

////Doc/english/umsrhc

COLLABORATORS

| | | | |
|---------------|--|-------------------|------------------|
| | <i>TITLE :</i> ////Doc/english/umsrhc | | |
| <i>ACTION</i> | <i>NAME</i> | <i>DATE</i> | <i>SIGNATURE</i> |
| WRITTEN BY | | February 12, 2023 | |

REVISION HISTORY

| NUMBER | DATE | DESCRIPTION | NAME |
|--------|------|-------------|------|
| | | | |

Contents

| | | |
|----------|--------------------------------|----------|
| 1 | ////Doc/english/umsrhc | 1 |
| 1.1 | ////Doc/english/umsrhc.guide | 1 |
| 1.2 | umsrhc.guide/Copyright | 2 |
| 1.3 | umsrhc.guide/Introduction | 3 |
| 1.4 | umsrhc.guide/UMS | 4 |
| 1.5 | umsrhc.guide/Message Format | 4 |
| 1.6 | umsrhc.guide/Addresses | 5 |
| 1.7 | umsrhc.guide/Attributes | 5 |
| 1.8 | umsrhc.guide/Conversion | 6 |
| 1.9 | umsrhc.guide/Message Base | 7 |
| 1.10 | umsrhc.guide/Users | 8 |
| 1.11 | umsrhc.guide/Modes | 9 |
| 1.12 | umsrhc.guide/RFC | 9 |
| 1.13 | umsrhc.guide/Message Formats | 10 |
| 1.14 | umsrhc.guide/MIME | 11 |
| 1.15 | umsrhc.guide/NNTP | 12 |
| 1.16 | umsrhc.guide/POP3 | 12 |
| 1.17 | umsrhc.guide/SMTP | 13 |
| 1.18 | umsrhc.guide/TCP-IP | 13 |
| 1.19 | umsrhc.guide/UUCP | 14 |
| 1.20 | umsrhc.guide/Requirements | 15 |
| 1.21 | umsrhc.guide/Directories | 16 |
| 1.22 | umsrhc.guide/Configuration | 16 |
| 1.23 | umsrhc.guide/Environment | 17 |
| 1.24 | umsrhc.guide/UMS Configuration | 17 |
| 1.25 | umsrhc.guide/rfc.domainname | 19 |
| 1.26 | umsrhc.guide/rfc.pathname | 19 |
| 1.27 | umsrhc.guide/rfc.username | 20 |
| 1.28 | umsrhc.guide/rfc.dstime | 20 |
| 1.29 | umsrhc.guide/rfc.export. | 21 |

| | | |
|------|-------------------------------|----|
| 1.30 | umsrhc.guide/rfc.import | 21 |
| 1.31 | umsrhc.guide/rfc.mailencoding | 22 |
| 1.32 | umsrhc.guide/rfc.newsencoding | 22 |
| 1.33 | umsrhc.guide/rfc.receivedname | 22 |
| 1.34 | umsrhc.guide/nntpd.access | 23 |
| 1.35 | umsrhc.guide/nntpd.log | 24 |
| 1.36 | umsrhc.guide/nntpget.groups | 24 |
| 1.37 | umsrhc.guide/smtpd.log | 25 |
| 1.38 | umsrhc.guide/uucp.mailexport | 25 |
| 1.39 | umsrhc.guide/uucp.newsexport | 26 |
| 1.40 | umsrhc.guide/uucp.nodename | 28 |
| 1.41 | umsrhc.guide/uucp.debugfile | 28 |
| 1.42 | umsrhc.guide/uucp.debuglevel | 28 |
| 1.43 | umsrhc.guide/uucp.dumbhost | 29 |
| 1.44 | umsrhc.guide/uucp.filtercr | 29 |
| 1.45 | umsrhc.guide/uucp.keepdupes | 30 |
| 1.46 | umsrhc.guide/uucp.logdupes | 30 |
| 1.47 | umsrhc.guide/uucp.mailroute | 30 |
| 1.48 | umsrhc.guide/uucp.recipients | 32 |
| 1.49 | umsrhc.guide/Commands | 32 |
| 1.50 | umsrhc.guide/umsnntp | 34 |
| 1.51 | umsrhc.guide/umsnntpd | 35 |
| 1.52 | umsrhc.guide/umsnntpget | 36 |
| 1.53 | umsrhc.guide/setnntpdate | 37 |
| 1.54 | umsrhc.guide/umspop3 | 38 |
| 1.55 | umsrhc.guide/umspop3d | 39 |
| 1.56 | umsrhc.guide/umssmtp | 39 |
| 1.57 | umsrhc.guide/umssmtpd | 40 |
| 1.58 | umsrhc.guide/ums2uucp | 41 |
| 1.59 | umsrhc.guide/uuxqt | 42 |
| 1.60 | umsrhc.guide/UMS options | 43 |
| 1.61 | umsrhc.guide/Protocol options | 43 |
| 1.62 | umsrhc.guide/Examples | 44 |
| 1.63 | umsrhc.guide/User | 45 |
| 1.64 | umsrhc.guide/Leaf Node | 45 |
| 1.65 | umsrhc.guide/Full Node | 46 |
| 1.66 | umsrhc.guide/Gateway | 47 |
| 1.67 | umsrhc.guide/Access | 48 |
| 1.68 | umsrhc.guide/Inetd | 49 |

| | | |
|------|--|----|
| 1.69 | umsrhc.guide/Services | 49 |
| 1.70 | umsrhc.guide/UUCPDirectories | 50 |
| 1.71 | umsrhc.guide/UUCPConfig | 50 |
| 1.72 | umsrhc.guide/L.Ports | 51 |
| 1.73 | umsrhc.guide/L.Sys | 51 |
| 1.74 | umsrhc.guide/UUCPPoll | 51 |
| 1.75 | umsrhc.guide/Telser | 52 |
| 1.76 | umsrhc.guide/Hosts | 52 |
| 1.77 | umsrhc.guide/Answers | 52 |
| 1.78 | umsrhc.guide/History | 53 |
| 1.79 | umsrhc.guide/TODO | 54 |
| 1.80 | umsrhc.guide/Authors address | 55 |
| 1.81 | umsrhc.guide/Credits | 55 |
| 1.82 | umsrhc.guide/Index | 55 |

Chapter 1

////Doc/english/umsrfc

1.1 ////Doc/english/umsrfc.guide

UMS RFC Documentation

Welcome to UMS RFC, a package to handle mail and news messages which conform to the

 RFC
 standards with
 UMS
 .

If you have used the UMS UUCP package you should reread the complete documentation. Almost everthing has changed!

Copyright

 Copyright and other legal stuff

Introduction

 A short tour through UMS, RFC and UUCP

Requirements

 Requirements for using UMS RFC

Directories

 Description of all directories in this distribution

Configuration

 How to configure UMS RFC

Commands

 Description of all UMS RFC commands

Examples

 Some example UMS RFC configurations

Answers
 Solutions to common UMS RFC problems

History
 History of UMS RFC

TODO
 Not yet implemented features

Authors address
 Where to send bug reports & comments

Credits
 The author wants to thank

Index
 Index for this document

1.2 umsrhc.guide/Copyright

Copyright and other legal stuff

COPYRIGHT
 =====

Copyright (C) 1992-95 Stefan Becker

Permission is granted to make and distribute verbatim copies of this manual provided the copyright notice and this permission notice are preserved on all copies.

No program, document, data file or source code from this software package, neither in whole nor in part, may be included or used in other software packages unless it is authorized by a written permission from the author.

NO WARRANTY
 =====

There is no warranty for this software package. Although the author has tried to prevent errors, he can't guarantee that the software package described in this document is 100% reliable. You are therefore using this material at your own risk. The author cannot be made responsible for any damage which is caused by using this software package.

DISTRIBUTION
 =====

This software package is freely distributable. It may be put on any media which is used for the distribution of free software, like Public Domain disk collections, CDROMs, FTP servers or bulletin board systems.

In order to ensure the integrity of this software package, distributors should use the original archive file umsrhc0_11.lha. The author cannot be made responsible if this software package has become unusable due to modifications of the archive contents or of the archive file itself.

There is no limit on the costs of the distribution, e.g. for the media, like floppy disks, streamer tapes or compact disks, or the process of duplicating. Such limits have been proven to be harmful to the idea of freely distributable software, e.g. instead of reducing the price of the floppy disk below the limit, the software was simply removed from the master disk.

Although the author does not impose any limit on the distribution of this software package, he would like to express his personal opinions on this matter:

- * This software package should be made available to everyone free of charge whenever it is possible.
- * If you have acquired this software package under normal conditions from a Public Domain dealer on a floppy disk at a price higher than 5DM or US \$5, then you have definitely paid too much. Please don't support this improper profit making any longer and switch to a cheaper source as soon as possible.

