UUPC/extended Command Reference

Overview

This section describes the syntax of the commands supplied with UUPC/extended. It assumes you have installed the programs and configured them as described in **Installation**, page 15, and that you have access to the Nutshell Handbook *Using UUCP and Usenet*.

Command Syntax

To obtain the syntax of the various command line options, enter the command name followed by '-?'; for example:

COMM34 -?
EXPIRE -?
FMT-?
GENSIG -?
MAIL -?
UUCICO -?
UUPOLL-?
UUCP -?
UUNAME -?
UUSTAT -?
UUSUB -?
UUX -?
UUXQT -?

Command Descriptions

Descriptions of the commands supplied as part of UUPC/extended follow in alphabetical order.

COMM34

Synopsis

COMM34 port3 [port4]

Availability

MS-DOS only (possible in OS/2 DOS box, see below)

Description

COMM34 defines non-default addresses of communications ports 3 and 4 (COM3 and COM4) under MS-DOS. It does this by poking the provided operands into the BIOS memory segment at hexadecimal address 0040:0000 where UUCICO and other programs such as Kermit search for port addresses. It also displays all defined port addresses for COM1-4.

> **Note:** This command is not available under OS/2, but may be used in an OS/2 DOS box. The preferred way to define the port addresses for COM3 and COM4 to both OS/2 and to DOS boxes under OS/2 is to add information to the relevant DEVICE= statements in the OS/2 CONFIG.SYS.

The syntax of the COMM34 command is:

COMM34 port3 [port4]

where:

port3 is the hexadecimal address for COM3, or 0 if omitted and port4 is specified.

> **Note:** COM3 must use IRQ 4.

port4 is the hexadecimal address for communications port 4.

> **Note:** COM4 must use IRQ 3.

COMM34 must be run before UUCICO, and only needs to be run once, perhaps by putting the appropriate line in your AUTOEXEC.BAT file. If all operands are omitted, COMM34 displays a help screen and the current port assignments.

EXPIRE

Synopsis

expire [-x *debuglevel*] [-e *days*] [-a *days*] [-n *newsgroup*]

Availability

MS-DOS, Windows 3.x, OS/2, and Windows NT

Description

EXPIRE archives and/or deletes articles created by RNEWS in normal mode after a specified period of days. For EXPIRE to run, the ACTIVE file must exist. (See RNEWS in **RNEWS**, page 52, for the layout of the ACTIVE file.) When received, articles are kept in the directory hierarchy defined by the **NewsDir** variable. After the period specified by the expire period, EXPIRE moves articles to the directory hierarchy defined by the **ArchiveDir** variable for the number of days specified by the archive period, after which EXPIRE deletes the articles.

> **Note:** Because the articles are moved from the main news directories to the archive directories via a rename call (the articles are not copied), the two directory hierarchies must reside on the same disk partition. If an article cannot be archived for this or any other reason, it will be deleted when the expire period is reached.

The syntax of EXPIRE is:

expire [-x *debuglevel*] [-e *days*] [-a *days*] [-n *newsgroup*]

Where:

-a *days*	Number of days news is to be archived (saved but inaccessible to the user) after it is expired. The default is zero (0) days, that is the news is be deleted without archiving.
-e *days*	Number of days news is be kept in a directories accessible to the user before being deleted or archived depending on the archive (-a) flag. The default period is seven (7) days.
-n *newsgroup*	The name of the newsgroup to scan for expired articles. The default is to scan all news groups in the active file for articles to expire.
-x *debug*	Display debug messages at or below level *debug*. The default value is 1, but the option may be set to 0 for unattended production use, or as high 20 for detailed debugging.

FMT

Synopsis

FMT [-#] [*input-file* [*output-file*]]

Availability

MS-DOS, OS/2, and Windows NT

Description

FMT is used as a simple paragraph formatter when entering mail in line mode. It effectively removes all carriage returns from within a paragraph and then writes the paragraph out in lines as close as possible to 72 characters in length without going over. Paragraphs are separated by blank lines in the input file and output files.

Note: If the first word on a new line is longer than the maximum line length, it is written by itself on the line. It is not truncated.

The command syntax is:

FMT [-#] [*input-file* [*output-file*]]

The optional flag -#, where "#" is a number greater than or equal to 0, overrides the default maximum length of 72 characters per line. The default input and output files are the console.

