster-Hilfe. Vielen Dank auch an Michael Jaritz, der anschließend die Pflege übernommen hatte.

Dank auch an alle, die Beiträge lieferten, und an alle Hinweisgeber, die auf Fehler und unklare Formulierungen hingewiesen haben.

Danke!

Lizenzen, Maintainer

```
================================================================================
Hamster, a free news- and mailserver for personal, family and workgroup use.
Copyright (c) 1999, Juergen Haible (<juergen.haible@t-online.de>).

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to
deal in the Software without restriction, including without limitation the
rights to use, copy, modify, merge, publish, distribute, sublicense, and/or
sell copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in
all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS
IN THE SOFTWARE.
================================================================================
```

Bei dieser Lizenz handelt es sich um die **MIT-Lizenz**, siehe http://de.wikipedia.org/wiki/MIT-Lizenz. Dort findet man auch eine deutsche Übersetzung.

Weitere Lizenzen & Patente

Achtung! Einige Zusatztools des Hamsters, insbesondere die OpenSSL-Bibliotheken, enthalten Softwaretechnologien, welche unter anderen Lizenzen stehen oder durch Patente geschützt sein könnten. Informieren Sie sich bitte vor der Verwendung der Programme über diese Problematik.

Maintainer

Der Maintainer des »Hamster Classic« ist Thomas G. Liesner (http://www.tglsoft.de/).

Zum Programm haben jedoch weitere Personen Code beigetragen. In den Dateien »changes_de.txt« bzw. »changes_en.txt« ist dies dokumentiert.

Siehe auch Informationen über die Deutsche Hamster-Hilfe.

FAQ: Häufige Fragen & Probleme

FAQ: Zusammenstellung der häufig gestellten Fragen und regelmässig vorkommenden Probleme

Wie häufig in FAQ-Listen sind auch hier auf den folgenden Seiten die Fragen nicht thematisch sortiert aufgeführt, sondern in der Reihenfolge, wie sie zufällig in die FAQ aufgenommen wurden. Siehe deshalb folgende ***Übersicht****:*

Sicherheit:

- Hamster und Sicherheit
- Hamster und SSL
- Hamster und Authentifikation
- Hamster und sichere Passwörter
- Wieso funktionieren meine Passworte plötzlich nicht mehr?
- Überschreitung des Nutzerlimits

Installation/Konfiguration/Nutzung:

- Hamster aktualisieren (Update)
- Maximale Dateigröße im Hamster
- Hamster und Reguläre Ausdrücke
- Hamster und FQDN
- Kann ich den Hamster über einen Router betreiben?
- Start des lokalen ⟨XXXX⟩-Servers misslungen
- Netzressourcen schonen
- Feeden von Newsgruppen
- Wie kann ich ältere Artikel nachladen?
- Unterschied zwischen »Gruppe zurücksetzen« und »Gruppe löschen«
- Zugriff auf den lokalen Newsserver ohne Username und Passwort
- Wieso lässt sich die Gruppe »internal.misc« nicht löschen?
- Mailinglisten in Newsgruppen konvertieren
- Zwei Hamster, geht das?
- Wie richte ich einen Archivhamster ein

- Zugriff auf den Hamster vom LAN aus?
- Hamster und Mailverteilung in Firmennetzen
- Wie kann ich bei einem Provider mehrere Postfächer abfragen?
- Scheduler-FAQ (Selbständiges regelmäßiges Holen und Versenden von News und Mails)
- Sofortiges Versenden von E-Mails
- E-Mails umleiten
- Hamster aktualisieren (Update)
- Hamster und IMAP
- Hamster und Events

Fehlersuche:

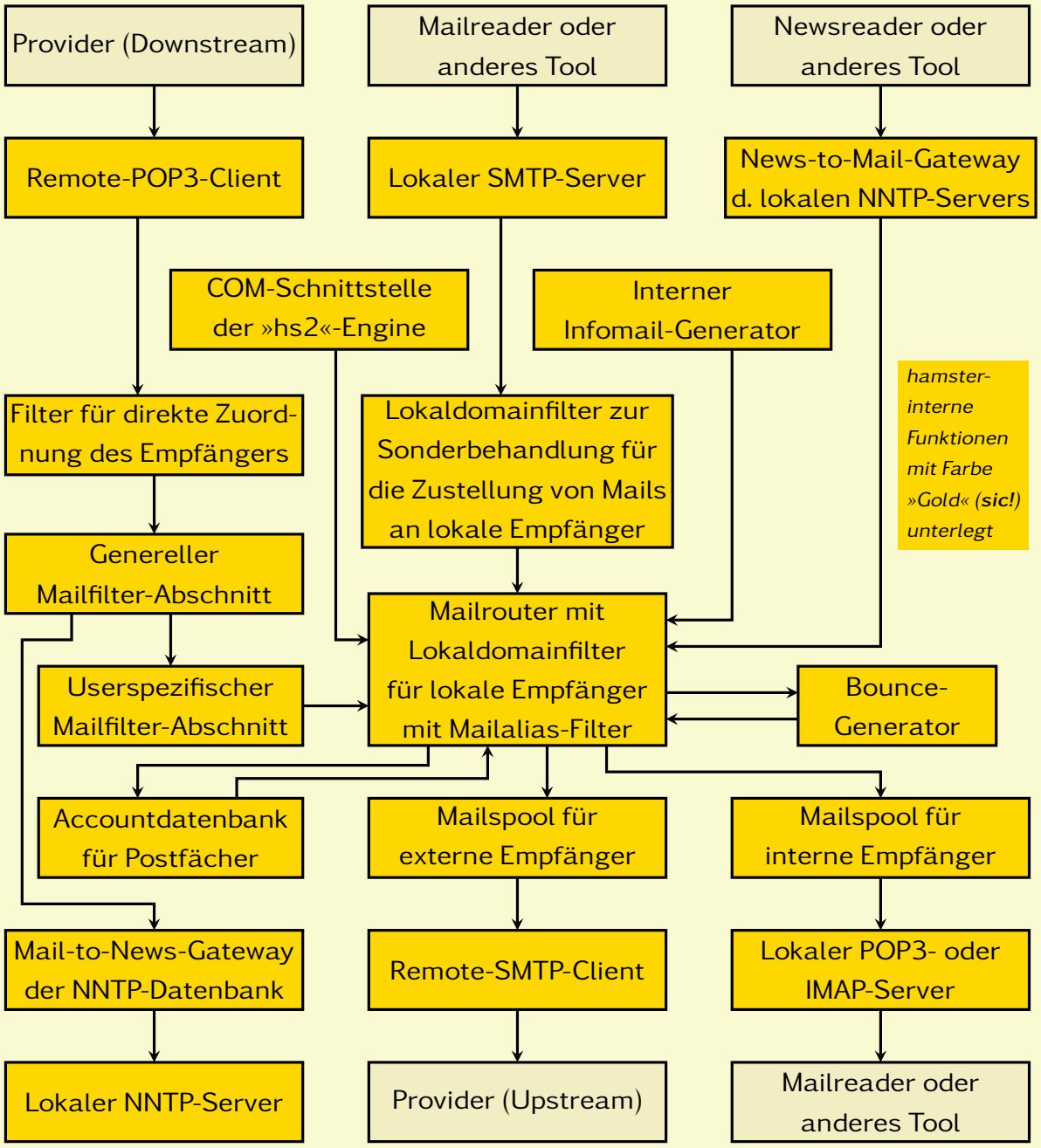
- Wie kann ich sehen, was im Hamster vorgeht?
- Wieso funktionieren meine Passworte plötzlich nicht mehr?
- Start des lokalen ⟨XXXX⟩-Servers misslungen
- Wie werden E-Mails im Hamster geroutet?
- Fehlermeldung »Missing AUTH« beim Versenden von E-Mails
- News und Mails mit Hilfe von Telnet kontrollieren
- Wo bekomme ich Hilfe her, wenn ich nicht weiter weiß?

Zusammenarbeit mit anderen Programmen:

- Hamster und Antivirenprogramme
- Hamster und große E-Mails
- Fehlermeldung »Missing AUTH« beim Versenden von E-Mails
- Wie importiere ich meine Mails in ein IMAP-Postfach?
- Zugriff auf den lokalen Newsserver ohne Username und Passwort

Einige veraltete Texte, die in früheren Hilfeversionen enthalten waren, sind weiterhin unter `http://speravir.website.org/files/hamsterhilfe/alt/` abrufbar.

Wie werden E-Mails im Hamster geroutet?



Ursprünglich von Jörn Weber am 20.03.2002 in Newsgroup `hamster.de.misc` *gepostet, vgl. Message-ID* `<news:43e80b65138fb92863cfb3e899bf2bba@blechtrottel.joernweber.de>`[1] *(Auszug, bearbeitet und ergänzt durch Steffen Hoffmann)*

Vgl. auch die Übersicht über den Datenfluss der Newsartikel im Hamster

[1] Siehe beispielsweise Fassung bei Google Groups.

Hamster und Antivirenprogramme

Antivirenprogramme können den Hamster in verschiedener Weise behindern:

Glossar: POP3

Zunächst einmal gibt es Programme, die sich selbst im Betriebssystem für den POP3-Standardport 110 registrieren, so dass der Hamster seinen lokalen POP3-Server zunächst nicht starten kann. Die Abhilfe hierfür ist, für den lokalen Server einen anderen Port zu verwenden, was zur Folge hat, dass auch die Mailreader umgestellt werden müssen. Beim Abrufen neuer E-Mail kann es dann zu einem Time-Out im Hamster oder Mailreader kommen, aber eine einmal geladene Mail erzeugt bei diesen Programmen später keine Störungen mehr, Usenet-News werden gar nicht beeinträchtigt.

Jedoch funktionieren inzwischen die meisten, wenn nicht alle Antivirenprogramme für den Privatanwender auf andere Art: Um eine Echtzeitscanfunktion zu ermöglichen, arbeiten sie als Systemdienst und installieren dazu einen speziellen Filtertreiber (On-Access- oder Background-Scanner). Sie können den Hamster sowohl beim Abruf von Mail als auch von News behindern. Darüber hinaus kann es ein zweites Mal Probleme geben, wenn mit dem Mail- oder Newsreader auf den Hamster zugegriffen wird. Man kann dann nur versuchen, die erlaubten Time-Out-Spannen zu erhöhen oder, wenn das Antivirenprogramm es ermöglicht, den Hamster zu den Ausnahmeprozessen hinzuzufügen (das würde dann jedoch nicht den Abruf durch die lokalen Reader betreffen).

Man kann sich auch grundsätzlich überlegen, die On-Access-Scanfunktion aller Netzwerkaktivitäten, soweit möglich, zu deaktivieren – das hängt aber vom Nutzungsverhalten der User ab: Im Hamster selbst werden keine potentiell schädlichen Dateien/Skripte ausgeführt, die in Mail oder News enthalten sind; das Lesen von Textmail und -news führt auch nicht dazu. Erst das Abspeichern angehängter Daten, das Lesen von HTML-Mail (oder seltener -News) oder die Anzeige korrupter Bilddateien ist potentiell gefährlich, dann sollte aber ein guter On-Access-Scanner von selbst aktiv werden.

Ein weiteres Problem kann sowohl durch ständig laufende Background-Scanner als auch beim manuellen Scan-Aufruf (On-Demand-Scanner) auftreten: Der Versuch des Antivirenprogramms, einen vermuteten oder wirklich vorhandenen Schädling aus einer »`data.dat`«, der Artikeldatenbank für eine Newsgruppe, zu entfernen, wird zu irreparablen Schäden an dieser Datei und damit dem Verlust aller lokal gespeicherten Artikel dieser Newsgruppe führen. Da E-Mails in den Hamster-Mailverzeichnissen als einzelne »`msg`«-Dateien gespeichert werden, betrifft dies in einem solchen Fall »nur« einzelne dort gespeicherte Mails. Daher sollte man, wenn es in dem Programm möglich ist, unbedingt die beiden Hamster-Unterverzeichnisse »`Groups`« und »`Mails`« vom Scannen ausschließen.

Überschreitung des Nutzerlimits

> *Ich bekomme immer die folgende Fehlermeldung angezeigt: »News-Aufträge für ⟨News-Server⟩ zurückgestellt wegen Überschreitung des Nutzerlimits.«*

Das, was Du da siehst, ist eine Info und keine Fehlermeldung. Wenn Du es nicht selbst anders eingestellt hast, erscheinen Fehlermeldungen im Hamster rot und Warnungen gelb.

Diese Info soll dir Folgendes sagen: Der Hamster versucht erst einmal, alle Gruppen parallel zu laden (der Hamster ist ein Multi-Thread-Programm). Da aber, je nach Provider, maximal 1–4 Threads pro Server zulässig sind, stellt der Hamster überzählige Ladeaufträge zurück und lädt sie dann nacheinander.

Fehlermeldung »Missing AUTH« beim Versenden von E-Mails

> *Beim Versenden von E-Mails erhalte ich die Fehlermeldung: »SMTP mail.example.net: Missing AUTH in EHLO-reply!«*

Glossar: SMTP, SMTP-AUTH

Hast Du vielleicht versehentlich das SMTP-AUTH-Verfahren für den SMTP-Server aktiviert? Ändern kannst Du das im Menü »Einstellungen« → »Mail: Server konfigurieren« ⇒ »SMTP-Mailserver«. Wenn die Authentifizierung gewollt ist, dann muss Dein Mailclient fehlkonfiguriert sein oder einen Bug im Programmcode besitzen.

Bekannt für einen ähnlichen Fehler wegen eines Bugs ist der Mailclient in »Xnews«; die Fehlermeldung lautet dort allerdings »*503 Bad sequence of commands (AUTH)*«, da »Xnews« sich trotz »SMTP-AUTH« mit »HELO« meldet (Fehlerbehebung nur durch Patchen der EXE-Datei möglich).

Start des lokalen ⟨XXXX⟩-Servers misslungen

> *Der lokale ⟨XXXX⟩-Server kann nicht gestartet werden!*

TCP, POP3, Port

Häufigste Ursache: Ein anderes Programm benutzt bereits den benötigten TCP-Port des Servers, z. B. einige Virus-Scanner für E-Mails, die sich selbst als POP3-Server beim Betriebssystem anmelden. Ein zweiter Hamster (aus einem anderem Verzeichnis als der schon laufende Hamster) könnte ebenfalls die Ursache sein. Abhilfe schafft hier die Verwendung einer anderen Port-Nummer für den lokalen Server. Diese Änderung muss dann natürlich auch im News- bzw. E-Mail-Client/-Reader eingetragen werden, da ansonsten nicht auf den Hamster, sondern eben auf das andere Programm zugegriffen würde. Eine weitere Möglichkeit ist auch eine Firewall

(z. B. »Zone Alarm«), deren Regeln für die Benutzung der Ports durch den Hamster u. U. erst angepasst werden müssen. In diesem Fall ist das Handbuch oder die Hilfe der entsprechenden Firewall zu Rate zu ziehen.

Eine weitere Ursache dafür, dass Hamster die Server nicht starten kann, kann eine fehlerhafte Bindung der Server an eine IP sein. In diesem Fall kommt im Logbuch die Fehlermeldung »10049« (Cannot assign requested address.)

Neben einer fehlerhaften manuellen Eingabe in den Feldern »An folgende IP binden« in den Menüs »Einstellungen« → »Lokale Server …« ⇒ »NNTP/POP3/SMTP«, kann eine dynamische Adressvergabe im Netzwerk Ursache für eine falsche Eintragung sein. Wenn es nicht möglich ist, dem Hamsterrechner eine feste IP zuzuweisen, kann man mit dem Skript »DHCP-bind« aus dem Skriptarchiv (siehe Linkliste) die Server an die jeweils aktuelle IP der Netzwerkkarte binden.

Glossar: NNTP POP3 SMTP

Eine weitere mögliche Fehlerursache für die Windows-Versionen 98, ME, 2000, XP und 2003 ist, dass durch den Media-Sense-Detektor des Netzwerkkarten-Treibers die IP-Adresse für die Bindung der lokalen Server temporär deaktiviert wird. Dieses passiert immer dann, wenn (d. h. solange) an der Netzwerkkarte, an welche die lokalen Server des Hamsters gebunden sind, die Gegenstelle fehlt (z. B. ein Hub oder Router). Das kommt meistens vor, wenn kein Hub/Switch benutzt wird, sondern nur zwei Rechner mit einem sogenannten Crossover-Kabel verbunden sind. Dieses »Feature« lässt sich jedoch deaktivieren. Siehe zu diesem Thema auch in der Microsoft-KB: http://support.microsoft.com/kb/q239924/.

Hamster und große E-Mails

Der Versand von E-Mail-Anhängen über den Hamster ist standardmäßig nur bis zu Größen von rund 2 MB möglich (genau 2 097 152 Bytes). Größere Mails rufen eine Fehlermeldung sowie den Abbruch der Nachricht hervor.

Dieses Verhalten liegt darin begründet, dass der Hamster eine Größenbeschränkung für E-Mails und News-Postings beinhaltet. Diese Größenbeschränkung ist in der Datei »hamster.ini« eingetragen, wobei zwischen E-Mail und News generell sowie zwischen Senden und Empfangen von E-Mails im Speziellen unterschieden wird. Es gilt für die Einträge (in der Sektion »[Setup]«) folgendes:

Limit für die Zeilenlänge der lokal empfangenen Artikel, getrennt nach Protokoll:

```
local.limit.linelen.nntp=1000
local.limit.linelen.pop3=1000
local.limit.linelen.smtp=1000
; (Für IMAP gibt es kein Zeilenlängenlimit.)
```

Limit für die Textgröße der lokal empfangenen Artikel, getrennt nach Protokoll:

```
local.limit.textsize.nntp=2097152
```

```
local.limit.textsize.pop3=2097152
local.limit.textsize.imap=2097152
local.limit.textsize.smtp=2097152
```

Um also die Beschränkungen auszuhebeln, können die Zahlen variiert werden; die Eingabe der 0 (Null) hebt die jeweilige Beschränkung ganz auf.

Die Änderung der Einträge für die Textgröße ist auch möglich über das Menü »Einstellungen« → »Lokale Server« ⇒ ⟨Reiter des jeweiligen Servers⟩.

Vor dem Aufheben oder Ändern der Beschränkungen sollte man sich jedoch über folgendes im Klaren sein:

Glossar: Base64

- E-Mail-Anhänge werden speziell base64-codiert, wodurch sich ihre Größe um rund ⅓ aufbläht (Faktor 1,36). Somit wird z. B. eine 2 MiB große Datei im Anhang einer E-Mail durch die Codierung auf 2,72 MiB aufgebläht.

SMTP
POP3
IMAP

- Derart große E-Mails belasten Mailserver und Datenleitungen, denn die Datei wird vom Versender über den SMTP-Server (meist des Providers) im Internet sowie über einen oder mehrere Mailserver bis letztlich zum POP3- oder IMAP-Server (des Empfängers) und von dort zum Empfänger transportiert. Damit wird einmal der SMTP-Server und einmal der POP3-Server über Gebühr und Aufgabe belastet, was einer DoS-Attacke (Denial-of-Service) gegenüber den Mailservern teilweise schon sehr nahe kommt (wer möchte, kann ja mal eine 90-MB-E-Mail an seinen Hamster schicken, während drei Leute versuchen, ihre E-Mails abzuholen …). Bleibt die E-Mail auf dem Server, wie bei IMAP meist üblich, aber auch bei POP3 möglich, so werden bei jedem Aufruf dieser E-Mail größere Datenmengen durch das Netz transportiert!

- Für die Übertragung solcher Datenmengen ist von Haus aus ein FTP-Server prädestiniert (gerade dann, wenn so etwas häufiger vorkommt); dessen Einrichtung sollte einem Hamster-Administrator nicht schwer fallen. Kostenlose FTP-Server für Windows sind erhältlich, z. B. »Filezilla Server« (Open Source). Dateien können zusätzlich auch komprimiert und dabei zur Sicherheit verschlüsselt werden. Dem gewünschten Empfänger muss dann nur noch das Kennwort bekanntgegeben werden.

- Auch der E-Mail-Client des Empfängers kann sich freuen, denn:

  Eingehende E-Mails werden zuerst in der Inbox gespeichert. Dort werden sie entweder vom Empfänger gelöscht oder in einen anderen Ordner verschoben. Beide Aktionen haben normalerweise zur Folge, dass zumindest im Ordner »Inbox« noch die komplette E-Mail enthalten ist, sofern man den Ordner nicht komprimiert. Somit liegen die aufgeblasene Datei in Form des E-Mail-Anhangs sowie als gespeicherte Form (der Empfänger wird die Datei ja wohl noch auf die Festplatte auslagern) vor, was dann allein schon rund 200 MB herumliegende Daten ausmacht. Wird die E-Mail verschoben und die Inbox nicht komprimiert, liegen schon

rund 300 MB auf der Platte (zweimal als Mail-Anhang, einmal als Datei) auf der Festplatte. Dann löscht man vielleicht die E-Mails auf dem Hamster nicht und schon liegt das ganze nochmal in einer rund 120 MB großen Textdatei vor.

- Die Belastung des Arbeitsspeichers ist enorm, da die E-Mail immer zweimal im Speicher liegt: Einmal durch den Hamster (Abholen/Senden) sowie einmal beim E-Mail-Client. Bei »richtig« großen E-Mails oder Systemen mit wenig Arbeitsspeicher (64 MB ist in diesem Zusammenhang wenig) kann das durchaus zum totalen System-Kill führen.
- Das FTP-Protokoll ist in der Lage, bei Abbrüchen den bereits übertragenen Teil zu sichern und bei einer neuen Verbindung nur den verbleibenden Rest nachzuladen.

  Im Mailprotokoll gilt dagegen: Ganz oder gar nicht.

Wie kann ich sehen, was im Hamster vorgeht?

(*Siehe auch* »News und Mails mit Hilfe von Telnet kontrollieren«.)

Als erstes im Menü »Einstellungen« → »Grundeinstellungen« ⇒ Reiter »Optik« (für die Protokollanzeige im Hauptfenster) oder besser Reiter »Protokolle« (für die auf dem Rechner gespeicherten Protokolldateien) die Kategorien »**Info**« und »**Detail**«, ggf. auch »**Debug**«, einschalten (*Vorsicht:* Debug erzeugt sehr viele Daten). Wird ein Skript verwendet, sollte als zweites der Befehl »trace(1)« vor dem kritischen Skriptabschnitt eingetragen werden, damit alle Skriptbefehle angezeigt werden. Der Befehl »trace(0)« hebt die Anzeige der Skriptbefehle wieder auf. Der Befehl »Dump« veranlasst den Hamster, alle Variablen eines Skriptes anzuzeigen.

Wo bekomme ich Hilfe her, wenn ich nicht weiter weiß?

Wenn Ihnen diese Hilfe-Datei keine ausreichenden Hinweise zur Lösung Ihrer Probleme liefert, schreiben Sie eine Anfrage in die Newsgruppe hamster.de.newuser. Dort lesen die Programmierer und erfahrenen Anwender mit und können Ihnen sicherlich jede Frage zum Hamster beantworten. Bitte beachten Sie, dass Ihnen nur so gut geholfen werden kann, wie präzise Ihre Fragestellung ist. Auf die Frage »Mein Hamster funktioniert nicht!!11elf« wird Ihnen kaum jemand helfen können. Versuchen Sie daher, Ihr Problem so präzise wie möglich zu schildern, im Zweifel lieber etwas ausführlicher als zu knapp. Sie ersparen Ihren freiwilligen Helfern damit unnötige Rückfragen. Schreiben Sie die folgenden wichtigen Angaben dazu:

- die Hamster-Version (siehe Menü »Hilfe« → »Info«), vor allem die letzten Stellen der Versionsnummer sind interessant,
- die Windows-Version (siehe auch das Menü »Hilfe« → »System-Information«),
- das Logfile oder, soweit es Ihnen möglich ist, besser der Ausschnitt aus dem Log, in dem Ihr Problem zu erkennen ist (**möglicherweise enthaltene Passwörter, unter Umständen nur base64-codiert, sollten anonymisiert werden!**),

Glossar: Base64

- den betroffenen externen Server,
- bei Problemen, die den Zugriff auf den Hamster betreffen, den Inhalt der Datei »IPAccess.hst«,
- bei Problemen, die den Download von E-Mails betreffen, den Inhalt der Datei »MailFilt.hst«,
- bei Problemen, die den Download von News betreffen, den Inhalt der Datei »Scores.hst«,
- bei Problemen, die ein Skript betreffen, den Inhalt der betreffenden Skriptdatei.

Siehe auch die Checkliste zur Fehlersuche und die Anleitung zum Posten eines Hamster-Logs (**jeweils enthaltene Links zum Teil veraltet!**), beide von Wilfried Kramer.

Hamster und Events

Allgemein

Events können genutzt werden, um anderen Prozessen (z. B. anderen laufenden Skripten) etwas mitzuteilen. Events können dabei über mehrere Programme genutzt werden, existieren also nicht hamster-intern, sondern windows-weit. Ein Hamsterskript kann so beispielsweise auf ein Event warten, das beispielsweise vom DFÜ-Netzwerk ausgelöst wird oder auch von einer zweiten Hamsterinstanz. Siehe dazu auch »Hamsterskript: Nebenläufige Skripte«.

Ein Prozess kann jeweils nur auf ein Event warten (entweder der Prozess wartet, oder er tut etwas; mehr geht nicht). Verschiedene Skripte stellen dabei verschiedene Prozesse dar, so dass man auf Events am besten in per »runscript()« gestarteten Skripten wartet. Den dazugehörigen Code in Subroutinen zu packen geht nicht!

Warum »While(true)«-Schleifen?

Ein »While(true)« bewirkt, dass die folgende Schleife immer wieder ausgeführt wird (die Bedingung ist immer »wahr«). Anderenfalls würde ein Skript nur einmal auf das entsprechende Event warten und dann nie wieder. So wird nach dem Abarbeiten der nachfolgenden Befehle wieder von vorne gestartet.

Anwendungsbeispiele:

```
# Hauptskript.hsc:

#!hs2

# ⟨...⟩

runscript( "AdminMails.hsc", "", false)
# Das Skript wird durch runscript() in einem eigenen Prozess ausgeführt.

# ⟨weitere Befehle⟩
```

```
# AdminMails.hsc:

#!hs2
while(true)

  # hier wartet der Hamster auf das Ereignis »E-Mail von außen eingegangen«

  EventWait( GetProcessIdentifier + "_mailin", 60000 )

  # Das Event mit dem zusammengesetzten Namen aus der Funktion »GetProcessIdentifier«
  # und dem String »_mailin« wird vom Hamster erzeugt und auf »Ein« gestellt, wenn
  # eine Mail von außen eintrifft, also von einem Server abgeholt wird.
  # Mehr Erklärungen darüber, wie sich der Name hier zusammensetzt, und über die ande-
  # ren hamsterinternen Events kann man unter »Synchronisation durch Events« finden.

  if FileExists( HamMailPath + "admin" )
  # Wir sind gerade nur an Mails für den Admin interessiert.
    MsgBox( "Eine Mail ist für den Admin eingegangen." )
  endif
endwhile

# Durch die While-Schleife wird wieder auf das nächste Ereignis gewartet.
```

Mit folgendem Beispiel geht der Hamster auf die Jagd, sobald eine Onlineverbindung aufgebaut wird. Das Skript wird wie immer aus dem Hauptskript mit »`runscript("WartenAufDFue.hsc", "", false)`« ausgeführt.

```
# WartenAufDfue.hsc:

#!hs2
while(true)

  EventWait( GetProcessIdentifier + "_rasconnected" )

  # Warten, dass das vom Hamster selbst gesetzte Event
  # »GetProcessIdentifier+"_rasconnected"« auf »Ein«, also
  # »Online-Verbindung steht«, gesetzt wird.

  # ⟨Hier nun die Befehle für das Abgleichen der Artikel/Mails
  # hinsetzen, bzw die entsprechende Subroutine aufrufen.⟩
endwhile
```

Eigene Events

Eigene Events kann man dazu nutzen, um mit anderen Hamstern oder anderen Programmen zu »kommunizieren«. Da Events windows-weit zu »sehen« sind, können andere Programme auf sie warten und dann entsprechende Aktionen ausführen.

Eigene Events werden mit »`$handle = EventCreate(⟨name⟩)`« erzeugt und hinterher wieder mit »`Eventclose($handle)`« geschlossen. Die Variable »`$handle`« enthält dabei die Speicheradresse des Events und ist nur zum Schließen des Events wichtig. »`⟨name⟩`« ist dabei der Name des Events, unter dem es systemweit angezeigt wird. Bei der Benennung sollte man sich sicher sein, dass kein anderes Programm den Namen benutzt. Zur Einmaligkeit des Namens kann hier der Mutex-String genommen werden, den man mit »`GetProcessIdentifier`« bekommt. Für die Kommunikation mit einem anderen Programm, wie in dem Beispiel unten, ist das aber nicht gerade nützlich, da dieser für beide Programme unterschiedlich ist.

Mit den Befehlen »`EventSet(⟨eventname⟩)`«, »`EventReset(⟨eventname⟩)`« und »`EventPulse(⟨eventname⟩)`« kann man den Zustand des Events verändern, also ein Event auslösen.

Mit »`EventWait(⟨eventname⟩)`« wird auf den Zustand »Ein« gewartet (auf »Aus« kann man nicht warten).

Warum »If«-Bedingungen?

Sowohl um »`Eventwait()`« als auch die das Event verändernden Befehle sollte man »If«-Bedingungen setzen, die darauf reagieren, wenn das Event noch nicht existiert (d. h. Rückgabewert != 0). Bei einem definierten Timeout bei »`Eventwait()`« sollte man auch auf den Rückgabewert »258« testen, der das anzeigt.

Anwendungsbeispiele:

```
#!hs2
varset( $a, Eventwait( "test", 180000 ))

if( $a != 0)
  if($a = 258 )
    warning( "Event nach 180 Sekunden nicht ausgeloest." )
  else
    warning( "Event nicht gefunden!" )
  endif
else
  warning _
  ( "Event ist auf 'ein' gestellt worden, arbeite entsprechende Befehle ab ..." )
  # weitere Befehle
endif
```

Bei Fällen ohne Timeout reicht folgender Skriptabschnitt aus:

```
#!hs2
if( eventpulse( "Test" ) != 0 )
  warning( "Event nicht gefunden!" )
endif
```

Hat man nun beispielsweise zwei Hamster gestartet (einen als normalen »Pullhamster« und einen als »Archivhamster«), dann kann man letzteren mit folgenden Skripten zum Abgleichen auffordern:

```
# Archivhamster.hsc (läuft auf dem Archivhamster):

#!hs2
var( $handle, $Fehler )
$handle = EventCreate( "abgleichen", True, False, $Fehler )

# Wir machen das Event mit folgenden Eigenschaften auf:
# »true« →  Nach einer Abfrage im Zustand »Ein« wird es automatisch zurückgesetzt.
# »false« →  Das Event bekommt den Zustand »Aus«.
#
# Sollte das Event hier schon existieren, dann wird die von Windows gemeldete
# Fehlermeldung in die Variable »$Fehler« gesetzt. Siehe dazu dann unter
# »Win32-Systemmeldungen«.
# Das Event selber wird aber nicht verändert, so dass dies in den meisten Fällen nicht
# stören sollte.

while(True)
  EventWait( "abgleichen" )
  # Nun wartet der Archivhamster, dass ihn irgendjemand zum Abgleichen auffordert,
  # also das Event »abgleichen« auf »Ein« setzt.

  # ⟨Hier die Befehle zum Abgleichen mit dem normalen »Pullhamster«.⟩
endwhile
# Lässt den Hamster wieder auf das Auslösen des Events warten.

EventClose($handle)
# Eigentlich überflüssig, da das Skript nicht mehr aus der »while(true)«-Schleife
# herauskommt, da die Bedingung immer wahr ist.


# ------------------------------------------
# Pullhamster.hsc (läuft auf dem Pullhamster):

#!hs2

# ⟨Hier die Befehle für das normale Newsabholen etc.⟩

hamwaitidle

if( eventset( "abgleichen" ) != 0 )
  Warning( "Event noch nicht vorhanden! Zweiter Hamster noch nicht gestartet." )
  Warning( "Oeffne Event und setze es auf 'Ein' ..." )
  Eventcreate( "abgleichen", True, True) # Das erste »True« macht ein »Eventreset()«
                                         # im Archivhamster überflüssig.
endif

# Der Hamster wartet, bis er alle Nachrichten etc. abgeholt hat und setzt dann
# das Event »abgleichen« auf »Ein«. Falls es noch nicht existiert hat, wird es
# entsprechend erstellt. Dafür ist die »If«-Schleife wichtig.
# »Eventset()« ist dabei »Eventpulse()« vorzuziehen, da so auch bei späterer Abfrage
# der zweite Hamster weiß, dass er etwas zu tun hat. Mit »Eventpulse()« würde er das
# »überhören«.
# Nun würde der »Archivhamster« mit seinen Abgleichbefehlen anfangen und danach wieder
# in Lauerstellung gehen.
```

Kann ich auf den lokalen Newsserver auch ohne Name und Passwort zugreifen?

Ja, das ist möglich. Im Hamster kann ein Account eingerichtet werden, dessen Einstellungen immer dann verwendet werden, falls sich der Newsreader *nicht* anmeldet. Dieser Account *muss* »nntpdefault« heißen und *muss* mit dem Passwort »*« (ein Stern) eingerichtet werden. Über Fähigkeiten für Mail (siehe Menü »Einstellungen« → »Benutzerverwaltung & Passworte« ⇒ »Lokale Benutzer«) sollte dieser Account aus Sicherheitsgründen nicht verfügen, also weder eine Mailbox besitzen noch Mail senden dürfen.

Zu beachten ist, dass damit jeder, der den Rechner überhaupt erreichen kann, Zugriff auf den Newsbereich des Hamsters erhält. Deshalb muss unbedingt darauf geachtet werden, dass über die Datei »IPAccess.hst« der Zugriff sorgfältig kontrolliert wird. In der Voreinstellung sollte diese Datei gar nicht existieren, so dass dann die Zugriffsmöglichkeit standardmäßig auf das lokale Netzwerk begrenzt ist.

Wieso funktionieren meine Passworte plötzlich nicht mehr?

Die Passworte werden vom Hamster mit dem »Blowfish«-Algorithmus verschlüsselt. Standardmäßig werden dazu noch einige Systemangaben wie Rechnername, Betriebssystem-Version und der Windows-Benutzername verwendet.

Wird der Benutzername geändert, oder neue Hardware/ein neues Betriebssystem installiert, kann Hamster die Passwörter nicht mehr lesen. In dem Fall müssen alle Passwörter neu eingegeben werden. Soll der Hamster dauerhaft mit mehreren Windows-Benutzernamen verwendet werden, muss die Verschlüsselungsstufe herabgesetzt werden. Dazu wählt man im Menü »Einstellungen« → »Benutzerverwaltung & Passworte« ⇒ »Passworte« → »Verschlüsselung« eine niedrigere Verschlüsselungsstufe aus, »Nutzt nur System-Infos« sollte für die meisten Fälle reichen. Danach müssen ***alle*** Passwörter neu eingegeben werden (auch die Benutzer- und Serverpassworte).

Unter Windows NT und Nachfolgern kann der Hamster auch als Dienst installiert werden und somit unabhängig vom Benutzer laufen (hierzu ist ein externes Programm notwendig, empfehlenswert ist »HamSvc« von Jürgen Haible, siehe Linkliste). Auch dann darf der Benutzername nicht zur Verschlüsselung herangezogen werden.

Wird der Hamster unter mehreren Betriebssystemen verwendet (z. B. Win,98/NT), muss die Verschlüsselungsstufe auf »Keine Nutzung von Systeminfos« gestellt werden. Auch danach müssen ***alle*** Passwörter neu eingegeben werden.

Wenn zur Verschlüsselung keine Systeminfos verwendet werden, kann jeder, der die Datei »Password.!!!« auf irgendeinem Wege bekommt, diese einfach verwenden, um z. B. Zugriff auf das POP3-Postfach zu bekommen!

Glossar: POP3

Wie kann ich ältere Artikel nachladen?

Es gibt mehrere Möglichkeiten. Welche zu verwenden ist, hängt von verschiedenen Einstellungen für das Nachladen ab:

- Solltest Du unter »Einstellungen« → »News, Server … « ⇒ »Einstellungen« die Voreinstellung[1] so geändert haben, dass stattdessen die XXX neuesten geladen werden, kannst Du die extra dafür vorgesehene Möglichkeit des Schiebereglers verwenden, der im unteren Bereich der Einstellungen für die einzelnen Newsgruppen zu finden ist unter »Einstellungen« → »News, Server … « ⇒ »Newsgruppen« → ⟨*Newsgruppe auswählen*⟩ → »Einstellungen«. Zum Nachladen älterer Artikel Schieberegler nach links bewegen.

  Zum Nachladen noch fehlender, aber jüngerer Artikel bei neu abonnierten Gruppen müsste der Newspull öfter wiederholt werden. Zusätzlich könnte das Artikellimit pro Newsabruf kurzzeitig höher gesetzt werden; wie hoch maximal, ist jedoch individuell unterschiedlich, da es von der Leistungsfähigkeit des eigenen Rechners und von der Stabilität der Verbindung zum Newsserver abhängig ist (man kann auch die Möglichkeit nutzen, Ladeaufträge zu teilen, siehe ebenfalls unter »Einstellungen« → »News, Server … « ⇒ »Einstellungen«, erweiterte Einstellungen müssen aktiviert sein).

- Artikel, die aufgrund einer Einstellung in der »Scores.hst« ignoriert, aber in das News-Killfile-Protokoll eingetragen wurden, können von dort aus nachgeladen werden. Ihre Message-IDs werden in die für jeden Server separat existierende »GetMIDs.txt« eingetragen, die beim nächsten Newspull abgefragt wird.

- In »Einstellungen« → »Automatische Abläufe« ⇒ »Lade Artikel über ihre MID« können auch direkt Message-IDs von Artikeln eingetragen werden, die beim nächsten Newspull abgerufen werden (es handelt sich hier um ein Front-End für die »GetMIDs.txt«). Wenn ein Ausrufezeichen an den Anfang einer Zeile gesetzt wird (»!<Message-ID>«), wird der entsprechende Artikel auch dann geladen, wenn er bereits in der Newshistory steht.

Zuletzt sei auch noch auf die Möglichkeit hingewiesen, *alle* Artikel einer Gruppe neu zu laden, die auf dem abgefragten Newsserver noch vorhanden sind: Das ist möglich durch das Zurücksetzen einer Gruppe unter »Einstellungen« → »News, Server … « ⇒ »Newsgruppen« → ⟨*Newsgruppe auswählen*⟩ → »Zurücksetzen« (siehe auch »Unterschied zwischen ›Gruppe zurücksetzen‹ und ›Gruppe löschen‹«).

Wieso lässt sich die Gruppe »internal.misc« nicht löschen?

Das ist so pauschal nicht richtig: Der Hamster benötigt eine Standardgruppe, damit er funktioniert, und *diese* ist nicht löschbar. Das ist in der Voreinstellung die Gruppe »internal.misc«, auch wenn in der »hamster.ini«, Sektion »[Setup]«, der dazu gehörende Eintrag »internalgroup.default« fehlt. Sie können diesen Eintrag aber manuell mit dem Verweis auf eine andere

[1] Die 100 ältesten Artikel werden geladen.

bestehende Gruppe eintragen bzw. ändern oder besser die Standardgruppe per Dialog über das Menü »Einstellungen« → »Grundeinstellungen« ⇒ »Interne Gruppen« ändern. Dann ist die andere, neu ausgewählte Gruppe nicht mehr löschbar, aber die Gruppe »internal.misc« kann entfernt werden.

Scheduler-FAQ

> *Wie bringe ich dem Hamster bei, dass er selbständig regelmäßig News und Mails holt und versendet?*

Das folgende Skript veranlasst den Hamster, in regelmäßigen Abständen News und Mail zu holen und zu versenden.

Sie können über das Menü »Skript« → »Skripte und Module verwalten« ⇒ Dialog »Skripte verwalten« → »Neu ... « ein neues Skript anlegen und den kopierten Skriptcode dort einfügen. Das Skript muss noch an Ihre Konfiguration angepasst werden, lesen Sie dazu die Anmerkungen weiter unten nach dem Skript durch. Damit das Skript beim Start des Hamsters mitgestartet wird, gibt es zwei Möglichkeiten: Tragen Sie es entweder als Action ein, siehe dazu unter Menü: »Einstellungen« → »Automatische Abläufe« ⇒ »Aktionen« oder hängen den Skriptnamen beim Start über die Kommandozeile an (bzw. erzeugen Sie eine Verknüpfung zur »`hamster.exe`« und ergänzen Sie den Skriptnamen so, wie im Folgenden zu lesen). Wenn das Skript »`MailNews.hsc`« hieße, sähe das also so aus:

```
hamster.exe MailNews.hsc
```

Zu jedem Befehl finden Sie weitere Informationen hier in der Hilfe-Datei – über den »Index der Hamsterskript-Anweisungen« finden Sie die Seite, wo der jeweilige Befehl erläutert wird.