USAGE RESTRICTIONS
=====

No program, document, data file or source code from this software package, neither in whole nor in part, may be used on any machine which is used

- * for the research, development, construction, testing or production of weapons or other military applications. This also includes any machine which is used in the education for any of the above mentioned purposes.
- * by people who accept, support or use violence against other people, e.g. citizens from foreign countries.

1.3 umsrhc.guide/Introduction

A short tour through UMS, RFC and UUCP

This chapter tries to give you a short overview about all UMS RFC related topics.

UMS

Universal Message System

Importers or Exporters. In UMS RFC the conversion is handled by the umsrfc.library which is used by all programs.

Advanced information:

- Addresses
 - UMS address formats
- Attributes
 - Additional message information
- Conversion
 - UMS <-> RFC conversion

1.6 umsrfc.guide/Addresses

UMS address formats

.....

RFC

addresses are mapped 1:1 to UMS addresses. Additionally UMS RFC currently understands the following two UMS address formats. It can convert them to and from

RFC

addresses:

<zone>:<hub>/<node>.<point>@fidonet

FIDO (FTN) address format

<box>.maus

Maus address format

1.7 umsrfc.guide/Attributes

Additional message information

.....

UMS RFC supports several parameters which can be specified in the UMS message field Attributes:

ALIAS

Use this alias for the address generation instead of rfc.username

.

UMS RFC checks if this alias is valid for the creator of the

message.

RECEIPT-REQUEST

Send a receipt message to the specified address. This creates the

RFC

header Return-Receipt-To. The address may be specified in one of the following four different forms:

- "" - Creates address for sender of message
- <name> - Creates address for local user
- <account@domain> - Any address may be specified
- <name>,<account@domain> - Any address may be specified

URGENT

Flag this message as urgent. This creates the following

RFC

header:

Priority: urgent

1.8 umsrhc.guide/Conversion

UMS <-> RFC conversion

.....

UMS itself is not a

RFC

system, which means that all messages must be transformed from the RFC format to the UMS format and vice versa. Not every piece of RFC information is used or stored by UMS, so the importer must preserve the original information. All RFC header fields which are not known to the importer are preserved.

There are two exceptions to this rule:

- The order of the RFC header fields is not always preserved. Luckily, the RFC standards do not impose a strict order for most of the header fields.
- If the message is a MIME message and the importer can decode the information, the exporter will not reconstruct the original message text. Instead it will create new MIME headers which specify the new format.

The exporters use the additional information which is stored by the importers to reconstruct the original messages. Here are some examples where preserving of the information is needed:

From
Reply-To

These RFC fields contain the address and maybe the real name of the user. Since this information is stored in two separate UMS fields these lines must be parsed to extract the needed information. This parsing process is not reversible.

Newsgroups

Crosspostings are stored as hard-linked messages, which appear in different groups. But an importer for a system may not have write access to all groups in the crossposting, so exporters can't reconstruct this line by reading all hard-linked messages.

References

UMS only uses one message ID for message threading.

Mapping of RFC headers to UMS fields:

| RFC header | UMS fields | Preserved |
|--------------|---------------------------------------|-----------|
| Control | - (cancel commands will be processed) | Yes |
| Date | Date & CDate | |
| Distribution | Distribution | |
| Followup-To | ReplyGroup | |
| From | FromAddr & FromName (MIME decoded) | Yes |
| In-Reply-To | ReferID | Yes |
| Message-ID | MessageID | |
| Newsgroups | Group (X messages hard linked) | Yes |
| Organization | Organization (MIME decoded) | |
| References | ReferID (the last id in the header) | Yes |
| Reply-To | ReplyAddr & ReplyName (MIME decoded) | Yes |
| Subject | Subject (MIME decoded) | |
| X-Mailer | Newsreader | |
| X-NewsReader | Newsreader | |
| <all others> | Comment | Yes |

All control messages will be written to a special newsgroup called rfc.control. You have to make sure that this newsgroup is included in the WRITEACCESS pattern of the importer user entries. You can allow the distribution of control messages by adding this newsgroup to the READACCESS pattern of a exporter user entry.

Additional UMS fields used for creation of RFC headers during export:

| UMS field | RFC header |
|--------------------|--|
| Attributes | (miscellaneous) |
| LogicalToAddr/Name | To (mail only, supersedes ToAddr) |
| RfcAttr | - (contents directly transferred into RFC message) |
| ToAddr/Name | To (mail only, also used as physical address) |

1.9 umsrfc.guide/Message Base

 UMS message base

All UMS messages are stored in a message base. This is a special database controlled by the umsserver. Knowledge about exact format of this database is not needed by the users because they don't access it directly. They have to go through the standard Interface ums.library.

Multiple message bases can be active at the same time, each controlled by its own umsserver. Even a message base on a remote machine can be accessed via network. All the user has to do is to specify the name of the message base:

<name>

Use the specified message base on the local machine.

<name>@<machine>

Use the specified message base on a remote machine using Envoy as network software.

<machine>:<service>

<machine>:<port>

Access a message base on a remote machine using TCP-IP as network software. The message base will be specified with the configuration on the remote machine.

1.10 umsrfc.guide/Users

Different types of users

UMS differentiates three types of users:

Users

They can read and write messages, but they don't have any special rights.

System Operators

Users with special rights, also called SysOps. They may access the headers of messages from other users to perform special tasks. Note that they can't read the message text itself, only the headers!

Exporters

Users with special read and write rights. They get read rights to all messages which should be exported. They may also import messages from remote systems.

UMS RFC needs one user with the name postmaster. It will send all error messages as mail to this user. Normally you should add the alias postmaster to a SysOp user entry.

1.11 umsrhc.guide/Modes

UMS operation modes

UMS can be operated in three different modes for news distribution:

Leaf node

The last system in a news message distribution chain. It imports news messages from one or more remote systems, but doesn't export those news messages to other systems. Only local news messages are exported to the remote systems. Most users should use this mode, because it's very easy to maintain.

To make UMS behave like a leaf node use

```
( NodeMode "%")
```

Full node

This node is located inside a news message distribution chain. It imports news messages from remote systems and exports them to other remote systems. The maintainer of such a system must control the message flow between the systems very carefully in order to prevent loops and duplicate messages. Only experienced users should use this mode.

With the UMS configuration variable `NodeMode` you can specify which newsgroups should be handled by UMS in full node mode, e.g.

```
( NodeMode "comp.#?" )
```

All messages in newsgroups whose names match this pattern may be read by other exporters. The actual message flow between the systems is controlled with the variable `READACCESS` in the exporter user entries.

Gateway

Such a node behaves like a full node, but additionally it can distribute news messages between networks with different technologies and newsgroup names. Such systems are very delicate to handle, can cause many headaches and create many enemies and flame wars. Only VERY VERY experienced users should use this mode. You have been warned!

To operate UMS as gateway you have to have a special key, the `GateKey`.

This scheme is only applied to news messages, because they can generate a very high amount of traffic in the networks. Mail messages are always freely distributed between all systems.

1.12 umsrhc.guide/RFC

Request for comments

=====

RFC are standards which define the Internet technology. Each RFC has a unique number by which it can be identified, e.g. RFC 822. Currently there are over 1700 RFC available, although some of the older ones have been obsoleted by newer ones.

The following RFC define the standards for mail and news messages and their transportation across networks.

Format standards:

Message Formats

The standard RFC message formats

MIME

Multi-Purpose Message Extensions

Service standards:

NNTP

Network News Transfer Protocol

POP3

Post Office Protocol (Version 3)

SMTP

Simple Mail Transfer Protocol

Other standards:

TCP-IP

Internet protocol stack

1.13 umsrhc.guide/Message Formats

The standard RFC message formats

Two standards define the format of messages on the Internet:

RFC 822 (1982)

Standard for the format of ARPA Internet text messages

Defines the format of messages on the Internet. It is therefore the most important standard for a RFC compliant messages handling system. A message consists of two parts: the header and the body. They are separated by an empty line.

The header consists of non-empty lines with a keyword, followed by a colon and the parameters. The standard defines many keywords, their semantics and the syntax of their parameters. Especially it specifies the format of time and address parameters. An example message header looks like this:

```
From: "Joe Dumb User" <jduser@test.foo.bar>
To: "John Doe" <jd@dummy.do.main>
Subject: Test message
Date: Fri, 26 May 1995 13:25:07 +0100
Message-ID: <54208434@test.foo.bar>
```

The standard doesn't define any format for the body of a message. The only restriction is that only 7 Bit ASCII characters may be used. This restriction may be circumvented with

MIME

.

RFC 1036 (1987)

Standard for Interchange of USENET Messages

This is the extension of RFC 822 for news messages. It defines some new keywords, like Newsgroups and Path. Also some of the many allowed variants in RFC 822 are narrowed down to a usable set of options.