To format all the text you have typed in from line mode, use it as a pipe at the question mark (?) prompt while sending mail:

~|FMT

GENSIG

Synopsis

GENSIG *fixed.inp variable.inp merged.out*

Availability

MS-DOS, OS/2, and Windows NT

Description

GENSIG reads a standard signature file and appends random text selected from a second file, writing the combined data to a third file. The format of the command is:

GENSIG *fixed.inp variable.inp merged.out*

Where *fixed.inp* is the fixed portion of the signature file containing your name and address, and *variable.inp* is a file which begins which a delimiter line followed by quotes or other text separated by additional delimiter lines. For example, the variable input file might look like this:

```
**
The above is a delimiter line.
**
Free the Intel 386!
**
"UUPC/extended" is "system crash" spelled sideways.
**
Don't quote me!
```

The file *merged.out* will contain the entire text of the *fixed.inp* file followed by one delimited text block from *variable.inp*. If you use this to generate a signature file, then your [*userid*].RC should reference the file *merged.out* as your signature file.

Note: To generate fresh quotes, this program should be run from your AUTOEXEC.BAT or other regularly run batch file.

MAIL

Command line interface for read and sending mail

Synopsis

mail [-x *debug*] [-p] [-t] [-f *name*]
mail [-x *debug*] [-p] [-t] [-u *userid*]
mail [-x *debug*] [-s *subject*] *address* ... [-c] *address* ... [-b] *address*

Availability

MS-DOS, Windows 3.x, OS/2, and Windows NT

Description

The MAIL command is used to both compose and read mail. It provides a human interface to the user mailboxes, and formats mail for the RMAIL command to perform delivery to local and remote users.

If MAIL is invoked interactively with one or more addresses, it prompts for the subject of the message (if not provided via the -s flag), and then prompts for the text of the message either directly or via a user specified editor. (See **The UUPC/extended UUPC.RC and [userid].RC files**, page 75, to find out how to specify an editor. The program then prompts for the action to take (list, edit, input additional lines of text directly, abort, or send) until the message is sent or aborted. When sent, the required RFC-822 headers are added to the mail and the entire message is passed to RMAIL for delivery.

If MAIL is invoked with one or more addresses and the input is not the keyboard, then the input is taken without prompting or editing and passed to RMAIL with the appropriate RFC-822 headers.

If invoked without addresses to send mail to, MAIL reads the user's system mailbox (or other mailbox if specified on the command line), displays a one line summary of each message, and prompts for a user command with a item number and a question mark (?). Commands can be issued to read, save, delete, or send mail until the program is terminated by the quit or exit commands.

The following option is always accepted by MAIL:

-x *debug* — Amount of debugging information to display. The default is taken from the environment variable UUPCDEBUG if set and is otherwise 0. The higher the debug level, the more information overwhelms the unsuspecting user.

The following options are accepted when composing mail:

-s *subject* — Subject of the message to send. If multiple words, this must be enclosed in quotes. This option must precede any addresses and the -c and/or -b flags, if supplied.

-c *address* — One or more carbon copy addresses. Such addresses are listed under the CC: field of the RFC-822 header rather than in the To: field. One or more addresses must be listed after the -c flag.

-b *address* — One or more blind carbon copy addresses. Such addresses are listed under the Bcc: field of the RFC-822 header, which is *dropped* by RMAIL after processing. Thus, no one sees these addresses in their mail headers.

The following options are accepted when reading mail:

-f *filename* Name of the file to read rather than the system mailbox. See below for the syntax of accepted file names.

-u *userid* Name of the user whose system mailbox is to be read.

-t Rather than display who mail is from in the one summary, display who the mail is to. This is automatically enabled for the file defined by the UUPC.RC variable **FileSent=** (See **The UUPC/extended UUPC.RC and [userid].RC files**, page 75, for a further explanation.), which defines where a copy of mail you send is saved. If the **-t** flag is specified for the **FileSent=** file, it reverts to normal processing (displaying whom the mail is from).

-p Print all the mail in the specified mailbox and exit.

Types of operands

In the list of commands available when reading mail, *messages* refers to one or more messages in the mail box. These items can be specified in one of the following ways:

Message Number	A single numeric value, the special character dot (.) to specify the current item, or the special character dollar sign ($) to specify the last item in the mailbox.
Message range	Two message numbers separated by a hyphen (-). The messages must be in ascending order; for example, 1-$ is valid, but $-1 is not valid.
List of messages	A list of message numbers and/or message ranges, separated by spaces. The list need not be in ascending order. An example is: 1-5 8 7 14-17.
Asterisk (*)	Selects all messages in the mailbox
Subject	A slash (/) followed by a subject to search for, optionally followed by a terminating slash. Each Subject: or Resent-Subject: will be searched for the specified string. An example is /UUPC bugs/
User id	A single word (no white space) appearing in the From or Resent-From lines of the mail header. Note that if the command accepts both an item and a userid, you cannot specify the item as a user id. **Note:** If the command accepts both an item and file name and you specify a user id for the item, you must explicitly provide the default filename unless the user id includes the characters at sign (@), percent sign (%), or exclamation point (!).