Weitere Skripte finden Sie im Skriptarchiv von Michael Gebert (siehe Linkliste).

```
### Skriptanfang ###
#!hs2

# Variablendefinition #
var( $Einwahl )
varset( $eingewaehlt, 0 )
varset( $try, 0 )

# Einstellungen #
$Einwahl = 1                    # [Anm. 1]

# Scheduler #
AtAdd( Mail1, "07:00", "19:00", "1111111", 30, 1 )  # [Anm. 2]
AtAdd( Mail2, "07:00", "19:00", "1111111", 15, 1 )
AtAdd( News1, "07:00", "19:00", "1111111", 60, 1 )
AtAdd( News2, "07:00", "19:00", "1111111", 15, 1 )

AtExecute
```

```
quit

# Subroutinen #
sub Mail1
  if( $Einwahl = 1 )
    Einwahlroutine
      if( $try >= 3 )
        return
      endif
  endif
  HamWaitIdle
  HamMessage( 4, 0 )          # [Anm. 5]
  HamMessage( 5, 0 )          # [Anm. 5]

  HamWaitIdle
  HamMailExchange             # [Anm. 3]
  HamWaitIdle
  HamMessage( 4, 1 )          # [Anm. 5]
  HamMessage( 5, 1 )          # [Anm. 5]
  if( $eingewaehlt = 1 )
    HamRasHangup
    $eingewaehlt = 0
  endif
endsub

sub News1
  if( $Einwahl = 1 )
    Einwahlroutine
    if( $try >= 3 )
      return
    endif
  endif

  HamWaitIdle
  HamMessage( 3, 0 )          # [Anm. 6]
  HamNewsJobsClear
  HamNewsJobsPostDef
  HamNewsJobsPullDef
  HamNewsJobsStart

  HamWaitIdle
  HamMessage( 3, 1 )          # [Anm. 6]
  if( $eingewaehlt = 1 )
    HamRasHangup
    $eingewaehlt = 0
  endif
endsub

sub Mail2
  if( HamGetStatus( 3, 7 ) > 0 )
    Mail1
  endif
endsub
```

```
sub News2
  if( HamGetStatus( 3, 5 ) > 0 )
    News1
  endif
endsub

sub Einwahlroutine
  $try = 1
  if( !RasIsConnected )
    $eingewaehlt = 1
    while( HamRasDial("Provider","$1") != 0 )  # [Anm. 4]
      if( $try >= 3 )
        $eingewaehlt = 0
        warning( "Einwahl fehlgeschlagen!" )
        return
      endif
      inc( $try )
      sleep( 10000 )
    endwhile
  endif
endsub
### Skriptende ###
```

Anm. 1: Der Wert der Variablen »$Einwahl« bestimmt, ob der Hamster eine DFÜ-Verbindung aufbauen soll oder nicht. Wenn als Wert eine »1« eingetragen ist, prüft der Hamster, ob schon eine DFÜ-Verbindung besteht, und wenn das nicht der Fall ist, wählt er sich ein (siehe hierzu auch Anm. 4). Falls man z. B. eine Standleitung hat und deshalb keine DFÜ-Verbindung aufbauen muss, setzt man den Wert auf »0«.

Anm. 2: Die erste »AtAdd«-Zeile veranlasst den Hamster, dass er von Montag bis Sonntag, zwischen 7.00 und 19.00 Uhr alle 30 Minuten die Subroutine »Mail1« abarbeiten soll. Falls Sie möchten, dass der Hamster z. B. nur Mails austauscht, setzen Sie vor die letzten 2 »AtAdd«-Zeilen ein »#«, damit der Hamster diese ignoriert (sie werden so »auskommentiert«). Die einzelnen Subroutinen haben folgende Aufgaben:

- »Mail1«: Empfangen und Versenden von E-Mails für alle im Menü »Einstellungen« → »Mail: Server konfigurieren« eingetragenen POP3- und SMTP-Server (siehe hierzu auch Anm. 3).
- »Mail2«: Schaut nach, ob E-Mails zum Versenden vorliegen, und falls das der Fall ist, wird die Subroutine »Mails1« ausgeführt.
- »News1«: Versenden und Empfangen von Artikeln für alle im Menü »Einstellungen« → »News: Server, Gruppen & Ladeaufträge« ⇒ »Newsserver« eingetragenen Newsserver und alle bei diesen abonnierten Gruppen.
- »News2«: Schaut nach, ob Artikel zum Versenden vorliegen, und falls das der Fall ist, wird die Subroutine »News1« ausgeführt.

Glossar: POP3 SMTP

Hinweis: Die im Skript angegebenen Intervalle, mit denen der Hamster den News- und Mailaustausch durchführt, sind vollkommen ausreichend. Da es sich bei News und Mail nicht um Echtzeitkommunikation handelt, sind kürzere Intervallzeiten unnötig, belasten nur übermäßig die News- und Mail-Server

und werden von den Server-Administratoren nicht gerne gesehen, siehe dazu auch »Netzressourcen schonen«.

Glossar: POP3 SMTP

Anm. 3: Um den Befehl »HamMailExchange« verwenden zu können, müssen die POP3- und SMTP-Server im Hamster unter »Einstellungen« → »Mail: Server konfigurieren« ⇒ »POP3-Mailserver« bzw. ⇒ »SMTP-Mailserver« eingetragen sein.

Wenn E-Mails abhängig vom Absender über verschiedene SMTP-Server verschickt werden sollen, muss man den Befehl »HamMailExchange« ersetzen. Löschen Sie die Zeile und setzen stattdessen Folgendes ein (je nach Konfiguration, hier als Beispiel für je ein Benutzerkonto bei `example.com` und `example.net`):

```
HamFetchMail( "pop.example.com" )
HamFetchMail( "pop.example.net" )
HamWaitIdle( 20000 )
HamSendMail( "smtp.example.com", "smtp", "Benutzer1@example\.com" )
HamSendMail( "mail.example.net", "smtp", "Benutzer2@example\.net" )
```

SMTP-AUTH

Falls der SMTP-Server ein Login über SMTP-AUTH benötigt, verwenden Sie anstatt »HamSendMail« den Befehl »HamSendMailAuth«. Für unser Beispiel würden die Zeilen also so aussehen:

```
HamSendMailAuth( "smtp.example.com", "smtp", "$2", "", "Benutzer1@example\.com" )
HamSendMailAuth( "mail.example.net", "smtp", "$3", "", "Benutzer2@example\.net" )
```

Mit den Variablen »$2« und »$3« wird die Anmeldename-Passwort-Kombination übergeben. Diese Variablen sind im Menü »Einstellungen« → »Benutzerverwaltung & Passworte« ⇒ »Passworte« zu definieren.

Hinweis: Damit der Befehl »HamSendMailAuth« funktioniert, müssen die entsprechenden SMTP-Server im Menü »Einstellungen« → »Mail: Server konfigurieren«⇒ »SMTP-Mailserver« auch als »SMTP-AUTH« markiert sein.

Anm. 4: Der Befehl »HamRasDial« baut eine mit »⟨Provider⟩« bezeichnete DFÜ-Verbindung auf und nimmt dazu die unter der Variablen »$1« definierte Benutzername-Passwort-Kombination. Sie müssen »⟨Provider⟩« mit dem Namen ihrer DFÜ-Netzwerkverbindung ersetzen, den sie unter »Arbeitsplatz« → »DFÜ-Netzwerk« (Telefonbuch des DFÜ Netzwerkes) definiert haben (Groß- und Kleinschreibweise beachten). Die Variable »$1« definieren Sie im Menü »Einstellungen« → »Benutzerverwaltung & Passworte« ⇒ »Passworte«.

Anm. 5: Mit den Befehlen »`HamMessage( 4, 0 )`« und »`HamMessage( 5, 0 )`« wird der interne POP3- bzw. SMTP-Server gestoppt, mit »`HamMessage( 4, 1 )`« und »`HamMessage( 5, 1 )`« wieder gestartet. Dies ist sicherer, falls Sie ausgehende Mails vor dem Versenden bzw. eingehende Mails nach dem Abholen überarbeiten lassen. Wenn Sie sicher sind, dass Sie die Abschaltung des internen POP3- bzw. SMTP-Servers nicht benötigen, setzen sie vor jede mit »`# [Anm. 5]`« markierte Zeile ein »#« (sie wird so »auskommentiert«).

Anm. 6: Mit dem Befehl »`HamMessage( 3, 0 )`« wird der interne NNTP-Server gestoppt und mit »`HamMessage( 3, 1 )`« wieder gestartet. Dies ist sicherer, falls Sie ausgehende Artikel vor dem Versenden bzw. eingehende Artikel nach dem Abholen überarbeiten lassen (z. B. von »Korrnews« und »Copyif«). Wenn Sie sicher sind, dass Sie die Abschaltung des internen NNTP-Servers nicht benötigen, setzen Sie vor jede mit »`# [Anm. 6]`« markierte Zeile ein »#« (sie wird so »auskommentiert«).

Glossar: NNTP

Feeden von Newsgruppen

Vorwort

Wer den Hamster als Feeder benutzen möchte, sollte fortgeschrittene Kenntnisse über das NNTP-Protokoll und die Funktionsweise des Usenet besitzen. Ohne diese Kenntnis läuft man Gefahr, in Fettnäpfe zu treten. Und diese sind bei dieser Technologie zur Genüge vorhanden.

Grundvoraussetzungen

Wird der Hamster als Feeder verwendet oder sollen Artikel von einem anderen Server in den Hamster gefeedet werden, so sollte er ausschließlich für diesen Zweck verwendet werden. Die Verwendung von Tools zur Nachbearbeitung von Artikeln ist zu unterlassen. Um den Hamster als lokalen Feeder betreiben zu können, ist es erforderlich, dass für den Hamster ein eigener Domain-Name verwendet wird. Der Rechnername gefolgt vom Domain-Namen (FQDN, z. B. »hamster.deine-domain.example.net«) ist im Menü »Einstellungen« → »Lokale Server« ⇒ »Allgemeines« → »FQDN für spezielle Header, Peering etc.« einzutragen. Des Weiteren sind im Menü »Einstellungen« → »Lokale Server« ⇒ »NNTP« unter dem Reiter »Header anpassen« die Optionen »Message-ID erzeugen, wenn noch keine vorhanden« und »Erzeugen oder ggf. Erweitern des Path-Headers« sowie unter dem Reiter »Zusätzliche Einstellungen« die Option »Neue Artikel direkt in die lokale Datenbank speichern« (auch bekannt als »local injection«) zu aktivieren; die Option »Erzeugen oder ggf. Erweitern des User-Agent-Headers« (unter »Header anpassen«) ist zu deaktivieren. Die Aktivierung des zusätzlichen »X-Trace:«-Headers mit (z. B.) »news.addxhtrace=X-Hamster-Trace« im Abschnitt »[Setup]« der »`hamster.ini`« ist zur Spambekämpfung zu empfehlen.

FQDN

Header

Warnung: Das Feeden von Artikeln ohne korrekten »Path:«-Header oder mit fehlendem oder defektem »Message-ID:«-Header führt zu schweren Störungen im Netz.

Feeden des Hamsters am lokalen Server

Um einem User das Feeden des Hamsters zu gestatten, ist unter »Einstellungen« → »Benutzerverwaltung & Passworte« die entsprechende Einstellung »Benutzer darf News peeren« des jeweiligen lokalen Benutzers oder der gesamten Benutzergruppe zu aktivieren. Mit der Option

»ihave.auth=0«, ebenfalls im Abschnitt »[Setup]« der »`hamster.ini`«, kann die generelle Erlaubnis für alle User, mit dem Hamster zu peeren, erteilt werden (***Achtung, Sicherheitsrisiko!***). Der Hamster kann sowohl mit den NNTP-Befehlen »IHAVE« als auch mit »TAKETHIS« gefeedet werden. Der Hamster speichert im Regelfall am lokalen Server eingehende Artikel im Ausgang des News-Clienten (Verzeichnis »`News.Out`«) ab. Mit der Einstellung »Nur als Feeder verwendet« im Einstellungsdialog der jeweiligen Gruppe unter »Einstellungen« → »News: Server, Gruppen & Ladeaufträge« ⇒ »Newsgruppen« kann dieses jedoch unterbunden werden, so dass die Artikel nur lokal – durch das Aktivieren der oben schon erwähnten Option »Neue Artikel direkt in die lokale Datenbank speichern« – gespeichert werden. In diesem Fall werden die am lokalen Server eingehenden Artikel nicht mehr beim Provider gepostet und können den Hamster nur noch verlassen, wenn dieser selbst einen anderen Server feedet. Weiterhin ist es zu empfehlen, über das Limit für die Anzahl der Zeilen und Bytes eines Artikels in der »`hamster.ini`« nachzudenken. Hierzu sind die Optionen »local.limit.linelen.nntp« und »local.limit.textsize.nntp« im Abschnitt »[Setup]« einzustellen. Ein zu kleiner Wert kann Probleme mit binären Dateianhängen und großen Artikeln verursachen und ein zu großer Wert ermöglicht das Fluten Deines Servers mit Datenmüll.

Glossar: NNTP

Feeden anderer Server durch den Hamster

Zum Feeden eines anderen Servers ist mit dem hs2-Befehl »HamNewsJobsFeed« ein entsprechender Job in die News-Queue einzustellen und mit dem Befehl »HamNewsJobsStart« die Abarbeitung der Job-Queue zu starten. Der zu feedende Server muss im Menü »Einstellungen → »News: Server, Gruppen & Ladeaufträge« ⇒ »Newsserver« eingetragen sein. Gefeedet werden alle Artikel der über »⟨group-select⟩« ausgewählten Gruppe(n) in der Hamsterdatenbank. Es werden keine Artikel aus dem Verzeichnis »`News.out`« auf diese Art und Weise transportiert. Jeder Artikel wird dem Peer genau einmal mittels »IHAVE«-Befehl angeboten. Der Hamster merkt sich, welche Artikel er welchem Server angeboten hat.

Weitere Antworten

> *„ Welche Server darf ich mit HamNewsJobsFeed ansprechen? “*

Jeden »RFC 977«-konformen Server, der bereit ist, mit Dir zu peeren. Der Hamster unterstützt beim Peeren auch »Original Authinfo«, wie in RFC 2980 beschrieben. Du kannst selbstverständlich auch einen zweiten Hamster, z. B. Deinen Archiv-Hamster, als Peer verwenden.

> *„ Brauche ich eine Standleitung? “*

Das hängt von den Anforderungen ab, welche Dein Peering-Partner an dich stellt. Einige Provider sind nicht in der Lage oder bereit, zeitversetzt zu feeden.

> *„ Gibt es Testserver, die ich feeden kann, ohne das Netz zu belästigen? “*

Verwende einen zweiten Hamster oder setze selbst einen zweiten, auf anderer Software beruhenden Server zum Testen auf, wie die im professionellen Umfeld eingesetzten Diablo (`http://www.openusenet.org/diablo/`) oder INN (`http://www.isc.org/software/inn`). Achte darauf, dass der Testserver einen anderen Domain-Namen verwendet als der zu testende Server.

> *„ Welche Provider sind bereit, meinen Hamster zu feeden? “*

Sprich mit Deinem lokalen Provider. Viele Provider stellen recht unterschiedliche Anforderungen an Peering-Partner. Meistens wird erwartet, dass Du eine feste IP-Adresse besitzt und eine Standleitung betreibst, da ein zeitversetztes Feeden durch sie nicht möglich ist.

> *„ Gibt es Clean Feed oder andere Filtermöglichkeiten? “*

Bisher noch nicht. Aber Du darfst Dir gerne Gedanken machen, wie so etwas mit einem hs2-Skript zu erledigen ist.

Kann der Hamster die Mailverteilung in einem Firmennetz erledigen?

Ja. Der Hamster kann externe und interne Mails für verschiedene Benutzer in kleineren und mittleren Teams bearbeiten. Wir empfehlen, die Installation zunächst mit Testaccounts durchzuführen (parallel zum bisherigen System). Für den Produktionsbetrieb ist fundiertes Wissen über den Hamster, seine Skript- und Filtermöglichkeiten und über die Netzwerkabläufe allgemein erforderlich. Der Einsatz von Entwicklerversionen (wie Hamster Future und Betaversionen) sollte unterbleiben.

Sie sollten sich auch über die rechtlichen Aspekte informieren, da Sie als Administrator potentiell Zugriff auf alle E-Mails haben.

Konvertieren von Mailinglisten in Newsgruppen

Diese Konfiguration benutzt die Skriptsprache des Hamsters für den Abruf der Mails sowie Korrnews (Download siehe Linkliste) für zusätzliche Schritte. Die Mails für die Liste werden zuerst einmal in einen speziellen lokalen Mail-Account namens »Liste« umgeleitet. Anschließend werden sie dort von Korrnews überarbeitet, wobei überflüssige Header gelöscht werden. Fehler von »eGroups«[1] (die hier nur als Beispiel dienen und über die auch die Hamster-Mailinglisten laufen) werden korrigiert, und die Werbung entsorgt. Diese überarbeiteten Mails holt sich der Hamster wieder selber ab, wobei sie dann wirklich umgeleitet werden.

Glossar: Header

[1] Inzwischen in »Yahoo! Groups« aufgegangen.

Aber nun geht es los:

1. Lokalen Benutzer anlegen mit dem Namen »Liste«. Dieser Account darf von keinem Client gepollt werden.

2. Folgende zwei Abschnitte in der »`Mailfilt.hst`« anlegen:

   ```
   [EGroups]
   # filter-rules for all mails fetched with identifier "EGroups"
   add(liste)  Reply-To: "usehamsternet@"
   add(liste)  Reply-To: "dafa-treff@"

   [EGListe]
   postto(hamster.de.usehamsternet) Reply-To: "usehamsternet@"
   postto(dafa.treff)               Reply-To: "dafa-treff@"
   ```

   Wie man sieht, werden hier zwei Listen bezogen. Hier ist ganz besonders zu bemerken, dass in beiden Abschnitten die Filterkriterien identisch sind. Hier ist es jeweils das »Reply-To:«, das kann im konkreten Fall aber auch das »Subject:« oder etwas anderes sein. Natürlich empfiehlt es sich, einen Account nur für die Mailinglisten zu verwenden. Aber absolut notwendig ist das nicht.

   Übrigens setzt der Hamster in den Postings dabei automatisch das Headerfeld »FollowUp-To: poster«.

   Glossar: Header

3. Für Korrnews eine spezielle Datei erstellen, und als »`EGroupsHeader.def`« speichern.

   ```
   Delete Header Received
   Delete Header Newsgroups
   Delete Header Return-Path
   Delete Header Delivered-To
   Delete Header X-eGroups-Return
   Delete Header X-Apparently-To
   Delete Header X-From_:
   Delete Header X-Hamster-To
   Delete Header Mailing-List
   Delete Header Precedence
   Delete Header List-Unsubscribe

   If not Header(X-eGroups-From) is Empty
     Set Header X-EG-From: %Header(From)%
     Set Header From: %Header(X-eGroups-From)%
     Delete Header X-eGroups-From
   endif
   ```

```
If Header(References) is Empty and not Header(In-Reply-To) is Empty
  Set Header References: \%Header(References)\% \%Header(In-Reply-To)\%
endif

Delete between last "^-+ eGroups Sponsor -+~-~>$","^-+_->$",true
```

Glossar: Header

Diese Angaben löschen eine Menge überflüssiger Header, zaubern, wenn nötig, den originalen »From:«-Header des Absenders wieder hervor, erzeugen References aus einem vorhandenen »In-Reply-To:« und löschen abschließend die Werbung heraus.

Dazu einen korrespondierenden Abschnitt »EGroups« in Korrnews erstellen, der diese Datei verwendet. Außerdem den Speicherort für den Account »Liste« in die »`korrnews.ini`« eintragen:

```
[EGroups]
MessagePath=..\Mails\Liste
Header.def=EGroupsHeader.def
```

4. Im Skript mittels »HamFetchMail« den externen E-Mail-Account abrufen, in dem die Beiträge der Mailingliste landen:

   ```
   hamfetchmail ( "pop-server.example.net", "pop3", _
                  "User", "Pass", "Liste", "EGroups" )
   ```

   Die Mails werden durch den obigen ersten Filter aus der »`MailFilt.hst`« im Account »Liste« abgelegt.

5. Nach allen weiteren Aktionen kann die Verbindung zum Provider getrennt werden, der Rest läuft lokal ab.

   ```
   hamwaitidle
   Execute ( "KorrNews\KorrNews.exe EGroups", HamPath,0,true )
   ```

   Dieser Abschnitt wartet, bis Hamster zur Ruhe gekommen ist. Dadurch sind garantiert alle interessanten Mails getrennt im Account »Liste« verfügbar. Die Verbesserungen werden vorgenommen, wie oben angegeben. Es fehlt nur noch der Import in die lokale NG.

6. Der nächste Abschnitt erledigt das. Der Hamster ruft die Mails aus »Liste« vom Mailserver ab, der er selber ist. Das läuft inzwischen unter »autoerotisch«.[1] Jetzt trifft das zweite Filterkriterium aus »`MailFilt.hst`« zu, so dass die Mails importiert werden.

   ```
   hamwaitidle
   hamfetchmail ( "localhost", "pop3", "liste", "liste", "admin", "EGListe" )
   hamwaitidle
   ```

[1] *Denn:* Er POPpt sich selbst. ;-)

Hier ist noch etwas verbesserungswürdig: »postto(. . .)« beachtet die History nicht. Deshalb können Beiträge mehrfach erscheinen, wenn sie entsprechend oft abgerufen werden. Dort sollte noch eine Anweisung hinein, die die einzelnen Dateien liest, und prüft, ob die MID schon in der History steht. Ist das der Fall, so soll die Datei gelöscht werden. Erst anschließend wird der Rest importiert.

MID = Message-ID

Vorteile: Falls wieder einmal ein allgemein zugänglicher Server so eine lokale Gruppe anbietet, so kann unbesorgt erst einmal von dort abgerufen werden. Die spezielle Maßnahme vermeidet dann Dubletten in der Gruppe. Es sei denn, die Mail wurde bei »eGroups« ohne MID eingeliefert. Aber das wird immer unwahrscheinlicher, je mehr Leser direkt mit dem Newsreader antworten.

Ursprüngliche Version von Wilfried Kramer (online unter `http://soscha.de/hamster/#ML2NG`), geändert durch Steffen Hoffmann

Zwei Hamster, geht das?

(*Vgl. auch* »Wie richte ich einen Archivhamster ein?«)

Aber ja! Hier ein Beispiel:

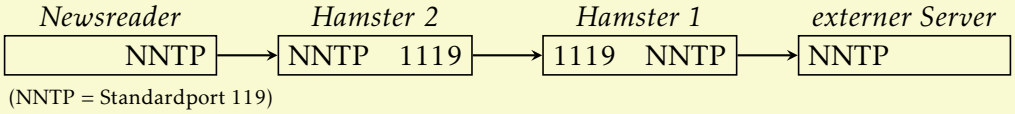
Der Hamster 1 ist der, welcher ans Netz geht und alles saugt. Gleichzeitig dient er als Newsarchiv. Dann gibt es noch den Hamster 2. Der zieht sich die News von Hamster 1 und behält sie nur eine Woche in seinen Backen.

Details der Einstellungen zu den zwei Hamstern:

- Hamster 1: lokaler NNTP-Port auf 1119.
- In Hamster 2 beim externen Server (hier: Hamster 1 auf »localhost«) ebenfalls Port 1119.

Glossar: NNTP Port

Das war's schon.



(NNTP = Standardport 119)

Warum das ganze? Erst einmal zum Rumexperimentieren mit dem Hamster selber (Optionen, Skripte etc. testen). Dann auch noch aus Performancegründen: Der hier verwendete Newsreader »Xnews« braucht bei vielen Artikeln deutlich länger zum Laden.

Ich kann im LAN nicht auf den Hamster zugreifen!

Wenn von einem anderen Rechner auf den Hamster zugegriffen werden soll, dann muss in der Clientanwendung (Newsreader, Mailprogramm) die Adresse des Rechners, auf dem der Hamster läuft, angegeben werden. Oft wird hier der Fehler gemacht, trotzdem auch dann die 127.0.0.1=»localhost« anzugeben.

Hierzu kann man auch einen Eintrag in der »hosts«-Datei des Clientrechners (im Windows-Verzeichnis oder dem Unterordner »\WINDOWS\system32\drivers\etc\« nach ihr suchen) machen, wenn der Hamster sich beispielsweise auf dem Rechner mit der IP-Adresse »192.168.0.1« befindet:

```
192.168.0.1    hamster
```

Dann braucht man z. B. im Newsreader nur »hamster« als Server anzugeben.

Im Hamster muss unter »Einstellungen« → »Lokale Server« ebenfalls für jeden Server, auf den man per LAN zugreifen will, die IP des Rechners »freigegeben« werden (jeweils unter dem Punkt »An folgende IP binden«). Hier gibt es drei Möglichkeiten:

- »127.0.0.1« bedeutet, dass man nur am gleichen Rechner auf den Hamster zugreifen kann (also nicht vom LAN aus). Dies ist die Standard-Einstellung.
- »⟨IP des Hamster-Rechners⟩« – z. B. 192.168.0.1 – bedeutet, alle Rechner, die auf diese IP im LAN zugreifen können, können die Server des Hamsters benutzen. Dies sollte die Einstellung für Hamster im LAN sein.
- »0.0.0.0« bedeutet, auf den Hamster kann von überall zugegriffen werden. Diese Einstellung ist nicht in der Vorauswahl zu sehen, sondern muss von Hand eingegeben werden. *Wer sie benutzen möchte, sollte sich der Risiken bewusst sein und über die Sicherheitsaspekte informieren!*

Zu beachten sind auch die IP-Adressen-Bereiche. In der Grundeinstellung kann der Hamster neben dem »localhost« die Adressen **192.168.0.0–192.168.255.255** verwalten. Wenn nur diese im LAN verwendet werden, sollte es jetzt schon gehen.

Werden andere IP-Adressen verwendet, müssen diese erst in der Datei »IPAccess.hst« freigeschaltet werden.

Wie kann ich bei einem Provider mehrere Postfächer abfragen?

Glossar: POP3 Port

Im Hamster kann man mehrere POP3-Server mit gleichem Namen anlegen, indem man unterschiedliche Aliasse ergänzt. Das Format für den Eintrag eines Servers ist folgender:

```
⟨Servername⟩[/Alias][,Port]
```

Der Alias steht für beliebigen Text, z. B. eine Numerierung oder Accountzuordnung:

```
pop.example.com/1
pop.example.com/2
pop.example.net/jane-doe
pop.example.net/john-doe
```

Zu jedem Server kann man dann seine eigenen Einstellungen (Benutzer/Passwort etc.) einstellen. Im Skript trägt man dann z. B. »HamFetchMail("pop.example.com/2")« ein.

Eine andere Lösung heißt: Alles im Skript behandeln und die Filtermöglichkeiten nutzen (Datei »MailFilt.hst«).

Der Befehl hierzu (Hamsterskript, hs2) ist »HamFetchMail«.

Beispiele:

```
HamFetchMail( "pop.example.org","pop3","$1","","hamster","listen" ) # Hamster-Listen
HamFetchMail( "pop.example.org","pop3","$2","","post","news" ) # News
```

Im Hamster braucht man dann keine POP-Server zu definieren (es ist aber aus anderen Gründen sinnvoll). Die verschiedenen Einstellungen erfolgen über die Variablen »$1«/»$2«, diese können im Hamster über das Menü »Einstellungen« → »Benutzerverwaltung & Passworte« ⇒ »Passworte« jeweils mit eigenem User und Passwort belegt werden.

Über den Befehl »HamFetchMail« können auch noch weitere Einstellungen vorgenommen werden, siehe dazu seine Befehlsdokumentation.

Hamster und FQDN

FQDN steht für »Fully Qualified Domain Name« und stellt den Idealfall der Namensauflösung in »TCP/IP«-basierenden Netzwerken dar. Per definitionem bezeichnet er exakt einen Rechner im Netzwerk. FQDN werden beispielsweise von VeriSign oder auch von DENIC vergeben (auch indirekt z. B. über Webhoster). Hier wollen wir uns auf den für den Hamster und das Usenet relevanten Teil der Definition beschränken.

Glossar: TCP

Damit ein Newsserver Newspostings verwalten kann, verwendet er so genannte Message-IDs. Diese müssen per definitionem im gesamten Usenet für den Zeitraum von mindestens zwei Jahren eindeutig sein, da sonst so unschöne Effekte wie überschriebene Artikel etc. entstehen könnten. Diese Message-IDs können auf unterschiedliche Weise erzeugt werden. In der überwiegenden Mehrzahl der Fälle übernimmt der erste empfangende Newsserver die Erzeugung der Message-IDs. Das macht er immer dann automatisch, wenn er eine Message ohne Message-ID bekommt, was viele Newsclients standardmäßig so handhaben.

Aber auch Newsclients können teilweise Message-IDs erzeugen. Jedoch ist längst nicht gewährleistet, dass diese das immer richtig machen (bekanntes Beispiel für fehlerhafte Message-ID: »Xnews« in der Voreinstellung). Eine kurze Recherche oder eine Headeranfrage in einer Newsgroup wie »de.newusers.infos« wäre also hilfreich, bevor man auf die sonstige Usenet-Gemeinde losgeht.

Header

Was gibt es nun für Möglichkeiten, die Eindeutigkeit der Message-ID zu garantieren? Richtig: Der FQDN kommt genau hier ins Spiel. Wie eingangs geschrieben, wird der FQDN zunächst von VeriSign, DENIC etc. vergeben. Nehmen wir als Beispiel »fqdn.example.net«.

Der Teil vor dem »fqdn.example.net«, also im Falle einer Webseite meist das »www«,[1] kennzeichnet den Rechner, der exakt adressiert werden soll und kann vom Besitzer des FQDN frei definiert werden. Für unseren Newsclient/-server bedeutet das: Wenn ich Verwechslungen ausschließen kann, kann ich jeden beliebigen Vorspann (z. B. »news«) vor meinen FQDN packen und den daraus resultierenden Namen für meine Newspostings verwenden. Ich trage also »news.fqdn.example.net« in das entsprechende Konfigurationsfeld meines Hamsters ein und – et voilà – mein Hamster erzeugt gültige eindeutige Message-IDs.

Alternativ: Falls man bei DENIC keine eigene Domäne reservieren kann oder möchte, kann man kostenlose Domain-Namen z. B. von einem DynDNS-Provider oder unter `http://www.my-fqdn.de/` (dort auch mehr Informationen) beziehen. Auch die Usenetprovider bieten oft diese Möglichkeit (häufig ausschließlich für News erlaubt!).

Hamster und Sicherheit

Die perfekte Sicherheit vor destruktiver Intelligenz gibt es in Netzwerken nicht. Völlige Sicherheit ist weder mit technischen noch mit konzeptionellen Mitteln auf irgendeinem hinreichend komplexen System zu erreichen.

Es ist aber ein mehr oder minder starker Schutz möglich. Bevor man mit der technischen Realisierung eines Schutzkonzeptes beginnt, muss man sich allerdings umfassend über das Thema informieren.

Die reine Installation einer sogenannten »Personal Firewall« bringt keinerlei Zugewinn an Schutz, weil sie suggeriert, dass der Benutzer keinerlei Kenntnisse braucht. Wer nicht versteht, was ein Programm einem mitteilen möchte, braucht dieses nicht. Es ist im Gegenteil so, dass sogenannte »Personal Firewalls« einem unerfahrenem Anwender mehr Probleme machen als dass sie ihn schützen.

Einen Schutz kann nur ein den jeweiligen Erfordernissen angepasstes Konzept bieten. In diesem Konzept müssen das jeweilige Betriebssystem und alle verwendete Hard- und Software mit einbezogen werden. Zu diesem Konzept gehören auch das Wissen des Betreibers um Gefahren und Risiken des Internets und den sicheren Umgang damit.

Eine besondere Beachtung verdienen die lokalen Server des Hamsters. Ein Schutz der lokalen Server ist notwendig, damit fremde User nicht unbefugt über den Hamster Artikel oder News posten und Daten ausspähen können.

Die lokalen Server verwenden in der Standardkonfiguration die TCP-Ports 23 für Telnet (ReCo-Server), 25 für SMTP, 110 für POP3, 143 für IMAP und 119 für NNTP. Diese Ports sind in der Voreinstellung an das lokale Netzwerkinterface (NIC) mit der IP-Adresse 127.0.0.1 (localhost/loopback-Adapter) gebunden. Diese Voreinstellung lässt sich über das Einstellungsmenü jedes lokalen Servers oder in der Datei »`hamster.ini`« im Abschnitt »[Setup]« ändern:

Glossar:
TCP
ReCo
SMTP
POP3
IMAP
NNTP

[1] Das ist stark vereinfacht, denn gerade bei Webadressen gibt es sehr oft *keine* 1 : 1-Übereinstimmung.

```
local.nntp.serverbind=127.0.0.1
local.pop3.serverbind=127.0.0.1
local.smtp.serverbind=127.0.0.1
local.imap.serverbind=127.0.0.1
local.reco.serverbind=127.0.0.1
```

Diese Einstellungen sollten allerdings nur, wenn unbedingt nötig, geändert werden. Mit der IP-Adresse 0.0.0.0 kann der Hamster an alle NIC des Rechners gebunden werden. *Aber Vorsicht:* Damit wird eventuell auch ein DoS-Angriff auf den Hamster möglich!

Um die lokalen Server zu schützen, weist der Hamster alle Zugriffe auf alle IP-Adressen ab, ausgenommen die des »localhost« mit der IP-Adresse 127.0.0.1 und die des für private Netze reservierten Bereiches 192.168.0.0–192.168.255.255. Diese Standardkonfiguration kann aufgehoben und manuell konfiguriert werden, indem im Hauptverzeichnis des Hamsters eine Datei mit dem Namen »`IPAccess.hst`« erstellt wird. Zum Aufbau und Format dieser lese man den entsprechenden Abschnitt der Hilfe.

Eine zweite Schutzstufe gegen die missbräuchliche Verwendung des Hamsters bietet der Passwortschutz der lokalen Server. Auch wenn der Hamster nur auf Einzelplatz-Systemen verwendet wird, sollten Sie diesen Schutz unbedingt aktivieren. Die News- und Mailreader sind in der Regel für die Verwendung des Passwort-Schutzes ausgelegt. Lediglich einige Mailreader beherrschen keine Passwortschutzverfahren für den SMTP-Server. Bei solchen Readern sollte man den Passwortschutz des POP3-Servers für den SMTP-Server mitbenutzen. Dieses Verfahren nennt sich »SMTP after POP3« und kann im Setup-Menü des lokalen SMTP-Servers eingeschaltet werden.

Glossar: SMTP, POP3

Die Passwörter der lokalen Server werden in der Standardkonfiguration mit dem »Blowfish«-Algorithmus gegen Diebstahl geschützt. Dieser Schutz sollte, wenn irgend möglich, nicht abgeschaltet werden. Dieses Verfahren bietet keinen völligen Schutz gegen Diebstahl der Passwörter, erschwert aber den Zugriff auf die Passwörter des Hamsters erheblich, wenn die Passwort-Datei übers Internet entwendet wird. Bei der Wahl der Passwörter sollten keine leicht zu erratenden Namen verwendet werden.

Als dritte Schutzstufe sollten die lokalen Server, wenn möglich, vor Beginn einer Online-Session beendet werden. Die lokalen Server lassen sich im Menü »Einstellungen« → »Lokale Server« ausschalten. Per Skript ist dieses mit den hs2-Befehlen

```
HamNNTPServer( 0 )
HamSMTPServer( 0 )
HamPOP3Server( 0 )
HamIMAPServer( 0 )
HamReCoServer( 0 )
```

NNTP, IMAP, ReCo

möglich. Sehen Sie sich folgende Skript-Vorschläge an. Diese Skripte können entweder in die eigenen Skripte integriert werden oder man legt sie im Hamsterskript-Verzeichnis ab und ruft sie dann bei Bedarf aus dem Menü heraus auf.

```
# Skript für Verbindungsaufbau:
#!hs2

# Bestehen noch Verbindungen von Clients?
if(HamGetStatus(2, 1) + HamGetStatus(2, 2) + HamGetStatus(2, 3) + _
    HamGetStatus(2, 4) + HamGetStatus(2, 5) > 0)
  error("Die Server-Sockets konnten nicht geschlossen werden, " + _
        "weil noch aktive Verbindungen zu Clients bestehen.")
  quit
endif

# Server-Sockets schließen
HamNNTPServer( 0 )
HamSMTPServer( 0 )
HamPOP3Server( 0 )
HamIMAPServer( 0 )
HamReCoServer( 0 )

# Läuft noch ein Server?
if(HamGetStatus(1, 1) + HamGetStatus(1, 2) + HamGetStatus(1, 3) + _
    HamGetStatus(1, 4) + HamGetStatus(1, 5) > 0)
  error("Mindestens ein Server-Socket konnte nicht geschlossen werden.")
  quit
endif

# Internetverbindung aufbauen
# Nur notwendig bei Telefon-Einwahl
HamRasDial ("Verbindungsname")
quit

## Weitere Aktionen ##

# Skript für Verbindungsabbau:
#!hs2

# Warten, bis der Hamster fertig ist
HamWaitIdle

# Internetverbindung trennen
# Nur notwendig bei Telefoneinwahl
HamRasHangup

# Server wieder starten
# Unbenutzte Server auskommentieren
HamNNTPServer( 1 )
HamSMTPServer( 1 )
HamPOP3Server( 1 )
HamIMAPServer( 1 )
HamReCoServer( 1 )

quit
```

Anmeldepasswörter für externe Server werden in der Regel ungeschützt im Klartext übertragen. Aus Sicherheitsgründen sollte Providern der Vorzug gegeben werden, die sichere Verbindungen via SSL/TLS und/oder bei POP3 wenigstens APOP unterstützen.

Glossar:
SSL
TLS
POP3

Netzressourcen schonen

> *Warum soll ich mit dem Hamster nicht einfach jede Minute Mail und News abru-“*
> *„fen?*

Auf diese Art und Weise verschwendest Du gleich mehrfach kostbare Ressourcen. Erst einmal wäre da die kostbare Netzbandbreite. Ungeachtet der Tatsache, dass immer neue Leitungen gelegt werden, ist sie immer noch begrenzt und wird es auch bleiben. Erschwerend kommt hinzu, dass nicht jeder Server, den Du benutzt, mit einer sehr guten Anbindung gesegnet ist. Selbst wenn doch, bleibt sie aber endlich. Eine andere Sache, die vielen nicht klar ist: Bandbreite und Traffic kosten den Betreiber richtig Geld. Je weniger er davon hat (im doppelten Wortsinne), desto eher wird er aufgeben. Wenn er gar kein Geld mit seinem Dienst verdient, mag das noch eher geschehen. Wenn Du nun – sinnlos – Bandbreite verbrauchst, kann sie von anderen Netzteilnehmern nicht genutzt werden. Das Internet funktioniert nur, wenn die Teilnehmer ein Mindestmaß an Vernunft zeigen – auch wenn die eigene Flatrate einem das Gegenteil suggerieren mag. Vergiss nicht: Der andere – das bist für fast alle Teilnehmer Du.

Eine andere, viel wichtigere Ressource ist die Auslastung bzw. Überlastung von Servern. Du kennst sicherlich aus dem Web das Phänomen, dass Server einfach nicht erreichbar sind. Das kann daran liegen, dass diese überlastet sind, weil zu viele gleichzeitig darauf zugreifen wollen. Du (und andere) bist dann der Gekniffene. Wenn Du in zu kurzen Intervallen pullst, stellst Du für Dein Gegenüber eine Dauerlast dar. Du alleine wirst in der Regel von Deinem Server verkraftet werden können. Aber denke daran, dass Du nicht alleine bist. Wenn sich zu viele daneben benehmen, ist das Ende der Fahnenstange sehr schnell erreicht. Besonders schlecht ist es, wenn Du auf diese Weise kostenlose Dienste über Gebühr belastest. Es wäre tragisch, wenn Hamster-User dazu beitragen würden, den Betreibern das Leben so schwer zu machen, dass sie es schließlich aufgeben. Dass dieses kein Geschwafel ist, kann man hier nachlesen:

> Ein Auszug aus `http://groups.google.com/groups?as_umsgid=serverwatch-24.04.2001@hsr.hda.nethamster.de`:
>
> »*da in letzter Zeit der Missbrauch(*) von wuff.inx.de (semi-offener Server für hamster.* über generische User/Pass-Kombination) in einem Maße zugenommen hat, daß die Menge von Beschwerden und Hinweisen, die ich schreiben muss, sowohl den zeitlichen als auch den nervlichen Rahmen dessen deutlich sprengt, was ich bereit bin, für eine Dienstleistung, die ich anderen unentgeltlich zur Verfügung stelle, aufzuwenden, werde ich den semi-offenen Zugang, den wuff.inx.de bisher bereitgestellt hat, in den nächsten Tagen schliessen.*
>
> *[...]*
>
> *(*) Es scheint irgendwie modern zu werden, einen Server mit x parallelen Verbindungen 24/7 alle 30 Sekunden oder alle Minute automatisiert und unbeaufsichtigt zu connecten,*

das nimmt seit einiger Zeit unverhältnismäßig stark zu. Wie eine Horde durchgeknallter Blutegel.«

Die gute Nachricht für Dich: Netzschonendes und professionelles Verhalten ist gleichzeitig auch für Dich vernünftig.