1.14 umsrc/guide/MIME

Multi-Purpose Message Extensions

RFC 822 concentrates on the format of the message headers and doesn't say much about the format of the message body. The body may therefore only be used for plain text. Additionally it allows only 7 Bit ASCII as charset for messages.

These restrictions, especially when the biggest hype of the 90's called Multimedia hit the message handling systems, led to development of several extensions to RFC 822 called MIME:

RFC 1521 (1993)

MIME Part I: Mechanisms for Specifying and Describing
the Format of Internet Message Bodies

This defines a structure for the message body. The body may contain Non-ASCII text and binary data. The encoding of the message body for transportation over non-transparent links is specified. Messages may be composed of several different types of data. A mechanism for splitting up large messages into several smaller ones is specified.

Additionally it defines some new RFC header fields which identify a MIME message and specify the contents and the encoding of the message body.

RFC 1522 (1993)

MIME Part II: Message Header Extensions for Non-ASCII Text

Specifies the encoding of Non-ASCII text in RFC message headers.

This allows you to use such texts in subjects.

UMS RFC currently only supports the decoding of message headers and bodies on import and the encoding of message bodies on export. So you may use all ISO-8859-1 characters in the message body, like ä, á and ß.

1.15 umsrhc.guide/NNTP

Network News Transfer Protocol

USENET, the biggest news distribution system of the world, originally used

UUCP

as transportation mechanism. But with the advent of the Internet with its online connections a new mechanism was needed. This led to the development of the NNTP:

RFC 977 (1986)

Network News Transfer Protocol

This standard specifies a synchronous protocol between two news exchanging entities, a client and a server. The client plays the active role and sends commands, like POST or GROUP, to the server. It then waits for a response of the server which contains status or error codes or the news message. Only after that the client may send the next command.

A special protocol is defined if the client is also a news server. The client announces which new messages are available from its system and the server can request or reject them. This lowers the network traffic because messages are only transferred once even if there are multiple redundant distribution paths available for a message.

1.16 umsrhc.guide/POP3

Post Office Protocol (Version 3)

On systems where user mailboxes are not available network-wide because of a missing network file system or security reasons the mail messages are usually received by one host. This central server plays the role of a post office which handles the mailboxes for its users. The server can be accessed using POP3.

RFC 1081 (1988)

Post Office Protocol - Version 3

Defines the synchronous protocol between a server and a client.

Note that the client can only retrieve messages. For sending mail messages he has to use another protocol like

SMTP

.

First the client has to send the user name and the password to gain access to the mailbox. The client may then retrieve messages from the mailbox. It may also mark messages for deletion, but the server will physically delete the messages only after the client has send the command to close the connection.

1.17 umsrhc.guide/SMTP

Simple Mail Transfer Protocol

One of the first applications for the Internet was the exchange of electronic mail. Several protocols for transferring mail messages over networks have been defined, but the most commonly used one is

SMTP

.

RFC 821 (1982)

Simple Mail Transfer Protocol

Defines a synchronous protocol between two mail exchanging

entities. The calling party plays the active role which can be

changed during the conversation, but this option seldomly used.

For each mail the active part first transmits the senders address

and then a list with the recipient addresses. These addresses are

checked by the receiving side. After that the mail message is

transferred.

1.18 umsrhc.guide/TCP-IP

Internet protocol stack

TCP/IP is the most commonly used abbreviation for protocol suite used on the Internet. Although there are several other protocols in use the most important are IP and TCP.

IP - Internet Protocol

Defines an unreliable, connectionless datagram service which is

the base for all Internet protocols and services. IP datagrams

contain a destination address which enables the interconnecting systems to route a datagram from the sender to the recipient. The addresses, the so called IP numbers, are 32 Bit wide. The arrival and the order of the datagrams is not guaranteed with IP.

IP can be used on many different types of network hardware. Examples are Ethernet, FDDI, serial lines (SLIP, CSLIP or PPP) or X.25.

TCP - Transmission Control Protocol

Defines a reliable, connection oriented stream protocol. From the point of the user the connection looks like a FIFO, that is the bytes arrive in the same order as he sends them. TCP splits the stream into datagrams and reassembles them at the receiving side. It also makes sure that the datagrams are placed correctly when they arrive out of order. Lost or errorneous datagrams are requested again from the sender.

1.19 umsrhc.guide/UUCP

Unix to Unix CoPy

=====

UUCP was originally written to transfer files to a remote UNIX machine. The syntax of the main command uucp is the same as the UNIX cp command hence the name. Each file transfer request creates a job which is stored in the so called spool directory for the remote system.

At certain times the local machine checks if there are any files waiting to be transferred to the remote system. It then starts the command uucico (Unix to Unix Copy In Copy Out) which automatically contacts the remote system via a modem line or a

TCP-IP
connection.

The remote system then starts its own uucico and both programs process their jobs and send the files forth and back.

Even remote command execution is possible with UUCP. To achieve this a job with three files is created. Two of them are data files with the prefix D.. One of these contains the data and the other the commands how to process the data file on the remote system. The third file with the prefix C. contains the commands for uucico to send both files to the remote system.

The command file is renamed to a file with the prefix X. on the remote system. After the connection has been closed uucico starts the command uuxqt (Unix to Unix eXeCuTion?). It looks into the spool directory for files with the prefix X. and processes the commands.

This mechanism can be also used to transfer mail and news messages (The news distribution system USENET was originally based on UUCP). The data file contains a message in the

RFC 822
format and the command file

contains mail and news processing commands, like rmail.

Only mail messages are transferred in this way nowadays, because small files degrade the performance of the file transfer. This can be avoided if the data file contains several messages (batching). News messages can simply be concatenated into one file, because they already contain the message length in the header. For mail messages an adaption of the

SMTP
format is used, called Batched SMTP (BSMTP).

To reduce the amount of data which needs to be transferred batched files are usually compressed. Currently in use are three compression methods: compress, freeze and gzip. Normally only the plain output of those programs is stored in the data file, but compressed news batches usually contain a small text header which identifies the compression method.

1.20 umsrfc.guide/Requirements

Requirements for using UMS RFC

This version of UMS RFC requires the following things:

- * AmigaOS 2.04 (V37) or better.
- *
 - UMS
 - V11 or better.
- * AmiTCP 3.0 or better if you are using
 - TCP-IP
 - .
- * A service provider for Mail and News via
 - UUCP
 - or
 - TCP-IP
 - (
 - NNTP
 - ,
 - POP3
 - or
 - SMTP
 -).

1.21 umsrhc.guide/Directories

Description of all directories in this distribution

UMS RFC uses the standard
 UMS
 directory structure, so you can easily
 unpack the distribution into your UMS directory.

UMS/Bin
 UMS RFC programs and some utility programs. Add this directory
 to your command path.

UMS/Devs
 Contains the telser.device.

UMS/Docs/english
 Documentation (ASCII and AmigaGuide) for UMS RFC in english.

UMS/Libs
 umsrhc.library and some utility libraries. Copy these to your
 LIBS: directory or add this directory to your LIBS: path with

 Assign LIBS: UMS/Libs ADD

UMS/Networks/RFC/convert
 Scripts to import existing mail and news messages from other
 UUCP
 packages into an
 UMS message base
 .

UMS/Networks/RFC/db
 Example configuration files for telser. Copy these to the directory
 AMITCP:db

UMS/Networks/RFC/lib
 Example configuration files for
 UUCP
 . Copy these to the directory
 UULIB:

UMS/Networks/RFC/src
 This directory contains the History file with the complete story
 about the development of UMS RFC and its predecessors.

1.22 umsrhc.guide/Configuration

How to configure UMS RFC

UMS RFC offers many options and this chapter explains the configuration. Please read it carefully because a proper configuration is essential for the correct operation of UMS RFC.

```
Environment
    AmigaDOS environment variables

UMS Configuration
    UMS configuration variables
```

1.23 umsrhc.guide/Environment

AmigaDOS environment variables

=====

Only one environment variable is currently used by UMS RFC:

```
UMSUUCP.mb
    This variable tells the command
        uuxqt
        which
        UMS message base
        it
    should use to import
        UUCP
        batches.
```

1.24 umsrhc.guide/UMS Configuration

UMS configuration variables

=====

Most of the options are configured using the UMS variables which are explained in this section.