Some commands also allow file names as operands. File names may look like any of the following:

Type of file	Description	Example
Simple file name	An MS-DOS file name with no path name	snuffles.spb

Relative path name	An MS-DOS file name with one or more back slashes (\) which does not begin with a back slash or drive letter.	mail\snuffles.spb
Absolute path name	An MS-DOS file name with one or more back slashes (\) which begins with a back slash or drive letter.	d:\uupc\snuffles.spb
UNC name	Universal Naming Convention name. A file located on a network server.	\\kewgate\uupc\mail\snuffles.spb
Mailbox	A simple file name prefixed by the plus sign (+) character. This will cause the file be referenced as if it is the mailbox for the specified user. For example, +postmast refers to the mailbox for user postmast.	+snuffles
Relative to your home directory	A simple file name preceded by a tilde and slash (~/). The file is then referenced by the prepending the home directory listed in your [*userid*].RC to the simple file name.	~/oldmail/snuffles.spb
Relative to another's home directory.	A simple file name preceded by a tilde, the userid, and a slash (~userid/). The file is then referenced by the prepending the home directory listed in the PASSWD file for user "userid" to the simple file name.	~snuffles/oldmail.spb

When sending mail from the command line or via the **mail** or **forward** commands, the destination address must be entered. Acceptable addresses look like any of the following:

Address type	Description	Example
userid	A simple user id, for delivery on the local system.	snuffles
userid@*node*	A user id and node combination, for delivery to *userid* on system *node*.	snuffles@kew.com
node!*userid*	A user id and node combination, for delivery to *userid* on system *node*	smersh!kendra!snuffles
nickname	A user or list defined in the user's ALIASES.TXT file or local system aliases file.	pbear

If the flag **-c** is inserted before a user id, then mail to the following users is sent as Carbon Copy (Cc:) addresses. If the flag **-b** is inserted before a user id, then mail to the following users is sent as blind carbon copies. Their addresses do not appear in the mail header. The blind carbon copy flag must follow any all normal addresses and carbon copy addressees.

Example: mail -s "Test message" snuffles -c athena!kewms -b software@kew.com

Commands available when reading mail

Commands are entered in response to the mail question mark prompt (?). All commands must be separated from their operands, if any, by white space. Most commands can be abbreviated to a single character; the commands which cannot be so abbreviated and their shortest allowable abbreviation are **debug** (**deb**), **dquit** (**dq**), **previous** (**pre**), **set** (**se**), and **status** (**st**).

The following commands can be used within MAIL:

empty line	If the current message has not been read, then an empty line acts as a **print** command; otherwise, it acts as a **next** command.
! *command*	Executes *command* as an MS-DOS, OS/2, or Windows NT command. If *command* is omitted, runs an inferior command processor (which gives the user a new command prompt.)
?	Prints a summary of commands available.
+ *integer*	Alias for **next** command.
- *integer*	Alias for **up** command.
alias *a1 a2 a3*	Displays alias information loaded from user's ALIASES.TXT file for nicknames *a1 a2 a3.* If an alias is list of other aliases, the list is recursively exploded and displayed. This command has no default operand. At least one alias must be specified.
copy *items file*	Copies mail *items* into *file* with headers. Default *file* is the printer on device **PRN**.
debug *integer*	Sets internal trace level to *integer.* The default is 0 when **mail** starts, and gets set to 1 if you just type "debug" without entering an *integer.* The internal trace level can also be set by the command line flag **-x.**
delete *messages*	Sets status of *messages* to deleted. Deleted messages are ignored when selecting mail by subject or user id, and are purged from the mailbox when the **quit** command is issued. See also **undelete**.
dquit *messages*	Short for **delete** followed by **quit.**
exit	Exits mail without updating mailbox; deleted items are left alone. Compare this to the **quit** command.
forward *messages address*	Resends *messages* to *address.* Note that if the **askcc** option is set, each *message* forwarded will be prompted for Carbon Copy addresses.
go *messages*	Positions to the last of the *messages* selected.
headers	Displays summary information for all items in the mailbox.
Headers *items*	Displays summary information for the selected *items* in the mailbox.