„ Wie saugen denn nun die Profis? “

Ein paar Tipps, ohne Anspruch auf Vollständigkeit:

- Lade erst, kurz bevor Du lesen willst.
- Es ist eigentlich völlig egal, wann Du einen bestimmten Artikel pullst, solange es geschieht, bevor Du ihn lesen willst. Davon, dass Du ihn schon lange vorher besorgst, hast Du genau nichts. Es ist also immer am vernünftigsten, wenn Du die Artikel kurz vor dem Lesen lädst. So hast Du die gerade aktuellsten und die älteren Artikel und vermeidest gleichzeitig unnütze Abfragen. Für Mail gilt das gleiche. Willst Du Instant Messaging betreiben, ist Mail nicht der richtige Weg.
- Wenn Du in Intervallen lädst, frage Dich, warum Du es tust.

  Wenn Du so viel lädst, dass Du es auf einen Schlag nicht schaffst, wäre ein Feed für Dich vielleicht die bessere Lösung. Du kannst damit u. U. bares Geld sparen.
- Wenn ein Feed nicht in Frage kommt: Frage Dich, ob die eine oder andere Abholung nicht überflüssig ist. Spätnachts? Ist es ganztägig wirklich sinnvoll? Kann man die Intervalle erweitern? Konsultiere zur Analyse das Hamster-Log.

„ Wie kann ich ansonsten noch Traffic sparen? “

Das geht schon beim Schreiben los. :-)

- Erfahrene Benutzer vermeiden unnötigen Ballast: HTML-Mail kann nicht von jedem gelesen werden, viele wollen sie auch gar nicht lesen. Eine Mail wird locker 4-mal so lang wie nötig. Das gilt noch mehr für Bildchen und dergleichen. Das wichtige ist der Text und nicht das Drumherum.
- Dasselbe gilt für umfangreiche Attachments: Es ist nicht wirklich sinnvoll, umfangreiche Dateien per Mail zu versenden. Das Mail-Protokoll SMTP schickt Deine Mail von Server zu Server, bevor sie ankommt und obendrein muss sie der Empfänger noch mal laden. Außerdem werden Binär-Dateien (das umfasst auch Word-Dokumente, PDF, Bilder . . .) codiert, damit sie auch unverfälscht ankommen, was die Mail noch mal um einen Faktor von ungefähr 1,3–1,5 aufbläht. Wenn man noch dazu nimmt, dass die meisten Mailserver eher langsam sind, ist es einsichtig: Upload auf einen Speicherplatz und Angabe der Adresse per Mail ist für alle Beteiligten besser.

Glossar: SMTP

- Apropos Attachments: Meide Binär-Newsgruppen wie der Teufel das Weihwasser. :-)

  Der Hamster ist denkbar ungeeignet, um Binär-Gruppen (i. d. R. Gruppen, die im Namen »binaries« führen) zu abonnieren. Zum einen, weil der Hamster grundsätzlich alle Artikel besorgt, zum anderen, weil er das regelmäßig tut. Binärgruppen sind aber auch ansonsten unsinnig. Das Usenet ist technisch ein Verbund von Servern, die ihre Daten weltweit replizieren, damit User auf »ihrem« Newsserver die Postings von anderen aus der ganzen Welt lesen können. Ein Posting wird also vielfach hin und her geschickt und gespeichert (Attachments mit einem ähnlichen aufgeblähten Format wie bei Mail!). Die Praxis zeigt, dass 97 % der Netzlast im Usenet allein durch Binary-Gruppen verursacht wird. Selbst in `de.All` kommen 50 % des Traffics aus einer Gruppe (»de.alt.dateien.weibsbilder«). Da nimmt es nicht wunder, dass es Bestrebungen gibt, Binärgruppen ganz abzuschaffen. Für Dich heißt das also: Benutze bitte andere Wege, um Dateien zu verteilen.

> *„ Was gibt es denn für andere Wege, um Dateien zu verteilen? “*

Am geeignetsten dürfte i. d. R. der erwähnte Upload auf einen passwortgeschützen FTP-Speicherplatz (oder der Upload einer passwortgeschützten Datei, wenn nur öffentlich zugänglicher Webspace zur Verfügung steht) und eine Bekanntmachung per Mail sein. Alternativ kann man auch (kurzzeitig) einen eigenen FTP- oder Webserver zu Hause betreiben, auf den dann die Empfänger zugreifen können. Auch eine Möglichkeit ist Instant-Messaging-Software wie ICQ, die einen direkten Datenaustausch ermöglicht, oder ein IRC-DCC-Transfer.

Die beiden letzteren Möglichkeiten sind allerdings mit besonderen Sicherheitsrisiken behaftet, da man auf dem eigenen Rechner einen Server betreibt. Über diese sollte man sich vorher informieren. Näheres dazu findet man auch unter dem Eintrag »Hamster und Sicherheit«.

E-Mails umleiten

Für die Aufteilung von Mails auf die unterschiedlichen Mailboxen im Hamster gibt es mehrere Möglichkeiten:

- Über das Menü »Einstellungen« → »Mail: Server konfigurieren« ⇒ »POP3-Mailserver« → »⟨POP3-Server⟩« → »Einstellungen« ⇒ »Standardbenutzer« unter »Ankommende Mails folgendem Benutzer zuordnen« den gewünschten Empfänger auswählen oder entsprechenden »⟨destuser⟩«-Parameter im Skriptbefehl »HamFetchMail« verwenden.

  Glossar: POP3

- Im oben genannten Menü unter »Spezieller Filterabschnitt … « oder im »⟨filter⟩«-Parameter in »HamFetchMail« einen speziellen Filterabschnitt in der »`MailFilt.hst`« angeben.
- Im allgemeinen Abschnitt »[*]« in der Filterdatei »`MailFilt.hst`« eine Umleitungs-Anweisung »(addaccounts)« auf einen lokalen Benutzer anlegen.

- Mailrouter mit Hilfe der Datei »MAlias.hst«.

Beachte, dass die drei ersten Umleitungsvarianten den Empfänger auf den Mailboxnamen ändern und erst dann an den Mailrouter übergeben. Es ist bei vom Remote-Server ankommenden E-Mails nicht möglich, diese mit dem Mail-Alias-Mechanismus direkt vom »To:«-Header der E-Mail ausgehend auf eine alternative E-Mail-Adresse eines anderen Servers umzuleiten. Aus Sicherheitsgründen muss immer eine Mailbox im Hamster dahinterstehen, auch wenn es nur eine virtuelle Mailbox ist. Dieses ist notwendig, um eventuell bei der Weiterleitung auftretende Fehler und sich daraus ergebende Fehlermitteilungen (»Bounce Messages«) auffangen zu können.

Glossar: Header

Die via POP3 einkommende Mail wird immer erst einer lokalen Mailbox zugeordnet – im Zweifelsfall ist das der Account »admin« – und dann erst dem Mailrouter übergeben. Das verhindert, dass ankommende E-Mails mit nicht zuzuordnenden Accounts durchrutschen und im »Mail.Out«-Verzeichnis gespeichert werden. Diese Sperre ist notwendig, damit der Hamster nicht als Relais für Spam missbraucht werden kann. Wer also eine externe Weiterleitung für via POP3 einkommende Mails möchte, muss für diese einen (virtuellen) Account anlegen, die per POP3 einkommenden Mails dann über eine der oben genannten Varianten diesem Account zuordnen und von diesem Account aus dann die E-Mails auf einen externen oder internen Account weiterleiten. Das unten stehende Beispiel demonstriert die Vorgehensweise.

POP3

Für am lokalen SMTP-Server einkommende E-Mails ist es aber durchaus möglich, E-Mails direkt von E-Mail-Adresse zu E-Mail-Adresse umzuleiten.

SMTP

Die Log-Daten sollten in der »DETAIL«-Einstellung für die Fehlersuche hinreichend auskunftsfreudig sein. Um dabei nicht von zu vielen Daten erschlagen zu werden, kann man im Hamster-Hauptfenster einen einzelnen Thread auch separat auswählen und betrachten.

Beispiele:

- E-Mails, welche an »Hoppel@hoppel.example.org« adressiert sind und vom POP3-Clienten des Hamsters beim Remote-Server abgeholt werden, sollen weitergeleitet werden an den externen Account »plumps@aua.example.net«.

  Vorgehensweise:

  1. Anlegen des lokalen Users »Plumps«
  2. Eintrag in der »MAlias.hst«:

     ```
     Plumps=plumps@aua.example.net
     ```

  3. Eintrag in der »MailFilt.hst«:

     ```
     [*]
     # Ignore-Regeln
     ⟨...⟩
     # Kill-Regeln
     ⟨...⟩
     ```

```
# Load-Regeln
⟨...⟩
# Postfächer füllen
add(Plumps) Any-Recipient: "Hoppel@hoppel.example.org"
```

- E-Mails am lokalen Server weiterleiten:
  Alle an den Postmaster der Domain »example.com« gerichteten und am lokalen SMTP-Server eingehenden E-Mails sollen an den hamster-internen Account »postmaster« weitergeleitet werden.

  Glossar: SMTP

  Eintrag in die »MAlias.hst«:

  ```
  postmaster@example.com=postmaster
  ```

Hamster und Reguläre Ausdrücke (RegEx)

Einleitung

Die im Hamster verwendeten »Regular Expression«-Funktionen nutzen eine Win32-Umsetzung von »Perl Compatible Regular Expression (PCRE)«, geschrieben von Philip Hazel, Copyright © 1997–2000 University of Cambridge (Näheres unter `http://www.pcre.org/`*).*

In den Dateien »Scores.hst«, »MailFilt.hst«, in einigen Einstellungen der grafischen Oberfläche bzw. in den dazugehörenden INI-Einstellungen und in einigen Hamsterskript-Funktionen werden »Regular Expressions« (»Reguläre Ausdrücke«, kurz auch RegEx/RegExp oder RE) verwendet. Hierzu im Folgenden eine Erläuterung ohne Anspruch auf Vollständigkeit.

Umfassende Informationen sind auf `http://perldoc.perl.org/perlre.html` zu finden (dies ist die Perl-Dokumentation für Reguläre Ausdrücke, die allerdings eine neuere Version beschreibt) oder sind z. B. im Buch »Mastering Regular Expressions« nachzulesen (»das Eulen-Buch« – Verlag O'Reilly Media; *Buch* ISBN 978-0-596-52812-6, *E-Book* ISBN 978-0-596-55899-4), auf deutsch erschienen als »Reguläre Ausdrücke« (*Buch* ISBN 978-3-89721-720-1, *E-Book* ISBN 978-3-89721-790-4). Eine kurze deutschsprachige Einführung ist bei SelfHTML unter http://wiki.selfhtml.org/wiki/Doku:Perl/Reguläre_Ausdrücke zu finden oder auch in der Wikipedia unter http://de.wikipedia.org/wiki/Regulärer_Ausdruck. Einen ausführlichen Einführungskurs gibt es bei den Regenechsen: Schritt-für-Schritt zu eigenen Regulären Ausdrücken. [1]

Zu beachten ist, dass im Hamster die Markierung Regulärer Ausdrücke anders erfolgt als in den genannten, auf Perl bezogenen Dokumentationen, siehe dazu die Definition Regulärer Ausdrücke im Hamster.

[1] Auf der dazugehörigen Downloadseite als PDF-Datei erhältlich.

Übersicht

- Anmerkungen zu den hier verwendeten Begriffen
- Definition von Regulären Ausdrücken
- Syntax
  - Syntax-Elemente, die außerhalb von »[…]« gelten
  - Syntax-Elemente, die innerhalb von »[…]« gelten
  - Beispiele für die Verwendung dieser Syntax
- Die Bedeutung einzelner Metazeichen
- Optionen der RegEx-Engine einstellen
- Quantifier
- Back-References
- Look-around-Assertions
- Konditionale Reguläre Ausdrücke
- Möglichkeiten & Beispiele für Reguläre Ausdrücke
  - Quantifier
  - Zeichenklassen
  - Weitere Möglichkeiten
- Reguläre Ausdrücke testen

Anmerkungen zu den hier verwendeten Begriffen

Wie oft in der Informationstechnik werden auch hier aus dem Englischen stammende Fachbegriffe oder davon abgeleitete Ausdrücke verwendet.

Englischer Ausdruck	Deutsche Bedeutung
assertion	»Behauptung/Erklärung«; in RegExp Metazeichen, die eine Aussage machen über den Kontext zweier Zeichen
caseless/caseinsensitive	Groß- und Kleinschreibung (*case*) wird ignoriert
casesensitive	Groß- und Kleinschreibung (*case*) wird beachtet
carriage return	Wagenrücklauf; erinnert an Schreibmaschine (Bedeutung: siehe »*newline*«)
character	Zeichen
escape character	»Fluchtzeichen«, maskiert die Metabedeutung des folgenden Zeichens; davon abgeleitet *escapen* für »Metabedeutung maskieren«
flag	»Flagge«, kennzeichnet einen Schalter, der gesetzt wurde (wie Flagge hoch oder runter)
greedy/greediness	»gierig«; bezieht sich auf das Verhalten von Quantifiern, Substantiv *greediness* = »Gierigkeit«
linefeed	Zeilenvorschub; erinnert an Schreibmaschine (Bedeutung: siehe newline)

Fortsetzung nächste Seite

Fortsetzung

Englischer Ausdruck	Deutsche Bedeutung
to match	passen; bezeichnet den Teil des untersuchten Strings, der auf den Suchstring passt; davon abgeleitet *matchen* (Regulärer Ausdruck matcht auf Suchstring) und Substantiv *Match* für passende Zeichenkette
newline	Zeilenendezeichen, je nach Betriebssystem nur *Carriage-Return* (kurz CR) oder nur *Linefeed* (LF) oder *Carriage-Return* und *Linefeed* (CRLF)
pattern	Suchmuster, bezeichnet kompletten Matchstring
quantifier	Operator, der angibt, wie oft der vorhergehende Ausdruck wiederholt werden soll/darf
string	Zeichenkette
subpattern	Logisch zusammengehöriger Abschnitt des Patterns (Gruppe)

Definition von Regulären Ausdrücken

Reguläre Ausdrücke sind eine Möglichkeit, Zeichenketten (*Strings*) mit einer Meta-Syntax zu beschreiben. Diese erlaubt es, Strings in Klassen einzuteilen und diese Klassen zu beschreiben. So kann man verschiedenartige Strings, die ein bestimmtes Element gemeinsam haben, anhand dieses Elementes klassifizieren.

Der fragliche String wird dabei von links nach rechts durchsucht und es wird der erste Match-String zurückgegeben. Standardmäßig wird dabei der längste Match zurückgegeben. Dieses Verhalten wird als gierig (*greedy*) bezeichnet und kann bei Bedarf auch abgeschaltet werden. Ebenso ist es (im Hamster) Standard, dass die Groß- und Kleinschreibung (*case*) nicht beachtet wird (*caseless*). In anderen Programmen, auch im gern verwendeten Kornnews (siehe in der Linkliste), wird jedoch in der Voreinstellung zwischen Groß- und Kleinschreibung unterschieden (*casesensitive*).

Reguläre Ausdrücke werden in den Hamster-Scoredateien (»`MailFilt.hst`« und »`Scores.hst`«) durch eine linke geschweifte Klammer »{« eingeleitet und mit einer rechten geschweiften Klammer »}« abgeschlossen. Alles dazwischen wird als Teil des Suchstrings betrachtet, die Klammern selbst gehören aber nicht zum Regulären Ausdruck. Wer Texte über Perl liest, wird dort oft Zeichenketten eingeschlossen in »/ . . ./« sehen. Die Schrägstriche erfüllen dabei den gleichen Zweck. In Hamsterskripten sind diese Klammern nicht notwendig, hier sind es nur bestimmte Funktionen, die Reguläre Ausdrücke erwarten, so dass bei diesen Funktionen die Regexp wie jeder andere String in Anführungszeichen eingeschlossen wird.

Achtung! In den Regulären Ausdrücken dürfen keine Kommentare aus der Syntax der Scoredateien oder von »hs2« stehen. Es dürfen aber logische Zeilenumbrüche verwendet werden, da die Zusammenfügung zu einer einzigen Zeile als allererster Schritt erfolgt.

Beispiel (»Scores.hst«):

Der Ausdruck findet wie zu erwarten alle Postings mit »heikos host« im »From«:

```
=-1234 from: {(?x)hei_
              kos\shost}
```

Der Ausdruck

```
=-1234 from: {(?x)hei  #Zeile 1_
              kos\shost}
```

dagegen findet, auf den ersten Blick unerwartet, alle Postings mit »hei« im »From«. Die Ursache ist, dass der Hamster aus den zwei Zeilen ein

```
=-1234 from: {(?x)hei  #Zeile 1kos\shost}
```

macht und der RegExp-Engine nur die Kette »`(?x)hei`« übergibt, da der gesamte letzte Teil »`#Zeile 1kos\shost`« ***als Kommentar interpretiert wird***.

Syntax

Syntax-Elemente, die außerhalb von »[...]« gelten

Zeichen	Kurzbeschreibung
\	Allgemeiner Escape-Character.
^	Steht für den Beginn eines Strings oder einer Zeile.
$	Steht für das Ende eines Strings oder einer Zeile.
.	Steht für jedes Zeichen außer newline.
[	Startet Definition von Sequenzen (Zeichenklassen).
]	Beendet Definition von Sequenzen (Zeichenklassen).
|	Steht für logisches Oder; hiermit werden Alternativen abgetrennt.
(	Startet subpattern (Gruppierung).
)	Beendet subpattern (Gruppierung).
?	Steht für 0 oder 1 Wiederholung des vorgehenden Elementes, hebt greediness auf; dient nach »(« als Schalter, um Optionen einzuleiten.
*	Steht für 0 oder mehr Wiederholungen des vorhergehenden Elementes.
+	Steht für 1 oder mehr Wiederholungen des vorhergehenden Elementes.

Fortsetzung nächste Seite

Fortsetzung

Zeichen	Kurzbeschreibung
{	Startet Quantifizierung (Minimum, Maximum) des vorhergehenden Elementes.
}	Beendet Quantifizierung (Minimum, Maximum) des vorhergehenden Elementes.

Syntax-Elemente, die innerhalb von »[...]« gelten

Zeichen	Kurzbeschreibung
\	Allgemeiner Escape-Character.
^	Als erstes Zeichen steht es für die Negation der beschriebenen Zeichenklasse: »[^a-zA-Z0-9]« steht für alle nichtalphanumerischen Zeichen.
-	Steht für einen Bereich »von...bis«: »[0-9]« steht für alle Ziffern. *(Steht es als erstes oder letztes Zeichen in einer Sequenz, hat es keine Sonderbedeutung.)*

Beispiele für die Verwendung dieser Syntax

Bevor die einzelnen Meta-Charaktere weiter erläutert werden, ein paar kurze Beispiele. Als Teststring soll hier der Satz »Regexp$ falsch angewandt erlauben einen wunderbaren Schuss in den eigenen Fuß.« dienen (Das »$«-Zeichen steht dort absichtlich, wie gleich zu sehen ist). Alle Regexps sind als eine Zeile zu lesen.

Regexp	Match	Erläuterung
Reg	Reg	Trivialer Fall; String ohne Metabedeutung.
[reg]	R	Caseless; wenn kein Quantifier angegeben wurde, verhält sich der Ausdruck wie die non-greedy-Variante (siehe unten).
[reg]+	Rege	Caseless; »+« ist hier greedy und es wird nach dem längstmöglichen String gesucht, der die drei Zeichen »reg« in beliebiger Reihenfolge und mindestens einmal enthält.
[reg]+?	R	Non-greedy-Variante.
[(reg)+]	R	»+« hat seine Bedeutung als Quantifier nur außerhalb von »[...]«.
[^reg]	x	Der erste Match, der nicht »r«, »e« oder »g« enthält.
[a-z]+	Regexp	»$« ist kein Zeichen zwischen »a« und »z«.
(?-i)[a-z]+	egexp	Case-sensitive; diese Option erstreckt sich soweit, wie der subpattern reicht, in dem das Flag gesetzt wurde.
((?-i)[a-z]+)\$.[A-Z]+	egexp$ falsch	Die Wirkung des Flags gilt nur in dem subpattern; »$« musste markiert werden, da es eine Metabedeutung besitzt; ».« (der Punkt) steht für genau ein beliebiges Zeichen, hier das Leerzeichen; caseless Match am Ende.
[0-9]*		Das hier ist etwas trügerisch. Wir sehen zwar keinen Matchstring, aber der Hamster erkennt dieses trotzdem als Treffer, da »*« (der Stern) auch für null Wiederholungen steht, und ein leerer String wird als Match erkannt. Hierauf ist bei der Verwendung von »*« als Quantifier immer zu achten.

Die Bedeutung einzelner Metazeichen

Eine Bedeutung des Backslash »\«, nämlich die als Escape-Character, kennen wir schon. Wie wir in den Beispielen sahen, hilft er uns, Zeichen mit einer Metabedeutung, wie etwa das »$«-Zeichen, zu maskieren. *(Wie schreibt man einen Backslash? Indem man ihn selbst maskiert:* »\\«*.)*

Aber der Backslash unterstützt auch unsere Faulheit und hilft uns gleichzeitig, RegExps lesbarer zu schreiben. Wir sahen oben, dass man Charakterklassen mit Hilfe von »[...]« beschreiben kann. Da dies oft vorkommt, gibt es einige Abkürzungen. Ebenso können wir mit dem Backslash nach Zeichen suchen, die wir bisher nicht finden konnten, nämlich alle Zeichen, die nicht direkt über die Tastatur eingegeben werden können.

Zeichen	Kurzbeschreibung
`\a`	Entspricht dem ASCII-Zeichen 7 (Bell).
`\cX`	Entspricht Control-X; hier steht X für ein beliebiges Zeichen. Vergegenwärtigen wir uns, dass Control-@ für das Zeichen mit dem ASCII-Wert 0, Control-a für das Zeichen mit dem ASCII-Wert 1 steht usw.; Control-i steht dann z. B. für das Zeichen mit dem ASCII-Wert 9 = Tabulator. Die Anwendung von `\cX` ist etwas komplizierter als hier ausgeführt, aber dieses liest man bei Bedarf am besten in der oben aufgeführten Perl-Dokumentation nach.
`\e`	Escape; ASCII-Wert 27.
`\f`	Formfeed; ASCII-Wert 12.
`\n`	Linefeed; ASCII-Wert 10.
`\r`	Carriage Return; ASCII-Wert 13.
`\t`	Tabulator; ASCII-Wert 9.
`\xHH`	Ein Zeichen, dessen Wert in hexadezimaler Notation (Zahlensystem mit Basis 16) `HH` ist; `\x09` wäre also der Tabulator und `\x41` ist das Zeichen »A«. Die Hex-Zahl bezieht sich auf die Position des Zeichens in der ASCII-Tabelle. Für die Zeichen mit einer Position > 127 gilt die ANSI-Tabelle. Funktioniert nur bis `\xFF` = dezimal 255.
`\ZZZ`	Ein Zeichen, dessen Wert in oktaler Notation (Zahlensystem mit Basis 8) `ZZZ` ist. Der Tabulator wäre also `\011` und das »A« `\101`. Funktioniert nur bis `\777` = dezimal 511.
`\d`	Entspricht `[0-9]`.
`\D`	Ist das Komplement zu `\d`, entspricht also `[^0-9]`.
`\s`	Steht für jeden White-Space-Character (Leerzeichen, Tabulator).
`\S`	Das Komplement zu `\s`, also alles, was kein White-Space-Character ist.

Fortsetzung nächste Seite

Fortsetzung

Zeichen	Kurzbeschreibung
\w	Jedes Zeichen, das als Word-Character betrachtet wird. Dieses entspricht der Sequenz »[0-9a-zA-Z_]« (alphanumerische Zeichen und der Unterstrich). **Buchstaben wie ä, ö, ü, é oder das ß gehören nicht dazu!**
\W	Das Komplement zu \w.
\⟨Zahl⟩	Verwendung als Back-Reference (Rückbezug) auf Gruppierung. »⟨Zahl⟩« muss eine natürliche Zahl sein (genauer: ein positiver Integer).
\b	Außerhalb von Sequenzen: Steht für eine Wortgrenze, d. h. auf einer Seite steht ein \w-Character und auf der anderen Seite ein \W-Character. Innerhalb von Sequenzen: Steht für den Backslash.
\B	Das Komplement zu \b.
\A	Beginn des Strings.
\Z	Ende des Strings oder Newline.
\z	Ende des Strings.

Die letzten Zeichen in der Tabelle dürfen mit Ausnahme des \b nicht in Sequenzen verwendet werden. Sie sind sogenannte Assertions (Aussagen). Sie beziehen sich nicht auf ein einzelnes Zeichen, sondern machen eine Aussage über den Kontext zweier Zeichen.

Beispielsweise matcht der Ausdruck \w+\b\w+\b\w+ nicht auf den Satz »Hamster ist klasse.« Das erste \w+ matcht zwar auf »Hamster«, dann müsste dem Ausdruck nach ein Zeichen der Klasse \W kommen. Dies stimmt noch, da ein Space (Leerzeichen) folgt, dann aber erwarten wir wieder Zeichen der Klasse \w. Aber unser Space finden wir so nicht, da das \b den Space nicht »einfing«. Als Hilfe könnten wir den Suchstring als \w+\s\w+\s\w+ schreiben. Jetzt fangen wir auch den Space.

\A, \Z und \z werden hier nicht weiter vertieft, da die Unterschiede zu »^« bzw. »$« für den Hamster unerheblich sind.

»^« bzw. »$«

Beides sind Assertions, wobei »^« außerhalb von Sequenzen für den Beginn des Strings und »$« für das Ende stehen. Innerhalb einer Sequenz bildet »^« als erstes Zeichen das Komplement (also die umgekehrte Wirkung) der Sequenz; steht es nicht zuerst, fehlt ihm diese Metabedeutung. »$« hat grundsätzlich keinerlei Metabedeutung in Sequenzen.

Eckige Klammern

Eine linke eckige Klammer »[« kennzeichnet den Beginn einer Sequenz, die eine Zeichenklasse beschreibt und geht bis zur letzten rechten eckigen Klammer »]«, der keine linke Klammer entspricht. Eine rechte eckige Klammer allein hat keine Metabedeutung, eine linke Klammer

muss mit einem Backslash maskiert werden. Innerhalb einer Sequenz kann eine rechte Klammer als erstes Zeichen stehen und muss nicht maskiert werden: `[]A-Za-z]+` matcht auf alphabetische Zeichen und die rechte Klammer.

Sequenzen

Innerhalb einer Sequenz dient der Bindestrich »-« dazu, den Bereich der Sequenz anzugeben. Das Intervall ist beidseitig geschlossen, `[a-z]` schließt also »a« und »z« mit ein. Um den Bindestrich selbst zu finden, schreibt man ihn einfach als erstes oder letztes Zeichen der Sequenz oder maskiert ihn mit einem Backslash. Sequenzen können auch mit den Oktalwerten (nicht vergessen: `\ZZZ`, mit `Z` von 0 bis 7, gibt den Oktalwert eines Zeichens wieder) oder den Hexadezimalwerten (`\xHH`, mit `H` von 0 bis F) der Zeichen notiert werden.

`\d`, `\D`, `\s`, `\S`, `\w` und `\W` können mit ihrer Metabedeutung in Sequenzen genutzt werden, `\b` steht innerhalb von »[...]« für den Backslash.

Das Pipe-Zeichen »|«

Wie wir oben sahen, steht »|« für ein logisches Oder (dies ist ein inklusives Oder, es entspricht der schrecklichen Sprachkonstruktion »und/oder«. Beispielsweise ist ein Satz »Heute Abend koche ich oder lese das Buch zu Ende.« in drei Fällen logisch wahr: Wenn ich nur koche, wenn ich nur lese, wenn ich koche und lese). Dieses Zeichen erlaubt es uns, verschiedene Alternativen für einen Match in einem Suchstring zu notieren. Mit Hilfe der runden Klammern »()« können wir Subpattern bilden, die wir dann mit dem Pipezeichen verknüpfen.

Optionen der RegEx-Engine einstellen

Der RE-Engine kann man bestimmte Optionen, häufig auch Flags genannt, zur allgemeinen Verarbeitung von Zeichenketten vorgeben. Diese Optionen sind in eine runde Klammer, beginnend mit einem Fragezeichen, gefolgt von den für die jeweilige Option stehenden Buchstaben, zu schreiben. Die Reichweite einer Option erstreckt sich auf das (Sub-)Pattern, in dem sie enthalten ist. Ein Minuszeichen vor der Option kehrt deren Bedeutung in das Gegenteil um.

Beispiel: »(?ms-ix)« – Es wird »m« und »s« eingeschaltet, »i« und »x« ausgeschaltet. Das Minus bezieht sich immer auf alle Optionen, die dahinter aufgeführt sind.

Folgende Optionen existieren:

Option	Bedeutung
i	Groß- und Kleinschreibung werden ignoriert.
m	Ändert die Interpretation von »^« und »$« so, dass bei mehrzeiligen Texten auf Beginn und Ende einer einzelnen Zeile geprüft wird.
s	Das Metazeichen ».« (Punkt) matcht auch auf das Newline-Zeichen.

Fortsetzung nächste Seite

Fortsetzung

Option	Bedeutung
x	Diese Option ermöglicht es, den Ausdruck durch Leerzeichen, Kommentare und Zeilenumbrüche in eine besser lesbare Form zu bringen. Im Hamster ist ein Zeilenumbruch damit aber nicht möglich. Deshalb ist es günstiger, auf diese Option zu verzichten. Statt dessen sollten Kommentare in der Form `(?#...)` benutzt und Umbrüche innerhalb einer Filterzeile mit dem Unterstrich am Zeilenende gekennzeichnet werden.
#	Kommentare.

Achtung! Im Hamster ist standardmäßig die Option »i« aktiviert, »m«, »s« und »x« sind deaktiviert. Um in einem Ausdruck nur groß- oder kleingeschriebene Buchstaben zu finden, muss deshalb die Option `(?-i)` benutzt werden.

Im gern zusammen mit dem Hamster eingesetzten Programm Korrnews *(siehe Linkliste) ist jedoch auch die Option »i« deaktiviert.*

Kommentare

Ein Kommentar beginnt mit der Zeichenfolge »`(?#`« und endet mit der schließenden runden Klammer »`)`«. Ist die Option »x« der RE-Engine aktiv, so werden alle Zeichen nach dem Sonderzeichen »#« außerhalb einer Zeichenklasse bis zum Zeilenende als Kommentar angesehen.

Quantifier

Quantifier sind Ausdrücke, die angeben, wie oft ein vorhergehender Ausdruck wiederholt werden darf oder muss. Wir haben schon »`*`«, »`+`« und »`?`« kennengelernt, diese sind eigentlich nur bequeme Kurzschreibweisen der allgemeinen Form eines Quantifiers: »`{n,m}`«, wobei dieser Ausdruck als »mindestens n Elemente und maximal m Elemente« zu lesen ist; »`{n,}`« heißt »minimal n und maximal unendlich viel Elemente«; »`{n}`« heißt »genau n Elemente«. Unsere drei bekannten Quantifier entsprechen in dieser Notation also:

`*` = `{0,}`
`+` = `{1,}`
`?` = `{0,1}`

Nach welchen Ausdrücken sind Quantifier erlaubt? Nach jedem einzelnen Character (sowohl escaped als auch nicht escaped), nach dem Punkt ».« (wir erinnern uns, er steht für genau ein beliebiges Zeichen), nach einer Sequenz oder nach einem Subpattern (ein durch runde Klammern eingeschlossener Teil des Suchpatterns). Außerdem sind sie noch nach sogenannten Backreferences erlaubt (siehe folgenden Abschnitt), was hier aber nicht behandelt wird. Quantifier sind greedy, versuchen also den längstmöglichen Matchstring zu finden. Da dies nicht in allen Fällen gewünscht ist, kann man dem Quantifier direkt das Fragezeichen »?« (oder die gerade

gelernte Variante davon) folgen lassen, dann wird der Quantifier genügsam und gibt sich mit dem kürzestmöglichen Matchstring zufrieden.

Quantifier sollten sehr mit Bedacht eingesetzt werden, da diese die Anzahl möglicher Kombinationen extrem erhöhen können, so dass ein Programm dann sehr lange braucht, um entscheiden zu können, ob ein Match vorliegt.

Back-References

Suchmuster, die in runden Klammern stehen, werden zwischengespeichert und können an anderer Stelle wiederverwendet werden, indem mit »\⟨Zahl⟩« darauf Bezug genommen wird:

```
{(Holz|Plastik) und \1fenster}
```

Das Regex findet entweder »`Holz`« oder »`Plastik`« und speichert diesen Wert, weil er in einer runden Klammer steht. Dann wird das Wort »und« gefunden. Wegen »`\1`« wird der gespeicherte Inhalt der Klammer, zusammen mit »`fenster`« zum Weitersuchen verwendet. Dadurch wird entweder »`Holzfenster`« oder »`Plastikfenster`« an dieser Stelle gefunden, je nachdem, welches der beiden Wörter weiter vorn gefunden wurde. Diese Regex findet also nur »`Holz und Holzfenster`« oder »`Plastik und Plastikfenster`«, aber *nicht* »`Holz und Plastikfenster`« oder »`Plastik und Holzfenster`«.

Der Rückbezug darf nicht in derselben Klammer stehen, auf die er sich bezieht:

```
{(a\1)}
```

führt nie zu einem positiven Ergebnis. Allerdings geht es, wenn sich ein Wiederholungszeichen dahinter befindet:

```
{(da|de\1)+}
```

findet »dadadada« oder »dadeda« oder auch »dadedadedada«.

Bei »⟨Zahl⟩« muss es sich um eine natürliche Zahl handeln, genauer: einen positiven Integer.

Look-around-Assertions

Mit einer Look-around-Assertion kann getestet werden, ob ein Muster der gefundenen Stelle vorangeht oder folgt, ohne dass die geprüfte Zeichenkette in den gefundenen String übernommen wird.

Als Beispiel soll die Buchstabenfolge »`foo`« gefunden werden, die von der Zeichenfolge »`bar`« gefolgt wird, ohne aber »`bar`« im Suchergebnis wiederzufinden: »`foo(?=bar)`«.

Das Ergebnis lautet »`foo`«, im Gegensatz zu einer »normalen« RegEx, bei der mit »`{foobar}`« das Ergebnis auch »`foobar`« lauten würde.

Mit »`{foo(?!bar)}`« lautet das Ergebnis dagegen nur dann »`foo`«, wenn »`foo`« *nicht* von »`bar`« gefolgt vorkommt, sondern z. B. von »`ter`« (also das Wort »`footer`« und Ableitungen davon).

Es gibt vier Arten der Assertion:

?=	vorwärtsgerichtet positiv *(positive look-ahead assertion)*
?!	vorwärtsgerichtet negativ *(negative look-ahead assertion)*

?<= rückwärtsgerichtet positiv *(positive look-behind assertion)*
?<! rückwärtsgerichtet negativ *(negative look-behind assertion)*

Die rückwärtsgerichteten Assertionen müssen eine feste Länge besitzen, d. h. in ihnen darf kein Quantifier vorkommen!

Beispiele für rückwärtsgerichtete Assertionen:

»`{(?<=foo)bar}`« findet »`bar`« nur, *wenn* »`foo`« davor steht.

»`{(?<!foo)bar}`« findet »`bar`« nur, wenn *kein* »`foo`« davor steht.

Ein praktisches Beispiel – Suche nach Newsgruppen mit ».de.« im Namen, aber nur, wenn kein »fr.« davor steht (französische Gruppen können auch ein ».de.« enthalten):

```
(?<!fr)\.\S*\.de\.
```

Konditionale Reguläre Ausdrücke

Eine weitere interessante Möglichkeit, um Assertionen anzuwenden, sind konditionale Reguläre Ausdrücke. Im Gegensatz zu der einfachen Assertion, die direkt mitentscheidet, ob der komplette Ausdruck matcht, lässt sich das hier im Sinne einer »if … then … else«-Konstellation relativieren. Der Aufbau ist folgender:

`(?(`*`⟨if⟩`*`) ⟨then⟩ | ⟨else⟩)`, dt.: `(?(`*`⟨wenn⟩`*`) ⟨dann⟩ | ⟨sonst⟩)`

oder nur

`(?(`*`⟨if⟩`*`) ⟨then⟩)`, dt.: `(?(`*`⟨wenn⟩`*`) ⟨dann⟩)`

Die Einleitung erfolgt immer mit einer geöffneten runden Klammer und einem Fragezeichen, gefolgt von einer »*wenn*«-Assertion, die nach den oberen Regeln aufgebaut ist. Dann folgen zwei Möglichkeiten, die mit einem Trennzeichen eingesetzt werden und mit einer sich schließenden Klammer enden. Je nach dem, wie der Test verläuft, wird dann die erste Möglichkeit bei positivem oder die zweite bei negativem Ergebnis durch die Regex-Engine ausgeführt.

Ein Beispiel für eine Anwendung ist, wenn nur der Name des Absenders in den Match übernommen werden soll, ganz gleich, ob die alte oder neue Schreibweise vorhanden ist:

Suchstring entweder »Vorname Name <Mail@Adresse>« oder »Mail@Adresse (Vorname Name)«

```
(?(?=.*<)[^<]*|[^\(]*\))
```

Die Assertion im konditionalen Regulären Ausdruck testet vorn, ob hier schon die neue Schreibweise angewandt wird, in dem sie eine geöffnete spitze Klammer sucht. Wenn dies der Fall ist, wird die erste, sonst die andere Variante genommen. Da der konditionale Reguläre Ausdruck hier allein steht, würde er auch nach diesem Muster »(?(?=<)…« (d. h. vorn ohne ».*«) die geöffnete spitze Klammer finden, wir hätten dann aber keine Möglichkeit, den vorderen Namen noch mit der ersten Variante durch ein Match zu greifen, da der interne Pointer in diesem Fall schon daran vorbei ist (z. B. `(?(?=<).*`) auf »test <123>«).

Leider haben wir noch das Problem, dass der Name noch nicht sauber abgetrennt ist. Entweder endet er noch mit einem Leerzeichen oder mit einer schließenden runden Klammer. Als Bonus für Fortgeschrittene können wir hier noch je eine weitere Assertion im Ergebnisausdruck mit einbauen, um das zu korrigieren:

```
(?(?=.*<).*\w(?=\s*<)|[^\(]*\w(?=\)))
```

Wie wir bereits oben gelernt haben, kann man mit einer einfachen Assertion testen, ob ihr bestimmte Zeichen folgen, ohne sie in den Match zu übernehmen. Und genau das passiert hier ebenfalls, wir haben also eine Kombination von einem klassischen konditionalen Regulären Ausdruck mit je einer normalen Assertion im Ergebnismatch. Außerdem kann durch das Einfügen von »\w« gleich noch auf alphanumerische Zeichen getestet werden, so dass es bei nicht vorhandenen Namen keinen Leerstring als Ergebnis gibt, sondern ein False.

Möglichkeiten & Beispiele für Reguläre Ausdrücke

Quantifier

Die einfachsten Beispiele sind die Quantifier »*«, »+«, »?« und »{…}«. Sie wirken auf das Element, das ihnen vorangeht, was z. B. ein Buchstabe sein kann.

»***« *bedeutet, dass ein Zeichen 0-mal oder öfter gefunden wird:*
»abc*« findet unter anderem »ab«, »abc« und »abccccc«.

»*+*« *bedeutet, dass ein Zeichen 1-mal oder öfter gefunden wird:*
»abc+« findet unter anderem »abc« und »abccccc«, aber nicht »ab«.

»*?*« *bedeutet, dass ein Zeichen 0- oder 1-mal gefunden wird:*
»abc?« findet »ab« und »abc«, aber nicht »abccccc«.

»*{n}*« *bedeutet, dass ein Zeichen n-mal gefunden wird:*
»abc{3}« findet nur »abccc«.

»*{n,}*« *bedeutet, dass ein Zeichen mindestens n-mal gefunden wird:*
»abc{3,}« findet unter anderem »abccc« und »abccccc«, aber nicht »abcc«.

»*{n,m}*« *bedeutet, dass ein Zeichen mindestens n- und höchstens m-mal gefunden wird:*
»abc{3,4}« findet nur »abccc« und »abcccc«, aber nicht »abcc« oder »abccccc«.

Zeichenklassen

(Zur Verdeutlichung werden nachfolgend Leerzeichen durch »␣« und der Tabulator durch »⟶« dargestellt.)

»*.* « *enthält alle Zeichen, kann alles darstellen:*
»␣…␣« findet alles, was 3 Zeichen lang ist (auch 3 Leerzeichen) und von Leerzeichen eingerahmt ist.