Mandantory variables:

```
rfc.domainname
    Domain name for your system

rfc.pathname
    Path name for your system
```

rfc.username
Account name for a local UMS RFC user

Optional variables (may be omitted):

rfc.dstime
Daylight Saving Time on/off

rfc.export.
Address conversions during message export

rfc.import.
Address conversions during message import

rfc.mailencoding
Encoding of mail message texts during export

rfc.newsencoding
Encoding of news message texts during export

rfc.receivedname
Name of your system for Received: lines

nntpd.access
Access rights for remote NNTP clients

nntpd.log
NNTP server log on/off

nntpget.groups
Group list for news message import

smtpd.log
SMTP server log on/off

Mandantory UUCP variables:

uucp.mailexport
Configuration for mail batches

uucp.newsexport
Configuration for news batches

uucp.nodename
UUCP node name for your system

Optional UUCP variables (may be omitted):

uucp.debugfile
Name of the file for debugging output

uucp.debuglevel
Debug level

```

uucp.dumbhost
    Smart or dump UUCP host

uucp.filtercr
    Filter CR characters

uucp.keepdups
    Keep dupe messages

uucp.logdups
    Log dupe messages

uucp.mailroute
    Mail routing configuration

uucp.recipients
    Limit number of recipients
    
```

1.25 umsrhc.guide/rfc.domainname

Domain name for your system

```

rfc.domainname
    The fully qualified domain name (FQDN) for your system. The FQDN
    consists of the hostname plus the name of your domain. This
    variable is mandantory for using UMS RFC.
    
```

Example:

```

( rfc.domainname "foobar.earth.sol.galaxy" )
    
```

1.26 umsrhc.guide/rfc.pathname

Path name for your system

```

rfc.pathname
    The name of your system in the "Path:" line for news messages. If
    your system is a registered UUCP node, then this variable only
    needs to contain the hostname of your system. Otherwise it must
    contain the FQDN of your system. This variable is mandantory for
    using UMS RFC. See also
        rfc.domainname
        .
    
```

Example for normal nodes:


```
( rfc.domainname "foobar.earth.sol.galaxy" )
( rfc.pathname   "foobar.earth.sol.galaxy" )
```

Example for registered UUCP nodes:

```
( rfc.domainname "foobar.earth.sol.galaxy" )
( rfc.pathname   "foobar" )
```

1.27 umsrhc.guide/rfc.username

Account name for a local UMS RFC user

rfc.username

This is the account name for a user. This variable is mandatory for every UMS user whose messages are exported with UMS RFC. The user entry also must contain an alias for this name.

Example:

```
( rfc.domainname "foobar.earth.sol.galaxy" )

( User
  ( Name "Joe Dumb User" )
  ( Alias "jduser" )
  ...
  ( rfc.username "jduser" )
)
```

The above example will result in the following Internet address for the user:

```
jduser@foobar.earth.sol.galaxy
```

1.28 umsrhc.guide/rfc.dstime

Daylight Saving Time on/off

rfc.dstime

This switch enables the Daylight Saving Time. The default is N.

Example:

```
( rfc.dstime "Y" )
```

1.29 umsrhc.guide/rfc.export.

Address conversions during message export

rfc.export.*

This set of variables specifies the domain names used in the address conversion for exporting messages from non RFC compliant nets. Currently defined are the conversions for Fido (FTN) and Maus, using the following variables:

```
rfc.export.fido (Default: .fidonet.org)
rfc.export.maus (Default: .maus.de)
```

Example:

```
( rfc.export.fido ".fido.de" )
( rfc.export.maus ".maus.sub.org" )
```

The above example will result in the following address conversion:

```
Joe User,1:2/3.4@fidonet -> Joe_User@p4.f3.n2.z1.fido.de
Joe User,HB.maus         -> Joe_User@HB.maus.sub.org
```

1.30 umsrhc.guide/rfc.import.

Address conversions during message import

rfc.import.*

This set of variables specifies the domain names used in the address conversion for importing messages arriving from a RFC compliant net, but which originate in a non RFC compliant net. You may specify several domain names for each type of net, separated by commas. Currently defined are the conversions for Fido (FTN) and Maus, using the following variables:

```
rfc.import.fido (Default: .fidonet.org)
rfc.import.maus (Default: .maus.de)
```

Example:

```
( rfc.import.fido ".fido.de,.fidonet.org" )
( rfc.import.maus ".maus.sub.org" )
```

The above example will result in the following address conversion:

```
Joe_User@p4.f3.n2.z1.fidonet.org -> Joe User,1:2/3.4@fidonet
Joe_User@p4.f3.n2.z1.fido.de     -> Joe User,1:2/3.4@fidonet
Joe_User@HB.maus.sub.org         -> Joe User,HB.maus
```

1.31 umsrfc.guide/rfc.mailencoding

Encoding of mail message texts during export

rfc.mailencoding

When exporting mail messages which contain non-ASCII characters for transportation over non-transparent (that is 8 bit clean) links, UMS RFC can encode them according to the

MIME

standard. The

following values are allowed for this variable:

- 0 Don't encode (Default)
- 1 Encode as quoted-printable

Example:

(rfc.mailencoding 1)

1.32 umsrfc.guide/rfc.newsencoding

Encoding of news message texts during export

rfc.newsencoding

When exporting news messages which contain non-ASCII characters for transportation over non-transparent (that is 8 bit clean) links, UMS RFC can encode them according to the

MIME

standard. The

following values are allowed for this variable:

- 0 Don't encode (Default)
- 1 Encode as quoted-printable

Example:

(rfc.newsencoding 1)

1.33 umsrfc.guide/rfc.receivedname

Name of your system for Received: lines

rfc.receivedname

With this variable you can set the name of your system for the Received: lines. If you don't set this variable then the contents

of
 rfc.domainname
 will be used. This variable is needed for
 systems with dynamic domain names.

Example:

```
( rfc.domainname "pop3host.domain" )
( rfc.receivedname "host.sub.domain" )
```

This will create the following lines in the
 RFC
 header:

```
...
Received: by host.sub.domain ...
...
From: user@pop3host.domain
...
```

1.34 umsrfc.guide/nntpd.access

Access rights for remote NNTP clients

nntpd.access

When a remote client tries to connect the
 NNTP
 service the
 umsnntpd
 server daemon reads the contents of this variable to find out
 which access rights the client has. The variable may contain
 multiple lines of the following format:

```
<Name pattern> <UMS user name>,<Post Y/N>,<Server Y/N>
```

The check is based on the domain name of the machine on which the
 client runs. The patterns from the first line down to the last
 line will be used to match the domain name. When a pattern matches
 then the access rights of this entry are used for the client. If
 no pattern matches the access is denied with an error message.

The account of the specified UMS user name is used to export and
 import the news messages. If the second parameter is set to Y then
 posting (that is importing) of news messages with the POST command
 is allowed. If the third parameter is also set to Y then the
 client is a

```
    NNTP  

    server and is allowed to use the IHAVE command  

    for transferring news messages.
```

Example:

```
( nntpd.access
  "localhost          NNTP1,Y,Y\n"
  "*.ping.de         NNTP2,Y,N\n"
  "*"                 NNTP3,N,N\n" )
```

The above example allows full access for a client running on the local machine. Clients running on machines in the domain ping.de are allowed to post news messages. All other machines are only allowed to read news messages. Using three different UMS users you can control which newsgroups are visible for each of the three categories using the READACCESS variable.

1.35 umsrhc.guide/nntpd.log

NNTP server log on/off

nntpd.log

This variable enables the command logging of the NNTP server to the UMS log file. The default is N.

Example:

```
( nntpd.log "Y" )
```

1.36 umsrhc.guide/nntpget.groups

Group list for news message import

nntpget.groups

The

NNTP

importer

umsnntpget

looks into this variable to find out

which newsgroups it should request from the remote server. It may contain multiple lines. If you are using the option -g each line must contain one newsgroup name. If you don't use this option then each line may contain multiple group name patterns separated by commas and even negations.

Example (use with option -g):

```
( nntpget.groups "group.a\n"
                  "group.b\n"
                  "group.d\n"
                  "de.test.a\n"
                  "foo.bar.a\n"
```

```
"foo.bar.b\n" )
```

Example (use without option -g):

```
( nntpget.groups "group.*,!group.c,de.test.a\n"
  "foo.bar.*\n" )
```

1.37 umsrhc.guide/smtpd.log

SMTP server log on/off

smtpd.log

This variable enables the command logging of the SMTP server to the UMS log file. The default is N.

Example:

```
( smtpd.log "Y" )
```

1.38 umsrhc.guide/uucp.mailexport

Configuration for mail batches

uucp.mailexport

This variable tells the UUCP exporter ums2uucp in which format mail messages should be exported. The syntax is as follows:

```
<command>[,<grade>[,<batch>[,<compress command>[,<header>[,<max size <->]]]]]
```

command

Command name for the C line in the UUCP command file. You have to talk to the administrator of your host which commands his software accepts and how it interprets them. The following table shows some common names for mail batches:

| Name | Meaning |
|---------|-------------------------------------|
| rmail | unbatched, uncompressed mail |
| rbsmtp | batched, uncompressed mail |
| rcbsmtp | batched, compressed (compress) mail |

```
rbcsmtp | batched, compressed (compress) mail
rfbsmtp | batched, compressed (freeze) mail
rbfsmtp | batched, compressed (freeze) mail
rgbsmtp | batched, compressed (gzip) mail
rbgsmtp | batched, compressed (gzip) mail
```

grade

Every UUCP job has a grade which is one letter out of 0..9, A..Z or a..z. It specifies the transfer priority for the job. 0 has the highest priority and z the lowest. The default is A.

batch

This optional parameter specifies if several mail message should be merged into one file (batching).

compress command

Name of a command for the optional compression of the data files. Legal values are compress, freeze and gzip. If you leave this field empty the data files won't be compressed.

header

If this field is not empty, then the following header line will be created in the data file:

```
#! <header text>
```

Mail batches normally don't contain any headers, so you may leave this field empty.

max size

Maximum size for one batch file. When it reaches this size it will be closed and a new batch file will be used. The default size is 65536 bytes.

Example for unbatched, uncompressed mail:

```
( uucp.mailexport "rmail" )
```

Example for batched, uncompressed mail, no header, batch size 65536 Bytes:

```
( uucp.mailexport "rbsmtp,A,Y" )
```

Example for batched, compressed mail, no header, batch size 300000 Bytes:

```
( uucp.mailexport "rcbsmtp,0,Y,compress,,300000" )
```

1.39 umsrfc.guide/uucp.newsexport

Configuration for news batches

uucp.newsexport

This variable tells the UUCP exporter ums2uucp in which format

news messages should be exported. The syntax is as follows:

```
<command>[,<grade>[,<batch>[,<compress command>[,<header>[,<max size <->]]]]]
```

command

Command name for the C line in the UUCP command file. You have to talk to the administrator of your host which commands his software accepts and how it interprets them. Normally you should set this to rnews.

grade

Every UUCP job has a grade which is one letter out of 0..9, A..Z or a..z. It specifies the transfer priority for the job. 0 has the highest priority and z the lowest. The default is A.

batch

This optional parameter specifies if several news message should be merged into one file (batching). Usually non-batched news transfer is a bad idea.

compress command

Name of a command for the optional compression of the data files. Legal values are compress, freeze and gzip. If you leave this field empty the data files won't be compressed.

header

If this field is not empty, then the following header line will be created in the data file:

```
#! <header text>
```

The following table shows some common header texts:

| Name | Meaning |
|----------|-------------------------------------|
| cunbatch | batched, compressed (compress) news |
| funbatch | batched, compressed (freeze) news |
| gunbatch | batched, compressed (gzip) news |

You should talk to the administrator of your host which headers and compression types are supported by the software of the host system.

max size

Maximum size for one batch file. When it reaches this size it will be closed and a new batch file will be used. The default size is 65536 bytes.

Example for batched, uncompressed news, no header, batch size 300000 Bytes:

```
( uucp.newsexport "rnews,A,Y,,,300000" )
```

Example for batched, compressed news, with header (! cunbatch), batch size 65536 Bytes:

```
( uucp.newsexport "rnews,B,Y,compress,cunbatch" )
```

1.40 umsrcf.guide/uucp.nodename

UUCP node name for your system

uucp.nodename

UUCP node name for your system. This name should not be longer than 7 characters.

Example:

```
( uucp.nodename "foobar" )
```

1.41 umsrcf.guide/uucp.debugfile

Name of the file for debugging output

uucp.debugfile

Output file for error logs. Useful only for debugging.

Example:

```
( uucp.logfile "t:uucp.log" )
```

1.42 umsrcf.guide/uucp.debuglevel

Debug level

uucp.debuglevel

Set the highest log level. Messages with a higher level are not logged. Useful only for debugging.

Example:

```
( uucp.loglevel 5 )
```

1.43 umsrhc.guide/uucp.dumbhost

Smart or dump UUCP host

uucp.dumbhost

Set this to Y when your host requires RFC976 style UUCP mail

envelopes. Default is to generate normal envelopes. With a given address and date, a normal envelope looks like this:

```
From <user>@<domain> <date>
```

A RFC976 envelope looks like this:

```
From <user> <date> remote from <domain>
```

Ask the administrator of your host what envelope type is expected by the software used on the host system.

Example:

```
( uucp.dumbhost "Y" )
```

1.44 umsrhc.guide/uucp.filtercr

Filter CR characters

uucp.filtercr

If this variable is set to Y then incoming UUCP

data files are stripped of CR characters before processing them.

Example:

```
( uucp.filtercr "Y" )
```

1.45 umsrhc.guide/uucp.keepdupes

Keep dupe messages

uucp.keepdupes

Normally duplicate messages are flagged as error and the UUCP job is not deleted. Sometimes dupes can't be avoided, because UUCP is

an offline protocol, that means there is no way to detect dupes on the sending hosts. On a system with several hosts and high traffic this can result in a high number of bad job files lying around, which do not contain real errors. If you set this variable to N then bad jobs, which had only dupe errors, are automatically deleted.

Example:

```
( uucp.keepdupes "N" )
```

1.46 umsrhc.guide/uucp.logdupes

Log dupe messages

uucp.logdupes

Normally duplicate messages are also logged in the UMS RFC error log. If you don't want to bother with dupes then you should set this to N. See also uucp.keepdupes

Example:

```
( uucp.logdupes "N" )
```

1.47 umsrhc.guide/uucp.mailroute

Mail routing configuration

uucp.mailroute

Sometimes it's important to specify static routes for outgoing mails. Normally this variable is empty so NO routing information will be generated. Each line in this variable consists of an address pattern and a list of hosts:

<address pattern> [<host1>[,<host2>....]]

When the recipient address matches the address pattern and the host list is not empty, then the

UUCP
exporter
ums2uucp
will

create an explicit mail route. The route information will be added AFTER the address conversion for non RFC compliant nets has been applied.

If the variable does contain several lines, then the patterns will be tried from the first line down to the last line until one pattern matches. If no pattern matches, then no route information will be generated.

Example:

```
( uucp.mailroute "#?.maus.de      host1\n"
    "#?.zer.sub.org    host2,host3\n"
    "#?.fidonet.org\n"
    "#?.de             host4\n" )
```

The following test cases:

```
user@box.maus.de
user@box.zer.sub.org
user@f7.n242.z2.fidonet.org
user@zauber.nase.de
user@dummy.blubb.edu
```

are translated to the following mail routes for non-batched mail transfer:

```
C rmail host1!box.maus.de!user
C rmail host2!host3!box.zer.sub.org!user
C rmail user@f7.n242.z2.fidonet.org
C rmail host4!zauber.nase.de!user
C rmail user@dummy.blubb.edu
```

and to the following mail routes for batched mail transfer:

```
RCPT TO: <@host1:user@box.maus.de>
RCPT TO: <@host2,@host3:user@box.zer.sub.org>
RCPT TO: <user@f7.n242.z2.fidonet.org>
RCPT TO: <@host4:user@zauber.nase.de>
```

RCPT TO: <user@dummy.blubb.edu>

1.48 umsrhc.guide/uucp.recipients

Limit number of recipients

uucp.recipients

Limits the number of recipients for one mail file. If there are more recipients than the UUCP exporter ums2uucp will create several mail files for one mail. The default is to put all recipients in one mail file.

Example:

```
( uucp.recipients 2 )
```

Using the above example on a mail with three recipients will result in two separate mails to be send. If unbatched mail transfer is used then the following two C lines are generated in the command files:

```
C rmail usera@test1.foo.bar,userb@test2.foo.bar
C rmail userc@test3.foo.bar
```

If batched mail transfer is used then the following commands are generated in the batch file:

```
MAIL FROM: <user@test.foo.bar>
RCPT TO: <usera@test1.foo.bar>
RCPT TO: <userb@test2.foo.bar>
DATA
...
MAIL FROM: <user@test.foo.bar>
RCPT TO: <userc@test3.foo.bar>
DATA
```

1.49 umsrhc.guide/Commands

Description of all UMS RFC commands

This section contains the description of all UMS RFC commands and their command line options. The following syntax will be used:

- a Mandatory command line option. You may not omit this option when using a command.
- b <...> Command line option with parameter. The space between the option and the parameter is optional.
- [-c <...>] Optional command line option. You may omit this option when using a command. The default value will be used instead.

The following UMS RFC commands are available:

Commands for NNTP:

- umsnntp NNTP exporter
- umsnntpd NNTP importer (server daemon)
- umsnntpget NNTP importer
- setnntpdate Tool to set date of the last NNTP access

Commands for POP3:

- umspop3 POP3 importer
- umspop3d POP3 exporter (server daemon)

Commands for SMTP:

- umssmtp SMTP exporter
- umssmtpd SMTP importer (server daemon)

Commands for UUCP:

- ums2uucp UUCP exporter
- uuxqt UUCP importer

Common command line options:

UMS options
 Common UMS related command line options

Protocol options
 Common protocol related command line options

1.50 umsrfc.guide/umsnntp

NNTP exporter

=====

The command umsnntp exports news messages via
 NNTP
 to a remote host.