»*\w*« *enthält alle alphanumerischen Zeichen (A…Z, a…z, 0…9 und den Unterstrich »_«):*
»\w{3}« findet unter anderem »123«, »ich«, »wir« und »A_B«.

»`\W`« *enthält alle Zeichen, die nicht alphanumerisch sind:*
»`\W{3}`« findet unter anderem »`///`«, »`---`« und »`|+|`«.

»`\s`« *findet ein »white space« (Leerzeichen, Tabulator, Newline, Return):*
»`a\sa`« findet unter anderem »a⟶a« (mit Tabulator) und »a␣a«, aber nicht »a␣␣a« (mit 2 Leerzeichen).

»`\S`« *findet alles, was kein »white space« ist:*
»`\S+\s\S+`« findet unter anderem »Hallo␣du«, aber nicht »A_A«.

»`\d`« *findet alle Ziffern (0…9):*
»`\d{3}`« findet alle 3-stelligen Zahlen.

»`\D`« *findet alle Zeichen, die keine Zahlen sind:*
»`\d-\D+`« findet unter anderem »3-stellig«.

»`[ab]`« *bedeutet, dass an dieser Stelle die Zeichen »a« und »b« vorkommen können:*
»`[ab]{3}`« findet: »aaa«, »aba«, »aab«, »abb«, »bbb«, »bab«, »bba« und »baa«.

Das Zeichen »-« kennzeichnet innerhalb von »[ ]« einen Bereich von Zeichen:
»`[A-Z]`« findet jeden Buchstaben zwischen »A« und »Z« *inkl.* »A« und »Z«.

Weitere Möglichkeiten

»`\b`« *ist die Grenze zwischen »`\w`« und »`\W`« (also eine Wortgrenze):*
»`Hall.\b`« findet »Hallo« in »Hallo du!« und »Hallo!«, aber nicht in »Halloren«; scheinbar paradox matcht jedoch auch »Hallö« in »Hallöle« auf den Suchstring, da das »ö« einerseits durch den Punkt gefangen wird, andererseits aber nicht als Wortzeichen gilt, so dass wir es hier zwischen »ö« und »l« für RegEx mit einer Wortgrenze zu tun haben!

»`^`« *ist der Anfang einer Zeichenkette:*
Im String »Hallo, Du halber Halbling!« wird mit »`^Hal..`« nur »Hallo« gefunden.

»`$`« *ist das Ende einer Zeichenkette:*
»`\s+$`« sucht nach (den meist überflüssigen) Leerzeichen am Zeilenende,
»`^\s+$`« sucht nach Zeilen, die nur aus White-Space bestehen, und »`^$`« nach Leerzeilen.

Reguläre Ausdrücke testen

Zum Testen von Regulären Ausdrücken existiert im Hamster-Skriptarchiv ein Skript »`Regexp.hsc`«.[1]

Weitere Freeware zum Testen Regulärer Ausdrücke (Auswahl):

- RegExTester: ursprünglich `http://www.steenet.de/regextester.html`, erhältlich unter `http://speravir.website.org/files/regex/`; verwendet die gleiche RegEx-Engine wie der Hamster (PCRE von Philip Hazel), und Programmsprache ist deutsch.

[1] Direktlink: `http://www.arcorhome.de/newshamster/micha/archive.php?n=HamsterScript.RegExp`

- RegExCoach: http://weitz.de/regex-coach/; ist etwas komfortabler, aber Programmsprache englisch und mit einer anderen Engine (CL-PPCRE) ausgestattet.
- RegExTester: http://www.christophm.de/software/RegExTester/ (benötigt Microsoft .NET Framework 1.0); Engine unbekannt, Programmsprache englisch.
- REGex TESTER (Online-Tester): http://www.regextester.com/; Website, englisch; kann so eingestellt werden, dass entsprechend den PCRE getestet wird (»Preg« aktivieren, Voreinstellung ist Dialekt »Javascript«).

Hamster und SSL

Warnung: *Das ist die FAQ zur Benutzung mit einem Hamster, der eine eingebaute SSL-Unterstützung besitzt und kein Stunnel mehr benötigt, also dem Hamster Classic ab Version 2.0.0.0. Für Hamster, die noch Stunnel benötigen (z. B. Hamster Classic vor Version 1.3.23.140), sollte die dafür gedachte FAQ benutzt werden. Sie ist zu finden unter* http://www.philippwendler.de/hamster/hamster-ssl-stunnel.txt.

Fragen

1. Wie benutze ich mit dem Hamster SSL?
   1.1. Wie kann ich über das Online-Menü SSL nutzen?
   1.2. Wie kann ich meine Mails mit Hamster-Skripten oder per OLE über eine SSL-Verbindung abholen und verschicken?
      1.2.1. Wenn HamMailExchange im Skript bzw. ControlRunMail per OLE verwendet wird.
      1.2.2. Wenn HamFetchMail im Skript bzw. ControlRunFetchMail per OLE verwendet wird.
      1.2.3. Wenn HamSendMail/HamSendMailAuth im Skript bzw. ControlRunSendMail/ControlRunSendMailAuth per OLE verwendet wird.
   1.3. Wie kann ich meine News über eine SSL-Verbindung abholen und verschicken?
2. Hamster als SSL-Server
3. Wie erhöhe ich die Sicherheit bei SSL?
4. Was sind Zertifikate und wozu brauche ich sie?
5. Wo bekomme ich fremde Zertifikate her?
   5.1. CA-Zertifikate
   5.2. Serverzertifikate
6. Wie erstelle ich mir ein eigenes Zertifikat?

7. Wie mache ich dem Hamster fremde Zertifikate bekannt?
7.1. Alle Zertifikate in einer Datei
7.2. Alle Zertifikate in einem Verzeichnis
8. Wie benutze ich Zertifikate mit Hamster und SSL?
8.1. Serverzertifikate
8.1.1. Einrichten mit dem Hamster als Server
8.1.2. Einrichten mit dem Hamster als Client
8.1.2.1. Überprüfen des CA-Zertifikates
8.1.2.2. Überprüfen des Serverzertifikates
8.2. Clientzertifikate
8.2.1. Einrichten mit dem Hamster als Server
8.2.2. Einrichten mit dem Hamster als Client
9. Was bedeuten diese Fehlermeldungen?
10. Wo bekomme ich OpenSSL her?
11. Was für interessante URLs gibt es zu diesem Thema?

Antworten

Glossar: SSL TLS

1. Wie benutze ich mit dem Hamster SSL?
Folgendes wird benötigt:

- Hamster Classic ab Version 2.0.0.0, siehe Linkverzeichnis.
- OpenSSL-DLLs, siehe Frage 10. Damit der Hamster sie findet, müssen die 2 Dateien folgende Namen tragen: »`libeay32.dll`« und »`libssl32.dll`«.

Diese ersten Schritte sind immer nötig:

Kopiere die beiden OpenSSL-DLLs, die Du kompiliert oder heruntergeladen hast, in das Hamsterhauptverzeichnis oder ein Verzeichnis, das in der Umgebungsvariable `%PATH%` enthalten ist. Starte dann den Hamster neu.

Port

Zuerst einmal musst Du herausfinden, ob der Server SSL/TLS auf dem Standardport (per SSL-Handshake mit dem »STARTTLS«-KOMMANDO bzw. bei POP3 »STLS«) oder über einen separaten Port unterstützt. Das steht oft auf der Website des Providers. Es gibt auch bei Remo Müllers »anytool«-Projekt (siehe Linkverzeichnis) im dortigen Wiki eine Übersicht über bekannte Freemail-Dienste, die SSL zumindestens auf separatem Port anbieten.

Glossar: Port

Wenn der Server SSL/TLS auf einem separaten Port unterstützt, und Du dir die Portnummern nicht merken kannst, öffne die Datei »services« (ohne Endung!). Bei Windows 95 bis Windows ME liegt sie im Windows-Verzeichnis, bei Windows NT und Nachfolgern im Verzeichnis »%SYSTEMDIR%\drivers\etc« (normalerweise »c:\windows\system32\drivers\etc«). Füge folgende Zeilen ein und speichere die Datei unter demselben Namen (*seit Windows Vista zum Teil bereits enthalten!*):

Glossar: SMTP, NNTP, POP3, TCP

```
smtps 465/tcp
nntps 563/tcp
imaps 993/tcp
pop3s 995/tcp
```

Jetzt kannst Du überall statt Port 465 »smtps«, statt Port 563 »nntps« und statt Port 995 »pop3s«, also den Standardportnamen mit angehängtem »s«, angeben.

Beachte im folgenden, dass es bei zwei oder mehr verschlüsselten Verbindungen gleichzeitig und wenig Hauptspeicher Probleme wegen Speichermangels geben kann. Sollte die Fehlermeldung »library has no ciphers - Creating SSL context failed« im Hamsterlog auftreten, musst Du die Zahl der gleichzeitigen verschlüsselten Verbindungen (auch mehrere Threads zu einem Newsserver zählen dazu) reduzieren.

1.1. Wie kann ich über das Online-Menü SSL nutzen?

Wenn der Server SSL/TLS auf dem Standardport unterstützt, trage den Server ganz normal ein und stelle die erste Auswahlliste in den »SSL-Einstellungen« auf »TLS immer nutzen«. Dazu müssen temporär die erweiterten Einstellungen aktiviert werden. Wenn der Server öfter mal Probleme mit dem TLS hat, und Du dann lieber über eine unsichere Verbindung abfragst, als gar nicht, kannst Du es auch auf »TLS nutzen, wenn möglich« stellen.

Wenn der Server SSL/TLS ausschließlich über einen separaten Port unterstützt, trage den Server in den Hamster ein, setze die Auswahlliste auf »SSL bei sicherem Port immer nutzen« und ersetze den Standardport (meist angegeben mit dem Protokollnamen) durch den SSL-Port. Das ist normalerweise »995« bei POP3-SSL, »465« bei SMTP-SSL und »563« bei NNTP-SSL.

Wenn Du jetzt über das Online-Menü eine Verbindung zu diesem Server aufbaust, wird eine SSL-Verbindung aufgebaut. Vergiss aber nicht, falls vorhanden, denselben Server ohne SSL aus der Gruppe im Online-Menü herauszunehmen. Das ist sonst ein Sicherheitsrisiko, da zusätzlich noch eine unverschlüsselte Verbindung aufgebaut wird und Dein Passwort wieder unverschlüsselt übertragen wird.

Beachte, dass Du dir ohne Zertifikat nicht sicher sein kannst, dass Du auch wirklich mit dem gewollten Server sprichst. Du solltest dir also das Zertifikat beschaffen (s. Frage 5) und benutzen (s. Frage 8.1.2). Mit der Auswahlbox »Wähle Überprüfungsverfahren« kannst Du die verwendete Stufe einstellen. Bei »Zertifikate immer überprüfen« wird nur das Zertifikat

der CA und die Signatur des Serverzertifikats überprüft. Bei »Peer mittels lokal vorhandener Zertifikate überprüfen« wird zusätzlich das Serverzertifikat selber überprüft. Zum Feld »Überprüfung über Datei« siehe Frage 7.

1.2. Wie kann ich meine Mails mit Hamster-Skripten oder per OLE über eine SSL-Verbindung abholen und verschicken?

1.2.1. Wenn HamMailExchange im Skript bzw. ControlRunMail per OLE verwendet wird.

Folge einfach den Anweisungen in Frage 1 und 1.1.

Beachte, dass in der Serverliste, die übergeben wird, eventuell der Port geändert werden muss.

1.2.2. Wenn HamFetchMail im Skript bzw. ControlRunFetchMail per OLE verwendet wird.

Ergänze den Befehl »HamFetchMail« im Skript um die SSL-Parameter. Mindestens der »SSLMode« muss bei SSL-Benutzung immer angeben werden, da der Hamster die SSL-Einstellungen, die eventuell im Menü für diesen Server gemacht wurden, nicht übernimmt. Die anderen SSL-Parameter können weggelassen werden, dann nimmt der Hamster die Standardwerte (SSLVerify = 0; SSLCaFile = "").

```
HamFetchMail( ⟨server⟩, ⟨port⟩, ⟨user⟩, ⟨pass⟩, ⟨destuser⟩, ⟨filter⟩, _
              ⟨LeaveOnServer⟩, ⟨SSLMode⟩, ⟨SSLVerify⟩, ⟨SSLCaFile⟩ )
```

Per OLE musst Du jetzt den Befehl »ControlRunFetchMailTLS« statt »ControlRunFetchMail« verwenden.

```
function ControlRunFetchMailTLS( ⟨server⟩: WideString; ⟨port⟩: WideString;
⟨user⟩: WideString; ⟨pass⟩: WideString; ⟨destuser⟩: WideString;
⟨SSLMode⟩:Integer; ⟨SSLVerify⟩:Integer; ⟨SSLCaFile⟩: WideString ): Integer
```

Die Wörter in den spitzen Klammern müssen dabei ersetzt werden, siehe die Erläuterungen zu den Befehlen in der Hilfe.

Bemerkungen:

Glossar: Port POP3

- ⟨port⟩: Der Port, auf dem man den SSL-Server erreichen kann. Das ist nur dann noch 110 oder »pop3«, wenn der Server SSL/TLS auf dem Standardport unterstützt, ansonsten normalerweise 995.
- ⟨SSLVerify⟩: Wird dieser Parameter weggelassen, wird, wie oben kurz erwähnt, die Standardeinstellung »0« = »Zertifikatsüberprüfung abgeschaltet« verwendet. Das ist zugleich aber auch die unsicherste Variante!
- ⟨SSLCaFile⟩: Der Pfad zum Zertifikat für die Überprüfung. Wenn leer, wird der im Menü eingestellte Standard genommen. Siehe dazu auch Frage 7.

Beispiele Hamster-Skript:

```
# Abruf vom SSL-Port 995, dazu passend der »SSLMode« = 1:
#!hs2
HamFetchMail( "pop3-server.example.com", "995", "$42", "", _
              "admin", "", 0, 1, 3, "" )

# Abruf vom Standardport 110, dazu passend der »SSLMode« = 3:
#!hs2
HamFetchMail( "pop3-server.example.com", "110", "", "", _
              "admin", "", 0, 3, 3, "" )
```

Beispiele VBScript:

```
' Abruf vom SSL-Port 995, dazu passend der »SSLMode« = 1:
Set Hamster = Wscript.CreateObject( "Hamster.App" )
ControlRunFetchMailTLS( "pop3-server.example.com", "995", "", "",
                        "admin", "", 1, 3, "" )
Wscript.DisconnectObject Hamster

' Abruf vom Standardport 110, dazu passend der »SSLMode« = 3:
Set Hamster = Wscript.CreateObject( "Hamster.App" )
ControlRunFetchMailTLS( "pop3-server.example.com", "110", "", "",
                        "admin", "", 3, 3, "" )
Wscript.DisconnectObject Hamster
```

Damit werden Mails aus einem Postfach auf dem Server »pop3-server.example.com« über SSL abgeholt. Das Serverzertifikat wird überprüft, es liegt im Standardpfad.

Das wär's. Vergiss aber nicht, falls vorhanden, den Befehl zum Mailholen vom selben Server, aber nicht über SSL, zu löschen oder auszukommentieren. Das ist sonst ein Sicherheitsrisiko, da zusätzlich noch eine unverschlüsselte Verbindung aufgebaut wird und Dein Passwort wieder unverschlüsselt übertragen wird.

1.2.3. Wenn HamSendMail/HamSendMailAuth im Skript bzw. ControlRunSendMail/ControlRunSendMailAuth per OLE verwendet wird.

Ergänze den Befehl »HamSendMail« bzw. »HamSendMailAuth« im Skript um die SSL-Parameter. Mindestens »SSLMode« muss bei SSL-Benutzung immer angeben werden, da der Hamster die SSL-Einstellungen, die eventuell im Menü für diesen Server gemacht wurden, nicht übernimmt. Die anderen SSL-Parameter können weggelassen werden, dann nimmt der Hamster die Standardwerte (SSLVerify = 0; SSLCaFile = "").

```
HamSendMail( ⟨server⟩, ⟨port⟩, ⟨from-select⟩, ⟨to-select⟩, ⟨SSLMode⟩, _
             ⟨SSLVerify⟩, ⟨SSLCaFile⟩ )
HamSendMailAuth( ⟨server⟩, ⟨port⟩, ⟨user⟩, ⟨pass⟩, ⟨from-select⟩, _
                 ⟨to-select⟩, ⟨SSLMode⟩, ⟨SSLVerify⟩, ⟨SSLCaFile⟩ )
```

Per OLE musst du jetzt den Befehl »ControlRunSendMailTLS« statt »ControlRunSendMail« bzw. »ControlRunSendMailAuthTLS« statt »ControlRunSendMailAuth« verwenden.

```
function ControlRunSendMailTLS( ⟨server⟩: WideString; ⟨port⟩: WideString;
⟨from-select⟩: WideString; ⟨to-select⟩: WideString; ⟨SSLMode⟩: Integer;
⟨SSLVerify⟩: Integer; ⟨SSLCaFile⟩: WideString ): Integer

function ControlRunSendMailAuthTLS( ⟨server⟩: WideString;
⟨port⟩: WideString; ⟨user⟩: WideString; ⟨pass⟩: WideString;
⟨from-select⟩: WideString; ⟨to-select⟩: WideString; ⟨SSLMode⟩: Integer;
⟨SSLVerify⟩: Integer; ⟨SSLCaFile⟩: WideString ): Integer
```

Die Wörter in den spitzen Klammern müssen dabei ersetzt werden, siehe die Erläuterungen zu den Befehlen in der Hilfe.

Bemerkungen:

Glossar: Port SMTP

- ⟨port⟩: Der Port, auf dem man den SSL-Server erreichen kann. Das ist nur dann noch 25 oder »smtp«, wenn der Server SSL/TLS auf dem Standardport unterstützt, ansonsten normalerweise 465.
- ⟨SSLVerify⟩: Wird dieser Parameter weggelassen, wird, wie oben kurz erwähnt, die Standardeinstellung »0« = »Zertifikatsüberprüfung abgeschaltet« verwendet. Das ist zugleich aber auch die unsicherste Variante!
- ⟨SSLCaFile⟩: Der Pfad zum Zertifikat für die Überprüfung. Wenn leer, wird der im Menü eingestellte Standard genommen. Siehe dazu auch Frage 7.

Beispiel Hamster-Skript:

```
# Abruf vom SSL-Port 465, dazu passend der »SSLMode« = 1:
#!hs2
HamSendMail( "smtp-server.example.com", "465", ".*", ".*", 1, 3, "" )
# oder:
# HamSendMailAuth("smtp-server.example.com", "465", "", "", ".*", ".*", 1, 3, "")

# Abruf vom Standardport 25, dazu passend der »SSLMode« = 3:
#!hs2
HamSendMail( "smtp-server.example.com", "25", ".*", ".*", 3, 3, "" )
# oder:
# HamSendMailAuth("smtp-server.example.com", "25", "", "", ".*", ".*", 3, 3, "")
```

Beispiele VBScript:

```
' Abruf vom SSL-Port 465, dazu passend der »SSLMode« = 1:
Set Hamster = Wscript.CreateObject( "Hamster.App" )
```

```
Hamster.ControlRunSendMailTLS( "smtp-server.example.com", "465",
                               ".*", ".*", 1, 3, "" )
' oder:
' Hamster.ControlRunSendMailAuthTLS( "smtp-server.example.com", "465", "", "",
'                                    ".*", ".*", 1, 3, "" )
Wscript.DisconnectObject Hamster

' Abruf vom Standardport 25, dazu passend der »SSLMode« = 3:
Set Hamster = Wscript.CreateObject( "Hamster.App" )
Hamster.ControlRunSendMailTLS( "smtp-server.example.com", "25",
                               ".*", ".*", 3, 3, "" )
' oder:
' Hamster.ControlRunSendMailAuthTLS( "smtp-server.example.com", "25", "", "",
'                                    ".*", ".*", 3, 3, "" )
Wscript.DisconnectObject Hamster
```

Damit werden alle Mails über den Server »smtp-server.example.com« per SSL verschickt. Das Serverzertifikat wird überprüft, es liegt im Standardpfad.

Das wär's. Vergiss aber nicht, falls vorhanden, den Befehl zum Mailsenden zum selben Server, aber nicht über SSL, zu löschen oder auszukommentieren. Das ist sonst ein Sicherheitsrisiko, da zusätzlich noch eine unverschlüsselte Verbindung aufgebaut wird und dein Passwort wieder unverschlüsselt übertragen wird.

1.3. Wie kann ich meine News über eine SSL-Verbindung abholen und verschicken?
Trage alle Daten in das Menü ein, wie unter Frage 1 und 1.1 beschrieben. Wenn Du alles im Menü richtig eingetragen hast (eventuellen Portwechsel von 119 auf 563 beachten) und Du jetzt im Skript oder per OLE den Befehl gibst, News von diesem Server abzuholen, benutzt der Hamster automatisch SSL.

2. Hamster als SSL-Server
Um SSL mit den lokalen Servern vom Hamster zu verwenden, brauchst Du ein Zertifikat mit geheimem Schlüssel. Solltest Du keines haben, erscheint beim Start des lokalen Servers mit aktiviertem TLS eine Warnung im Log und der Server wird ohne TLS gestartet. Die Verbindung vom Client zum Hamster wird also nicht verschlüsselt! Wie Du ein eigenes Zertifikat erzeugen kannst, wird in Frage 6 beschrieben.

Nun zur Hamster-Konfiguration:

Im Hamster unter »Einstellungen« → »Grundeinstellungen« müssen eventuell temporär die erweiterten Einstellungen aktiviert werden und dann auf der Registerkarte »SSL« im Feld »Schlüsselpaar« die Datei, die Dein eigenes Zertifikat mit dem dazugehörigen privaten Schlüssel im PEM-Format (wird beim Erstellen, wie in Frage 6 beschrieben, automatisch benutzt) enthält, angegeben werden. Wenn Du den privaten Schlüssel mit einem Passwort geschützt hast, musst Du dieses Passwort ebenfalls eingeben. Wenn Du es nicht speichern lassen willst, kannst Du

Glossar: PEM

auch »?« angeben, dann fragt der Hamster nach jedem Start einmal nach dem Passwort. Die nächsten zwei Einstellfelder sind nur interessant, wenn Du Clientzertifikate benutzen willst.

Auf der Registerkarte des jeweiligen Servers unter »Einstellungen« → »Lokale Server« einstellen, dass TLS verwendet werden soll. Wenn alle Deine Clients TLS (eigentlich korrekt: SSL/TLS auf dem Standardport) unterstützen, solltest Du das Auswahlfeld auf »Erzwinge TLS« stellen, ansonsten auf »Erlaube TLS«. Bei letzterem wird aber nur dann SSL (also Verschlüsselung) verwendet, wenn der Client das auch unterstützt und so konfiguriert ist!

Ab dem nächsten Start des jeweiligen Servers, also beim Starten des Hamsters oder beim Benutzen des jeweiligen Eintrags im Menü »Lokale Server«, wird der Server mit SSL-Unterstützung gestartet.

Beachte bitte, dass es nicht möglich ist, Hamster als SSL-Server auf einem separaten SSL-Port zu benutzen. RFC 2595 (Link s. Frage 11), Abschnitt 7, nennt gute Gründe, warum eine solche Konfiguration nicht mehr verwendet werden sollte. Möchtest Du SSL trotzdem über einen separaten SSL-Port verwenden, brauchst Du dazu einen externen SSL-Wrapper, z. B. das Programm Stunnel. Stunnel muss dann an »0.0.0.0:⟨SSL-Port⟩« (je nach Protokoll) gebunden werden und eingehende verschlüsselte Verbindungen an »127.0.0.1:⟨Port⟩« (wenn der Hamster auf demselben Computer wie Stunnel läuft) weiterleiten. Das Zertifikat muss ebenfalls bei Stunnel angegeben werden. Den Hamster bindest Du dann an 127.0.0.1 und stellst ihn auf »Kein TLS«. Zu einer ausführlichen Erklärung lies dir die Hilfe zu Stunnel durch (siehe ebenfalls Frage 11).

Port

Beachte bitte, dass, wenn der Hamster als Server Verbindungen vom Internet annehmen soll, auf jeden Fall weitere Schutzmaßnahmen notwendig sind. In dieser Hinsicht hilft kein SSL, außer Du verwendest Client-Zertifikate. Also, für *jeden* Hamsteraccount starke Passwörter verwenden. Schalte in den Einstellungen der lokalen Server auf der Seite »Mail« »SMTP-AUTH« ein, wenn Du einen Mailreader hast, der dies beherrscht. Dabei muss der Mailreader sich mit einer gültigen Username-Passwort-Kombination beim Hamster authentifizieren, bevor er Mails verschicken kann. Wenn Dein Mailreader dies nicht kann, schalte »SMTP-after-POP3« ein. Dabei muss der Mailreader, bevor er Mails verschicken kann, Mails abholen. Wenn Du keines von beiden machst, kann jeder, ohne ein Passwort kennen zu müssen, Spam über Deinen Rechner verschicken. Erstelle auch einen Benutzer mit dem Usernamen »nntpdefault« und dem Passwort »*« (ein Stern). Dieser Benutzer wird immer dann verwendet, wenn der Client keinen Benutzernamen und kein Passwort beim lokalen Newsserver angibt. In den Einstellungen dieses Benutzers darf dann in keinem Eingabefeld mit weißem Hintergrund außer »voller Name« irgendetwas stehen und der Benutzer darf keine Mailbox haben, so dass dieser Benutzer (und damit auch ein eventueller Spammer oder Angreifer ohne Benutzernamen und Passwort) gar nichts machen kann. Es wäre zwar nicht unbedingt notwendig, diesen Benutzer einzurichten, da sich der Hamster ohne diesen ebenso verhält, aber es ist trotzdem zu empfehlen, damit nicht später einmal dieser Benutzer mit zu vielen Berechtigungen eingerichtet wird und dabei eine große Sicherheitslücke entsteht.

SMTP-AUTH

SMTP
POP3

Zur Einrichtung von Client-Zertifikaten siehe Frage 8.2.1.

3. Wie erhöhe ich die Sicherheit bei SSL?
Für eine SSL-Verbindung braucht OpenSSL und damit der Hamster, denn der benutzt die OpenSSL-Biliotheken, möglichst zufällige Zufallsdaten. Dazu wird automatisch die Datei »randseed.!!!« im Hamsterverzeichnis angelegt, in der die Zufallsdaten aus dem internen Pseudozufallszahlengenerator (PRNG) von OpenSSL gespeichert werden. Wenn Du eine Quelle guter Zufallsdaten hast, die du stattdessen verwenden möchtest, kannst Du unter »Einstellungen« → »Grundeinstellungen« auf der Registerkarte »SSL« im Feld »Externe Datei mit Zufallsdaten für PRNG« eine andere Datei angeben, aus der Zufallsdaten gelesen werden. Bitte beachte dabei, dass Hamster in die Datei »randseed.!!!« nach jeder Benutzung neue Zufallsdaten schreibt, nicht aber in die zusätzlich angegebene Datei.

Vielleicht solltest Du auch mal über Server- und Client-Zertifikate nachdenken. Siehe dazu Frage 4 und folgende bis inkl. Frage 8.

4. Was sind Zertifikate und wozu brauche ich sie?
Zertifikate kann man vielleicht mit einem Pass vergleichen. Du zeigst es/ihn vor, wenn Du willst, dass andere sich darauf verlassen können, dass Du wirklich der bist, der Du zu sein vorgibst. Dafür bürgt jemand, dem Du traust: eine Certificate Authority (CA) oder der Staat. Du kannst zwar nicht wie bei einem Pass mit dem Bild prüfen, ob das Zertifikat wirklich dem gehört, der es benutzen will, aber das macht nichts, da der andere ohne den zum öffentlichen Schlüssel (= Zertifikat) passenden geheimen Schlüssel oder eine sehr, sehr große Rechenpower nichts entschlüsseln kann, was Du an den Inhaber des Zertifikats schickst.

Self-Signed-Zertifikate musst Du dir vorstellen wie den Pass eines Königs: Der Ausstellende (hier: der Staat = König) bürgt für sich selbst. Du brauchst also Zertifikate, wenn Du sichergehen willst, dass der, mit dem Du sprichst, wirklich der ist, mit dem Du sprechen willst, und nicht irgendein x-beliebiger Betrüger. Technisch gesehen ist ein Zertifikat ein öffentlicher RSA-Schlüssel. Weil im Normalfall nur der richtige Server den dazu passenden privaten Schlüssel hat, weißt Du, wenn der Server Deine Texte richtig entschlüsseln kann, es ist der richtige.

Voraussetzung dafür ist natürlich, dass Du das richtige Zertifikat (= öffentlicher Schlüssel) hast. Wenn Du dir dieses über das Internet holst, wie in Frage 5 beschrieben, kannst Du dir dessen erstmal *nicht* sicher sein! Eine Möglichkeit, eine größere Gewissheit zu bekommen, sind Signaturen. Nehmen wir an, es hat schon mal jemand (z. B. eine Certificate Authority = CA) überprüft, ob dieses Zertifikat echt ist, und dann das Zertifikat signiert. Wenn Du jetzt den (garantiert) richtigen öffentlichen Schlüssel von diesem Jemand besitzt und diesem vertraust, dass er z. B. keine gefälschten Zertifikate signiert, kannst Du dir auch über das Zertifikat sicher sein.

Nochmal eine Zusammenfassung: Ein Zertifikat ist normalerweise dazu da, dass man sich sicher sein kann, mit wem man spricht. Dies ist z. B. beim Serverzertifikat der Fall. Ein Zertifikat einer

CA ist dazu da, die Richtigkeit eines anderen Zertifikats zu bestätigen. Ein Self-Signed-Zertifikat ist ein beliebiges Zertifikat, dessen Richtigkeit nur vom Aussteller bestätigt wird, nicht noch zusätzlich von jemand anderem. Wenn man ein solches bekommt, muss man dieses Zertifikat also genau prüfen, während es normalerweise ausreicht, das Zertifikat der CA zu überprüfen, weil diese ja schon das Serverzertifikat überprüft hat.

5. Wo bekomme ich fremde Zertifikate her?

5.1. CA-Zertifikate

Wenn Du schon weißt, welche CA das Serverzertifikat signiert hat, kannst Du die nächsten Schritte überspringen. Ansonsten brauchst Du das Serverzertifikat (s. Frage 5.2) und OpenSSL (s. Frage 10), um das herauszufinden. Rufe OpenSSL mit folgender Befehlszeile auf:

```
openssl x509 -issuer -noout -in ⟨Serverzertifikat.pem⟩
```

Die Ausgabe sollte dann den Namen der CA enthalten. Eine Gesamtübersicht über das Zertifikat erhältst Du mit:

```
openssl x509 -text -noout -in ⟨Serverzertifikat.pem⟩
```

Wenn Du den Namen der CA hast, brauchst Du noch deren Zertifikat. Meistens bekommst Du es auf deren Website. Die am häufigsten verwendeten sind auch bei den meisten Browsern mitgeliefert. Wenn Du dem Browserhersteller vertraust und dir sicher bist, dass Du wirklich die Originalversion des Browsers hast, kannst Du sie auch von dort exportieren:

- Die beim *Internet Explorer* mitgelieferten Zertifikate siehst Du unter »Systemsteuerung« ⇒ »Internetoptionen« → Registerkarte »Inhalte« → Button »Zertifikate«. Beim Exportieren des Zertifikats muss das Format »Base64-codiert X.509 (.CER)« gewählt werden. Der Dateiname ist egal, der Übersichtlichkeit halber ist aber die Endung »`.pem`« (von PEM) zu empfehlen (der Dateiname muss in doppelte Anführungszeichen gesetzt werden, sonst wird in jedem Falle die Endung »`.cer`« gewählt).

  Glossar: Base64 X.509 PEM

- Im *Mozilla Firefox* musst Du auf »Extras« → »Einstellungen« ⇒ Reiter »Erweitert« → Reiter »Zertifikate« → Button »Zertifikate anzeigen« gehen (bei ausgeblendeter Menüleiste musst Du zunächst das Menü entweder durch Klick auf den »Firefox«-Button oder durch Drücken der Alt-Taste einblenden und dann den Menüpunkt »Einstellungen« auswählen). Beim Exportieren des Zertifikats muss der Dateityp »X.509-Zertifikat (PEM)« ausgewählt werden.

- In *Opera bis Version 12* findest Du die Zertifikate im Menü »Extras« → »Einstellungen« ⇒ Reiter »Erweitert« → Abschnitt »Sicherheit« → Button »Zertifikate verwalten« (bei ausgeblendeter Menüleiste musst Du das Menü zunächst entweder durch Klick auf den »Opera«-Button oder durch Drücken der Alt-Taste einblenden und dann den Menüpunkt »Einstel-

lungen« auswählen). Beim Exportieren muss der Dateityp »PEM-Datei (*.pem)« ausgewählt werden.[1]

Glossar: PEM

- In *Opera seit Version 15*, *Google Chrome*, *SRWare Iron* und vermutlich allen anderen Browsern, die die HTML-Rendering-Engine Blink verwenden,[2] wird letztendlich die Zertifikatsverwaltung von Windows aufgerufen, so dass auch von Anfang an die oben für den Internet Explorer beschriebene Vorgehensweise genutzt werden kann.

Remo Müller bietet unter `http://code.google.com/p/anytools/HamCert` Rootzertifikate häufig genutzter Mail- und Usenetserver in der Form an, dass sie entsprechend Frage 7.2 direkt eingesetzt werden können.

Beachte jedoch unbedingt, dass Du dir der Echtheit der Zertifikate so *nicht* sicher sein kannst. Du solltest daher unbedingt den Fingerprint des Zertifikats, den dir OpenSSL mit dem Parameter »`x509 -info`« anzeigt, überprüfen. Das kann z. B. per Telefon geschehen oder Du besuchst ein Büro der jeweiligen Firma.

5.2. Serverzertifikate

Manchmal findest Du das Serverzertifikat auf der Website des Serverbetreibers oder Du holst es dir mit der »`OpenSSL.exe`« (s. Frage 10) direkt vom Server.

Wenn der Server SSL auf einem separaten Port unterstützt, starte OpenSSL, während eine Onlineverbindung besteht, mit dieser Kommandozeile:

Port

```
openssl s_client -connect ⟨Server⟩:⟨Port⟩ -showcerts > OpenSSL.log
```

Für »`⟨Server⟩:⟨Port⟩`« gib die Adresse und den Port des SSL-Servers an, dessen Zertifikat du bekommen willst.

Bei einem POP3- oder SMTP-Server, der SSL/TLS auf dem Standardport unterstützt, bekommst du das Zertifikat so (in der Eingabeaufforderung *eine* Zeile):

POP3 SMTP

```
openssl s_client -connect ⟨Server⟩:⟨Port⟩ -starttls ⟨Protokoll⟩
        -showcerts > OpenSSL.log
```

Wenn Du noch eine ältere Version von OpenSSL benutzt (*was nicht zu empfehlen ist!*), dann unterstützt diese leider den »`-starttls`«-Parameter nur für SMTP. Bei einem anderen Protokoll musst du daher dann zuerst eine Klartextverbindung aufbauen und die Verschlüsselung händisch starten. Von Martin Germann gibt es das Programm SSL_Tool, das genau das für die Protokolle SMTP, POP3, NNTP und IMAP automatisch macht (Achtung, Direktlink!):

NNTP IMAP

```
http://www.ximera.de/hamster/SSL_Tool.zip
```

Auch bei Verwendung dieses Programms benötigst Du OpenSSL. Starte das Programm, trage die Daten des Servers unter »Remote« und einen freien lokalen Port unter »local« ein, und starte den Wrapper mit einem Klick auf »Start Wrapper«. Jetzt verbinde dich mit OpenSSL auf dem lokalen Port, den Du angegeben hast:

Glossar: Port

[1] Frühere Versionen von Opera unterstützten dieses Format nicht, dann musste das Zertifikat mit Hilfe der »`OpenSSL.exe`« (s. Frage 10) aus dem »CER«-Format in das »PEM«-Format gewandelt werden.

[2] Vgl. Blink (Software) in der Wikipedia.

```
openssl s_client -connect localhost:⟨Port⟩ > OpenSSL.log
```

Das Programm handelt daraufhin eine TLS-Verbindung mit dem angegebenen Server aus und tunnelt dann die Verbindung zu OpenSSL.

Öffne anschließend die Datei »OpenSSL.log« im OpenSSL-Verzeichnis mit einem beliebigen Texteditor und kopiere alles von einschließlich der Zeile

```
-----BEGIN CERTIFICATE-----
```

bis zu einschließlich der Zeile

```
-----END CERTIFICATE-----
```

in eine neue Datei. Gib dieser Datei einen möglichst aussagekräftigen Namen. Der Übersichtlichkeit halber ist zu empfehlen, für alle Zertifikatsdateien mit diesem Format die Endung ».pem« zu nehmen (von PEM). PEM

Bei einigen Servern erfolgt die Zertifizierung mehrstufig. Daher werden (bei korrekter Serverkonfiguration) auch mehrere Zertifikate ausgeliefert, die allesamt abgespeichert werden müssen. Das Serverzertifikat ist dabei das zuerst aufgeführte.

Die »OpenSSL.log« kannst Du anschließend löschen.

Remo Müller bietet unter http://code.google.com/p/anytools/HamCert Serverzertifikate häufig genutzter Mail- und Usenetserver in der Form an, dass sie entsprechend Frage 7.2 direkt eingesetzt werden können.

Genau wie beim CA-Zertifikat kannst Du dir jedoch so *nicht* sicher sein, dass Du das richtige Zertifikat hast. Du kannst jedoch die Signatur mit dem CA-Zertifikat überprüfen, oder du prüfst wieder den Fingerprint über andere Kanäle wie Telefon oder persönliches Erscheinen.

6. Wie erstelle ich mir ein eigenes Zertifikat?

Als erstes ein kurzer Hinweis: Du benötigst meist kein eigenes Zertifikat, wenn Du nur SSL-Verbindungen zu einem fremden Server wie Web.de oder GMX aufbauen willst, da diese meistens keine Clientzertifikate unterstützen.

Zum Zertifikat-Erstellen benötigst Du OpenSSL (s. Frage 10) sowie die 2 Dateien »MakeCert.bat« und »cert.cnf« von Martin Germann, die Du unter folgendem URL bekommst (Achtung, Direktlink!):

http://www.ximera.de/hamster/MakeCert.zip

Kopiere die beiden Dateien aus dem Zip-Archiv in das Verzeichnis von »OpenSSL.exe«, führe »MakeCert.bat« aus und folge den Anweisungen. Das Passwort sollte allerdings wegen möglichen Kompatibilitätsproblemen keine Umlaute und Sonderzeichen enthalten. Der öffentliche Schlüssel (= Zertikat) und ein dazugehöriger privater RSA-Schlüssel befindet sich dann in der Datei »certificate.pem« im OpenSSL-Verzeichnis.

Das Zertifikat alleine kann bedenkenlos weitergegeben werden, aber der RSA-Schlüssel nicht! Wenn ein Angreifer beides hat, kann er so tun, als wäre er Du und die ganzen Zertifikate wären wirkungslos. Also auf keinen Fall die Zeilen

```
-----BEGIN RSA PRIVATE KEY-----
...
----END RSA PRIVATE KEY-----
```

weitergeben. Auf dem lokalen Computer müssen sie natürlich trotzdem vorhanden sein.

Achtung!
Der RSA-Schlüssel darf *nur* auf dem Computer sein, der sich mit diesem Zertifikat authentifizieren soll, also auf dem Server bei Verwendung von Serverzertifikaten und auf dem Client bei Verwendung von Clientzertifikaten. Gib ihn deshalb nicht weiter! Gelangen der Schlüssel und das Zertifikat in fremde Hände, kann der Dritte ohne irgendein Problem so tun, als wäre er Du! Der ganze Sinn von Zertifikaten wäre also hinfällig.

Zum Einbinden des Zertifikates in den Hamster siehe Frage 8.1.1 (als Serverzertifikat) oder Frage 8.1.2 (als Clientzertifikat).

7. Wie mache ich dem Hamster fremde Zertifikate bekannt?
Beide hier beschriebenen Methoden sind technisch gleichwertig, jede hat ihre Vor- und Nachteile. Die Auswahl bleibt daher dem Anwender überlassen.

Grundsätzlich gilt, dass, wenn in den Servereinstellungen oder im Skript keine Angaben gemacht werden, die Standardpfade aus den entsprechenden Feldern auf der Registerkarte »SSL« in den Grundeinstellungen des Hamsters übernommen werden. In den Grundeinstellungen darf im Feld »Datei mit Zertifikaten zwecks Überprüfung« nur eine einzelne Datei (s. Frage 7.1) und im Feld »Pfad mit Zertifikaten zwecks Überprüfung« nur ein Verzeichnis (s. Frage 7.2) stehen. Das nicht benutzte Feld kann leergelassen werden. In dem Feld im Servermenü oder im Parameter »⟨SSLCaFile⟩« im Skript kann entweder eine einzelne Datei oder ein ganzes Verzeichnis angegeben werden.