Syntax:

```
umsnntp  -h <remote host>
          [-S <Service name or port number>]

          [-p <UMS password>]
          [-s <UMS message base>]
          [-u <UMS user name>]

          [-b <select bit>]
          [-i]

          [-c]
          [-r <n>]
          [-w <n>]
```

Defaults:

```
-S  Service nntp (port 119)
-p  No password
-s  Default message base
-u  User NNTP
-b  No select bit
-r  360
-w  60
```

Explanation of the command line options:

- b
 Bit number of an additional select bit. Only messages with this bit set in the user flags will be exported. This can be used to run a special filter program before exporting with umsnntp.
- i
 Use the IHAVE command to send the news messages to the remote host. If you don't specify this option, then the POST command is used. Most

NNTP

servers don't allow the IHAVE command for clients, so you may not be able to use this option.

-c

Export news messages continuously. After exporting all new messages umsnntp will wait a short while and then will start to export all new messages which arrived in the meantime.

-r

Causes umsnntp to rescan the whole message base for new messages after n export runs. All messages which couldn't be sent until now will be sent again. This option is only useful in conjunction with -c.

-w

Causes umsnntp to wait n seconds between two export runs. This option is only useful in conjunction with -c.

See also

UMS options
and
Protocol options
.

1.51 umsrfc.guide/umsnntpd

NNTP importer (server daemon)

=====

The command umsnntpd is a server daemon which offers a transfer service for news messages via

NNTP

to remote clients. It can only be

executed by the Internet Super Daemon

Inetd

.

Syntax:

```
umsnntpd [<UMS user name>]
          [<UMS password>]
          [<UMS message base>]
```

Defaults:

User NNTPD
No password
Default message base

1.52 umsrfc.guide/umsnntpget

NNTP importer

=====

The command umsnntp imports news messages via
 NNTP
 from a remote
 host.

Syntax:

```
umsnntpget -h <remote host>
           [-S <Service name or port number>]

           [-p <UMS password>]
           [-s <UMS message base>]
           [-u <UMS user name>]

           [-g]
           [-P <n>]
           [-c]
           [-q]
```

Defaults:

```
-S Service nntp (port 119)
-p No password
-s Default message base
-u User NNTP
-P 1
```

Explanation of the command line options:

- g
 Use the commands GROUP, STAT and NEXT to scan for new news messages. This may lower the load on the remote NNTP server but the transfer overhead may be significantly higher, especially if the server uses long expiration periods for news messages. If you don't specify this option the command NEWNEWS is used which only transfers information about new messages.
- P
 Use n (up to 20) parallel processes to transfer news messages. This is useful to make the transfer of news messages over a slow serial link more efficient by using most of the available bandwidth.
- c
 Use parameters on the command line after the last valid command line option as input parameters. You may request news messages by message ID:

```
umsnntpget -h nntphost -c 1234@dummy.do.main 6789@test.foo.bar
```

or by group names:

```
umsnntpget -h nntpghost -c -g comp.sys.amiga.misc comp.sys.amiga.datacomm
```

-q

Don't print message ids.

See also

```
    UMS options
    and
    Protocol options
    .
```

1.53 umsrfc.guide/setnntpdate

Tool to set date of the last NNTP access

=====

The command setnntpdate sets the date of the last NNTP access for

```
umsnntpget
.
```

Syntax:

```
setnntpdate <UMS user name>
            <UMS password>
            [<UMS message base>]
            [DATE <date string>]
```

Defaults:

Default message base

Explanation of the command line options:

DATE

You have to specify the date as defined in RFC 822. setnntpdate will transform it to the format which is needed for

```
umsnntpget
.
```

If you don't specify this option the last access date will be printed in RFC 822 format.

Example:

```
setnntpdate NNTP secret DATE "18 Nov 1994 20:05:30"
```

1.54 umsrfc.guide/umspop3

POP3 importer

=====

The command umspop3 imports mail messages via
POP3
from a remote
host.

Syntax:

```
umspop3 -h <remote host>
        [-S <Service name or port number>]

        [-p <UMS password>]
        [-s <UMS message base>]
        [-u <UMS user name>]

        -n <mailbox name>
        -i <mailbox password>
        [-k]

        [-c]
        [-w <n>]
```

Defaults:

```
-S Service pop3 (port 110)
-p No password
-s Default message base
-u User POP3
-w 60
```

Explanation of the command line options:

- n
Name of the mailbox on the remote host.
- i
Password for the mailbox on the remote host.
- k
If this option is used umspop3 won't delete the messages in the mailbox on the remote host. Note that these messages will show up as dupes next time you use umspop3!
- c
Import mail messages continuously. After importing all new messages umspop3 will wait a short while and then will start to import all new messages which arrived in the meantime.
- w
Causes umspop3 to wait n seconds between two export runs. This option is only useful in conjunction with -c.

See also

UMS options
and
Protocol options
.

1.55 umsrc.guide/umspop3d

POP3 exporter (server daemon)

=====

The command umspop3d is a server daemon which offers a transfer service for mail messages via

POP3
to remote clients. It can only be
executed by the Internet Super Daemon
Inetd
.

Syntax:

```
umspop3d [<UMS user name>]
          [<UMS password>]
          [<UMS message base>]
```

Defaults:

```
User POP3D
No password
Default message base
```

1.56 umsrc.guide/umssmtp

SMTP exporter

=====

The command umssmtp exports mail messages via
SMTP
to a remote host.

Syntax:

```
umssmtp -h <remote host>
        [-S <Service name or port number>]

        [-p <UMS password>]
        [-s <UMS message base>]
        [-u <UMS user name>]
```

[-b <select bit>]

[-c]

[-r <n>]

[-w <n>]

Defaults:

-S Service smtp (port 25)
 -p No password
 -s Default message base
 -u User SMTP
 -b No select bit
 -r 360
 -w 60

Explanation of the command line options:

- b
 Bit number of an additional select bit. Only messages with this bit set in the user flags will be exported. This can be used to run a special filter program before exporting with umssmtp.
- c
 Export mail messages continuously. After exporting all new messages umssmtp will wait a short time and then will start to export all new messages which arrived in the meantime.
- r
 Causes umssmtp to rescan the whole message base for new messages after n export runs. All messages which couldn't be sent until now will be sent again. This option is only useful in conjunction with -c.
- w
 Causes umssmtp to wait n seconds between two export runs. This option is only useful in conjunction with -c.

See also

UMS options
 and
 Protocol options
 .

1.57 umsrfc.guide/umssmtpd

SMTP importer (server daemon)

=====

The command umssmtpd is a server daemon which offers a transfer service for mail messages via

SMTP

to remote clients. It can only be
 executed by the Internet Super Daemon
 Inetd
 .

Syntax:

```
umssmtpd [<UMS user name>]
          [<UMS password>]
          [<UMS message base>]
```

Defaults:

```
User SMTPD
No password
Default message base
```

1.58 umsrfc.guide/ums2uucp

UUCP exporter

=====

The command ums2uucp exports mail and news messages and creates
 batch files for the transfer via
 UUCP
 to a remote host.

Syntax:

```
ums2uucp -h <remote host>

          [-p <UMS password>]
          [-s <UMS message base>]
          [-u <UMS user name>]

          [-b <select bit>]
          [-d <n>]
```

Defaults:

```
-p No password
-s Default message base
-u User UUCP
-b No select bit
-d 0
```

Explanation of the command line options:

- h
UUCP node name of the remote host.
- b
Bit number of an additional select bit. Only messages with this

bit set in the user flags will be exported. This can be used to run a special filter program before exporting with ums2uucp.

-d
UUCP log level: 0, 1, 2 or 3.

See also
 UUCP options
 .

1.59 umsrfc.guide/uuxqt

UUCP importer

=====

The command uuxqt imports mail and news messages from batch files which were transferred via

UUCP
 from a remote host. This command is normally only executed by the UUCP transfer program uucico.

Syntax:

```
uuxqt [-p <UMS password>]
      [-d <n>]
      [<subdirectory>]
```

Defaults:

```
-p No password
-d 0
```

Explanation of the command line options:

-p
Password for the UMS user. This will be used for ALL logins.

-d
UUCP log level: 0, 1, 2 or 3.

subdirectory
Name of the subdirectory in the UUCP spool directory. This parameter is supplied by uucico when a hierarchical spool directory structure is used.

uuxqt uses the following system for the UMS user names
:

uucp.default Used as the first login to read the configuration.
 uucp.<name> Used when importing a batch from node name.