7.1. Alle Zertifikate in einer Datei
Bei nur einem oder wenigen Zertifikaten ist es am einfachsten, diese alle in eine Datei zu kopieren. Diese sieht dann z. B. so aus:

```
-----BEGIN CERTIFICATE-----
...
-----END CERTIFICATE-----

-----BEGIN CERTIFICATE-----
...
-----END CERTIFICATE-----
```

Der Dateiname und der Ort der Datei ist egal, der Übersichtlichkeit wegen ist aber ein möglichst aussagekräftiger Name mit der Endung »`.pem`« zu empfehlen (von PEM).

Glossar: PEM

Außer den Zertifikaten kann auch noch anderer Text in der Datei stehen, z. B. um zu notieren, zu welchem Server welches Zertifikat gehört, aber nur außerhalb der Zeilen von »`-----BEGIN CERTIFICATE-----`« bis »`-----END CERTIFICATE-----`«. Um möglichen Problemen aus dem Weg zu gehen, sollte der eigene Text auch nur aus ASCII-Zeichen bestehen, d. h. keine Umlaute usw. enthalten.

Das könnte dann so aussehen:

```
news.example.com:
-----BEGIN CERTIFICATE-----
...
-----END CERTIFICATE-----

pop3.example.com:
-----BEGIN CERTIFICATE-----
...
-----END CERTIFICATE-----

smtp.example.com:
-----BEGIN CERTIFICATE-----
...
-----END CERTIFICATE-----

CA-Zertifikat von ⟨Zertifizierungsstelle⟩ fuer example.com:
-----BEGIN CERTIFICATE-----
...
-----END CERTIFICATE-----
```

Im Hamster wird dann an der entsprechenden Stelle der Name der Datei inklusive komplettem Pfad angegeben.

In Abwandlung dieses Vorgehens kann man z. B. die Zertifikate für alle Server eines Providers und das dazugehörende CA-Zertifikat (denn gewöhnlich ist es je Provider nur eines) in je einer Datei speichern und diese in den Einstellungen für die dazugehörigen Server angeben.

Vorteil: Vollständige manuelle Pflege der Zertifikate möglich.

Nachteil: Manuelle Pflege mit deutlich größerem Aufwand als in Methode 2 notwendig. Vor allem, wenn *alle* Zertifikate in *einer* Datei abgelegt werden, wird diese schnell unübersichtlich. Letzteres wird durch die Aufteilung in mehrere Dateien aber deutlich abgemildert.

7.2. Alle Zertifikate in einem Verzeichnis

Mit vielen Zertifikaten kann eine Datei leicht unübersichtlich werden. Dann empfiehlt sich

diese Methode: Alle Zertifikate stehen jeweils einzeln in einer Datei. Damit der Hamster sie findet, müssen sie im gleichen Verzeichnis liegen und ein bestimmtes Dateinamensformat haben, nämlich »⟨Hash-des-Zertifikats⟩.0«. Von Martin Germann gibt es das Skript »SSL-Cert-Hash«, das mit Hilfe der »OpenSSL.exe« (s. Frage 10) allen Zertifikaten in einem Verzeichnis, die in ».pem«-Dateien vorliegen, automatisch den richtigen Dateinamen gibt (Direktlink):

http://www.ximera.de/hamster/SSL-Cert-Hash.hsc

Im Hamster wird dann nur der Pfad zu diesem Verzeichnis angegeben. Natürlich funktioniert diese Methode auch dann, wenn Du nur ein Zertifikat hast.

Vorteil: Sehr wenig Aufwand nötig, einfach aktuelle Zertifikate im entsprechenden Format ins Verzeichnis ablegen.

Nachteil: Das Aufräumen des Verzeichnisses wird durch das kryptische Dateinamensformat massiv erschwert, vor allem, wenn die jeweils dazugehörige Textdatei gelöscht wurde. Es muss in jeder einzelnen Textdatei nachgelesen werden, ob das dazugehörige Zertifikat noch gültig ist. Fehlt die Textdatei, muss die entsprechende Zertifikatsdatei wieder die Endung ».pem« erhalten und noch einmal mit »SSL-Cert Hash« umgewandelt werden.

8. Wie benutze ich Zertifikate mit Hamster und SSL?

Mit Serverzertifikaten stellt der Client sicher, dass er mit dem richtigen Server spricht. Das wird z. B. dann verwendet, wenn der Hamster einen fremden Server abfragen soll (s. Frage 8.1.2) oder als TLS-Server mit einem Zertifikat versehen werden soll (s. Frage 8.1.1). Clientzertifikate werden verwendet, wenn der Server sicherstellen will, dass er mit dem richtigen Client spricht (also das, wozu normalerweise Benutzername und Passwort verwendet werden). Das wird von externen Servern leider selten unterstützt.

Du kannst natürlich auch ohne Probleme Serverzertifikate und Clientzertifikat gleichzeitig verwenden.

8.1. Serverzertifikate

8.1.1. Einrichten mit dem Hamster als Server

Wie Du ein eigenes Zertifikat erstellen kannst, steht in Frage 6. Eingebunden wird es, indem Du den Dateinamen bei temporär aktivierten erweiterten Einstellungen in den Grundeinstellungen auf der Registerkarte »SSL« in der ersten Box eingibst. Außerdem musst Du eventuell noch das Passwort angeben. Dann authentifiziert sich der Hamster dem Client gegenüber automatisch.

8.1.2. Einrichten mit dem Hamster als Client

Dabei gibt es zwei Stufen. Die erste Stufe überprüft nur das Zertifikat der CA und die Signatur dieser CA auf dem Serverzertifikat. Das Serverzertifikat selber wird nicht überprüft. Wenn Du der CA vertraust, reicht das normalerweise aus. Wenn Du allerdings der CA nicht vertraust, solltest Du unbedingt die zweite Stufe nutzen, dabei ist das Vertrauen in die CA nicht mehr so wichtig. Du solltest dann aber das Serverzertifikat besonders genau überprüfen, auf jeden Fall auch durch einen Kanal außerhalb des Internets. Die zweite Stufe baut auf der ersten auf und überprüft zusätzlich noch das Serverzertifikat selber. Wie Du ein Zertifikat von einer CA oder von einem externen Server bekommst, steht in Frage 5.

8.1.2.1. Überprüfen des CA-Zertifikates

Wenn Du nun das Zertifikat der CA hast, sagst Du dem Hamster, dass er es beim Aufbau der Verbindung überprüfen soll. Dazu setzt Du im Skript den Parameter »⟨SSLVerify⟩« auf 2 oder im Menü die Auswahlbox »Wähle Überprüfungsverfahren« auf »Zertifikate immer überprüfen«. Was Du in das Feld »Überprüfung über Datei« eingeben musst, ist in Frage 7 erklärt.

8.1.2.2. Überprüfen des Serverzertifikats per lokaler Kopie

Im Skript setzt Du nun den Parameter »⟨SSLVerify⟩« auf 3 oder im Menü die Auswahlbox »Wähle Überprüfungsverfahren« auf »Peer mittels lokal vorhandener Zertifikate überprüfen«. Was Du in das Feld »Überprüfung über Datei« eingeben musst, steht in Frage 7. Beachte aber, dass Du dem Hamster jetzt zwei Zertifikate (das Server- und das CA-Zertifikat) auf einmal bekannt machen musst.

8.2. Clientzertifikate

8.2.1. Einrichten mit dem Hamster als Server

Du benötigst auf dem Server das Zertifikat des Clienten (nicht jedoch dessen privaten Schlüssel!), der sich beim Server authentifizieren soll. Außerdem muss der Hamster, wie in Frage 2 beschrieben, vollständig für den Betrieb als TLS-Server eingerichtet sein.

In den Grundeinstellungen auf der Registerkarte »SSL« aktivierst Du eventuell temporär die erweiterten Einstellungen und gibst entweder bei »Datei mit Zertifikaten zwecks Überprüfung« eine Datei an, die alle Zertifikate gesammelt enthält oder bei »Pfad mit Zertifikaten zwecks Überprüfung« ein Verzeichnis, in dem alle Zertifikate in Einzeldateien liegen. Siehe dazu auch Frage 7.

Jetzt kann der Hamster zwar die Zertifikate vergleichen, aber er fängt mit dem Ergebnis noch nichts an. Damit der Hamster nur Verbindungen mit gültigem Clientzertifikat akzeptiert, musst Du in der »`hamster.ini`« im Abschnitt [SSL] (siehe auch »Hamster.ini, sonstige Sektionen«) den Eintrag »VerifyLevel« auf 2 oder 3 stellen.

8.2.2. Einrichten mit dem Hamster als Client

Wie Du ein eigenes Zertifikat erstellen kannst, steht in Frage 6. Eingebunden wird es, indem Du den Dateinamen bei temporär aktivierten erweiterten Einstellungen in den Grundeinstellungen auf der Registerkarte »SSL« in der ersten Box eingibst. Außerdem musst Du eventuell noch das Passwort angeben.

9. Was bedeuten diese Fehlermeldungen?

```
[...] Verify error: unable to get local issuer certificate depth=0 [...]
[...] OpenSSL error: SSL_connect: error:14090086:SSL
                     routines:SSL3_GET_SERVER_CERTIFICATE:certificate verify failed
```

Wenn diese Fehlermeldungen auftauchen, ist höchstwahrscheinlich »Zertifikate überprüfen« eingestellt (oder der Parameter »⟨SSLVerify⟩« auf ≠ 0 gesetzt), aber der Hamster kann das Zertifikat der CA nicht finden. Stelle sicher, dass Du es hast, dass es im richtigen Format vorliegt (s. Frage 7) und dass alle Pfade stimmen. Wenn Du keine Möglichkeit hast, das Zertifikat der CA zu bekommen, musst Du die Zertifikatsüberprüfung leider ausschalten.

```
[...] OpenSSL error: SSL_CTX_new: error:140A90A1:SSL routines:SSL_CTX_new:library
                     has no ciphers
[...] Creating SSL context failed
```

Hier konnte OpenSSL nicht initialisiert werden. Wahrscheinlich ist es ein Speicherproblem. Prüfe, ob Du mehrere SSL-Verbindungen gleichzeitig aufbaust (auch mehrere Threads zu einem Newsserver sind verschiedene Verbindungen!) und stelle auf weniger bis eine gleichzeitige SSL-Verbindung um.

10. Wo bekomme ich OpenSSL her?

Den OpenSSL-Sourcecode gibt es auf `http://www.openssl.org/`.

Eine Anleitung zum Kompilieren unter Windows ist darin enthalten.

Prinzipiell solltest Du OpenSSL selber kompilieren, denn nur so kannst Du sicher sein, dass derjenige, von dem das Binary kommt, keine Schadfunktion oder einen Trojaner eingefügt hat.

Für diejenigen, die sich OpenSSL nicht selber kompilieren können, gibt es zum Zeitpunkt des Schreibens das »Shining Light Productions – Win32 OpenSSL« unter `http://www.slproweb.com/products/Win32OpenSSL.html`. Auch einige netzwerkbezogene Software wird mit kompilierten OpenSSL-Binaries ausgeliefert, allerdings sind diese nicht immer auf dem aktuellsten Stand, was unter Umständen ein Sicherheitsproblem darstellt.

Unter `http://de.wikipedia.org/wiki/OpenSSL` ist mit hoher Wahrscheinlichkeit immer eine aktuelle Downloadmöglichkeit zu finden.

11. Was für interessante URLs gibt es zu diesem Thema?
Als erstes nochmal die Download-URLs:

- OpenSSL-Sourcecode: `http://www.openssl.org/`
- Shining Light Productions – Win32 OpenSSL: `http://www.slproweb.com/products/Win32OpenSSL.html`
- Skript zur Zertifikatserstellung: `http://www.ximera.de/hamster/MakeCert.zip` (Direktlink!)
- Skript »SSL-Cert-Hash« zur richtigen Benennung der Zertifikatsdateien im Verzeichnis (s. Frage 7.2): `http://www.ximera.de/hamster/SSL-Cert-Hash.hsc` (Direktlink!)
- SSL_Tool (OpenSSL mit TLS nutzen): `http://www.ximera.de/hamster/SSL_Tool.zip` (Direktlink!)
- Root- und Serverzertifikate, angeboten von Remo Müller: `http://code.google.com/p/anytools/HamCert`

URLs zum Einlesen:

- Wikipedia: »Transport Layer Security« (TLS): `http://de.wikipedia.org/wiki/Transport_Layer_Security`
- Bundesamt für Sicherheit in der Informationstechnik: `http://www.bsi.bund.de/`
- DFN-PKI, Zertifizierungsinstanz für das Deutsche Forschungsnetz: `https://www.pki.dfn.de/`
- TeleTrusT, Verein zur Förderung der Vertrauenswürdigkeit von Informations- und Kommunikationstechnik: `http://www.teletrust.de/`
- OpenSSL: `http://www.openssl.org/`
- ModSSL: `http://www.modssl.org/`
- RFC-Sammlung: z. B. unter `http://www.rfc-editor.org/`
- Stunnel: `http://www.stunnel.org/`

Achtung! Der in früheren Versionen der SSL-FAQ aufgeführte Netzauftritt des Entwicklers der OpenSSL-Einbindung in den Hamster (`http://www.ximera.de/hamster.html`) ist leider weitgehend veraltet, insbesondere die Links für die OpenSSL-Binaries und die Anleitung zum Selbstkompilieren.

Newsgroups:

- Allgemeine Diskussionen rund um Computersicherheit: `news:de.comp.security.misc`
- SSL in Verbindung mit dem Hamster: `news:hamster.de.config`

(*Letzte von Philipp Wendler bearbeitete Version 3.5 dieser FAQ:* `http://www.philippwendler.de/hamster/hamster-ssl-classic.html`. **Achtung: zum Teil veraltet!**)

Kann ich den Hamster über einen Router betreiben?

Der Router sorgt dafür, dass der Hamster eine permanente Verbindung zum Netz hat. Entweder hast Du eine Standleitung, die natürlich über einen »richtigen« Router geht, oder Du hast einen Dial-in-Router, der bei Bedarf die Verbindung herstellt. Für die angeschlossenen Systeme sieht es in beiden Fällen wie eine permanente Verbindung aus.

Der Hamster kann hier den Aufbau der Verbindung nicht direkt beeinflussen. Daher musst Du in Deinen Skripten die Befehle zum Wählen und Trennen einfach weglassen. Wenn Du keine Skripte benutzt, dann geht das natürlich auch. Die Menübefehle funktionieren ebenfalls.

Mit folgenden Skripten kann man testen, ob eine Verbindung ins Internet möglich ist, zunächst als Subroutine, die in andere Skripte einzubetten ist, dann als eigenständiges Skript.

- Subroutine:

```
sub CheckbyPing
# Note: Unter Windows 2000 ist »ping« nicht verwendbar, da es dort immer dasselbe
# Errorlevel zurückgibt. Als Alternative bietet sich »Fping« (»Fast Pinger«) an
# (Freeware), Weblink: http://www.kwakkelflap.com/fping.html, die »Execute«-
# Funktion muss dann ein wenig angepasst werden, s. unten.

  varset ( $rechner , "Name.oder.IP-Adresse.des.Referenzrechners.angeben" )
  varset ( $response , $try, 1 )
  varset ( $maxtry , 3 )

  if ($rechner = "Name.oder.IP-Adresse.des.Referenzrechners.angeben" )
    warning ( "LART: " + $rechner )
    warning ( "Niemals ein Skript starten, ohne es sich vorher anzusehen!" )
    quit
  endif

  while ($response = 1)
    Execute ("ping -n 1 " + $rechner, "" , 0 , True, $response)
    # Execute ("fping " + $rechner + " -n 1", "" , 0 , True, $response)
    If ( $try = $maxtry )
      warning ( $rechner + " kann aktuell nicht gefunden werden!" )
      quit
    else
      print ( $try + " Ping an " + $rechner + " geschickt." )
      inc ($try)
    endif
  endwhile
print ( $rechner + " hat auf PING reagiert." )
endsub
```

- Eigenständiges Skript:

```
# Online-Check.hsc

# Verwendet eine Komponente des MS Internet Explorer, funktioniert laut
# Dokumentation zur »wininet.dll« (WinINet Reference) erst ab Windows 2000
#
```

```
# Prüft erst, ob der IE sich im Offline-Modus befindet. Wenn ja, kommt eine Anfra-
# ge, ob man online gehen will. Anschließend wird auf Erreichbarkeit einer Adresse
# getestet, die am Anfang dort einzutragen ist, wo in der Voreinstellung die
# Startseite der deutschsprachigen Wikipedia steht.

#!hs2
varset( $testURL,"http://de.wikipedia.org/" )
#                ^^^^^^^^^^^^^^^^^^^^^^^^
#                 Hier Adresse eintragen.

if InternetGetConnectedStateEx($testURL)
  AddLog( "* InternetGetConnectedStateEx: ++ OK!", $logOK )
else
  AddLog( "* InternetGetConnectedStateEx: -- Failed!",$logFail )
endif

# Die Message-Boxen können auch auskommentiert werden.
if CheckOpenUrl($testURL)>0
  MsgBox ("Test-URL " + $testURL + " erreichbar.", "Online-Check: Online", _
          0x40|0x0)
  AddLog( "* CheckOpenUrl: ++ online!", $logOK )
else
  MsgBox ("Test-URL " + $testURL + " NICHT erreichbar!", "Online-Check: Offline", _
          0x30|0x0)
  AddLog( "* CheckOpenUrl: -- offline!",$logFail )
endif

sub InternetGetConnectedStateEx($strUrl)
  varset( $buf, MemAlloc(len($strUrl)+1) )
  MemSetStr( $buf, $strUrl )
  varset( $p1, MemAlloc(4) )
  MemSetInt( $p1, 0 )
  varset( $p2, MemAlloc(4) )
  MemSetInt( $p2, 0 )

  varset( $result, DllCall($InternetGetConnectedStateEx,$p1,$p2,$buf,0) )

  varset( $dwFlags, MemGetInt($p1) )
  varset( $inetConnection, "<unknown>" )
  if ($p2)
    $inetConnection= MemGetStr($p2)
  endif
  MemFree($p2)
  MemFree($p1)
  MemFree($buf)

  print( "* InternetGetConnectedStateEx( 0x" + hex($dwFlags,2) + ", " _
        + $inetConnection + "): " + iif($result,"TRUE","FALSE") )
  if ( ($dwFlags & 0x02 ) = 0 )
    print( "Lokales LAN ist offline. Check Hardware." )
    return(0)
  endif

  if ( $dwFlags & 0x20 )
    print( "IE ist im Offline-Modus." )
```

```
    return( InternetGoOnline($strUrl) )
  endif
  return($result)
endsub

sub InternetGoOnline($strUrl)
  varset( $buf, MemAlloc(len($strUrl)+1) )
  MemSetStr($buf, $strUrl)
  varset( $result, DllCall($InternetGoOnline,$buf,0,0) )
  MemFree($buf)
  return($result)
endsub

sub CheckOpenUrl($strUrl)
  var($io_h, $iou_h)
  varset( $buf, MemAlloc(len($strUrl)+1) )
  MemSetStr($buf, $strUrl)
  $io_h = DllCall( $InternetOpen, 0, 1, 0, 0, 0 )
  $iou_h = DllCall( $InternetOpenUrl, $io_h, $buf, 0, 0, 0x80000000, 0 )
  DllCall( $InternetCloseHandle, $io_h )
  MemFree($buf)
  return($iou_h)
endsub

#!initialize
  varset( $InternetOpen, "wininet.dll|InternetOpenA|n|nnnnn" )
  varset( $InternetOpenUrl, "wininet.dll|InternetOpenUrlA|n|nnnnnn" )
  varset( $InternetCheckConnection, _
         "wininet.dll|InternetCheckConnectionA|n|nnn" )
  varset( $InternetGetConnectedStateEx, _
         "wininet.dll|InternetGetConnectedStateExA|n|nnnn" )
  varset( $InternetGoOnline, "wininet.dll|InternetGoOnlineA|n|nnn" )
  varset( $InternetCloseHandle, "wininet.dll|InternetCloseHandle|n|n" )
  varset( $hWinInet, DllLoad("wininet.dll") )
  if $hWinInet=0
    error( "DLL ""wininet.dll"" not found." )
    return(-1)
  endif

  if HamVersion(0) > "2.1.0.11"
    varset( $logOK, 9 )
    varset( $logFail, 10 )
  else
    varset( $logOK, 3 )
    varset( $logFail, 3 )
  endif

return(0)

#!finalize
  DllFree( $hWinInet )
return(0)
```

Hamster und Authentifikation

1. Einführung in die Grundlagen der Authentifikation

1.1. Kryptografische Algorithmen

Grundlage sind die RFC 1320 und 1321. Sie definieren ein MD4 und MD5, welche als Ausgabe einen MD4 bzw. MD5 als Digest liefern.

MD2	RFC 1319
MD4	RFC 1320
MD5	RFC 1321
SHA1	RFC 3174

MD4 besitzt inzwischen nur noch historischen Wert. Im Jahre 2001 wurde auch SHA1 in RFC 3174 standardisiert. SHA1 liefert ebenfalls einen Digest, der im Gegensatz zu MD5 mit wesentlich weniger Konstanten auskommt und deshalb theoretisch sicherer ist als MD5. Gegen SHA1 ist bisher auch, im Gegensatz zu MD5, kein theoretischer Angriff bekannt. SHA1 liefert einen 160-Bit-Digest (20 Byte), MD5 hingegen nur einen 128-Bit-Digest (16 Byte). Der Angriff gegen MD5 ist ausschließlich theoretischer Natur, da es nicht ausreichend Rechenzeitkapazität gibt, ihn zu brechen. Setzt man die Lebensdauer von MD5 in ein Verhältnis zur Lebensdauer von Passwörtern, ist MD5 auch in den nächsten Jahrzehnten sicher. Bei SHA1 muss man bezüglich der Sicherheit ebenfalls bedenken, dass er in allen Aspekten noch nicht so gut untersucht ist wie MD5. Meiner Einschätzung nach ist der Unterschied zwischen MD5 und SHA1 bezüglich Sicherheit eher paranoid und eine Glaubensfrage. Ein weiterer Aspekt bezüglich der Sicherheit dieser Verfahren ist die praktische Implementation der Hash-Algorithmen. Um die Hash-Algorithmen auch für kurze Textfetzen, wie zum Beispiel Passwörter, unangreifbar zu machen, wird der jeweilige Hash-Algorithmus im HMAC-Verfahren doppelt auf das Passwort angewendet. Die Formel hierfür lautet:

Im Folgenden in Bezug auf MD5 leider veraltet!

```
ipad = String gefüllt mit dem Byte 0x36 entsprechend der Länge des Hash
opad = String gefüllt mit dem Byte 0x5C entsprechend der Länge des Hash
   K = Das zu sichernde Passwort
   H = Hash-Funktion, zum Beispiel MD5 oder SHA1

Hash = H(K XOR opad, H(K XOR ipad, text))
```

Weiterhin ist entscheidend, ob nur der Server den Client oder auch der Client den Server authentifiziert, um einen »Man in the middle«-Angriff zu vermeiden. Das CRAM-Verfahren authentifiziert nur den Client, nicht aber den Server. Erst das DIGEST-Verfahren (z. B. DIGEST-MD5) ermöglicht auch eine Authentifizierung des Servers.

1.2. Login-Verfahren

Um das Login sicherer zu gestalten, wurden im Laufe der Zeit unterschiedliche Methoden für die einzelnen Protokolle entwickelt:

Glossar:
IMAP
SASL
SMTP-AUTH
SMTP
POP3

IMAP4-Auth	RFC 2060
POP-Auth	RFC 1734
POP-SASL-AUTH	RFC 2449
APOP	RFC 1939
HTTP-Auth	RFC 2069
SMTP-Auth	RFC 2554
SMTP after POP3	(nicht standardisiert)

Diese Protokolle nutzen unterschiedliche Methoden für das sichere Login, aber teilweise dieselben Hash-Funktionen und Key-Verwaltungsverfahren.

HMAC	RFC 2104
CRAM	RFC 2195
SESS	RFC 2617
SASL	RFC 2222

Zum Beispiel verwendet APOP den MD5-Digest (nicht zu verwechseln mit dem SASL-Verfahren MD5-DIGEST). Zu Beginn wurden das unverschlüsselte LOGIN-Verfahren und mehr oder minder sichere Verfahren wie KERBEROS, S/Key und GSSAPI verwendet. Für POP und IMAP wurde dann erstmalig das für beide Protokolle gemeinsame HMAC-Verfahren eingeführt, welches von CRAM verwendet wird. CRAM definiert selber nur CRAM-MD5, da zu diesem Zeitpunkt SHA1 noch nicht als RFC standardisiert war. Für HTTP wurde parallel ein weiterentwickeltes Verfahren namens MD5-SESS eingeführt, was ebenfalls für andere Hash-Funktionen wie SHA1 vorbereitet war, ohne sie selbst zu definieren.

Um dem Wildwuchs der einzelnen Verfahren und Hash-Methoden einen vernünftigen Rahmen zu geben und das vom jeweiligen Server angebotene Verfahren automatisch auswählen zu können, wurde dann mit SASL ein Auswahlverfahren für Login-Mechanismen eingeführt. Dieser RFC definierte sofort die SASL-Mechanismen für S/Key, KERBEROS und GSSAPI und übernahm LOGIN und CRAM-MD5 von IMAP/POP. Beim LOGIN-Mechanismus handelt es sich um ein primitives, unverschlüsseltes Verfahren, welches nur im Notfall verwendet werden sollte. In SMTP wurde SASL mit RFC 2554 übernommen. In RFC 2449 wurde SASL endgültig für POP3 definiert und in den CAPA-Befehl verlagert. Bis zu diesem Zeitpunkt wurde die Authentifikation bei POP3, wie bei IMAP, im AUTH-Befehl platziert. Es wird heute noch von vielen POP3-Servern die Authentifikation ohne SASL-Mechanismus angeboten. Inzwischen wurden weitere SASL-Mechanismen standardisiert, welche für alle Protokolle gültig sind. Das sind:

PLAIN und EXTERNAL	RFC 2595	SSL/TLS unverschlüsselt, da SSL an sich schon sicher ist.
SECURID	RFC 2808	Für RSA-Chipkarten-Anwendungen. SECURID ermöglicht auch andere HASH-Funktionen als MD5, z. B. SHA1.

SSL
TLS

Glossar: SASL

DIGEST-MD5 und MD5-SESS	RFC 2831	Übernimmt die Digest-Methoden von HTTP und macht HTTP konform zu SASL. DIGEST ermöglicht auch andere HASH-Funktionen als MD5, z. B. SHA1. Dieses Verfahren ermöglicht es auch, den Server zu authentifizieren.
OTP-MD5	RFC 2444	Für »One Time Passwords«. OTP ermöglicht auch andere HASH-Funktionen als MD5, z. B. SHA1.
ANONYMOUS	RFC 2245	Für anonymes Login.
ISO/IEC 9798-3	RFC 3163	Für die Benutzung von Zertifikaten als Ausweis fürs Login. Basiert auf der PKI-Infrastruktur.

Dazu kommen dann die schon bekannten Mechanismen

KERBEROS	RFC 2222	Für homogene Umgebungen. Basierend auf Ticketservern.
S/Key	RFC 2222	Für auf einander folgende abhängige Keys. Das Verfahren ist nur schwer handhabbar bei öffentlichen Servern und eher gut für Intranetze.
LOGIN	RFC 2222	Klartext-LOGIN
GSS-API	RFC 2222	Für Anwender-Applikationen in Verbindung mit Kerberos.
CRAM-MD5	RFC 2195	Weit verbreitetes praktisch sicheres Verfahren. CRAM ist auch mit SHA1 als Hash-Algorithmus möglich. Methodischer Nachfolger ist DIGEST-SHA1.

1.3. Andere Authentitifizierungsverfahren

Eine Ausnahme bei den Authentifizierungsverfahren ist das »SMTP after POP3«-Verfahren. Bei diesem Verfahren wird ein dem eigentlichen SMTP-Login vorhergehender erfolgreicher POP3-Zugriff zur Authentifikation verwendet. Es ist im eigentlichen Sinne kein selbständiges Authentifizierungsverfahren, sondern die Mitverwendung des Authentifizierungsverfahren eines anderen Protokolls. Dieses Verfahren besitzt erhebliche Sicherheitsmängel und sollte nach Möglichkeit nicht mehr verwendet werden.

SMTP
POP3

2. Authentifikationsverfahren für die lokalen Server des Hamsters

Folgende Authentifikationsverfahren sind an allen lokalen Servern des Hamsters möglich:

LOGIN, CRAM-MD5, DIGEST-MD5, CRAM-SHA1, PLAIN

Der Mechanismus LOGIN ist unsicher und sollte nach Möglichkeit nicht verwendet werden. Er lässt sich im Setup-Menü aller lokalen Server abschalten. Der Mechanismus PLAIN kann nur zusammen mit dem SSL-Verfahren verwendet werden. Für den lokalen POP3-Server kann auch POP3-AUTH ohne SASL-Mechanismus und das APOP-Verfahren zur Authentifikation verwendet werden. Für den lokalen SMTP-Server kann als Authentifikationsverfahren auch das »SMTP after POP3«-Verfahren verwendet werden. Eine gemeinsame Nutzung dieses Verfahrens mit SMTP-AUTH ist technisch bedingt nicht möglich. **Allgemein ist aber von der Verwendung des »SMTP after POP3«-Verfahrens aus Sicherheitsgründen abzuraten.**

Glossar:
SSL
POP3
SASL
SMTP
SMTP-AUTH

3. Authentifikationsverfahren für die externen Server
Für die Authentifikation des Hamsters bei den Servern der Provider können folgende Verfahren verwendet werden:

LOGIN, CRAM-MD5, DIGEST-MD5, CRAM-SHA1, PLAIN

Der Mechanismus LOGIN ist unsicher und sollte nach Möglichkeit nicht verwendet werden. Um die Verwendung dieses unsicheren Verfahrens zu vermeiden, lässt sich im Setup-Menü der externen Server bei der Passworteingabe ein sicheres Authentifizierungsverfahren, wie zum Beispiel CRAM-MD5, erzwingen. Der Mechanismus PLAIN kann nur zusammen mit dem SSL-Verfahren verwendet werden. Das »SMTP after POP3«-Verfahren kann mittels eines Skriptes nach folgenden Beispiel realisiert werden:

```
HamFetchMail( ⟨server⟩ )
HamWaitIdle
HamSendMail( ⟨server⟩ )
```

Allgemein ist aber von der Verwendung des »SMTP after POP3«-Verfahrens aus Sicherheitsgründen abzuraten.

Für Verbindungen zu POP3-Servern kann das APOP-Verfahren verwendet werden. Um die Verwendung eines bestimmten Login-Verfahrens zu erzwingen, kann dem Passwort der Name des jeweiligen Verfahrens, gefolgt von einem Doppelpunkt, vorangestellt werden. Zulässige Präfixe sind:

PASS:	Klartext-Passwort	AUTH:	Auth-Verfahren für SMTP oder POP3 erzwingen
APOP:	POP3-APOP-Verfahren	SASL:	SASL-Auth-Verfahren für SMTP oder POP3 erzwingen
LOGIN:	LOGIN-Verfahren (Klartext)	CRAM-MD5:	CRAM-MD5-Verfahren
PLAIN:	SSL-PLAIN-Verfahren (Klartext innerhalb eines SSL-Tunnels)	DIGEST-MD5:	DIGEST-MD5-Verfahren
		CRAM-SHA1:	CRAM-SHA1-Verfahren

Achtung! Die Präfixe müssen groß geschrieben werden!

Hamster und sichere Passwörter

> *„ Was sind sichere Kennwörter? “*

Unter einem starken Kennwort versteht man eine Zeichenkombination, die einem Angriff durch Raten über sehr lange Zeit standhält.

Bei einem Wörterbuchangriff versucht ein angreifendes Programm sämtliche Wörter, die irgendwann und irgendwo geschrieben wurden (vom Namenslexikon über Film, Musik und Reiseführer in jeder Sprache und jedem Dialekt), solange, bis es Zugriff erhält oder weitere Scans unterbunden werden. Als Steigerung dazu gibt es den Brute-Force-Angriff, bei dem jede beliebige Zeichenfolge versucht wird.

Wie stark die bislang von Dir benutzten Kennwörter sind, kannst Du herausfinden, wenn Du sie oder besser sehr ähnliche Zeichenketten auf `https://passwortcheck.datenschutz.ch/` einem Test unterziehst. *(Beachte den Hinweis dort, keine tatsächlich benutzten Kennwörter zu verwenden. Falls doch, solltest Du nach dem Test unbedingt neue Kennwörter vergeben.)*

Aus heutiger Sicht gilt eine Zeichenfolge mit einer Kombination aus 10 Buchstaben, Ziffern und Sonderzeichen als relativ sicher.

> *„ Wie kann ich mir ein scheinbar kryptisches Kennwort merken? “*

Es ist einfach, wenn Du einen Satz bildest und daraus das Kennwort ableitest:

`IbH&XfdN,nafE-M.` = *Ich benutze Hamster und Xnews für das Nutznetz, nicht aber für E-Mail.*

Auch die berüchtigte Kennwortfrage bei manchen Webaccounts »Wen hast Du zum erstem Mal geküsst?« solltest Du nicht mit einem simplen »Manuel(a)« beantworten – Namen und Geburtstage gelten als besonders unsicher, weil damit sogar ein menschlicher Angreifer Zugang zu Deinen Accounts erhalten kann (sozialer Angriff). Beflügle stattdessen Deine Phantasie: `A9.S#bP#mM`.

Solche Geschichten vergisst man nicht – da braucht man auch keine gelben POST-IT-Zettel am Monitor.

Hamster aktualisieren (Update)

Das ist einfach: Zuerst die neue Version aus dem Netz laden oder sonst irgendwie besorgen. Dann den Hamster beenden, und die Dateien aus der neuen ZIP- oder 7Z-Datei in das Hamsterverzeichnis kopieren. Bei einer Standard-Installation gehen also alle Dateien direkt zur »`hamster.exe`«. Wer etwas mehr konfiguriert hatte, und verschiedene Pfade für z. B. Gruppen und Skripte nutzt, muss das eben nach seiner Konfiguration entsprechend aufteilen. ***Wichtige Ausnahme:*** Wenn die Sprachdatei »`hamster_XX.dat`«[1] bearbeitet wurde, würden beim

[1] »XX« steht für ein Sprachenkürzel, »de« für deutsch.

einfachen Kopieren der Standardversion dieser Datei alle Änderungen verloren gehen. Vorher ein Backup wenigstens des geänderten Teils machen und diesen hinterher in der neuen Datei an den richtigen Stellen wieder eintragen bzw. entsprechende vorhandene Einstellungen überschreiben.

> *„ Wie ist das mit den Beta-Versionen? “*

Das ist etwas schwieriger: Die Testversionen enthalten nur die geänderten Dateien. Daher ist angeraten, auf jeden Fall vorher auf die letzte Stable-Version aufzurüsten, und anschließend die Beta-Version einzuspielen. Am sichersten wäre, sukzessive die Beta-Versionen einzuspielen.

In beiden Fällen einfach anschließend den Hamster neu starten. Etwaige Änderungen konfiguriert er, soweit möglich, selbst; anderenfalls wird der Administrator aufgefordert, das selber zu übernehmen.

Hamster und IMAP

> *„ Hamster soll IMAP verstehen!? Wozu und wie benutze ich das? “*

Glossar: IMAP

Der Hamster hat ab der Version 1.3.23.160 (beta) bzw. 2.0 einen IMAP4rev1-Server implementiert. Damit kannst Du mit einem Client, welcher IMAP versteht, Deine Mailbox lesen und verwalten. Mehrere Clienten – auch auf unterschiedlichen Rechnern – greifen auf dieselbe Ordnerstruktur zurück und können auch den Status einzelner Mails über den Server austauschen. – So die Theorie von IMAP.

Im Hamster gibt es in der Ordnerstruktur eine Besonderheit: Es gibt keine Unterordner des »Hauptordners« INBOX. Jeder Versuch, Ordner dorthin zu verschieben, wird mit einer Fehlermeldung quittiert.

Auch hat der Hamster hat *keinen* IMAP-Clienten implementiert, mit dem Du Mails von anderen IMAP-Servern abrufen kannst. Diese müssen weiterhin per POP3 abgerufen oder direkt per SMTP-MX in den Hamster eingeliefert werden.

POP3
SMTP

Einrichtung

- Neuen Benutzer einrichten über das Menü »Einstellungen« → »Benutzerverwaltung und Passworte« ⇒ »Lokale Benutzer« *(oder bestehenden Nutzer konfigurieren)*.
  - Gruppe auswählen oder neue Gruppe einrichten *(bei Neueinrichtung sollten die Einstellungen, die hier für einen Nutzer angegeben werden, direkt in der Gruppe getätigt werden)*.
  - Neuen Nutzer erstellen *(oder bestehenden Nutzer auswählen)*, im neuen Fenster anschließend unter dem Reiter »Mail(POP3/IMAP+SMTP)« die Option »Individuelle Einstellung bzgl. Postfach« aktivieren und in der daraufhin erscheinenden Auswahlbox »Benutzer hat Postfach, Nutzung per IMAP-Client« auswählen.

Glossar: IMAP SSL TLS

- IMAP-Server einstellen über das Menü »Einstellungen« → »Lokale Server« ⇒ »IMAP«

  Die Voreinstellung bei »SSL-Benutzung« (»Kein TLS«) erzeugt eine Warnung beim Start des Servers, da sie eigentlich gegen die Definition im RFC verstößt! Um jedoch SSL in Verbindung mit dem Hamster nutzen zu können, werden zwei zusätzliche DLL-Dateien benötigt; mehr zum Thema siehe in der SSL-FAQ.

- Außerdem empfehlenswert:
  - IMAP-Server bei Hamster-Start aktivieren: Unter »Einstellungen« → »Automatische Abläufe« ⇒ »Allgemeines« unter »Folgende lokale Server starten« die Option für den IMAP-Server aktivieren
  - Im Mailscore-/Killfile eine Regel zum Einsortieren in das IMAP-Konto einfügen, siehe in der Beschreibung der Datei »`MailFilt.hst`«

Jetzt kann man in seinem bevorzugten IMAP-Clienten Verbindung zum Hamster aufnehmen: entsprechend der Voreinstellung `127.0.0.1:143` oder `localhost:143` (allgemein ausgedrückt ⟨IP-Bindung⟩:⟨Port⟩). Port

Hamster 2.0 unterstützt IMAP4rev1 nach reinem RFC 2060, d.h. Erweiterungen wie »Namespace« und einige andere sind (noch) nicht möglich.

Sofortiges Versenden von E-Mail

Wenn der Hamster eine Mail von einem Mailclient bekommt und er diese Mail nicht einem lokalen Benutzer zustellen kann, legt er sie im Verzeichnis »`Hamster\Mails\Mail.Out`« ab (oder dem entsprechenden Verzeichnis, das in der »`hamster.ini`«, Sektion »[Directories]« mit dem Schlüssel »Mail.out« festgelegt wurde), wo sie dann auf die Versendung wartet.

Um diese Mail zu versenden, gibt es 3 Möglichkeiten:

1. *Möglichkeit:* Man versendet sie manuell entweder über das Menü »Online« oder über ein Skript.
2. *Möglichkeit:* Ein Skript, das regelmäßig nachschaut, ob Mails zum Versenden vorliegen, und diese dann versendet. Ein Beispiel eines solchen Skriptes findet man in der FAQ unter »Selbständiges regelmäßiges Holen und Versenden von News und Mails«.
3. *Möglichkeit:* Man versendet die Mail sofort, nachdem sie vom Hamster in »`Mail.Out`« abgelegt wurde. Dazu muss man zum einen ein Skript haben, das die Mail versendet, zum anderen muss man den Hamster so konfigurieren, dass das Skript aufgerufen wird, sobald eine Mail in »`Mail.Out`« abgelegt wurde. Das soll hier im Folgenden beschrieben werden.

Die einfachste Form dieses Skriptes wäre:

```
#!hs2
HamSendMailAuth( ⟨server⟩, ⟨...⟩ )
quit
```

Allerdings empfiehlt es sich, den Ablauf mit Events zu steuern, ein Beispiel dazu steht weiter unten.

Wenn der Hamster die Onlineverbindung selbst aufbauen muss,ist das oben angegebene Beispiel nicht verwendbar. In dem Fall ist auch eine Eventsteuerung notwendig, auch dazu gibt es weiter unten ein Beispiel.

Glossar:
SMTP
SMTP-AUTH
POP3

Wenn man nicht die Möglichkeit hat, die Mails per SMTP-AUTH zu verschicken, sondern nur ein »SMTP after POP3«-Postfach hat, muss man das Postfach vorher erst freischalten:

```
#!hs2
HamFetchMail
HamWaitIdle( 10000 )
HamSendMail( ⟨server⟩, ⟨...⟩ )
quit
```

Wenn man alleiniger Benutzer des Hamsters ist, mag das obige noch akzeptabel sein. In einem Netzwerk mit mehreren Benutzern könnte das bei größerem Mailaufkommen aber dazu führen, dass der POP3-Server in zu kurzen Intervallen abgefragt wird, womit man sich beim Provider nicht beliebt macht. In diesem Fall sollte man auf die Lösung zurückgreifen, die in »Selbständiges regelmäßiges Holen und Versenden von News und Mails« beschrieben ist.