1.60 umsrhc.guide/UMS options

Common UMS related command line options

=====

The following UMS related comand line options are understood by most of the programs:

- p Password for the UMS user .
- s Name of the UMS message base used for importing or exporting.
- u Name of the UMS user .

1.61 umsrhc.guide/Protocol options

Common protocol related command line options

=====

The following protocol related comand line options are understood by most of the programs:

- h Name or IP number of the remote host.
- S Name or port number of the TCP-IP service. All services have default port numbers so this option should normally not be used.

Hosts
Host descriptions

1.63 umsrfc.guide/User

Configuration for an UMS RFC user
=====

Excerpt from the file ums.config:

```
( User
  ( Alias
    jduser
    postmaster
  )
  ( Readaccess "#?" )
  ( Writeaccess "#?" )
  ( Netaccess "#?" )
  ( Import "%" )
  ( Name "Joe Dumb User" )
  ( Password "top secret" )
  ( rfc.username "jduser" )
)
```

1.64 umsrfc.guide/Leaf Node

Configuration for a simple node
=====

Excerpt from the file ums.config:

```
( Aka "#?@ttest.(foo.bar|uucp)" )
( NodeMode "%" )

( Exporter
  ( Alias
    uucp.default
    NNTP
    POP3
    SMTP
    UUCP
  )
  ( Readaccess "#?" )
  ( Writeaccess "#?" )
  ( Netaccess "#?" )
  ( Import "#?" )
  ( Export "#?" )
  ( Distribution "#?" )
)
```

```
( Name uucp.dummy )
( Password )
( rfc.domainname "test.foo.bar" )
( rfc.mailencoding 0 )
( rfc.newsendcoding 0 )
( rfc.pathname "test.foo.bar" )
( uucp.dumbhost "Y" )
( uucp.keepdupes "Y" )
( uucp.logdupes "Y" )
( uucp.mailexport "rmail,N" )
( uucp.newsexport "rnews,Y,compress,cunbatch,300000" )
( uucp.nodename "test" )
)
```

This system receives all messages from the node dummy. It will send out unbatched mail messages and batched, compressed news messages to the node dummy. The transmission link to the remote node is transparent, so no encoding is needed. The users on this system will have the address

```
<user>@test.foo.bar
```

To poll the node dummy with
 UUCP
 the user has to use the following

commands:

```
ums2uucp -h dummy
uucico -sdummy
```

To send out mail and news message via
 TCP-IP
 and to fetch mail

messages the user has to use the following commands:

```
umssmtp -h dummy.foo.bar
umsnntp -h dummy.foo.bar
umspop3 -h dummy.foo.bar -n jduser -i more_top_secret
```

1.65 umsrfc.guide/Full Node

Configuration for a full node

=====

Excerpt from the file ums.config:

```
( Aka "#?@test.(foo.bar|uucp)" )
( NodeMode "(comp|sci).#?" )

( rfc.domainname "test.foo.bar" )
( rfc.pathname "test.foo.bar" )

( Exporter
```

```

( Alias
  uucp.default
  UUCP
)
( Readaccess "(comp|sci).#?" )
( Writeaccess "(comp|sci).#?" )
( Netaccess "#?" )
( Import "#?" )
( Export "#?.foo.bar" )
( Distribution "#?" )
( Name uucp.dummy )
( Password )
( rfc.mailencoding 1 )
( rfc.newsencoding 1 )
( uucp.dumbhost "Y" )
( uucp.keepdups "N" )
( uucp.logdups "Y" )
( uucp.mailexport "rsmtp,Y" )
( uucp.newsexport "rnews,Y,gzip,gunbatch" )
( uucp.nodename "test" )
)

( Exporter
  ( Alias
    NNTPD
    POP3
    POP3D
    SMTP
    SMTPD
  )
  ( Readaccess "#?" )
  ( Writeaccess "#?" )
  ( Netaccess "#?" )
  ( Import "#?" )
  ( Export "~(#?.foo.bar)" )
  ( Distribution "#?" )
  ( Name NNTP )
  ( Password )
  ( rfc.mailencoding 0 )
  ( rfc.newsencoding 0 )
)

```

In this setup the newsgroup hierarchies comp and sci are distributed between two systems. Mail messages to the domain foo.bar are send out with

```

UUCP
, all others are send out with
SMTP
.

```

1.66 umsrhc.guide/Gateway

Configuration for a gateway

=====

Excerpt from the file ums.config:

```
( Netgroup
  "comp.sys.amiga.misc"
  "fidonet.comp.sys.amiga.misc"
)

( Exporter
  ( Alias
    uucp.default
    NNTP
    NNTPD
    POP3
    POP3D
    SMTP
    SMTPD
    UUCP
  )
  ( Readaccess "comp.#?" )
  ( Writeaccess "comp.#?" )
  ( Netaccess "#?" )
  ( Import "#?" )
  ( Export "~(#?@fidonet)" )
  ( Distribution "#?" )
  ( Name uucp.dummy )
  ( Password )
  ( rfc.domainname "test.foo.bar" )
  ( rfc.mailencoding 0 )
  ( rfc.newsencoding 0 )
  ( rfc.pathname "test.foo.bar" )
  ( uucp.dumbhost "Y" )
  ( uucp.keepdups "N" )
  ( uucp.logdups "Y" )
  ( uucp.mailexport "rbsmtp,Y" )
  ( uucp.newsexport "rnews,Y,gzip,gunbatch" )
  ( uucp.nodename "test" )
)
```

The example shows the UMS RFC part of a gateway configuration. One newsgroup is distributed between networks with different technologies (gating).

1.67 umsrhc.guide/Access

Security Configuration

=====

If you are using the commercial version of AmiTCP you may limit the access to your system. The file AMITCP:db/inet.access contains the rules which services may be used on your system. The following two lines allow the access to the services

```

        NNTP
        and
        SMTP
        from any host on the
Internet:

        smtp    *    allow    LOG
        nntp    *    allow    LOG
    
```

1.68 umsrhc.guide/inetd

Configuration for the Internet Super Daemon

=====

The daemons for the
 NNTP
 ,
 POP3
 and
 SMTP
 services are started
 automatically when a remote host accesses those services. Add the
 following three lines to the inetd configuration file
 AMITCP:db/inetd.conf:

```

        smtp    stream  tcp  nowait  root  UMS:Bin/umssmtpd  umssmtpd
        pop3    stream  tcp  nowait  root  UMS:Bin/umspop3d  umspop3d
        nntp    stream  tcp  nowait  root  UMS:Bin/umsnntp  umsnntp
    
```

1.69 umsrhc.guide/Services

Services configuration

=====

To use the services
 NNTP
 ,
 POP3
 and
 SMTP
 the file AMITCP:db/services
 must contain the following lines:

```

        smtp    25/tcp
        pop3    110/tcp
        nntp    119/tcp
    
```

1.70 umsrhc.guide/UUCPDirectories

UUCP directories

=====

UUCP
needs three directories for its work:

Spool directory

Directory for command, data and log files. You should assign UUSPOOL: to this directory. Each remote site has its own subdirectory. In each subdirectory there should be a directory bad-jobs for erroneous jobs. Example with two remote systems named dummy and test:

```
1> Dir UUSPOOL: all
    dummy (dir)
        bad-jobs (dir)
    test (dir)
        bad-jobs (dir)
logfile
XferStat
TimeLog
```

Configuration directory

UUCP configuration file reside here. You should assign UULIB: to this directory.

```
1> Dir UULIB:
    config
    L.Sys
    seq
L.Ports
security
```

Public directory

Download and upload area for UUCP. You should assign UUPUB: to this directory.