Hat man nun das Skript, muss man nur noch dafür sorgen, dass es im richtigen Moment aufgerufen wird. Mit dem Hamster geht das sehr komfortabel über den Menü-Dialog »Einstellungen« → »Automatische Abläufe« ⇒ »Aktionen«. Unter »Wähle Aktion« wählt man »E-Mails → Verarbeitung → ausgehender E-Mails« aus. In dem rechten Teil des Fensters trägt man nun im Feld »Starte (hs2-)Skript« dieses Skript ein und markiert die Option »Auf das Ende des Skripts warten«.

Jetzt versendet der Hamster herausgehende Mails sofort. Etwas Wichtiges muss man aber noch beachten: Sollte der Versand einer Mail aus irgendeinem Grund nicht funktionieren (z. B. der Mailserver antwortet nicht) wird der nächste Sendeversuch erst unternommen, wenn der Hamster die nächste Mail bekommt, die er rausschicken soll. Wenn der Hamster auf einem Server läuft und ein Benutzer noch schnell eine letzte Mail für diesen Tag abschickt, könnte sie aus den oben angegebenen Gründen erst einmal liegen bleiben. Da man sowieso noch ein Skript benötigt, welches regelmäßig Mails abholt, wäre es ratsam, dass dieses Skript auch regelmäßig prüft, ob noch Mails zum Versenden vorliegen. Damit nicht beide Skripte gleichzeitig versuchen, Mails zu versenden, sollte man Events einsetzen, siehe den FAQ-Beitrag »Hamster und Events« und die unten stehenden Beispiele.

Für Newsartikel muss man natürlich die folgenden Skripte anpassen und die entsprechenden Skriptbefehle für Abruf und Posten verwenden:

```
#!hs2
HamNewsJobsClear
HamNewsJobsPostDef
HamNewsJobsStart
quit
```

Um das Skript für den Newsversand aufzurufen, muss man es in »Einstellungen« → »Automatische Abläufe« ⇒ »Aktionen« unter »Artikel/Postings → Verarbeitung → ausgehender Artikel« eintragen. Hierbei gelten auch die gleichen Anmerkungen wie zum Versenden von Mails.

Eventsteuerung (ohne Verbindungsaufbau durch den Hamster)

Die Eventsteuerung funktioniert in diesem Fall so, dass zwei Skripte, eines zum Versenden (»Event_Versenden.hsc«) und eines zum Abholen (»Event_Abholen.hsc«) von Mails, ständig laufen und darauf warten, dass ein Event durch andere Skripte ausgelöst wird.

Das Skript »Event_Trigger.hsc« sorgt dafür, dass der Event zum Holen von Mails ausgelöst wird, sobald eine Mail zum Versenden bereit liegt. Dazu muss dieses Skript im Menü »Einstellungen« → »Automatische Abläufe« ⇒ »Aktionen« unter »E-Mails → Verarbeitung → ausgehender E-Mails« eingetragen werden.

Das Skript »Event_Scheduler.hsc« läuft auch ständig im Hintergrund und prüft zum einen regelmäßig, ob Mails liegen geblieben sind, die zum Versand anstehen, und zum anderen sorgt es dafür, dass regelmäßig Mails abgeholt werden.

Wenn man die Skripte startet, müssen zuerst »Event_Versenden.hsc« und »Event_Abholen.hsc« gestartet werden und erst dann »Event_Scheduler.hsc«. Um den Startvorgang zu automatisieren, kann man auch das Startskript »Event_Start.hsc« verwenden. Wenn man dieses Skript im Menü »Einstellungen« → »Automatische Abläufe« ⇒ »Aktionen« unter »Hamster → Start« im Feld »Starte (hs2-)Skript« einträgt, werden die Skripte automatisch beim Hamsterstart gestartet.

Beenden sollte man die Skripte ***nicht*** durch den Menüeintrag »Skript« → »Alle Skripte beenden«, sondern mittels des Skriptes »Event_Beenden.hsc«. Dieses Skript sorgt dafür, dass die Skripte »Event_Versenden.hsc« und »Event_Abholen.hsc« innerhalb einer Minute beendet werden. Das Skript »Event_Scheduler.hsc« beendet sich selbst, wenn es merkt, dass die zwei letztgenannten Skripte nicht mehr aktiv sind.

Falls man die Skripte doch über »Skript« → »Alle Skripte beenden« beendet hat, muss der Hamster neu gestartet werden, bevor man die Skripte wieder verwenden kann.

```
#--- Event_Start.hsc ---
#!hs2
runscript( HamHscPath + "Event_Versenden.hsc", "", false )
runscript( HamHscPath + "Event_Abholen.hsc", "", false )
sleep( 2000 )
runscript( HamHscPath + "Event_Scheduler.hsc", "", false )
quit
#-----------------------
```

```
#--- Event_Versenden.hsc ---
#!hs2
varset( $ScriptIdent, GetProcessIdentifier + "FAQActionScript" )
varset( $EventVersenden, "Versenden" + $ScriptIdent )
varset( $EventBeendenVersenden, "BeendenVersenden" + $ScriptIdent )
varset( $Handle_EventVersenden, 0 )
varset( $Handle_EventBeendenVersenden, 0 )
varset( $EventFehlercode, 0 )

$Handle_EventVersenden = EventCreate( $EventVersenden, 1, 0 )
$Handle_EventBeendenVersenden = EventCreate( $EventBeendenVersenden, 1, 0 )

do
  $EventFehlercode = EventMultipleWait( 60000, $Handle_EventVersenden, _
                     $Handle_EventBeendenVersenden )
  if( $EventFehlercode = 1 )
    HamSendMailAuth( ⟨...⟩ ) # Hier einstellen!
  endif
  if( $EventFehlercode = 2 )
    beenden
  endif
loop

sub beenden
  EventClose( $Handle_EventVersenden )
  EventClose( $Handle_EventBeendenVersenden )
  quit
endsub
#--------------------------

#--- Event_Abholen.hsc ---
#!hs2
varset( $ScriptIdent, GetProcessIdentifier + "FAQActionScript" )
varset( $EventAbholen, "Abholen" + $ScriptIdent )
varset( $EventBeendenAbholen, "BeendenAbholen" + $ScriptIdent )
varset( $Handle_EventAbholen, 0 )
varset( $Handle_EventBeendenAbholen, 0 )
varset( $EventFehlercode, 0 )

$Handle_EventAbholen = EventCreate( $EventAbholen, 1, 0 )
$Handle_EventBeendenAbholen = EventCreate( $EventBeendenAbholen, 1, 0 )

do
  $EventFehlercode = EventMultipleWait( 60000, $Handle_EventAbholen, _
                     $Handle_EventBeendenAbholen )
  if( $EventFehlercode = 1 )
    HamFetchMail( ⟨...⟩ ) # Hier einstellen!
  endif
  if( $EventFehlercode = 2 )
    beenden
  endif
loop

sub beenden
  EventClose( $Handle_EventAbholen )
```

```
  EventClose( $Handle_EventBeendenAbholen )
  quit
endsub
#-------------------------

#--- Event_Scheduler.hsc ---
#!hs2
varset( $ScriptIdent, GetProcessIdentifier + "FAQActionScript" )
varset( $EventVersenden, "Versenden" + $ScriptIdent )
varset( $EventAbholen, "Abholen" + $ScriptIdent )

AtAdd( versendenundholen, "00:00", "24:00", "1111111", 15, 1 )
AtExecute
quit

sub versendenundholen
  if( HamGetStatus( 3, 7 ) > 0 )
    if( EventSet( $EventVersenden ) = -1 )
      addlog("Das Versendeskript ist nicht gestartet, Mails werden nicht versendet," _
             + " Scheduler wird beendet.", 5 )
      return(1)
    endif
  endif
  if( EventSet( $EventAbholen ) = -1 )
    addlog( "Das Abholskript ist nicht gestartet, Mails werden nicht geholt," _
           + " Scheduler wird beendet.", 5 )
    return(1)
  endif
endsub
#--------------------------

#--- Event_Trigger.hsc ---
#!hs2
varset( $ScriptIdent, GetProcessIdentifier + "FAQActionScript" )
varset( $EventVersenden, "Versenden" + $ScriptIdent )

if( EventSet( $EventVersenden ) = -1 )
  addlog( "Das Versendeskript ist nicht gestartet, Mails werden nicht versendet.", 5 )
else
  addlog( "Event zum Versenden wird ausgelöst.", 4 )
endif
quit
#-------------------------

#--- Event_Beenden.hsc ---
#!hs2
varset( $ScriptIdent, GetProcessIdentifier + "FAQActionScript" )
varset( $EventBeendenVersenden, "BeendenVersenden" + $ScriptIdent )
varset( $EventBeendenAbholen, "BeendenAbholen" + $ScriptIdent )

if( EventSet( $EventBeendenVersenden ) = -1 )
  addlog( "Das Versendeskript ist nicht gestartet.", 5 )
else
  addlog( "Das Versendeskript wird innerhalb der nächsten 60 Sekunden beendet.", 4 )
endif
```

```
if( EventSet( $EventBeendenAbholen ) = -1 )
  addlog( "Das Abholskript ist nicht gestartet.", 5 )
else
  addlog( "Das Abholskript wird innerhalb der nächsten 60 Sekunden beendet.", 4 )
endif
quit
#------------------------
```

Eventsteuerung (mit Verbindungsaufbau durch den Hamster)

Wenn der Hamster die Onlineverbindung selbst aufbauen soll, dann ist ein zeitgesteuerter Mailaustausch zu bevorzugen, wie es im FAQ-Beitrag »Selbständiges regelmäßiges Holen und Versenden von News und Mails« beschrieben ist.

Wer trotzdem ein eventgesteuertes Skript zum sofortigen Versenden einsetzen möchte, kann folgende Skripte verwenden (es gelten die gleichen Anmerkungen wie bei den Skripten ohne Verbindungsaufbau durch den Hamster):

```
#--- Event_Start.hsc ---
#!hs2
runscript( HamHscPath + "Event_Versenden_und_Holen.hsc", "", false )
sleep( 2000 )
runscript( HamHscPath + "Event_Scheduler.hsc", "", false )
quit
#----------------------

#--- Event_Versenden_und_Holen.hsc ---
#!hs2
varset( $ScriptIdent, GetProcessIdentifier + "FAQActionScript" )
varset( $EventVersenden, "Versenden" + $ScriptIdent )
varset( $EventAbholen, "Abholen" + $ScriptIdent )
varset( $EventBeenden, "Beenden" + $ScriptIdent )
varset( $Handle_EventVersenden, 0 )
varset( $Handle_EventAbholen, 0 )
varset( $Handle_EventBeenden, 0 )
varset( $EventFehlercode, 0 )
varset( $eingewaehlt, 0 )
varset( $try, 0 )

$Handle_EventVersenden = EventCreate( $EventVersenden, 0, 0 )
$Handle_EventAbholen = EventCreate( $EventAbholen, 0, 0 )
$Handle_EventBeenden = EventCreate( $EventBeenden, 0, 0 )

do
  $EventFehlercode = EventWait( $EventVersenden, 60000 )
  if( $EventFehlercode = 258 )
    if( EventWait( $EventBeenden, 100 ) = 0)
      addlog( "Skript soll beendet werden.", 4 )
      beenden
    endif
    if( EventWait( $EventAbholen, 100 ) = 0)
      if( Verbindungsaufbau != 1)
        EventReset( $EventAbholen )
```

```
        HamFetchMail( ⟨...⟩ ) # Hier anpassen!
        if( EventWait( $EventVersenden, 100 ) = 258)
          Verbindungsabbau
        endif
      endif
    endif
  else
    if( Verbindungsaufbau != 1)
      while( EventWait( $EventVersenden, 1 ) = 0 )
        EventReset( $EventVersenden )
        HamSendMailAuth( ⟨...⟩ ) # Hier anpassen!
        if( EventWait( $EventAbholen, 100 ) = 0)
          EventReset( $EventAbholen )
          HamFetchMail( ⟨...⟩ ) # Hier anpassen!
        endif
      endwhile
      Verbindungsabbau
    else
      EventReset( $EventVersenden )
    endif
  endif
loop

sub Verbindungsaufbau
  $try = 1
  if( !RasIsConnected )
    $eingewaehlt = 1
    while( HamRasDial( ⟨...⟩ ) != 0 ) # Hier anpassen!
      if( $try >= 3 )
        $eingewaehlt = 0
        warning( "Einwahl fehlgeschlagen." )
        return(1)
      endif
      inc( $try )
      sleep( 10000 )
    endwhile
  endif
  return(0)
endsub

sub Verbindungsabbau
  if( $eingewaehlt = 1 )
    HamRasHangup
    $eingewaehlt = 0
  endif
endsub

sub beenden
  EventReset( $EventBeenden )
  EventClose( $Handle_EventVersenden )
  EventClose( $Handle_EventAbholen )
  EventClose( $Handle_EventBeenden )
  quit
endsub
#-------------------------------------
```

```
#--- Event_Scheduler.hsc ---
#!hs2
varset( $ScriptIdent, GetProcessIdentifier + "FAQActionScript" )
varset( $EventVersenden, "Versenden" + $ScriptIdent )
varset( $EventAbholen, "Abholen" + $ScriptIdent )

AtAdd( versendenundholen, "00:00", "24:00", "1111111", 15, true )
AtExecute
quit

sub versendenundholen
  if( HamGetStatus( 3, 7 ) > 0 )
    if( EventSet( $EventVersenden ) = -1 )
      addlog("Das Versendeskript ist nicht gestartet, Mails werden nicht versendet," _
             + "Scheduler wird beendet.", 5 )
      return(1)
    else
      addlog( "Event zum Versenden wird ausgelöst.", 4 )
    endif
  endif
  if( EventSet( $EventAbholen ) = -1 )
    addlog( "Das Versendeskript ist nicht gestartet, Mails werden nicht geholt," _
           + "Scheduler wird beendet.", 5 )
    return(1)
  else
    addlog( "Event zum Holen wird ausgelöst.", 4 )
  endif
endsub
#---------------------------

#--- Event_Trigger.hsc ---
#!hs2
varset( $ScriptIdent, GetProcessIdentifier + "FAQActionScript" )
varset( $EventVersenden, "Versenden" + $ScriptIdent )

if( EventSet( $EventVersenden ) = -1 )
  addlog( "Das Versendeskript ist nicht gestartet, Mails werden nicht versendet.", 5 )
else
  addlog( "Event zum Versenden wird ausgelöst.", 4 )
endif
quit
#-------------------------

#--- Event_Beenden.hsc ---
#!hs2
varset( $ScriptIdent, GetProcessIdentifier + "FAQActionScript" )
varset( $EventBeenden, "Beenden" + $ScriptIdent )

if( EventSet( $EventBeenden ) = -1 )
  addlog( "Das Versendeskript ist nicht gestartet.", 5 )
else
  addlog( "Das Versendeskript wird innerhalb der nächsten 60 Sekunden beendet.", 4 )
endif
quit
#-------------------------
```

Wie richte ich einen Archivhamster ein?

(*Vgl. auch* »Zwei Hamster, geht das?«)

> *„ Wie richte ich einen »Archivhamster« im lokalen Netzwerk ein? “*

Irgendwann kommt für jeden Nutznetznutzer der Augenblick, in dem er »seine« Lieblings-Newsgroups auf ewig archivieren möchte. Dafür bietet sich natürlich der Hamster an.

Aber warum jetzt einen zusätzlichen Archivhamster? Nun, diese Vorgehensweise bietet den Vorteil, dass der Arbeitshamster immer schön schlank und somit auch entsprechend schnell bleibt. Vor allem beim »Putzen«.

Wie richte ich nun einen Archivhamster auf einem anderen PC in meinem kleinen Heimnetzwerk ein?

1. Installation Archivhamster
Installiere auf dem gewünschten PC einfach Deine Hamsterversion neu.

Eine Neuinstallation statt eines Starts mit einer 1:1-Kopie des Arbeitshamsters ist aus folgenden Gründen empfehlenswert:

1.1. »Datenmüll«
Die ganzen evtl. vorhandenen Skripte und Newsgroups, die nicht archiviert werden sollen, werden gar nicht erst mitkopiert. Somit entfällt ein manuelles »Bereinigen«.

1.2. Einstellungen
Du brauchst in Deinem Archivhamster keine Newsserver Deines Internet-Service-Providers (ISP). Schließlich willst Du ja nur von Deinem Arbeitshamster pullen und dessen Bestand archivieren. Auch ist oft bei einem Arbeitshamster die Haltezeit der News-History nur auf eine begrenzte Dauer eingestellt. Im Archivhamster sollte sie jedoch auf unendlich (Wert: 0) stehen. Schließlich willst Du ja auch mal auf alte Artikel in Deinem Archiv schnell zugreifen können.

Wenn Du mit einer 1:1-Kopie startest und dort nur die Haltezeit der News-History änderst, muss die History neu erstellt werden. Und das dauert …

Also am besten mit einem »jungfräulichen« Hamster starten. Aber das ist Geschmackssache. Theoretisch geht es auch mit einer 1:1-Kopie des Arbeitshamsters. Im Folgenden gehe ich jedoch von einer Neuinstallation aus.

2. Einstellungen im Archivhamster

2.1. Newsserver
Richte in Deinem Archivhamster einen neuen Newsserver ein (Menü »Einstellungen« → »News: Server, … « ⇒ »Newsserver«). Als Adresse gibst Du den Namen des Rechners ein,

auf dem Dein Arbeitshamster läuft (z. B. »HauptPC«) bzw. dessen IP-Adresse – sofern fest vergeben – im Netz (z. B. »192.168.47.11«), gefolgt von »⟨XYZ⟩«. Dabei gibt »⟨XYZ⟩« den Port an, auf dem Dein Arbeitshamster seinen NNTP-Server für die Clients zur Verfügung stellt (in der Regel »119«). Als Benutzer setzt Du hier »Archivhamster« o. ä. und ein Passwort. Diese Einstellungen bitte merken, Du benötigst sie bei 3.2.

Glossar: Port NNTP

Unter »Einstellungen« → »News: Server, . . . « ⇒ »Einstellungen« solltest Du die Artikellimits auf »keine Begrenzung« setzen. Schließlich soll Dein Archivhamster nachher »unendlich« archivieren. Sonst wäre er ja kein richtiger Archivar. Daher muss auch unter »Einstellungen« → »Lokale Server« ⇒ »NNTP« bei Haltedauer überall (History usw.) eine »0« für unendlich gesetzt werden!

2.2. Benutzer

Lege einen neuen Benutzer im Archivhamster an, z. B. »Arbeitshamster«. Vergib hier auf jeden Fall ein Passwort. Schließlich soll Dein Arbeitshamster auch mal auf den Archivhamster zugreifen können, um eine ältere MID mal zügig zu finden.

MID = Message-ID

2.3. IP-Bindung

Den Archivhamster musst Du nun an »0.0.0.0« binden. Beachte hierzu bitte auch die Beschreibung der Datei »IPAccess.hst« in dieser Hilfe! Nur so kann sowohl der Arbeitshamster als auch ein evtl. auf dem Archivhamster-PC installierter News-Client auf Deinen Archivhamster zugreifen.

2.4. Lokale Newsgroup

Lege eine neue lokale Newsgroup auf Deinem Archivhamster an (z. B. »local.mid-suche«) an. So kann Dein Arbeitshamster schnell und komfortabel eine MID auf dem Archivhamster suchen. Dazu später mehr.

3. Einstellungen im Arbeitshamster

3.1. Newsserver

Analog zu 2.1 musst Du hier jetzt den Archivhamster als neuen Newsserver angeben. Dabei solltest Du die Option »Server ist schreibgeschützt d.h. kein Posten erlaubt« aktivieren. Dies ist Voraussetzung, damit Dein Arbeitshamster nachher eine gesuchte MID beim Archivhamster anfragen kann. Als Benutzer und Passwort gibst Du hier die Daten aus 2.2 ein.

3.2. Benutzer

Im Arbeitshamster muss nun ein neuer Benutzer mit den Daten aus 2.1 angelegt werden (Passwort nicht vergessen!), damit Dein Archivhamster auf den Arbeitshamster zugreifen kann.

3.3. IP-Bindung

Sollte hier ebenfalls auf »0.0.0.0« gesetzt werden. Bitte unbedingt die Sicherheitshinweise aus der Hilfe zur Datei »IPAccess.hst« beachten!

4. Archivhamster »füttern«
Bitte beachten: Bei einer Interaktion zwischen den beiden Hamstern müssen natürlich immer beide auf dem entsprechenden Rechner gleichzeitig gestartet sein!

4.1. Erstbefüllung
Aktiviere beim Archivhamster in den Einstellungen des Newsservers »Arbeitshamster« unter dem Reiter »Beim nächsten Abholen von News« die Optionen

- »Liste der Newsgruppen nachladen«
- »Liste der Gruppenbeschreibungen nachladen«

Stoße nun den Arbeitshamster unter dem Menü »Online« → »NNTP-Server« → »Arbeitshamster« an. Es sollte nun ein kurzer Pull erfolgen, bei dem der Archivhamster die entsprechenden Informationen abruft.

Nun kannst Du unter bei »Einstellungen« → »News: Server, . . . « ⇒ »News-Ladeaufträge« die Newsgroups auswählen, die Du archivieren willst. Ein erneuter Anstoß unter »Online« → »NNTP-Server« → »Arbeitshamster« sollte nun den ersten »richtigen« Pull anstoßen. Dieser kann natürlich etwas dauern, weil Dein Archivhamster nun die kompletten zu archivierenden Gruppen pullt.

4.2. Regelmäßige Fütterung
Wie, ist Geschmackssache. Entweder skriptgesteuert oder manuell über das »Online«-Menü wie unter 4.1. Natürlich sollten die Abstände zwischen den Pulls kürzer sein als die Haltezeit der Newsgroups im Arbeitshamster.

5. Arbeitshamster sucht Artikel zu alter MID

MID = Message-ID

5.1. Nötige Einstellungen im Arbeitshamster
Aktualisiere analog zu 4.1 die Gruppenliste der auf dem Archivhamster verfügbaren Newsgroups in Deinem Arbeitshamster. Unter »News-Ladeaufträge« bestellst Du nun bei Deinem Server »Archivhamster« die Newsgroup »local.mid-suche« o. ä. entsprechend 2.4.

5.2. MID suchen
Wenn Du eine »alte« MID suchst, die Dein Arbeitshamster nicht mehr vorrätig hat, gibt es folgende Möglichkeiten:

5.2.1. Newsclient: Du merkst es beim Stöbern in Deinem Newsclient, der Deinen Arbeitshamster automatisch danach fragt. Dann trägt der Arbeitshamster diese MID automatisch in die »`GetMIDs.txt`« ein.

5.2.2. Manuell: Natürlich kannst Du eine MID auch händisch in die »`GetMIDs.txt`« eintragen, siehe dazu auch das Hamster-Menü »Einstellungen« → »Automatische Abläufe« ⇒ »Lade Artikel über ihre MID«.

5.3. Artikel pullen

Grundsätzlich gibt es jetzt zwei Möglichkeiten, die Artikel aus der »`GetMIDs.txt`« zu pullen:

5.3.1. Skript: Du baust in Deine Pull-Skripte einen Ladeauftrag ein, der den Archivhamster pullt.

5.3.2. Manuell: Über das Menü »Online« → »NNTP-Server« → »Archivhamster«.

Beides sollte nicht viel Zeit in Anspruch nehmen, da ja vom Archivhamster nur die Newsgroup »local.mid-suche« gepullt wird (die keine Artikel enthält oder je enthalten wird …) und ansonsten auf dem Archivhamster nur die MID gesucht wird.

Abschlussbemerkung:

Natürlich kann man den Archivhamster noch um Mailarchivierung o. ä. erweitern oder auch ganz anders konfigurieren. Wie immer beim Hamster sind der Phantasie keine Grenzen gesetzt!

Viel Spaß beim Archivieren.

News und Mail mit Hilfe von Telnet kontrollieren

Einführung

Wenn es Probleme in der Kommunikation zwischen Server und Client gibt, kann man versuchen, mit Telnet auf den jeweiligen Server zuzugreifen. Wie das geht, soll im Folgenden gezeigt werden (mit Beschränkung auf die wichtigsten Möglichkeiten).

Die Bildschirmausgabe kann kopiert und zum Beispiel in Newspostings veröffentlicht werden und so bei der Fehlersuche behilflich sein.

Wesentlich besser als das mit Windows mitgelieferte Telnet ist im Übrigen das Tool PuTTY, auf das hier aber nicht eingegangen wird (das Vorgehen ist auch im Wesentlichen dasselbe). PuTTY kann direkt mit verschlüsselten Verbindungen umgehen, bei Telnet muss dazu Stunnel dazwischen geschaltet werden (siehe SSL-FAQ, Frage 11). PuTTY kann unter `http://www.chiark.greenend.org.uk/~sgtatham/putty/` heruntergeladen werden. Eine weitere, sehr viel einfachere (und deutschsprachige) Alternative ist TT von Th. Gohel (herunterzuladen unter `http://www.pbhq.de/filebase/fdb141.html#TT.ZIP`.

Seit Windows Vista muss Telnet erst aktiviert werden, siehe dazu die Anleitungen »Install Telnet Client« (englisch) auf Microsoft TechNet oder »Aktivierung des Telnet-Clients unter Windows 7« (deutsch, bebildert), die auch für Windows Vista funktioniert.

Bei den folgenden Beispielen sind Nutzereingaben in einem Grünton gefärbt und mit »> « markiert, Serverausgaben sind in dunklem Blauton dargestellt mit davorgesetztem »< « (Markierungen in Anlehnung an das Hamsterlog).

Ein paar Worte noch zum hier verwendeten Server-Namen »*pausbacke.hamster.localhost*«:

Dies ist ein für die Beispiele ausgedachter Name (ebenso wie es einen Hamster in solch einer hohen Version wohl nie geben wird). Bei Hamstern, die nicht nur innerhalb eines LAN erreichbar sind bzw. nicht auf Rechnern arbeiten, die per Telefon an das Internet angebunden sind, muss der Name ein »Fully Qualified Domain Name« (FQDN) sein (Forderung in SMTP, s. u.); der Eintrag erfolgt über das Menü »Einstellungen« → »Lokale Server« ⇒ »Allgemeines« → »FQDN für spezielle Header, Peering etc.« (siehe unbedingt auch Hilfe dazu und zum »FQDN für Message-ID«). Die Top-Level-Domain *localhost* genügt dieser Forderung nicht!

Glossar: FQDN SMTP

Gehe direkt zu:

- News mit Telnet (NNTP)
- Mail abholen mit Telnet – IMAP
- Mail abholen mit Telnet – POP3
- Mail versenden mit Telnet (SMTP)

NNTP IMAP POP3

Allgemeines

Telnet ist über die Kommandozeile zugängig. Entweder man öffnet die DOS-Konsole (»Start« → »(Alle) Programme« → »Zubehör« → »Eingabeaufforderung« [bzw. bei Windows 9x »Start« → »Programme« → »MS-DOS-Eingabeaufforderung«]), oder man geht gleich über »Start« → »Ausführen…« und gibt dort folgendes ein:

```
telnet ⟨Server⟩ ⟨Port⟩
```

Wichtig: Die Parameter »⟨Server⟩« und »⟨Port⟩« werden hier nicht, wie sonst oft üblich, durch einen Doppelpunkt, sondern durch Leerzeichen getrennt.

Glossar: Port

⟨Server⟩: Ersetzen durch Name oder IP-Nummer des Rechners, der angesprochen werden soll. Der Hamster ist meist auf demselben Rechner installiert, an dem man sitzt. Bei Kommunikation mit seinem Nager kann man dann »localhost« oder »127.0.0.1« für »⟨Server⟩« setzen (hier und im Folgenden natürlich immer ohne Anführungszeichen).

⟨Port⟩: Ersetzen durch Portnummer. Wenn sie von der jeweiligen Standardportnummer abweicht, sollte das bei der Einrichtung bekannt gegeben worden sein.
Für die Standardportnummern gibt es auch Portbezeichnungen, die man hier ebenso verwenden kann. Welche Bezeichnung welchem Port zugeordnet ist (z. B. »smtp« statt »25« oder »nntp« statt »119«), entnimmt Windows einer Datei namens »`services`« (*ohne* Extension), die bei neueren Windowsversionen in »`%Windir%\System32\Drivers\etc\`« zu finden ist, bei älteren im Windows-Hauptverzeichnis.

Die folgende Serverantwort kann sehr verschieden aussehen. Für die Kommunikation untereinander ist aber – mit einigen Ausnahmen – nur die Zeichenkette am Anfang wichtig (meist dreistellige Zahlen), der sehr oft folgende Text ist nur für menschliche Leser gedacht.

Das weitere Vorgehen hängt von dieser Serverantwort ab und davon, was man überprüfen will: Meldet sich beispielsweise statt des erwarteten Servers ein anderes Programm, muss man in dieser Richtung weitersehen.

Ansonsten: Je nachdem, wo es Probleme gibt, siehe unten weiter unter »News mit Telnet«, unter »Mail abholen mit Telnet – IMAP« bzw. »Mail abholen mit Telnet – POP3« oder aber unter »Mail versenden mit Telnet«.

IMAP
POP3

Wie oben bereits erwähnt, handelt es sich hier jeweils nur um einen Überblick. Ausführliche Informationen findet man direkt im jeweiligen RFC bzw. in den jeweiligen RFCs, die z. B. unter `http://www.ietf.org/rfc.html` oder `http://www.rfc-editor.org/rfc.html` abgerufen werden können.

Alle folgenden Eingaben, insbesondere die Anmeldung, sollte man immer einigermaßen zügig hintereinander machen. Sonst kommt es leicht zu einem Timeout, und die Verbindung wird vom Server beendet.

Es ist übrigens egal, ob man die Kommandos in Groß- oder Kleinschreibung eingibt. Server, die nur eine Schreibvariante akzeptieren, sind fehlkonfiguriert. Wenn man aber bei einer Schreibweise ausschließlich Fehlermeldungen erhält, kann man es auch einmal mit der anderen Variante probieren.

Damit man im Telnet-Fenster die eigenen Nutzereingaben sehen kann, muss man bei Windows 9x das lokale Echo aktivieren über das Menü »Terminal« → »Einstellungen« → »[x] Lokales Echo«. Bei Windows-NT und Folgeversionen muss man dafür unter Umständen einmalig etwas anders vorgehen, als oben beschrieben: Nur »`telnet`« eingeben, die Anzeige ändert sich zu »`Microsoft Telnet>` «; dort eingeben: »`set localecho`« und nach Betätigung der Enter↵-Taste entweder mit »`open ⟨Server⟩ ⟨Port⟩`« fortfahren oder mit »`quit`« den Telnet-Client beenden.

Die Korrektur der Eingaben funktioniert bei allen Windows-Versionen nicht – auch wenn nach einem Korrekturversuch im Telnet-Fenster der richtige Befehl angezeigt wird, antwortet der Server mit einer Fehlermeldung (bei PuTTY funktioniert das Löschen dagegen).

Bei Windows 9x kann zusätzlich der Inhalt des Telnet-Fensters nicht gescrollt werden. Will man nachträglich die vorher gemachten Eingaben und Antworten darauf sehen (um sie z. B. in ein Newsposting einzufügen), empfiehlt sich ein Log. Dieses kann mit »Terminal« → »Protokoll starten…« erstellt werden; zum Abschluss dann auf »Terminal« → »Protokoll beenden« gehen (bei TT wird dagegen zwangsweise eine Logdatei erstellt).

News mit Telnet

Glossar:
NNTP
Port

Das zugrundeliegende Protokoll ist NNTP, der Standardport ist 119.

Zur Verbindungsaufnahme also eingeben:

```
> telnet pausbacke.hamster.localhost 119
```

Wenn nichts kaputt ist, sieht die Antwort etwa so aus:

```
< 200 NNTP-Server Classic Hamster Version 42.0 (Build 42.0.8.15) (post ok)
    on pausbacke.hamster.localhost says: Hi!
```

Meldet sich jetzt nicht der erwartete Newsserver, kann man hier aufhören weiterzulesen und in Abhängigkeit von der Antwort an anderer Stelle nach Lösungen suchen.

Zur Bedeutung der dreistelligen Zahlen (Statuscodes):

- 1xx – Informative Mitteilung
- 2xx – Kommando in Ordnung
- 3xx – Kommando bis jetzt in Ordnung, erwarte weitere Eingaben
- 4xx – Kommando korrekt, konnte aber aus verschiedenen Gründen nicht ausgeführt werden
- 5xx – Kommando inkorrekt oder nicht implementiert, oder schwerer Programmfehler aufgetreten

Genaueres s. unter »Servermeldungen bzw. Protokoll-Antwortcodes« bzw. im RFC 977. Nur soviel: Oben steht als erstes eine »200« – d. h., dass man auf diesem Server posten darf. Ein Server, der nur Lesezugang gewährt, würde eine »201« melden.

Übersicht über unten aufgeführte Kommandos:

- Hilfe ausgeben lassen → **HELP**
- Authentifizierung → **AUTHINFO USER & AUTHINFO PASS**
- Sich als Newsclient ausgeben → **MODE READER**
- Anzeige der auf dem Server vorhandenen Newsgroups und Zusatzinformationen → **LIST / LIST ACTIVE & LIST NEWSGROUPS**
- Anfrage nach neuen Gruppen → **NEWGROUPS**
- Wechseln in eine Gruppe → **GROUP**
- Übersicht über Artikel einer Gruppe → **XOVER**
- Anzeige eines einzelnen Artikels → **ARTICLE**

hier auch:

  - Anzeige einzelner Header → **HEAD**
  - Anzeige einzelner Textbodys → **BODY**
- Absetzen eines Postings → **POST**

  hier auch:
  - Absetzen eines **Cancel**s
  - Absetzen eines **Supersedes**
- Ordnungsgemäßes Verlassen → **QUIT**

Bei Eingabe von `HELP` erhält man meist eine Anzeige, welche Kommandos der Server unterstützt. (Das funktioniert gewöhnlich ohne Authentifizierung. Siehe sonst folgenden Punkt »**AUTHINFO ...**«.)

```
> help
< 100 Implemented commands follow:
< [...]
```

Das Kommando muss von allen Servern unterstützt werden, die ausgegebene Liste kann aber auch leer sein!

Zur Authentifizierung dienen die Kommandos `AUTHINFO USER` und `AUTHINFO PASS`, s. folgendes Bsp.; »⟨User⟩« durch den Benutzernamen ersetzen und »⟨Pass⟩« durch das Passwort.

Achtung! Das Passwort ist im Klartext zu sehen und sollte bei einer Veröffentlichung des Dialogs anonymisiert werden (z. B. durch Sterne).

```
> authinfo user ⟨User⟩
< 381 More authentication information required
> authinfo pass ⟨Pass⟩
< 281 Authentication accepted
```

Alles gut, wir sind schon drin (Das war ja einfach!).

Bei manchen Servern ist es von Vorteil, durch Eingabe des Kommandos `MODE READER` mitzuteilen, dass man ein Client ist. Bei den meisten ist das aber die Voreinstellung.

```
> mode reader
< 200 You are already in this mode. Ignored.
```

Hier hätte man es also auch weglassen können.

Eingabe von `LIST` oder `LIST ACTIVE` liefert eine Anzeige der auf dem Server vorhandenen Newsgroups (pro Zeile eine), zum Format s. u.

Achtung! Das sind bei vielen Servern Zehntausende von Gruppen!

```
> list
< 215 list of newsgroups follows
```

»list active ⟨Gruppenmuster⟩« liefert ebenfalls eine Anzeige der auf dem Server vorhandenen Newsgroups (wieder pro Zeile eine), aber auf die in »⟨Gruppenmuster⟩« angegebene(n) Gruppe(n) eingeschränkt. Als Wildcards erlaubt sind »*« und »?« (und weitere, siehe in RFC 2980).

Achtung! Lässt man »⟨Gruppenmuster⟩« weg, hat man dasselbe Ergebnis wie oben!

```
> list active hamster.de.t*
< 215 list of newsgroups follows
< hamster.de.talk 54321 12345 y
< hamster.de.tools 9876 6789 y
< .
```

Verallgemeinert sieht jede Zeile so aus:

»⟨Newsgroup c b x⟩«, wobei ›c‹ für die höchste vorhandene Artikelnummer steht, ›b‹ für die niedrigste und ›x‹ für einen der Buchstaben »y«, »n« oder »m«, was wiederum bedeutet »Posten erlaubt« (y) bzw. »nicht erlaubt« (n) oder »moderierte Gruppe« (m).

Bei einer leeren Liste würde nur der Punkt als Antwort erscheinen.

Das Kommando »LIST NEWSGROUPS ⟨Gruppenmuster⟩«, liefert für die auf dem Server existierende(n) und in »⟨Gruppenmuster⟩« angegebene(n) Gruppe(n) den (die) Gruppennamen und eine evtl. vorhandene Kurzbeschreibung. Hier sind dieselben Wildcards wie oben erlaubt.

Achtung! Lässt man »⟨Gruppenmuster⟩« weg, bekommt man *alle* existierenden Gruppen aufgelistet!

```
> list newsgroups hamster.de.t*
< 215 information follows
< hamster.de.talk Gespraeche mit und ohne Hamster.
< hamster.de.tools        Zusatztools fuer den Hamster.
< .
```

Eine leere Liste liefert wieder nur den Punkt.

Mit dem Kommando NEWGROUPS kann man erfahren, ob seit einem bestimmten Datum neue Gruppen angelegt wurden. Das Kommando muss man dazu in folgender Form eingeben: »newsgroups ⟨JJMMTT HHMMSS⟩« mit ›JJ‹ = Jahr, ›MM‹ = Monat, ›TT‹ = Tag , ›HH‹ = Stunde, ›MM‹ = Minute, ›SS‹ = Sekunde (alles zweistellig).

```
> newgroups 050523 233205
< 231 list of new newsgroups follows
< gruppe.gibt.es 32 23 y
< .
```

Hier gibt es seit dem 23. Mai 2005, 23:32:05, nur eine neue Gruppe. Das Format der Antwort ist dasselbe wie oben bei **LIST**. Eine leere Liste liefert wieder nur den Punkt.

Mit der Eingabe von »GROUP ⟨Newsgroup⟩« wechselt man in die mit »⟨Newsgroup⟩« angegebene Gruppe, sofern vorhanden. Die Serverantwort liefert dabei noch einige Informationen:

```
> group gruppe.gibt.es
< 211 5 23 32 gruppe.gibt.es
```

Verallgemeinert sieht die Serverantwort so aus:

»⟨Statuscode⟩ a b c ⟨Newsgroup⟩«. Dabei steht ›a‹ für die Anzahl der Artikel in der Gruppe, ›b‹ für die niedrigste vorhandene Artikelnummer, ›c‹ für die höchste.

Wenn man sich die Differenz von ›b‹ und ›c‹ im Beispiel ansieht, sollte man erwarten, dass 10 Artikel vorhanden sind. Der Server meldet aber, dass er nur 5 Artikel besitzt. Die Erklärung für diesen scheinbaren Widerspruch ist, dass einzelne Artikel bereits wieder nachträglich gelöscht wurden (hier genau 5), zum Beispiel durch einen Cancel. Die Folge ist, dass keine fortlaufende und ununterbrochene Numerierung mehr existiert und es bei Anforderung einiger Artikel nach Nummer (s. u. bei **ARTICLE**) Fehlermeldungen geben würde. (Nur fehlkonfigurierte Server füllen die Löcher in der Numerierung wieder auf.)

Mit dem Kommando »XOVER ⟨Bereich⟩«, erhält man eine Übersicht der mit »⟨Bereich⟩« angegebenen Artikel in der zuvor ausgewählten Gruppe. Diese Übersicht enthält beim Hamster die Headerfelder »Subject:«, »From:«, »Date:«, »Message-ID:«, »References:« (nur, wenn nötig), »Bytes:«, »Lines:« und »Xref:«, und zwar in dieser Reihenfolge ohne, dass die Bezeichnungen dieser Headerfelder dabei stehen (Ausnahme Xref). »⟨Bereich⟩« kann »⟨niedrigereNummer⟩-⟨höhereNummer⟩« oder eine einzelne Artikelnummer enthalten.

Glossar: Header

Welche Headerfelder in der Übersicht ausgegeben werden und in welcher Reihenfolge, kann man bei den meisten Servern durch Eingabe von »LIST OVERVIEW.FMT« erfahren.

```
> list overview.fmt
< 215 information follows
< Subject:
< [... ]
< Xref:full
< .

> group local.test
< 211 66 33 99 local.test
> xover 42
< 224 Overview information follows
< 42  Testposting  <user@example.net>  Tue, 20 Apr 2004 20:04:04 +0200
<  <message-id.des.postings.nr.4711@mid.example.net>
<  <message-id.des.referenzierten.postings@mid.example.net>  123  5
<  Xref: pausbacke.hamster.localhost local.test:42
```

```
    [Je nach Einstellung des Servers kann das im Original eine lange Zeile
     sein. Wie man sieht, steht als erstes (und letztes in Xref:) die
     Artikelnummer.]
< .
```

Hier markiert der allein stehende Punkt das Ende der Ausgabe.

Sollte man das **GROUP**-Kommando zuvor noch nicht ausgeführt haben, gibt es eine Fehlermeldung wie diese:

```
< 412 No news group selected
```

Zur Auswahl und Anzeige eines Artikels mittels `ARTICLE` gibt es zwei Varianten:

Entweder kann man in der Gruppe, in der man sich gerade befindet, einen Artikel über seine Nummer aufrufen oder unabhängig davon, welche Gruppe gerade eingestellt ist, jeden Artikel über seine Message-ID.

Will man statt des gesamten Artikels nur die Headerzeilen bzw. den Body erhalten, gibt man statt **ARTICLE** jeweils entweder `HEAD` oder `BODY` ein. Die drei Kommandos funktionieren auf identische Weise.

Variante 1: »article ⟨nummer⟩«

```
> group gruppe.gibt.es
< 211 5 23 32 gruppe.gibt.es
> article 23
< 220 23 <message-id_des_artikels23_in_gruppe.gibt.es@mid.example.net>
    article
< [...]
< .
```

Der allein stehende Punkt ist hier die Markierung für das Artikelende.

Wie die Serverausgabe zeigt, steht bei der Antwort nach dem Statuscode immer zunächst die Artikelnummer, dann die Message-ID.

Sollte man bei dieser Variante das **GROUP**-Kommando zuvor noch nicht ausgeführt haben, gibt es eine Fehlermeldung wie diese:

```
< 412 no newsgroup has been selected
```

Variante 2: »article ⟨message-id⟩«

```
> article
    <message-id.eines.artikels.irgendwo.auf.dem.Newsserver@mid.example.net>
< 220 42
    <message-id.eines.artikels.irgendwo.auf.dem.Newsserver@mid.example.net>
    article
< [...]
< .
```

Zum Absetzen eines Postings – per Telnet am besten nur in Testgruppen – muss man zunächst das Kommando POST eingeben, nach der Serverantwort dann auf eigenen Zeilen mindestens »From:«, »Subject:« und »Newsgroups:« angeben, anschließend eine Leerzeile einfügen, dann den Text und am Ende einen Punkt auf einer eigenen Zeile eingeben (jede Zeile mit Enter↲ abschließen). Es ist nicht nötig, zuvor das **GROUP**-Kommando auszuführen.

```
> post
< 340 OK, recommended ID <vorgeschlagene.message-id@mid.example.net>
> from: <user@example.net>
> subject: Testposting
> newsgroups: local.test
> [Leerzeile]
> Nur'n Test.
> .
< 240 article posted ok
```

Um einen **Artikel** zu **canceln** (*genau genommen:* um die Newsserver zu bitten, ob sie so nett sein könnten, den Artikel zu löschen), muss außer den oben genannten noch ein weiteres Headerfeld »Control:« mit folgendem Inhalt angegeben werden: »control: cancel ⟨Message-ID_Des_Zu_Cancelnden_Postings⟩«.

Glossar: Header

Entweder im »From:« oder in einem zusätzlichen Headerfeld »Sender:« muss dieselbe E-Mail-Adresse verwendet werden, die in dem zu cancelnden Posting im »From:« bzw., sofern vorhanden, im »Sender:« stand.

```
> post
< 340 OK, recommended ID
    <vorgeschlagene.message-id.fuer.cancel@mid.example.net>
> from: <alternative-adresse@example.net>
> sender: <user@example.net>
> subject: cmsg <message-id.des.zu.cancelnden.postings@mid.example.net>
> control: cancel <message-id.des.zu.cancelnden.postings@mid.example.net>
> newsgroups: local.test
> [Leerzeile]
> Cancelmessage.
> .
< 240 article posted ok
```

Server, die Cancel ausführen (siehe auch später folgenden Absatz!), sollten danach auf das Kommando »article ⟨NummerDesGecanceltenPostings⟩« eine Fehlermeldung ausgeben. Da aus Performancegründen bei vielen großen Servern die Datenbank nur in größeren Abständen, unter Umständen nur einmal täglich (gewöhnlich nachts) gesäubert (»gepurget«) wird, kann es sein, dass der Artikel noch über »article ⟨Message-ID_Des_Gecancelten_Postings⟩« für einige Zeit abrufbar ist.

Will man dagegen einen geposteten Artikel durch eine neuere Version ersetzen, gibt man als zusätzliches Headerfeld »**Supersedes:**« mit folgendem Inhalt an: »supersedes: ⟨Message-

ID_Des_Zu_Ersetzenden_Postings⟩«. Ideal ist, wenn man in diesem Fall auch die Message-ID des zu ersetzenden Postings als Referenz angibt (Headerfeld »References:«).

Glossar: Header

```
> post
< 340 OK, recommended ID
    <vorgeschlagene.message-id.fuer.supersede@mid.example.net>
> from: <user@example.net>
> subject: Supersede-Test, 2. Posting
> references: <message-id.des.zu.ersetzenden.postings@mid.example.net>
> supersedes: <message-id.des.zu.ersetzenden.postings@mid.example.net>
> newsgroups: local.test
> [Leerzeile]
> Achtung, Test, Test, Test!
> Dieses Posting soll das zuvor gesendete Posting mit der Message-ID
> <message-id.des.zu.ersetzenden.postings@mid.example.net> und dem
> Subject 'Supersede-Test, 1. Posting' ersetzen.
> .
< 240 article posted ok
```

Auch hier ist kein Server dazu gezwungen, ein solches Supersede auszuführen. Geschieht das nicht und hat man die References mit angegeben, bilden die Postings einen Thread (wobei das auch von der eventuellen Änderung des Subjects und den entsprechenden Einstellungen im Newsreader abhängt). Siehe auch folgenden Absatz!

Sowohl das Canceln als auch das Superseden per Telnet funktioniert *nicht* bei Servern, die den Cancel-Lock-Mechanismus benutzen, mehr dazu siehe im Wikipedia-Artikel Canceln und den Anleitungen der einzelnen Server.

Zum ordnungsgemäßen Verlassen »QUIT« eingeben:

```
> quit
< 205 Closing connection.
```

Mail abholen mit Telnet – IMAP

Der Standardport ist 143.

Port

Das Protokoll IMAP ist sehr umfangreich (RFC 3501 umfasst 108 Seiten), und vor allem sind die Kommandos zum Teil komplex. Deshalb meint der maßgebliche Entwickler des IMAP-Servers im Classic-Hamster: »*Zum Teil per Telnet machbar, aber man will es nicht, es ist auch nicht dazu gedacht.*« Für umfangreiche Tests sollte man sich lieber einen alternativen IMAP-Clienten mit Logging-Fähigkeiten zulegen.

IMAP

Beschränken wir uns auf die etwas einfacheren Kommandos:

Zur Verbindungsaufnahme eingeben:

```
> telnet pausbacke.hamster.localhost 143
```

Eine positive Antwort sieht z. B. so aus:

```
< * OK IMAP4rev1 Server Classic Hamster Version 42.0 (Build 42.0.8.15) on
    pausbacke.hamster.localhost greets you!
```

Glossar: IMAP

Meldet sich jetzt nicht der erwartete IMAP-Server, kann man hier aufhören weiterzulesen und in Abhängigkeit von der Antwort an anderer Stelle nach Lösungen suchen.

Bevor es weitergeht, ein wichtiger Hinweis:

Bei IMAP muss jedem Client-Kommando eine eindeutige alphanumerische Zeichenkette vorangehen, genannt Tag (englisch und deshalb als »Tägg« ausgesprochen). Am einfachsten ist es, von Null oder Eins hochzuzählen, evtl. noch zur Unterscheidung von anderen Zahlen durch einen oder wenige Buchstaben ergänzt. Die Serverantwort beginnt dann jedes Mal mit diesem Tag oder stattdessen einem Stern.

Vergisst man das Tag, wird der Server das Kommando irrtümlich für das Tag halten!

Mögliche Statusanzeigen sind:

⟨Tag⟩ OK: Kommandoausführung erfolgreich

* OK: Informative Mitteilung

⟨Tag⟩ NO: Fehlermeldung – korrektes Kommando, Ausführung aber nicht erfolgreich

* NO: Warnmeldung – Kommandoausführung aber möglich

⟨Tag⟩ BAD: Fehlermeldung – Kommando bekannt, Ausführung aber wegen falscher Eingabe nicht möglich (z. B. unvollständiges Kommando)

* BAD: Fehlermeldung – unbekanntes Kommando oder interner Serverfehler

* PREAUTH: bereits mit Verbindungsaufnahme ist auf andere Weise Authentifizierung erfolgt (z. B. über Einwahl) – *erscheint nur zu Beginn der Verbindung*

* BYE: Antwort nach ordnungsgemäßer Abmeldung

Übersicht über unten aufgeführte Kommandos:

- Hilfe anzeigen lassen → **HELP**
- Auflistung der Fähigkeiten des Servers → **CAPABILITY**
- Authentifizierung → **LOGIN**
- Anzeige der Ordner → **LIST**
- Anlegen von Ordnern → **CREATE**
- Löschen von Ordnern → **DELETE**
- Umbenennen von Ordnern → **RENAME**
- Anzeige der Abonnements → **LSUB**

- Abonnieren von Ordnern bzw. Abonnement aufheben → **SUBSCRIBE & UNSUBSCRIBE**
- Ordnungsgemäßes Verlassen → **LOGOUT**

Obwohl es im RFC nicht definiert ist, kann man beim Hamster und evtl. einigen anderen Servern das Kommando `HELP` eingeben, um eine Anzeige zu erhalten, welche Kommandos der Server unterstützt.

```
> Tag1 help
< * Implemented Commands follow
< Tag1 {42}
< [...]
```

Um eine Auflistung der Fähigkeiten des Servers zu erhalten, muss man `CAPABILITY` eingeben.

```
> Tag2 capability
< * CAPABILITY IMAP4rev1 AUTH=CRAM-SHA1 AUTH=CRAM-MD5 AUTH=DIGEST-MD5
< + AUTH=LOGIN IDLE LITERAL ID
< Tag2 OK I'm ready sending capabilities!
```

Wenn bei einem Server nicht aus Sicherheitsgründen `LOGIN` deaktiviert wurde (das sollte bei **CAPABILITY** als »LOGINDISABLED« deklariert werden), kann man sich mit »⟨Tag⟩ login ⟨User⟩ ⟨Pass⟩« anmelden. »⟨User⟩« natürlich durch den Benutzernamen ersetzen und »⟨Pass⟩« durch das Passwort.

Achtung! Das Passwort ist im Klartext zu sehen und sollte bei einer Veröffentlichung des Dialogs anonymisiert werden (z. B. durch Sterne).

```
> Tag3 login ⟨User⟩ ⟨Pass⟩
< Tag3 OK LOGIN completed.
```

Bei **LOGINDISABLED** sähe die Antwort etwa so aus:

```
> Tag3 login ⟨User⟩ ⟨Pass⟩
< Tag3 BAD LOGIN is switched off by server-admin. Please use AUTHENTICATE.
```

Allerdings sind die anderen Authentifizierungsmethoden (vgl. oben bei **CAPABILITY** die Angaben »AUTH=... «) nicht für eine Telnet-Session geeignet.

Das Kommando `LIST` dient zur Anzeige der Ordner. Die Angabe erfolgt in der Form »⟨Tag⟩ list "⟨Referenz⟩" "⟨Mailbox⟩"«, wobei »⟨Referenz⟩« für einen Pfad zur Mailbox steht (die genaue Implementation ist serverabhängig!) und »⟨Mailbox⟩« für einen Ordner steht. Bei beiden sind die Wildcards »*« und »%« erlaubt, die sich nur dadurch unterscheiden, dass »*« den Hierarchietrenner einschließt (welcher das ist, hängt vom Server ab), »%« dagegen nicht (d. h. z. B., mit »⟨Tag⟩ list "*" "*"« wird die *gesamte* Ordnerliste angefordert!); ob Groß-/Kleinschreibung unterschieden wird, ist abhängig vom Betriebssystem des Servers (Windows und damit dem Hamster ist es egal). Leere Strings werden durch »""« (2x ") angegeben.

```
> Tag4 list "*" "*"
< * LIST () "/" "INBOX"
< [...]
< * LIST () "/" "Test"
< * LIST () "/" "Test/Mail1"
< * LIST () "/" "Test/Mail2"
< Tag4 OK You have now the List!

> Tag5 list "" ""
< * LIST (\NOSELECT) "/" ""
< Tag5 OK You have now the List!

> Tag6 list "test" "*"
< * LIST () "/" "Test"
< * LIST () "/" "Test/Mail"
< * LIST () "/" "Test/Mail2"
< Tag6 OK You have now the List!

> Tag7 list "test" "%"
< * LIST () "/" "Test"
< Tag7 OK You have now the List!

> Tag8 list "test" "%/%"
< * LIST () "/" "Test/Mail"
< * LIST () "/" "Test/Mail2"
< Tag8 OK You have now the List!
```

Mit »⟨Tag⟩ CREATE ⟨Ordner⟩« wird ein Ordner angelegt. Wenn der Server die Erstellung von Unterordnern unterstützt, dann kann man hier bei »⟨Ordner⟩« den Pfad mit angeben. Wenn der übergeordnete Ordner dabei noch nicht existiert, wird er mitangelegt. Für den Namen sollten bei einer Telnet-Session nur ASCII-Zeichen von 0x21 (dez. 33) bis 0x7E (dez. 126) mit Ausnahme des »&« verwendet werden. Ob zwischen Groß- und Kleinschreibung unterschieden wird, ist abhängig vom Betriebssystem des Servers.

```
> Tag9 create Testordner/Unterordner
< Tag9 OK Mailbox created!
> Tag10 list * testo*
< * LIST () "/" "Testordner"
< * LIST () "/" "Testordner/Unterordner"
< Tag10 OK You have now the List!
```

Gelöscht werden Ordner mit »⟨Tag⟩ DELETE ⟨Ordner⟩«. Enthält der ausgewählte Ordner jedoch noch Unterordner, so kann er nicht gelöscht werden, sondern es wird dabei sein Status geändert. Ebenso wenig kann der Ordner **INBOX** gelöscht werden. Ob für »⟨Ordner⟩« zwischen Groß- und Kleinschreibung unterschieden wird, ist abhängig vom Betriebssystem des Servers.

```
> Tag11 delete testordner
< Tag11 OK Mailbox deleted!
```

```
> Tag12 list * testo*
< * LIST (\NOSELECT) "/" "Testordner"
< * LIST () "/" "Testordner/Unterordner"
< Tag12 OK You have now the List!

> Tag13 delete testordner/unterordner
< Tag13 OK Mailbox deleted!
> Tag14 delete testordner
< Tag14 OK Mailbox deleted!
> Tag15 list * testo*
< Tag15 OK You have now the List!
```

Mit dem Kommando **RENAME** können Ordner umbenannt werden. Die Syntax ist »⟨Tag⟩ rename ⟨alterOrdnername⟩ ⟨neuerOrdnername⟩«. Wenn der Server Unterordner unterstützt, dann muss man hier jeweils zusätzlich den Pfad angeben. Wenn bei »⟨neuerOrdnername⟩« der übergeordnete Ordner noch nicht existiert, wird er mitangelegt. Für den neuen Namen sollten bei einer Telnet-Session nur ASCII-Zeichen von 0x21 (dez. 33) bis 0x7E (dez. 126) mit Ausnahme des »&« verwendet werden. Ob zwischen Groß- und Kleinschreibung unterschieden wird, ist abhängig vom Betriebssystem des Servers.

```
> Tag16 rename Test/Mail Test/Mail1
< Tag16 OK Mailbox renamed.
> Tag17 list "test" "%/%"
< * LIST () "/" "Test/Mail1"
< * LIST () "/" "Test/Mail2"
< Tag17 OK You have now the List!
```

Mit dem Kommando **LSUB** kann man sich die abonnierten Ordner anzeigen lassen. Die Angabe erfolgt in der Form »⟨Tag⟩ lsub "⟨Referenz⟩" "⟨Mailbox⟩"«, wobei »⟨Referenz⟩« für einen Pfad zur Mailbox steht (die genaue Implementation ist serverabhängig!) und »⟨Mailbox⟩« für einen Ordner steht. Bei beiden sind die Wildcards »*« und »%« erlaubt, die sich nur dadurch unterscheiden, dass »*« den Hierarchietrenner einschließt (welcher das ist, hängt vom Server ab), »%« dagegen nicht (d. h. z. B., mit »⟨Tag⟩ lsub "*" "*"« wird die *gesamte* Liste der abonnierten Ordner angefordert!); ob Groß-/Kleinschreibung unterschieden wird, ist abhängig vom Betriebssystem des Servers (Windows und damit dem Hamster ist es egal). Leere Strings werden durch »""« (2x ") angegeben.

```
> Tag18 lsub "*" "*"
< * LSUB () "/" "Test"
< Tag18 OK You have now the List!
```

Hier ist nur der Ordner »Test« abonniert. Bei einer leeren Liste würde nur »⟨Tag⟩ OK ...« angezeigt werden.

Mit dem Kommando SUBSCRIBE abonniert man einen Ordner. Es muss für jeden zu abonnierenden Ordner ein eigenes »⟨Tag⟩ subscribe ⟨Ordner⟩« eingegeben werden, wobei »⟨Ordner⟩« inklusive Pfad anzugeben ist.

Umgekehrt wird mit »⟨Tag⟩ UNSUBSCRIBE ⟨Ordner⟩« ein Abonnement aufgehoben. Auch das muss für jeden Ordner einzeln durchgeführt werden.

Ob für »⟨Ordner⟩« Groß-/Kleinschreibung unterschieden wird, ist abhängig vom Betriebssystem des Servers.

```
> Tag19 subscribe test/mail1
< Tag19 OK Mailbox subscribed
> Tag20 subscribe test/mail2
< Tag20 OK Mailbox subscribed
> Tag21 lsub "*" "mail*"
< * LSUB () "/" "Test/Mail1"
< * LSUB () "/" "Test/Mail2"
< Tag21 OK You have now the List!

> Tag22 unsubscribe test/mail1
< Tag22 OK Mailbox unsubscribed
> Tag23 unsubscribe test/mail2
< Tag23 OK Mailbox unsubscribed
> Tag24 lsub "*" "mail*"
< Tag24 OK You have now the List!
```

Zum ordnungsgemäßen Verlassen dient das Kommando LOGOUT:

```
> Tag25 logout
< * BYE IMAP4rev1 closing connection - goodbye!
< Tag25 OK Closing.
```

Mail abholen mit Telnet – POP3

Glossar: POP3 Port

Der POP3-Standardport ist 110.

Zur Verbindungsaufnahme also eingeben:

```
> telnet pausbacke.hamster.localhost 110
```

Eine positive Antwort sieht z. B. so aus:

```
< +OK POP3-Server Classic Hamster Version 42.0 (Build 42.0.8.15)
    pausbacke.hamster.localhost greets you!
```

Meldet sich jetzt nicht der erwartete POP3-Server, kann man hier aufhören weiterzulesen und in Abhängigkeit von der Antwort an anderer Stelle nach Lösungen suchen.

Mögliche Statusanzeigen (hier nur zwei verschiedene ohne Zahlen):

`+OK` – Alles in Ordnung
`-ERR` – Fehler

Übersicht über unten aufgeführte Kommandos:

- Hilfe ausgeben lassen → **HELP**
- Optionale Kommandos anzeigen lassen → **CAPA**
- Authentifizierung → **USER & PASS**
- Übersicht über Mailbox → **STAT**
- Aufstellung des Mailboxinhaltes → **LIST**
- Auflistung der UID → **UIDL**
- Header abrufen (und optional Teile des Bodys) → **TOP**
- Eine Nachricht vollständig abrufen → **RETR**
- Nachrichten löschen → **DELE**
- Markierung zum Löschen rückgängig machen → **RSET**
- Ordnungsgemäßes Verlassen → **QUIT**

Obwohl es im RFC nicht definiert ist, kann man bei vielen Servern das Kommando `HELP` eingeben, um eine Anzeige zu erhalten, welche Kommandos der Server unterstützt.

```
> help
< +OK Implemented commands follow:
< [...]
< .
```

Mit dem Kommando `CAPA` kann man sich bei den meisten Servern die Authentifizierungsmöglichkeiten und optionale Kommandos anzeigen lassen.

```
> capa
< +OK Capability of non-authorization follows:
< TOP
< USER
< [...]
< UIDL
< IMPLEMENTATION HamsterPOP3Server
< .
```

Die Angaben können variieren in Abhängigkeit davon, ob man authentifiziert ist oder nicht, deshalb auch der Hinweis in der ersten Zeile der Serverantwort.

Zur Authentifizierung dienen die Kommandos `USER` und `PASS`, s. folgendes Bsp.; »⟨User⟩« durch den Benutzernamen ersetzen und »⟨Pass⟩« durch das Passwort. (Es gibt weitere Authentifizierungsmethoden, die aber für eine Telnet-Session nicht geeignet sind.)

Achtung! Das Passwort ist im Klartext zu sehen und sollte bei einer Veröffentlichung des Dialogs anonymisiert werden (z. B. durch Sterne).

```
> user ⟨User⟩
< +OK More authentication information required
> pass ⟨Pass⟩
< +OK mailbox locked, 5 messages
```

Alles ist gut – wir sind drin und bekommen bei diesem Server auch gleich gemeldet, dass es 5 Nachrichten in der Mailbox gibt.

Mit dem Kommando `STAT` erhält man eine Übersicht über seine Mailbox.

```
> stat
< +OK 5 3220
```

Verallgemeinert sieht die Antwort so aus: »⟨Statusanzeige⟩ n m«, wobei ›n‹ die Anzahl der Nachrichten in der Mailbox darstellt (hier also 5) und ›m‹ die Gesamtgröße in Bytes angibt (hier 3220 Bytes).

Mit dem Kommando `LIST` kann man eine detailliertere Aufstellung erhalten. Für jede Nachricht wird dann in einer eigenen Zeile erst die Nummer und dann die Größe in Bytes aufgeführt. Wenn man viele Nachrichten in der Mailbox hat, kann die Liste also sehr groß werden!

Mit »`list ⟨Nummer⟩`« kann man die Angabe auf eine einzelne Nachricht einschränken. Die Numerierung erfolgt bei jeder Anmeldung neu, ändert sich während jeder Session aber nicht.

```
> list
< +OK 5 messages
< 1 999
< 2 666
< 3 1111
< 4 333
< 5 111
< .
> list 3
< +OK 3 1111
```

Ein optionales Feature, das viele Server nutzen, ist die Vergabe von UID (»Unique Identification Numbers« = eindeutige Identifikationsnummern). Wie der Name vermuten lässt, bleibt die UID für jede Nachricht immer konstant. Dadurch erst wird es zum Beispiel möglich, bereits einmal geladene Mail bei späteren Sessions zu ignorieren oder nach einem festen Zeitraum zu löschen.

Um eine Auflistung dieser UIDs zu bekommen, muss man das Kommando UIDL (für »Unique ID Listing«) eingeben. Ob ein Server dieses Kommando unterstützt, erfährt man durch Eingabe von **CAPA**.

Mit »uidl ⟨Nummer⟩« wird die Ausgabe wieder auf die mit »⟨Nummer⟩« angegebene Nachricht beschränkt, wobei man »⟨Nummer⟩« durch das Kommando **LIST** erhält.

```
> uidl
< +OK
< 1 abcdefghijklmnopqrstuvwxyz1234567890
< 2 zyxwvutsrqponmlkjihgfedcba0987654321
< 3 1234567890abcdefghijklmnopqrstuvwxyz
< 4 abcdefghijklmnopqrstuvwxyz0987654321
< 5 zyxwvutsrqponmlkjihgfedcba1234567890
< .
> uidl 3
< +OK 3 1234567890abcdefghijklmnopqrstuvwxyz
```

Das erlaubte Format der UID ist: 1 bis 70 Zeichen von 0x21 (dez. 33) bis 0x7E (dez. 126), also aus dem ASCII-Bereich.

Manche Server haben in der Vergangenheit das Gebot der Eindeutigkeit verletzt, indem sie immer wieder neue »UID« (die hier diesen Namen nicht verdienten) vergaben.

Auch das Kommando TOP ist zwar laut RFC optional, wird aber von fast allen Servern unterstützt, wenn auch zum Teil nicht ganz korrekt (mehr dazu unten). Ob ein Server dieses Kommando unterstützt, erfährt man durch Eingabe von **CAPA**.

Glossar: Header

Mit »top ⟨Nummer⟩ ⟨Zeilen⟩« erhält man alle Header und die durch »⟨Zeilen⟩« angeforderte Anzahl an Zeilen des Bodys der mit »⟨Nummer⟩« angegebenen Nachricht. »⟨Nummer⟩« erhält man durch das Kommando **LIST**. Besitzt die Nachricht weniger Zeilen, als angefordert wurden, wird der gesamte Inhalt ausgegeben, ohne dass Leerzeilen ergänzt werden.

```
> top 3 1
< +OK 1111 octets
< Received: from localhost (HELO [127.0.0.1]) [127.0.0.1]
<         by pausbacke.hamster.localhost (192.168.0.1)  (userid 1)
<         with ESMTP (Classic Hamster Version 42.0 Build 42.0.8.15) ;
<         Tue, 20 Apr 2004 20:04:04 +0200
< From: Hamster-Administrator <admin@hamster.example.net>
< [weitere Headerzeilen]
< [Leerzeile]
< Du, mein liebster User!
< .
```

Hier ist »Du, mein liebster User!« die erste und als einzige auch übertragene Zeile des Bodys. Manche Server interpretieren dagegen »⟨Zeilen⟩« fälschlicherweise unter Einbeziehung der Header, so dass diese unter Umständen nur zum Teil ausgegeben werden. Bei dem obigen

Beispiel mit »top 3 1« wäre dann nur die erste Headerzeile (»Received: [...]«) übertragen worden!

Ein weiteres Problem mit diesem Kommando gibt es oft – gerade in der Zusammenarbeit des Hamsters – mit externen Spamfilterprogrammen.

Mit »RETR ⟨Nummer⟩« ruft man eine Nachricht immer vollständig ab. An »⟨Nummer⟩« gelangt man durch das Kommando **LIST**.

```
> retr 3
< +OK 1111 octets
< Received: from localhost (HELO [127.0.0.1]) [127.0.0.1]
<         by pausbacke.hamster.localhost (192.168.0.1)  (userid 1)
<         with ESMTP (Classic Hamster Version 42.0 Build 42.0.8.15) ;
<         Tue, 20 Apr 2004 20:04:04 +0200
< From: Hamster-Administrator <admin@hamster.example.net>
< To: User <user@hamster.example.net>
< [Rest der Nachricht]
< .
```

Gelöscht werden Nachrichten durch Eingabe von »DELE ⟨Nummer⟩«. »⟨Nummer⟩« erhält man durch das Kommando **LIST**.

```
> dele 3
< +OK message 3 marked for deletion
> dele 5
< +OK message 5 marked for deletion
> list
< +OK 3 messages
< 1 999
< 2 666
< 4 333
< .
```

Das bewirkt während der Session zunächst nur, dass die durch »⟨Nummer⟩« angegebene Nachricht zum Löschen *vorgemerkt* wird. Der eigentliche Löschvorgang erfolgt erst, wenn man sich ordnungsgemäß abgemeldet hat (**QUIT**). Bricht die Verbindung zum Server ab oder schließt man Telnet ohne ordnungsgemäße Abmeldung, bleiben die Nachrichten erhalten. Trotzdem wird man bei den meisten Servern noch während der Session nicht mehr auf die jeweilige Nachricht zugreifen können, d. h. bei allen oben erwähnten Kommandos wird die Nachricht nicht mehr erscheinen bzw. wird der Server eine Fehlermeldung ausgeben.

Will man die Markierung zum Löschen rückgängig machen, muss man das Kommando RSET angeben. Die Markierung wird damit für *alle* zum Löschen markierten Nachrichten entfernt!

```
> rset
< +OK mailbox has 5 messages
> list
```

```
< +OK 5 messages
< 1 999
< 2 666
< 3 1111
< 4 333
< 5 111
< .
```

Zum ordnungsgemäßen Verlassen »QUIT« eingeben:

```
> quit
< +OK closing connection - goodbye!
```

Mail versenden mit Telnet

Glossar: SMTP, Port, SMTP-AUTH

Das zugrundeliegende Protokoll ist SMTP, der Standardport ist 25.

Später wurde das Protokoll erweitert (ESMTP = Extended SMTP), auf dessen Grundlage unter anderem auch die Möglichkeit zur Authentifizierung (SMTP-AUTH) basiert. Die Portnummer hat sich dabei aber nicht verändert. Viele Server bieten aber zusätzlich auf Port 587 einen Zugang exklusiv per SMTP-AUTH.

Zur Verbindungsaufnahme also eingeben:

```
> telnet pausbacke.hamster.localhost 25
```

Eine positive Antwort sieht z. B. so aus:

```
< 220 SMTP-Server Classic Hamster Version 42.0 (Build 42.0.8.15)
    pausbacke.hamster.localhost is ready.
```

Meldet sich jetzt nicht der erwartete Mailserver, kann man hier aufhören weiterzulesen und in Abhängigkeit von der Antwort an anderer Stelle nach Lösungen suchen.

Zur Bedeutung der dreistelligen Zahlen (Statuscodes):

- 1xx – Kommando bis jetzt in Ordnung, erwarte zusätzliche Eingaben, um weitermachen zu können *(nur bei ESMTP)*
- 2xx – Kommando in Ordnung
- 3xx – Kommando bis jetzt in Ordnung, erwarte weitere Eingaben (nur bei Kommandosequenzen)
- 4xx – temporärer Fehler
- 5xx – permanenter Fehler

Genaueres s. unter »Servermeldungen bzw. Protokoll-Antwortcodes« bzw. im RFC 2821.

Übersicht über unten aufgeführte Kommandos:

- Hilfe ausgeben lassen → **HELP**
- Anmeldung nach klassischem SMTP → **HELO**
- Anmeldung gemäß ESMTP → **EHLO**
- Authentifizierung → **AUTH**
- Mail verschicken → **MAIL FROM, RCPT TO & DATA**
- Ordnungsgemäßes Verlassen → **QUIT**

Bei Eingabe von `HELP` erhält man meist eine Anzeige, welche Kommandos der Server unterstützt.

```
> help
< 214-Implemented commands follow:
< 214-helo
< 214-ehlo
< [...]
```

Das Kommando muss von allen Servern unterstützt werden, die ausgegebene Liste kann aber auch leer sein!

Nach klassischem SMTP erfolgt die Anmeldung mit »`HELO ⟨Hostname⟩`« (von »Hello«). »⟨Hostname⟩« muss dabei mindestens bei Rechnern, die nicht nur innerhalb eines LAN erreichbar bzw. nicht nur per Telefon an das Internet angebunden sind, ein »Fully Qualified Domain Name« (siehe Glossareintrag FQDN) sein oder wenigstens eine IP-Nummer. Eine Authentifizierung ist nicht erforderlich und standardmäßig auch nicht möglich.

```
> helo gold.hamster.localhost
< 250 helo gold.hamster.localhost
```

Von den meisten SMTP-Servern wird aber inzwischen die Anmeldung gemäß ESMTP unterstützt. Dazu ist das Kommando »`EHLO ⟨Hostname⟩`« (von »Extended HELO«) einzugeben.

```
> ehlo gold.hamster.localhost
< 250-pausbacke.hamster.localhost
< 250-8BITMIME
< 250-AUTH DIGEST-MD5 CRAM-SHA1 CRAM-MD5 LOGIN
< 250-AUTH=DIGEST-MD5 CRAM-SHA1 CRAM-MD5 LOGIN
< 250 HELP
```

Der Server gibt daraufhin in seiner Antwort – gewöhnlich in mehreren Zeilen – zuerst seinen Domainnamen an und danach bekannt, welche Erweiterungen er unterstützt. Jede Zeile mit Ausnahme der letzten muss so aussehen: »250-⟨unterstützte_Erweiterung⟩ [optionale Parameter]«; nur in der letzten Zeile hat statt des »-« ein Leerzeichen zu stehen, wie auch im Beispiel zu sehen ist. Ein Server, der keine Erweiterungen besitzt, müsste dem entsprechend so antworten: »`250 ⟨Servername⟩`«.

Aufgrund der Spamproblematik muss man sich bei den meisten Servern inzwischen authentifizieren, sonst nehmen sie nur noch Mail für Ihre eigene Domain an (und das zum Teil auch nur von bestimmten IP-Adressen).

Läuft die Authentifizierung nicht über die per Telefoneinwahl zugewiesene IP-Nummer des eigenen Providers, ist besonders die obige Zeile beim **EHLO**-Kommando mit »**AUTH**« von Interesse (die zweite Zeile fängt einen Bug in einigen Netscape-Versionen ab). Hier teilt der Server mit, welche Authentifizierungsmechanismen er kennt. Die einfachste ohne SSL angebotene Variante ist »**LOGIN**«, bei der Username und Passwort base64-codiert werden, wodurch sie zwar verschleiert, aber auch leicht wieder zu decodieren sind.

Glossar: SSL, Base64

Die anderen Varianten sind für die Arbeit mit Telnet jedoch nicht praktikabel. So bleibt nur die Authentifizierung durch Eingabe von »`AUTH LOGIN`«. Man benötigt dafür allerdings ein externes Tool, das Zeichenketten, hier den Usernamen und das Passwort, base64-codieren kann, wie beispielsweise UUDeview, den Totalcommander oder etwas Spezielles wie `base64.exe`.

Achtung! Das Passwort ist leicht zu decodieren und sollte bei einer Veröffentlichung des Dialogs anonymisiert werden (z. B. durch Sterne).

```
> auth login
< 334 VXNlcm5hbWU6
  [D. h. base64-decodiert »334 Username:«.]
> dXNlcg==
  [D. h. »user«, also verallgemeinert der Username in Base64.]
< 334 UGFzc3dvcmQ6
  [D. h. »334 Password:«.]
> Z2VoZWlt
  [D. h. »geheim«, also verallgemeinert das Passwort in Base64.]
< 235 Authentication successful.
```

Um eine Nachricht zu verschicken, muss man folgendermaßen vorgehen:

Zunächst muss man den sogenannten »Envelope« (Umschlag) angeben (s. u. dazu mehr), nämlich »`MAIL FROM:`« für den (wahren) Absender und »`RCPT TO:`« für den oder die (wahren) Empfänger, und zwar für jeden Empfänger in einer eigenen Zeile. Das Kommando `DATA` leitet dann die Eingabe der Nachricht ein. Dann muss mindestens das Headerfeld »From:« angegeben werden,[1] »Subject:« und »To:« sind aber auch zu empfehlen bzw. viele Server verlangen die Angabe von »To:«. Alles immer auf eigenen Zeilen eingeben. Anschließend eine Leerzeile einfügen, dann den Text und am Ende einen Punkt auf einer eigenen Zeile eingeben (jede Zeile mit Enter↲ abschließen). Die Nachricht darf in der unten angegebenen Form nur ASCII-Zeichen enthalten, was für Testmail meistens ausreichen sollte.

Glossar: Header

```
> mail from: User Hase <hase@hamster.localhost>
< 250 OK
```

[1] Laut RFC 2822 muss »Date:« ebenso unbedingt angegeben werden, aber dieser Header wird üblicherweise auch durch die Server gesetzt, wenn er fehlt.

```
> rcpt to: Hamsteradmin <admin@hamster.localhost>
< 250 OK
> rcpt to: User Maus <maus@hamster.localhost>
> data
< 354 Start mail input; end with <CRLF>.<CRLF>
> from: Hase <hase@example.net>
> to: Goldhamster <goldhamster@example.net>
> subject: Testmail
> [Leerzeile]
> Das ist nur eine Testnachricht.
> Euer Hasilein.
> .
< 250 OK
```

Am Ende vor dem Punkt auf eigener Zeile nicht noch eine Leerzeile eingeben, das soll laut RFC 2821 unbedingt vermieden werden (ist allerdings bei einer Testnachricht nicht so tragisch).

Bemerkenswert ist in diesem Zusammenhang, dass die in der Nachricht angegebenen Absender- und Empfängeradressen, obwohl sie Pflichtangaben sind, für den Transport keinerlei Bedeutung haben, sondern nur die bei »**MAIL FROM:**« und »**RCPT TO:**« angegebenen, die bei den meisten Servern nicht in der Nachricht angegeben werden, da sie, wie oben erwähnt, den »Umschlag« darstellen (Ausnahmen sind solche Angaben wie »X-Envelope-From:« bzw. »X-Envelope-To:«).

Im obigen Beispiel ist die wahre Absenderadresse deshalb »<hase@hamster.localhost>«, und die Mail geht an »<admin@hamster.localhost>« und »<maus@hamster.localhost>«, obwohl in »To:« und »From:« etwas anderes steht. Bei dem Großteil der Spamflut, die fast jeder von uns erhält, ist das nicht anders.

Zum ordnungsgemäßen Verlassen »`QUIT`« eingeben:

```
> quit
< 221 closing connection - goodbye!
```

Import von E-Mail in IMAP-Postfach

„ Wie importiere ich meine Mails in ein IMAP-Postfach? “

Glossar: IMAP

Achtung: Beim Umgang mit großen Datenmengen können unerwartete Probleme auftreten. *Du solltest auf jeden Fall vorher ein Backup anlegen.*

Möglichkeit 1: Import über einen IMAP-Client

Sofern Du Deine Mails bereits in einem IMAP-fähigen Client untergebracht hast, kopiere diese doch direkt in einen/den IMAP-Ordner. Ich persönlich bin aber eher vorsichtig und würde

nur einen Thread gleichzeitig an der Mailbox arbeiten lassen oder zumindest IDLE (das ist ein interner IMAP-Mechanismus) ausschalten, sofern möglich.

Möglichkeit 2: Direktimport auf Dateiebene

Du kannst Mails als Textdatei (»*.eml*«; für jede Mail eine einzelne Datei) speichern, exportieren oder konvertieren. Beende Hamster oder fahre den IMAP-Server herunter. Wenn auch das nicht möglich ist, trenne zumindest alle Clients von der Mailbox (und lasse sie getrennt). Erstelle Dir dann einen neuen (!) Unterordner in der IMAP-Mailbox mit beliebigem (7-Bit-ASCII-)Namen. Dort kannst Du die Mails (die einzelnen Textdateien) reinkopieren. Benenne diese Dateien um in »`nnnn.msg`« (siehe Hamster-Namensformat). Starte den IMAP-Server und abonniere in deinem IMAP-Client den neuen Ordner. Der Hamster erzeugt dabei automatisch eine neue Datenbank, wodurch die E-Mails zur Verfügung gestellt werden. Danach existieren natürlich keine Statusinformationen, d.h. Du kannst damit keine Information importieren, ob die E-Mails gelesen/beantwortet wurden.

Glossar: IMAP

Falls erwünscht, kannst Du die E-Mails im Client noch in den eigentlichen Zielordner verschieben.

Möglichkeit 3: Abrufen aus einem POP3-Konto

POP3

Du hast die Mails noch in einer POP3-Box des Hamsters (als »`*.msg`«). Dann lasse den Hamster sich selbst POPpen (d.h. als POP3-Server ist »localhost« anzugeben) und schreibe entsprechende Mailfilter-Regeln, um diese Mails in Deinen (neuen) IMAP-Ordner zu kopieren.

Schlussfolgerung:

Am einfachsten kopierst Du die Mails direkt mittels eines IMAP-fähigen Clients (diese haben meistens auch Import-Funktionalität) oder Du konvertierst die Mails in das Hamsterformat und machst bei Möglichkeit 2 weiter. Möglichkeit 3 ist ein Spezialfall, bei dem man aber Kontrolle hat, wohin die Mails sortiert werden.

Maximale Dateigröße im Hamster

> *„Kann es sein, dass der Hamster über einer Dateigröße von 2 GiB nicht mehr hamstert?“*

Ja, im Hamster Classic beträgt die absolute Maximalgröße der Dateien 2^{31} (= 2 147 483 648) Bytes = 2 GiB.

Im Hamster Playground ist die absolute Maximalgröße dagegen 2^{32} (= 4 294 967 296) Bytes = 4 GiB.

Beide arbeiten zwar mit 32-Bit-Werten (Integern), aber im Hamster Classic sind sie vorzeichenbehaftet, d.h. es wird 1 Bit für die Vorzeicheninterpretation verwendet.

Unterschied zwischen »Gruppe zurücksetzen« und »Gruppe löschen«

> *„Was ist der Unterschied zwischen »Gruppe zurücksetzen« und »Gruppe löschen« und wann sollte ich welche Möglichkeit von beiden benutzen?“*

»Gruppe zurücksetzen« bewirkt, dass eine bereits bestellte Gruppe in einen »jungfräulichen« Zustand versetzt wird, so als ob sie eine neu bestellte Gruppe wäre. Es werden alle Einträge dieser Gruppe aus der History entfernt und die Artikeldatenbank geleert. Schließlich werden auch noch die Zähler für den Artikelstand gelöscht. Die Gruppe selbst – d. h. der Eintrag in der »`groups.!!!`«, das entsprechende Verzeichnis und die zugehörigen Ladeaufträge – bleibt jedoch erhalten.