1.71 umsrhc.guide/UUCPConfig

Main UUCP configuration

=====

The file UULIB:config contains the UUCP configuration. A minimal configuration only needs your node name, a dummy user name and a setting for the debug level:

```
NodeName    foobar
UserName    root
Debug       0
```

1.72 umsrhc.guide/L.Ports

UUCP ports description

=====

The file UULIB:L.Ports describes the devices which uucico uses to contact remote systems. Normally this is a serial port with a modem, so the file should look like this:

```
acu=ser device=serial.device unit=0
```

If you want to use

UUCP

over

TCP-IP

you must add another device to

this file:

```
acu=tcp device=telser.device unit=0
```

1.73 umsrhc.guide/L.Sys

Remote UUCP Systems

=====

The file UULIB:L.Sys describes how uucico can contact remote systems. This includes the device to use, the serial line speed, the phone number and the login script:

```
dummy Any ser 19200 ATDP123456 ogin: foobar ssword: hello
```

To contact a remote system with

UUCP

over

TCP-IP

you replace the

phone number with the hostname of the remote system. So the line should look like this:

```
dummy Any tcp 19200 dummy ogin: foobar ssword: hello
```

1.74 umsrhc.guide/UUCPPoll

UUCP poll script
 =====

The following script first exports all messages to the remote system and then contacts it. You can specify the name of the remote system as parameter:

```
.KEY SYSTEM
.DEF SYSTEM dummy
.BRA {
.KET }

ums2uucp -h {SYSTEM}
uucico -s{SYSTEM}
```

1.75 umsrhc.guide/Telser

Telser configuration
 =====

The file AMITCP:db/telser.conf describes the configuration for every unit of the telser.device. For the unit 0 it should look like this:

```
0 "" NoOperation NOOPENWIN NOLINGER NODEBUG T:telser.log
```

1.76 umsrhc.guide/Hosts

Host descriptions
 =====

The file AMITCP:db/telser.hosts contains the descriptions of the remote systems. If you dial a remote system using the telser.device it looks into this file to find out the hostname and the parameters. For our example system it looks like this:

```
dummy dummy.do.main 540 "" 00000000001
```

Please make sure that the 11th bit is set, because it activates the transparent mode of telser.device which is a must for

```
UUCP
over
TCP-IP
.
```

1.77 umsrhc.guide/Answers

Solutions to common UMS RFC problems

Error: No rfc.username for...
 You have forgotten to add the variable
 rfc.username
 to the user
 entry of a lokal user.

Error: Can't write article, check WRITEACCESS!
 UMS RFC couldn't write the article to any newsgroup. You have
 requested a newsgroup but forgotten to add it to the WRITEACCESS
 patter of the importer entry. If you are receiving control
 messages you must also add the newsgroup rfc.control to this
 pattern!

Error: Can't get connection data!
 You forgot to update the service file of your AmiTCP
 installation. See
 Services
 .

You should also read the FAQ about
 UMS
 .

1.78 umsrfc.guide/History

History of UMS RFC

0.11 (09-Dec-1995)

- POP3
 server daemon
 umspop3d
 .
- Support for UMS' binary field.
- Implemented UUCP grades. The format of the variables
 uucp.mailexport
 and
 uucp.newsexport
 was changed!
- Implemented CTRL-C handling for
 umsnntpget
 .
- Command logging for

NNTP
and
SMTP
server daemons.

- Switch for Daylight Saving Time.
- Several bug fixes and new command switches.

0.10 (04-Jun-1995)

- First official release of the UMS RFC package the successor of the UMS UUCP package.
- All RFC from/to UMS conversions are handled by the umsrfc.library now.
- Support for NNTP, POP3, SMTP, UUCP.
- New documentation.

For all older versions please read the file History in the source directory.

1.79 umsrfc.guide/TODO

Not yet implemented features

The following things have not yet been implemented. The order shown below doesn't imply any priority for an item:

- * MX support for the
SMTP
exporter
umssmtp
.
- * Encoding/Decoding with base64 in
MIME
messages.
- *
MIME
encoding of messages headers.
- * The address conversion should be freely configurable. Does anyone have a good reference for a pattern search & replace algorithm (like in perl)?

1.80 umsrcf.guide/Authors address

Where to send bug reports & comments

The author can be reached at the following addresses:

Postal address:

Stefan Becker
 Verkehrsstrasse 11
 44809 Bochum
 GERMANY

Internet Electronic Mail:

stefanb@yello.ping.de

1.81 umsrcf.guide/Credits

The author wants to thank

I want to thank all users of the public beta version for testing this package. Especially I want to thank Matthias Scheler and Bernhard Möllemann who tested intermediate versions on their systems. Many thanks also to the members of the UMS-Dev mailing list for the discussions and the ideas.

Thanks go to Kai 'wusel' Siering for allowing me to include his version of uucico in this package. You may find the latest version of the program in his

UUCP

package called wUUCP on Aminet in the directory comm/uucp.

Thanks also to Sam Yee for allowing me to include parts of his telser package. You may find the latest version of his package on Aminet in the directory comm/tcp. telser is Shareware, so if you use it regularly PLEASE do register it!

1.82 umsrcf.guide/Index

Index

Account name

rfc.username

| | |
|-----------------------|-----------------|
| Address | Authors address |
| Attributes | Attributes |
| Bug reports | Authors address |
| Command Line Options | Commands |
| Command Syntax | Commands |
| Comments | Authors address |
| Common Problems | Answers |
| Configuration | Configuration |
| Conversion | Conversion |
| Copyright | Copyright |
| Credits | Credits |
| Daylight Saving Time | rfc.dstime |
| Directories | Directories |
| Distribution | Copyright |
| Domain Name | rfc.domainname |
| E-Mail | Authors address |
| Environment variables | Environment |
| Examples | Examples |
| Full Node | Modes |

Full Node Configuration
 Full Node

Gateway
 Modes

Gateway Configuration
 Gateway

History
 History

Host Descriptions
 Hosts

Inetd Configuration
 Inetd

Internet address
 Authors address

Internet protocols
 TCP-IP

Introduction
 Introduction

Leaf Node
 Modes

Leaf Node Configuration
 Leaf Node

Legal stuff
 Copyright

MIME
 MIME

Multi-Purpose Message Extensions
 MIME

Network News Transfer Protocol
 NNTP

NNTP
 NNTP

NNTP exporter
 umsnntp

NNTP importer
 umsnntpget

NNTP server daemon
 umsnntpd

| | |
|----------------------|-----------------|
| nntpd.access | nntpd.access |
| nntpd.log | nntpd.log |
| nntpget.groups | nntpget.groups |
| Path name | rfc.pathname |
| Permissions | Copyright |
| POP3 | POP3 |
| POP3 importer | umspop3 |
| POP3 server daemon | umspop3d |
| Post Office Protocol | POP3 |
| Postal address | Authors address |
| Prohibitions | Copyright |
| Protocol stack | TCP-IP |
| Questions & Answers | Answers |
| Remote UUCP Systems | L.Sys |
| Requirements | Requirements |
| RFC | RFC |
| RFC 1036 | Message Formats |
| RFC 1081 | POP3 |
| RFC 1521 | MIME |

| | |
|------------------------|------------------|
| RFC 1522 | MIME |
| RFC 821 | SMTP |
| RFC 822 | Message Formats |
| RFC 977 | NNTP |
| RFC Message Formats | Message Formats |
| rfc.control | Conversion |
| rfc.domainname | rfc.domainname |
| rfc.dstime | rfc.dstime |
| rfc.export.fido | rfc.export. |
| rfc.export.maus | rfc.export. |
| rfc.import.fido | rfc.import. |
| rfc.import.maus | rfc.import. |
| rfc.mailencoding | rfc.mailencoding |
| rfc.newsencoding | rfc.newsencoding |
| rfc.pathname | rfc.pathname |
| rfc.receivedname | rfc.receivedname |
| rfc.username | rfc.username |
| Security Configuration | Access |
| Services configuration | Services |

| | |
|--|----------------|
| setnntpdate | setnntpdate |
| Simple Mail Transfer Protocol SMTP | |
| SMTP | SMTP |
| SMTP exporter | umssmtp |
| SMTP server daemon | umssmtpd |
| smtpd.log | smtpd.log |
| TCP/IP | TCP-IP |
| Telser Configuration Telser | |
| TODO | TODO |
| UMS | UMS |
| UMS Address Formats Addresses | |
| UMS configuration variables UMS Configuration | |
| UMS Message Base | Message Base |
| UMS Message Format | Message Format |
| UMS Operation Modes Modes | |
| ums2uucp | ums2uucp |
| umsnntp | umsnntp |
| umsnntpd | umsnntpd |
| umsnntpget | umsnntpget |

| | |
|------------------------|-----------------|
| umspop3 | umspop3 |
| umspop3d | umspop3d |
| umssmtp | umssmtp |
| umssmtpd | umssmtpd |
| Unix to Unix CoPy | UUCP |
| User Configuration | User |
| Users | Users |
| UUCP | UUCP |
| UUCP Configuration | UUCPConfig |
| UUCP Directories | UUCPDirectories |
| UUCP exporter | ums2uucp |
| UUCP importer | uuxqt |
| UUCP poll script | UUCPPoll |
| UUCP ports description | L.Ports |
| uucp.debugfile | uucp.debugfile |
| uucp.debuglevel | uucp.debuglevel |
| uucp.dumbhost | uucp.dumbhost |
| uucp.filtercr | uucp.filtercr |
| uucp.keepdups | uucp.keepdups |

| | |
|-----------------|-----------------|
| uucp.logdupes | uucp.logdupes |
| uucp.mailexport | uucp.mailexport |
| uucp.mailroute | uucp.mailroute |
| uucp.newsexport | uucp.newsexport |
| uucp.nodename | uucp.nodename |
| uucp.recipients | uucp.recipients |
| uuxqt | uuxqt |