Im Gegensatz dazu wird mit »Gruppe löschen« lediglich der entsprechende Ladeauftrag entfernt und das zur Gruppe gehörige Verzeichnis samt darin enthaltenen Dateien gelöscht, ***nicht*** *aber die Einträge in der Artikelhistory*. Das bedeutet bei einem erneuten Abonnieren der Gruppe, dass die bisherigen Artikel dieser Gruppe immer noch bekannt sind, und daher nicht erneut geladen werden. Wird ein gelöschter Artikel angefragt, wird seine Message-ID auch nicht in die Liste nachzuladender MIDs eingetragen, sondern der Hamster wird das mit einer normalen NNTP-Fehlermeldung quittieren.

Glossar: NNTP

Wenn man sichergehen will, sollte man eine Gruppe vor dem Löschen immer zurücksetzen. Wenn man aber eine Gruppe nur kurzzeitig abonniert hatte und sich sicher ist, dass man sie in kürzerer Zeit nicht wieder lesen wird, reicht aber trotz allem das Löschen aus und man kann auf das Zurücksetzen verzichten.

Achtung! *Das Zurücksetzen einer Gruppe setzt voraus, dass die »*`data.idx`*« der betreffenden Gruppe noch vorhanden ist. Ein Entfernen der alten Einträge aus der History ist nicht mehr möglich nach dem Löschen der Gruppe, auch wenn sie danach neu angelegt wurde.*

Glossar

Base64 Verfahren zur Codierung von 8-Bit-Binärdaten in eine Zeichenfolge, die nur aus lesbaren codepage-unabhängigen ASCII-Zeichen besteht (d. h. 7-Bit-Zeichencodierung). Vorher lesbarer Text ist nicht mehr erkennbar, aber leicht wieder zu decodieren (vgl. Quoted-printable).

FQDN Fully Qualified Domain Name:
Der FQDN ist die einmalige und universelle Beschreibung eines Rechners oder einer Gruppe von Rechnern. Er existiert weltweit einmalig und dient zur eindeutigen Identifizierung eines Rechners bzw. einer Rechnergruppe und der von diesem Rechner bzw. Rechnergruppe gesendeten Artikel.

Ein FQDN kann nicht selbst ausgedacht werden, sondern man braucht entweder eine eigene Domain oder muss eine Subdomain vom Provider oder einem anderen Domain-Inhaber zur Verfügung gestellt bekommen. Kostenlose Domain-Namen kann man z. B. von einem DynDNS-Provider oder `http://www.my-fqdn.de/` (dort auch mehr Informationen) beziehen. Ebenso bieten die Usenetprovider oft diese Möglichkeit.

Header Header von E-Mail oder Usenet-Artikeln:
Als Header eines Artikels oder einer E-Mail werden die jeweiligen Kopfzeilen als einzelne Zeile oder in ihrer Gesamtheit bezeichnet. Header steuern die Weiterverbreitung von Artikeln und E-Mails, stellen ihren Status dar oder transportieren technische Informationen zur Darstellung des Artikels bzw. der E-Mail.

IMAP Internet Message Access Protocol, Version 4, revision 1,
ursprünglich RFC 2060 (M. Crispin, 1996), ersetzt durch RFC 3501 (M. Crispin, 2003):
Anwendungsprotokoll für den Zugriff auf und die Verwaltung von empfangenen E-Mails, die sich in einem Postfach auf einem Mailserver befinden, also nicht auf den lokalen Rechner heruntergeladen werden müssen.

NNTP Network News Transfer Protocol,
ursprünglich RFC 977 (Brian Kantor, Phil Lapsley, 1986), ersetzt durch RFC 3977 (C. Feather, 2006):
Softwareprotokoll zum Austausch von News im Usenet.

PEM Privacy Enhanced Mail Security Certificate:
Format zum Transport von kryptografischen Schlüsseln, bei dem zur Erhöhung der Transportsicherheit durch Base64-Codierung Byte(=8-Bit)-Sequenzen in 7-Bit-Sequenzen umgewandelt werden.

POP3 Post Office Protocol, Version 3,
STD 53/RFC 1939 (J. Myers, M. Rose, 1996):
Softwareprotokoll zum Abholen von E-Mails aus Postfächern.

Port Im Zusammenhang mit den Internet-Protokollen TCP und UDP wird der Begriff Port für die Adressierung eines Datenkanals innerhalb dieser beiden Protokolle verwendet. Der Wertebereich der Ports ist 0–65535. Der Port 0 kann bei vielen Systemen nicht verwendet werden.

Viele Ports, insbesondere im Bereich von 0–1023, sind für bestimmte Dienste reserviert. Die Liste der reservierten Ports kann bei der IANA eingesehen werden (`http://www.iana.org/assignments/port-numbers`, vgl. auch in der Wikipedia: Liste der standardisierten Ports). Für Ports können bei vielen Betriebssystemen auch Namen verwendet werden. Die Zuordnung zwischen Namen und Port-Nummern wird in der Regel in einer Datei mit dem Namen »`services`« verwaltet.

Beispiele:

Name	*Port*	*Name*	*Port*
SMTP	25	SMTPS	465
POP3	110	NNTPS	563
NNTP	119	IMAPS	993
IMAP	143	POP3S	995

Quoted-printable Verfahren, bei dem 8-bit-codierte Zeichen außerhalb des ASCII-Zeichensatzes durch 7-bit-codierte Zeichenfolgen ersetzt werden. Die Lesbarkeit des Textes bleibt so meistens einigermaßen erhalten (vgl. Base64).

ReCo Remote-Control-Server:
Server zur Fernsteuerung des Hamsters mittels Telnet-Clienten.

SASL Simple Authentication and Security Layer:
Verfahren zur einheitlichen Verwendung verschiedener Authentifizierungsmechanismen. Bekannte Authentifizierungsmechanismen sind CRAM-MD5, LOGIN, PLAIN usw.

SMTP Simple Mail Transfer Protocol,
ursprünglich RFC 821 (Jonathan B. Postel, 1982), ersetzt durch RFC 2821 (J. Klensin, 2001):
Softwareprotokoll zum Versenden von E-Mails (SMTP mit Authentifizierung: SMTP-AUTH).

SMTP-AUTH Simple Mail Transfer Protocol Authentication,
ursprünglich RFC 2554 (J. Myers, 1999), ersetzt durch RFC 4954 (R. Siemborski, A. Melnikov, 2007), RFC 2831 (P. Leach, C. Newman, 2000):
Softwareprotokoll-Erweiterung zur Berechtigungsüberprüfung beim Versenden von E-Mails, nutzt SASL-Mechanismen.

SSL Secure Socket Layer:
Kryptografisches Verfahren zum Schutz von TCP-Verbindungen zwischen Server und Client. Der Nachfolger wird als TLS bezeichnet. Für nähere Informationen siehe auch bei `http://www.openssl.org/` oder in der Wikipedia unter Transport Layer Security.

TCP Transmission Control Protocol:
TCP ist das am weitesten verbreitete Protokoll für die Netzwerkverbindung zwischen Client und Server. Es setzt auf dem Transportprotokoll IP (*Internet Protocol*) auf und ermöglicht eine fehlerfreie Verbindung. TCP sorgt dafür, dass die Datenpakete, welche über IP ankommen, in die richtige Reihenfolge gebracht und gegebenenfalls korrigiert werden (die Kombination wird gewöhnlich als *TCP/IP* bezeichnet).

TLS Transport Layer Security:
Kryptografisches Verfahren zum Schutz von TCP-Verbindungen zwischen Server und Client, Weiterentwicklung von SSL (TLS 1.0 = SSL 3.1). Für nähere Informationen siehe auch bei `http://www.openssl.org/` oder in der Wikipedia unter Transport Layer Security.

X.509 ITU-Standard für digitale kryptografische Zertifikate. Sie dienen der sicheren Identifizierung von Personen, E-Mail-Adressen oder Internet-Servern.

Register

Hinweis: Anders als innerhalb der Hilfe erfolgt die Verlinkung vom gesamten Register aus immer auf den Beginn einer Seite. Der gesuchte Begriff kann dort aber auch ganz am Ende zu finden sein. Deshalb ändert sich der Fokus nach einem Klick auf einen Seitenlink so, dass die gesamte Seite angezeigt wird.

Gesamtindex

A

Actions 214
ActiveX-Schnittstelle ... *s.* OLE-Schnittstelle
Alias-Namen
– Externe Server
• Newsserver 33
• POP3-Mailserver 43
• SMTP-Mailserver 47
– Lokale Benutzer 18

C

COM-Schnittstelle ..... *s.* OLE-Schnittstelle

D

data.ini, Optionen
– Created 116
– description 116
– feedonly 115
– LastClientRead 116
– Local.Max 116
– Local.Min 116
– Moderator 115
– postserver 115
– pull.limit 115
– purge.articles.keepcount 116
– purge.articles.keepdays 115
– ⟨Server⟩.FirstPullDone . 116
– ⟨Server⟩.Low 116
– ⟨Server⟩.Max 116
– ⟨Server⟩.Min 116
– Type 115
Dateien 61
– Aufräumen 7
– ~ im Hauptverzeichnis .. 61
• hamster.ini ..... 63, *s. auch Indexeintrag* hamster.ini
• IPAccess.hst 87
• MailFilt.hst 96
• MAlias.hst 106
• Password.!!! 109
• Scores.hst 89
• Zeichensatzkonvertierungsdateien 108
– ~ in »⟨Hauptverzeichnis⟩\Groups« 114
– ~ in »⟨Hauptverzeichnis⟩\Groups\News.err« 114
– ~ in »⟨Hauptverzeichnis⟩\Groups\News.out« 114
– ~ in »⟨Hauptverzeichnis⟩\Groups\⟨Newsgruppenname⟩« 115
• data.ini ..... 115, *s. auch* data.ini, Optionen
– ~ in »⟨Hauptverzeichnis⟩\Logs« 119
– ~ in »⟨Hauptverzeichnis⟩\Mails« 117
– ~ in »⟨Hauptverzeichnis⟩\Mails\admin« 117
– ~ in »⟨Hauptverzeichnis⟩\Mails\⟨Benutzer⟩« 117
– ~ in »⟨Hauptverzeichnis⟩\Mails\Mail.out« 117
– ~ in »⟨Hauptverzeichnis⟩\Server« 110
– ~ in »⟨Hauptverzeichnis⟩\Server\⟨Servername⟩« . 110
• server.ini ..... 110, *s. auch* server.ini, Optionen
– News-History erneuern ... 7
– Spezielle Dateien bearbeiten 52
Datenfluss
– News 236
DCOM-Schnittstelle .... *s.* OLE-Schnittstelle

E

Einstellungen . *s.* Konfiguration
Errorcodes *s.* Fehlermeldungen
Event-Objekte 215
Events ..... *s.* Hamsterscript
Externe Server
– NNTP
• Einstellungen 32
• Newsserver 33
– POP3
• Einstellungen 42
• Mailserver 43
– SMTP
• Einstellungen 47
• Mailserver 47

F

FAQ 287
Fehlermeldungen 236
– DFÜ 240
– häufige ~ 237
– RAS 240
– Serverfehler 274
– SSL/TLS 269
– Win32-Netzwerk 263
– Win32-System 243
– WinSock 239
Fehlernummern .. *s.* Fehlermeldungen
Fernsteuerung 276
– Installation 276
– Syntax des Servers 277

G

Globale Gruppenliste, Erzeugen/Erneuern 7

H

Häufige Fragen u. Probleme (FAQ) 287
Hamster-Mailinglisten 281
hamster.ini
– Optionen *s. eigenen* Index d. »Hamster.ini«-Optionen
– Sektionen 63
• [Actions] 79
• [Directories] 64
• [Main] 78
• [Setup] 65

• Sonstige ~ 83
Hamsterscript ..... 120, *s. auch* Index d. Hamsterscript-Anweisungen
– Actions 214
– Anweisungen 121
• Präprozessor-~ 122
– Ausdrücke 126
• arithmetisch 126
• boolesch 129
– Befehle 121
– Boolesche Werte 126
– Concatenation 126
– Datentypen
• dynamisch 126
• elementar 125
– Events 206
– Funktionen 135
• Definition 135
• eigene ~ 136
• eingebaute ~ ..... *s. eigene Verweise:* allgem. Funktn. und Serverfunktn.
• Parameter 138
• Variablen 136
– Funktionen, eingebaute (allgemein) 141
• Artikelverwaltung 165
• Dateien 152
• DFÜ-Netzwerk 171
• E-Mail-Versand und -Empfang 190
• Ein- und Ausgabe 162
• Fehlerbehandlung 151
• Hamster-Steuerung allgemein 181
• Listen 157
• Newsartikel-Verwaltung 201
• Newsgruppen-Verwaltung 199
• Newsjobs-Liste bearbeiten 197
• News-Versand und -Empfang 194
• numerische ~ 143
• Pfade 188
• RAS-~ 171
• Remote-Access-Steuerung 188
• Scheduler 168
• Scoring 204
• sonstige ~ 174
• Statusabfragen 186
• String-~ 144
• Variablenmanipulation 142
• Verzeichnisse 152, 188
• Zeichenketten-~ 144
• Zeiger-~ 172
• Zeit-~ 149
• Zeitgeber 168
– Funktionen, eingebaute (Server) 181
– Integer 125
– Kommentare 121
– Konstanten 125
– Module 212
• Definition 212
• Gebrauch 214
– nebenläufige Skripte ... 205
• Events 206
• Synchronisation 205
– Operatoren 126
• arithmetisch 126
• Assoziativität 128
• binär 126
• boolesch 129
• logisch 129
• Priorität 128
• unär 126
– Parameter 138
• Rückgabe-~ 138
– Prozeduren 135
• Definition 135
• eigene ~ 136
• eingebaute ~ 139
• Funktionen 135
• Parameter 138
• Rekursion 136
• Variablen 136
– Rekursion 136
– Schleifen 133
• For-Schleifen 135
• postkonditionale ~ ... 134
• präkonditionale ~ 134
• Zählschleifen 135
– Sprünge 130
– Strings 125
– Synchronisation 205
• Events 206
– Variablen 123
• global 136
• lokal 136
• Sichtbarkeit 136
– Verzweigungen 131
– Wahrheitswerte 126
– Zeichenketten 125
Hamsterskript *s.* Hamsterscript
Hauptfenster
– Ausgewählter Auftrag ... 58
– hs2-Shell 59
• hs2Shell.hsm 59
– Newsjobsliste 58
History ..... *s.* News-History
hs2 ..... *s.* Hamsterscript
hs2-Shell 59
– hs2Shell.hsm 59

I

Installation 5
Interne Gruppen ..... *s.* Lokale Gruppen

K

Kommandozeilenoptionen . 279
Konfiguration
– Automatische Abläufe
• Aktionen 31
• Allgemeines 30
• Artikel über MID laden 32
– Benutzer & Passworte
• Alias-Namen 18
• Lokale Benutzer 15
• Passworte 18
– Erweiterte Einstellungen de-/aktivieren 52
– Grundeinstellungen
• Interne Gruppen 12
• Online-Menü 11
• Optik 8
• Protokolle 10
• SSL 13
• Verschiedenes 14
– Lokale Server
• Allgemeines 19
• Fernzugriff 29
• IMAP 25
• IP-Access 20
• NNTP 21
• POP3 24
• SMTP 27
– Mail-Killfile-Protokoll ... 49
– Mail-Server
• POP3-Einstellungen ... 42
• POP3-Mailserver 43
• SMTP-Einstellungen ... 47
• SMTP-Mailserver 47
– Neu laden 52
– News-Killfile-Protokoll/-Einstellungen 38
• Datei (Submenü) 40
• Einträge (Submenü) 41
• Kill-/Scorefile (Reiter) . 41
– News-Server & -Gruppen

• Einstellungen 32
• News-Ladeaufträge 37
• Newsgruppen 36
• Newsserver 33
– Spezielle Dateien bearbeiten 52

L
Linkliste 280
Lizenz 286
Log-Dateien, Konfigurieren .. 10
Lokale Benutzer 15
– Alias-Namen 18
– nntpdefault 15
– Passworte 18
Lokale Gruppen 12
Lokale Server 19, 54
– Allgemeines 19
– IMAP 25
• Starten/Beenden 54
– IP-Access 20
– NNTP 21
• Starten/Beenden 54
– POP3 24
• Starten/Beenden 54
– SMTP 27
• Starten/Beenden 54
– Telnet-Server .... 29, *s. auch* Fernsteuerung
• Starten/Beenden 55

M
Mail-Verzeichnisse 55
Mailinglisten 281
Menüs 7
– Datei 7
• Aktualisiere Menüeinträge 8
• Aufräumen 7
• Beenden 8
• Globale Gruppenliste ... 7
• Kennworte neu laden ... 8
• News-History erneuern .. 7
• Zähler/Protokoll löschen 8
– Einstellungen 8, *s. auch* Konfiguration
• Automatische Abläufe . 30
• Benutzerverwaltung & Passworte 15
• De-/Aktivieren erweiterte ~ 52
• Grundeinstellungen 8
• Konfiguration neu laden 52
• Lokale Server 19
• Mail: Killfile-Protokoll . 49
• Mail: Server konfigurieren 42
• News: Killfile-Protokoll/-Einstellungen 38
• News: Server, Gruppen & Ladeaufträge 32
• Spezielle Dateien 52
– Haupt-Kontextmenü
• Einzelzeile kopieren ... 58
• Protokoll kopieren 58
• Protokoll(e) leeren 59
– Lokale Server 54
• News-/Mail-Verzeichnisse 55
• Start/Stop IMAP-Server 54
• Start/Stop News-Server 54
• Start/Stop POP3-Server 54
• Start/Stop SMTP-Server 54
• Start/Stop Telnet-Server 55
– Online 52
• 1–9 52
• Alle Aufträge beenden . 53
• Alle NNTP-Server 53
• Alle POP3-Server 53
• Alle Server (News+Mail) 53
• DFÜ-Netzwerk 53
• DFÜ-Netzwerk schließen 53
• ⟨*Einzelner Server*⟩ 53
– Skript 53
• Alle Skripte beenden ... 54
• Skripte und Module verwalten 53
• ⟨*Verfügbare Skripte*⟩ 53
Module ..... *s.* Hamsterscript
Mutex-Objekte 216

N
News-Verzeichnisse 55
News-History, Erneuern 7
nntpdefault 15

O
OLE-Schnittstelle 217
– Funktionen ..... *s. auch* Index d. OLE-Schnittstellen-Anweisungen
• allgemein 218
• Auftrags-~ 222
• DFÜ-Netzwerk 230
• News-~ 230
• RAS-~ 230
• sonstige ~ 234
• Task-~ 222
– Hamster als Service 218
Ordner ..... *s.* Verzeichnisse

P
Protokoll-Antwortcodes 274
Protokolle, Konfigurieren ... 10

R
ReCo-Server .. *s.* Fernsteuerung
Remote-Control-Server . *s.* Fernsteuerung

S
Server, extern . *s.* Externe Server
Server, lokal ... *s.* Lokale Server
server.ini, Optionen
– 4-Digit-Years 112
– AUTH 113
– Connections.failed 113
– Connections.ok 113
– Disabled 111
– FilterSection 113
– ForcedAuthMech 112
– GetMaxMailsPerSession 113
– LastConnect.Count.failed 114
– LastConnect.Count.ok . 114
– LastConnect.failed 114
– LastConnect.ok 114
– LastGMT 112
– LeaveOnServer 113
– LoadDescs 112
– LocalUser 113
– PostDir 111
– ReadOnly 111
– remote.timeout.command 111
– remote.timeout.connect 111
– SASL 113
– Servertype 112
– SSLCaFile 111
– SSLMode 111
– SSLVerifyLevel 111
– Threads 112
– UsePostDir 111
– UseTOPCommand 112
Servermeldungen 274
Skript ..... *s.* Hamsterscript
Synchronisation .... *s.* Hamsterskript

T
Telnet-Server . *s.* Fernsteuerung

V
Versionsinformationen 281
Verzeichnisse 61
– Hauptverzeichnis 61

– ⟨Hauptverzeichnis⟩\
Groups 114
• News.err 114
• News.out 114
• ⟨Newsgruppenname⟩ . 115
– ⟨Hauptverzeichnis⟩\
Logs 119
– ⟨Hauptverzeichnis⟩\
Mails 117
• admin 117
• ⟨Benutzer⟩ 117
• Mail.out 117
– ⟨Hauptverzeichnis⟩\
Server 110
• ⟨Servername⟩ 110

Z

Zusatzprogramme 280

Optionen der »Hamster.ini«

A
AddUserAgent 68
AspectListToPreview 85
ateveryhour.exec 79
ateveryhour.pars 79
ateveryhour.script 79
ateveryhour.wait 79

B
BottomlineFormat 79

C
CaFile 86
CaPath 86
Charsets 64
color.debug.brush 78
color.debug.text 78
color.detail.brush 78
color.detail.text 78
color.else.brush 78
color.error.brush 78
color.error.text 78
color.info.brush 78
color.info.text 78
color.system.brush 78
color.system.text 78
color.warning.brush 78
color.warning.text 78
ColoredTabs 78
ColWidths 84
ComServername 70
config.local.treefull 65

D
dun.beforedial.exec 79
dun.beforedial.pars 79
dun.beforedial.script 79
dun.beforedial.wait 79
dun.connected.exec 79
dun.connected.pars 79
dun.connected.script 79
dun.connected.wait 79
dun.dialfailed.exec 79
dun.dialfailed.pars 79
dun.dialfailed.script 79
dun.dialfailed.wait 79
dun.hangup.exec 79
dun.hangup.pars 79
dun.hangup.script 79
dun.hangup.wait 79

E
editor.app 65
editor.⟨ext⟩.app 65
editor.⟨ext⟩.params 65
editor.params 65
EmptyTrashAfterXDays 85

F
FilterNewGroupsInfo 77
FQDN 69
FQDN2 69

G
GlobalList 85
group.doubleclick.exec 79
group.doubleclick.pars 79
group.doubleclick.script 79
group.doubleclick.wait 79
Groups 64

H
HighEncryption 86
HintFormat 79
Hist 87
history.chunkbits 65

I
ihave.auth 70
imap.7bit 72
imap.delay 72
imap.LOGINDisable 72
imap.NcBrain 73
imap.SASL.LOGINDisable ... 73
internalgroup.cancelnotify .. 70
internalgroup.default 70
internalgroup.newgroups 70
internalgroup.posterrors 70
internalgroup.postok 70
internalgroup.pullerrors 70
internalgroup.statistics 70
internalgroup.unknowngroup 70

K
Kill 87

L
Language.cached 65
language.savefile 65
Language.Update.en.dat 65
LastDir 85
LastInfoMsg 87
Load 87
local.autostart.nntp 71
local.autostart.pop3 74
local.autostart.reco 76
local.autostart.smtp 74
local.codepage 69
local.imap.id 73
local.imap.serverbind 73
local.imap.timeout.login 73
local.imap.tlsmode 73
local.limit.bytessend 69
local.limit.linelen.nntp 71
local.limit.linelen.pop3 74
local.limit.linelen.smtp 75
local.limit.sendfails 69
local.limit.textsize.imap 73
local.limit.textsize.nntp 71
local.limit.textsize.pop3 74
local.limit.textsize.smtp 75
local.list.require.auth 71
local.mail.
BounceIfUnknownUser .. 75
local.mail.LocalMIDFQDN .. 69
local.mail.reqnotauth 75
local.nntp.feedjunk 71
local.nntp.infomail 71
local.nntp.infopostfail 71
local.nntp.infopostok 71
local.nntp.serverbind 71
local.nntp.timeout.login 71
local.nntp.tlsmode 71
local.pop3.sasl 74
local.pop3.serverbind 74
local.pop3.timeout.login 74
local.pop3.tlsmode 74
local.port.imap 73
local.port.nntp 71
local.port.pop3 74
local.port.reco 76
local.port.smtp 75
local.postmaster 69
local.reco.sasl 76
local.reco.serverbind 76
local.reco.timeout.login 76
local.reco.usetls 76
local.smtp.sasl 75
local.smtp.serverbind 75
local.smtp.timeout.login 75
local.smtp.tlsmode 75
local.smtpafterpop3.period .. 69
local.timeout.inactivity 69

local.timeout.quitdelay 69
local.timewait.sendfails 69
local.usenet 69
log.FormatFileName 66
Log.IgnoreConnectionLost .. 66
logfilter.FileByKey 66
logfilter.FileByText 66
logfilter.ThreadByKey 66
logfilter.ThreadByText 66
logfilter.ViewByKey 66
logfilter.ViewByText 66
Logs 64
Logs.ShowHints 78
LowEncryption 86

M

mail.addreceived 68
mail.addxhamster 68
mail.DefColumnSort 83
mail.DefColumnSortUp 84
mail.ext 66
mail.ext.err 66
mail.filter.mailsize.delete 77
mail.filter.mailsize.ignore ... 77
mail.filter.mailsize.notify 77
mail.filter.toplines 77
mail.flupforgate 69
mail.GateUseHamster-
Envelope 70
mail.getheader.exec 79
mail.getheader.pars 79
mail.getheader.script 79
mail.getheader.wait 79
mail.in.exec 79
mail.in.pars 79
mail.in.script 79
mail.in.wait 79
mail.infomail.
addheadertosubject 70
mail.internal.exec 79
mail.internal.pars 79
mail.internal.script 79
mail.internal.wait 79
mail.local.exec 79
mail.local.pars 79
mail.local.script 79
mail.local.wait 79
Mail.out 64
mail.out.exec 79
mail.out.pars 79
mail.out.script 79
mail.out.wait 79
mail.removemids 70
Mail.ScoreFile.ExpiredEntries.
Delete 67
mail.ShowHeaders 84
mail.smtpneedbrackets 75
Mailboxes 84
Mails 64
Main 64
main.AskIfOutboxNotEmpty 67
Main.HideMenuItems 67
MaxLocalIMAPServers 73
MaxLocalIMAPServersPerIP . 73
MaxLocalNNTPServers 72
MaxLocalNNTPServersPerIP 72
MaxLocalPOP3Servers 74
MaxLocalPOP3ServersPerIP . 74
MaxLocalRECOServers 76
MaxLocalRECOServersPerIP 76
MaxLocalSMTPServers 75
MaxLocalSMTPServersPerIP . 75
MaxTasks 70
MaxUnknownGroupsInRe ... 77
MediumEncryption 86
mid.crypt 69
Modules 64

N

news.addpath 72
news.addxhamster 68
news.addxhtrace 72
news.cancelcontrolmsg 72
news.DefColumnSort 83
news.DefColumnSortUp 84
News.err 64
news.ext 66
news.ext.err 66
news.feededcancel 72
news.feededcancelcontrolmsg 72
news.feededcancelverify 72
news.feededsupersedes 72
news.gateway.exec 79
news.gateway.pars 79
news.gateway.script 79
news.gateway.wait 79
news.GenerateDate 72
news.in.nntp.exec 79
news.in.nntp.pars 79
news.in.nntp.script 79
news.in.nntp.wait 79
news.local.exec 79
news.local.pars 79
news.local.script 79
news.local.wait 79
news.localinjection 72
news.makeparts 77
News.out 64
news.out.exec 79
news.out.pars 79
news.out.script 79
news.out.wait 79
news.preprocess.exec 79
news.preprocess.pars 79
news.preprocess.script 79
news.preprocess.wait 79
News.ScoreFile.ExpiredEntries.
Delete 67
news.ShowHeaders 84
news.sizeofparts.max 77
news.sizeofparts.min 77
news.verifycancel 72
news2mail.MIDExtension ... 70
nntp.autogetserverinfos 77
nntp.dropresidualjobs 77
nntp.ModeReader 77
NoMailboxes 84

O

OwnTaskbarEntry
– ([EditDirs]) 85
– ([EditScripts]) 85

P

password.codebase 67
pop3.delay 74
Preview_AllHeader 84
Preview_FixedPitchFont 84
Preview_MaxLines 84
Preview_TextAttr 84
Preview_Wordwrap 84
PrivateKeyPair 86
purge.articles.keepcount 67
purge.articles.keepdays 67
purge.daily 67
purge.history.keepdays 67
purge.lastdailypurge 67
purge.mailinlog.keeplines ... 68
purge.mailoutlog.keeplines .. 68
purge.mhistory.keepdays 67
purge.newsoutlog.keeplines . 68
purge.rasdiallog.keeplines ... 68

R

RandFile 86
Remote.Timeout.Command . 76
Remote.Timeout.Connect 76

S

score.DeleteThread.expire ... 68

score.DeleteThread.section .. 68
score.DeleteThread.value 68
score.RetrieveThread.expire . 68
score.RetrieveThread.section 68
score.RetrieveThread.value .. 68
Script.Listbox.Height 68
Scripts 64
ScriptsRC 64
searchmid.exec 79
searchmid.pars 79
searchmid.script 79
searchmid.wait 79
SendInfoMailLocalOnly 70
sendmail.attempts.del 78
sendmail.attempts.max 78
Server 64
Server⟨x⟩ 86
ShellHist.Log 78
ShowDirectly 84
ShowFileTypes 83
ShowIMAPSubDirs 85
ShowShell 78
ShowTrash 85
shutdown.exec 79
shutdown.pars 79
shutdown.script 79
shutdown.wait 79
smtp.delay 76
startup.exec 79
Startup.OLEServer 70
startup.pars 79
startup.script 79
StartUp.ScriptTimeout 68
startup.splash 68
startup.wait 79
status.click.exec 79
status.click.pars 79
status.click.script 79
status.click.wait 79
status.doubleclick.exec 79
status.doubleclick.pars 79
status.doubleclick.script 79
status.doubleclick.wait 79

T

TabColorErrorBrush 78
TabColorErrorText 78
TabColorWarningBrush 78
TabColorWarningText 78
TaskbarTitle 78
Timeout.ExecInit 83
Timeout.ExecRun 83
Title 79
Title⟨x⟩ 86
Trash 64
trayicon.click.exec 79
trayicon.click.pars 79
trayicon.click.script 79
trayicon.click.wait 79
trayicon.doubleclick.exec 79
trayicon.doubleclick.pars 79
trayicon.doubleclick.script .. 79
trayicon.doubleclick.wait 79
trayicon.middleclick.exec 79
trayicon.middleclick.pars 79
trayicon.middleclick.script .. 79
trayicon.middleclick.wait 79
trayicon.middledoubleclick.
exec 79
trayicon.middledoubleclick.
pars 79
trayicon.middledoubleclick.
script 79
trayicon.middledoubleclick.
wait 79

U

UpdateIntervalms 84
Use3DES 86
UseADH 87
UseDES 86
UseExportAlgorithms 87
UseIDEA 86
UseMD5 86
UseRC2 86
UseRC4 86
UseServerModeV2 87
UseSHA1 86
UseSmallIcons 84
UseSSLv3 86
UseTLSv1 86

V

VerifyLevel 86

Hamsterscript-Anweisungen

#! (Präprozessor-Anweisgn.)
#!debug 122
#!finalize 122
#!hs2 122
#!initialize 122
#!load 122
#!menu 123
#!trace 122

A
abs 143
Addlog 163
ArtAddHeader 167
ArtAlloc 165
ArtDelHeader 168
ArtFree 165
ArtGetBody 166
ArtGetHeader 167
ArtGetHeaders 166
ArtGetText 166
ArtHeaderExists 167
ArtLoad 166
ArtSave 166
ArtSetBody 166
ArtSetHeader 167
ArtSetHeaders 166
ArtSetText 166
assert 139
AtAdd 168
AtClear 169
AtCount 170
AtEveryMins 171
AtExecute 169
AtFrom 170
AtOnDays 170
AtSubFunction 170
AtUntil 170

B
beep 139
break 133

C
chr 144
ClearXCounter 177
ClipRead 175
ClipWrite 175
const 125
ConstEnum 125
continue 133
copy 145

D
debug 140
dec 142
DecodeBase64 147
DecodeMIMEHeaderString . 147
DecodeQP 147
decodetime 150
DecodeToLocalCharset 147
DecXCounter 177
delete 146
DeleteHostsEntry 176
digest 175
DirChange 156
DirCurrent 156
DirExists 155
DirMake 156
DirRemove 156
DirSystem 156
DirWindows 156
DiskFreeKB 153
DLLCall 178
DLLFree 177
DllLastError 178
DllLoad 177
do ... loop 133
dump 140

E
else 132
elseif 132
encodetime 150
endfor 135
endif 131
endsub 136
endwhile 134
entercontext 124
ErrCatch 151
ErrLine 152
ErrLineNo 152
ErrModule 152
ErrMsg 151
ErrNum 151
error 139
ErrSender 152
eval 148
EventClose 209
EventCreate 208
EventMultipleWait 209
EventPulse 208
EventReset 207
EventSet 207
EventWait 207
execute 174

F
false 143
FileCopy 154
FileDelete 154
FileExists 153
FileMove 155
FileRename 154
FileSize 153
FileTime 154
for ... endfor 135

G
GetEnvironment 174
GetProcessIdentifier 175
GetTasksActive 176
GetTasksRun 176
GetTasksWait 176
GetUptimeDays 150
GetUptimeHours 150
GetUptimeMins 150
GetUptimeSecs 150
gosub 130
goto 130

H
HamAddGroup 200
HamAddPull 200
HamArtCount 201
HamArtDeleteMID 204
HamArtDeleteNrInGroup .. 204
HamArtImport 203
HamArtLocateMid 203
HamArtLocateMidInGroup 203
HamArtNoMax 202
HamArtNoMin 202
HamArtText 202
HamArtTextExport 202
HamChangePassword 181
HamCheckPurge 185
HamDelGroup 200
HamDelPull 200
HamDialogAddPull 185
HamDialogEditDirs 184
HamDialogKillFileLog 185
HamDialogMailKillFileLog . 185
HamDialogScripts 184
HamEnvelopeFrom 187
HamExePath 188
HamFetchMail 191

HamFlush 185
HamGetStatus 186
HamGroupClose 201
HamGroupCount 199
HamGroupDescriptionBy-
Handle 201
HamGroupIndex 199
HamGroupName 200
HamgroupNameByHandle . 201
HamGroupOpen 201
HamGroupsPath 188
HamHscPath 188
HamHsmPath 188
HamIMAPServer 183
HamIsIdle 189
HamLogsPath 188
HamMailExchange 190
HamMailPath 188
HamMailScoreList 204
HamMailsOutPath 188
HamMainFQDN 187
HamMainWindow 184
HamMessage 181
HamMIDFQDN 187
HamNewMail 193
HamNewsErrPath 188
HamNewsJobsAdd 199
HamNewsJobsCheckActive . 199
HamNewsJobsClear 194
HamNewsJobsDelete 199
HamNewsJobsFeed 197
HamNewsJobsGetCounter .. 197
HamNewsJobsGetParam ... 197
HamNewsJobsGetPriority .. 198
HamNewsJobsGetServer ... 197
HamNewsJobsGetType 198
HamNewsJobsPost 196
HamNewsJobsPostDef 196
HamNewsJobsPull 195
HamNewsJobsPullDef 195
HamNewsJobsSetPriority ... 198
HamNewsJobsStart 194
HamNewsOutPath 188
HamNewsPost 196
HamNewsPull 195
HamNNTPServer 183
HamPath 188
HamPOP3Server 183
HamPostmaster 187
HamPurge 185
HamRasDial 189
HamRasHangup 189
HamRCPath 188
HamRebuildGlobalLists 186
HamRebuildHistory 186
HamReCoServer 183
HamReloadConfig 184
HamReloadIPAccess 184
HamReloadPassword 181
HamResetCounters 183
HamRotateLog 163
HamScoreList 204
HamScoreTest 204
HamSendMail 192
HamSendMailAuth 192
HamServerPath 188
HamSetLogin 188
HamShutDown 183
HamSMTPServer 183
HamStopAllTasks 183
HamThreadCount 190
HamTrayIcon 184
HamUsenetAcc 188
HamVersion 186
HamWaitIdle 190
hex 145

I
icase 132
if ... else ... endif 132
if ... elseif ... else ... endif 132
if ... endif 131
iif 132
inc 142
IncXCounter 177
IniDelete 153
IniEraseSection 153
IniRead 152
IniWrite 153
InputBox 164
InputPW 165
int 143
isint 143
isstr 144

L
label 130
leavecontext 124
len 145
ListAdd 159
ListAlloc 157
ListAppend 161
ListBox 165
ListClear 158
ListCount 158
ListDelete 160
ListDirs 162
ListExists 157
ListFiles 161
ListFree 157
ListGet 158
ListGetKey 159
ListGetTag 158
ListGetText 160
ListIndexOf 161
ListInsert 160
ListLoad 161
ListRasEntries 162
ListSave 161
ListSet 158
ListSetKey 159
ListSetTag 159
ListSetText 160
ListSort 160
localhostaddr 176
localhostname 176
lookuphostaddr 177
lookuphostname 177
loop 133
lowercase 146

M
MemAlloc 172
MemForget 173
MemFree 173
MemGetInt 173
MemGetStr 173
MemSetInt 173
MemSetStr 173
MemSize 173
MemVarPtr 173
MsgBox 163

O
ord 144

P
paramcount 175
paramstr 176
PopupBox 164
pos 145
print 162

Q
quit 139

R
random 175
RasDial 172
RasErrText 171
RasGetConnection 171

RasGetIP 172
RasHangup 172
RasIsConnected 171
RasLastError 171
RE_Extract 148
RE_Match 148
RE_Parse 148
RE_Split 149
repeat ... until 134
replace 147
return 139
RFCTimeZone 150
runscript 140

S
ScriptPriority 175
set 142
SetHostsEntry_ByAddr 176
SetHostsEntry_ByName 176
SetTaskLimiter 176
SetXCounter 177
sgn 144
Shell 174
sleep 140
StopThread 176
str 144
sub ... endsub 136
SysErrorMessage 151

T
TestMailfilterLine 204
ticks 149
time 149
timegmt 149
trace 140
trim 146
true 143

U
until 134
uppercase 146
UTF7toUCS16 147
UTF8toUCS32 147

V
var 123
varset 142

W
warning 163
while ... endwhile 134

X
XCounter 177

Anweisungen der OLE-Schnittstelle

C

ClearXCounter 235
ControlAddGroup 230
ControlAddLog 222
ControlAddPull 231
ControlChangePassword ... 222
ControlDelGroup 231
ControlDelPull 231
ControlFlush 223
ControlGetGroupsPath 218
ControlGetHscPath 218
ControlGetHsmPath 218
ControlGetInfo 218
ControlGetLogsPath 218
ControlGetMailsOutPath ... 218
ControlGetMailsPath 218
ControlGetNewsErrPath ... 218
ControlGetNewsOutPath ... 218
ControlGetPath 218
ControlGetProcessIdentifier 234
ControlGetServerPath 218
ControlGetStatus 220
ControlGetVersion 218
ControlIsIdle 222
ControlMessage 219
ControlNewMail 226
ControlNewsJobsAdd 229
ControlNewsJobsDelete 230
ControlNewsJobsGet-
Counter 228
ControlNewsJobsGetParam . 228
ControlNewsJobsGet-
Priority 229
ControlNewsJobsGetServer . 228
ControlNewsJobsGetType .. 229
ControlNewsJobsSet-
Priority 229
ControlRotateLog 235
ControlRunFetchMail 224
ControlRunFetchMailTLS .. 224
ControlRunMail 223
ControlRunNewsPost 227
ControlRunNewsPull 227
ControlRunPurge 223
ControlRunPurgeGroup 223
ControlRunRebuildGlobal-
Lists 223
ControlRunRebuildHistory . 223
ControlRunScript 222
ControlRunSendMail 224
ControlRunSendMailAuth .. 224
ControlRunSendMailAuth-
TLS 226
ControlRunSendMailTLS ... 225
ControlSetLogin 218
ControlThreadCount 222
ControlWaitIdle 222

D

DecodeBase64 235
DecodeMIMEHeaderString . 235
DecodeQP 235
DecXCounter 235

H

hs2AsyncCommand 235

I

IncXCounter 235

N

NewsArtCount 232
NewsArtNoMax 233
NewsArtNoMin 232
NewsArtText 233
NewsArtTextExport 233
NewsArtXOver 233
NewsDeleteByMID 234
NewsGrpClose 231
NewsGrpCount 231
NewsGrpIndex 231
NewsGrpInformation 232
NewsGrpName 231
NewsGrpNameByHandle .. 232
NewsGrpOpen 231
NewsGrpXOver 232
NewsImport 234
NewsJobsClear 227
NewsJobsFeed 228
NewsJobsPost 228
NewsJobsPostDef 227
NewsJobsPull 227
NewsJobsPullDef 227
NewsJobsStart 228
NewsLocateMID 234
NewsLocateMID2 234
NewsScoreListFor 233
NewsScoreTest 233

R

RasDial 230
RasDynGetConnection 230
RasGetIP 230
RasHangup 230
RasIsConnected 230
RasLastError 230
RasListEntries 230

S

SetXCounter 235

X

XCounter 234