help	Prints this long help text.
mail -s *subject address* mail *address*	Interactively send mail to *address* with optional *subject.*
next *integer*	Move down in mailbox by *integer* items. Default is 1.
previous *integer*	Alias of **up** command.
print *messages*	Display *messages* from the mailbox, using the external pager if defined in the configuration file and ignoring (not printing) a standard list of RFC-822 fields in the mail header such as Received: and Message-Id:. Compare this to the **Print**, **type**, and **Type** commands.
Print *messages*	Display *messages* from mailbox, using internal pager and ignoring (not printing) a standard list of RFC-822 fields in the mail header.
quit	Terminates the reading of mail. All deleted messages are purged, and all other messages are saved in the original mailbox or in ~/mbox depending on the setting of the **save** flag.
reply *messages*	Interactively sends mail to the authors of each of the *messages* requested. Note that each item is replied to separately, specifying "reply 1-10" will send ten pieces of mail to ten people.
save *messages file*	Saves the specified *messages* complete with mail headers into *file*, and then deletes the *messages.* Compare this to the **copy** and **write** commands.
set	Sets various Boolean options. These option can also be set in your [*userid*].RC file. The options are listed in **The UUPC/extended UUPC.RC and [userid].RC files**, page 75.
status	Reports miscellaneous program information, including: The version and creation time of the program The operating system version The current user address and related information The current file name, size, and date last updated
type *messages*	Display *messages* from mailbox, using external pager if available and displaying RFC-822 fields suppressed by the **print** command. Compare this to the p**rint**, **Print**, and **Type** commands**.**
Type *messages*	Display *messages* from mailbox, using internal pager and displaying RFC-822 fields suppressed by the **print** and **Print** commands.
undelete *messages*	Changes status of *messages* to unread.
write *messages file*	Writes *messages* to *file* without the RFC-822 headers and then deletes them. The default *file* is **~/mbox**. Compare this to the **copy** and **save** commands.
xit	Alias for the **exit** command.

Subcommands available while sending mail:

~a	Insert standard signature file.
~A	Insert alternate signature file.
~e	Invoke editor on current message.
~m *item*	Include body of message(s) defined by *item*, indented.
~M *item*	Include message(s) define by *item* with headers, indented.
~f *item*	Include body of message(s) defined by *item*, unindented.
~F *item*	Include message(s) define by *item* with headers, unindented.
~p	Print message entered so far using external pager.
~P	Print message entered so far.
~r *file*	Read in an arbitrary file.
~s *subject*	Add new subject or replace existing subject with *subject*
~v	Alias for ~e command (for compatibility with UNIX sendmail).
~?	Display this mail subcommand list.
~|*cmd*	Filter message entered so far through *cmd*
~!*cmd*	Execute command (does not alter message)
~~	Enter a data line beginning with a tilde (~)

When entering mail at the ? prompt, use these commands by putting the ~ in column 1.

Note: The **~m** and **~M** commands use the same syntax to describe for items as the main mail parser; this allows specifying message number(s), user id, or subject. Type "help" at a MAIL prompt for a detailed description of the allowed syntax.

Note: The exact meaning of **~p** and **~P** commands can be inverted by the use of the **pager** option.

Files

/uupc/tilde.hlp	Help file for tilde (~) commands when sending mail.
/uupc/mail/*[userid]*.spb	System Mailbox for *[userid]*.

Bugs

Command line option checking is not as robust as it could be.

The automatic advancing of the current item pointer to an undeleted item makes multiple operations on a deleted item interesting.

Under MS-DOS, redirection from NUL looks like a device, not a file, and thus the program prompts for input when it has no business doing so.

The copy command does not properly copy files to the printer under OS/2.

MAIL incorrectly parses an return address with a quoted exclamation point, such as:

> "Smarter than the average bear!" <snuffles@kendra.kew.com>

Some editors, like EDLIN, cause the signature file to be lost because the editor appends a ^Z (Ctrl-Z) to the file.

The reply command gets confused by mixed mode addresses.

How MAIL determines return addresses is sometimes flaky.

The list of RFC-822 headers which are suppressed by the print command cannot be altered by the user; they are displayed in an all or nothing fashion.

There is no 'Replyall' command to reply to all interested parties to a message.

In MAIL, entering '1' should print the first item in the mailbox if the option 'autoprint' is set. Currently, it doesn't.

The list of ignored headers is not configurable.

The user cannot configure which header lines are used by MAIL in determining From or Replyto addresses.

See Also

RMAIL

History

This is the 20th revision of this document and was last updated on October 1, 1993 by Katherine Derbyshire.

1993UUPC/extended

NOVRSTRK

Synopsis

NOVRSTRK [input-file [output-file]]

Availability

MS-DOS, OS/2, and Windows NT

Description

NOVRSTRK is used to strip overstrikes from files to allow viewing on a terminal. If used to display the UUPC/extended documentation, it will drop the overstrikes created by back spaces which are used to create bold and underscored text on a printer. The syntax of NOVRSTRK is:

NOVRSTRK [input-file [output-file]]

The default input and output files are the console.

RMAIL

Synopsis

RMAIL [[-f | -F] *filename*] [-x *debug*] *address(es)*
RMAIL [[-f | -F] *filename*] [-x *debug*] [-s *subject*] -w *address(es)* [-c *address(es)*] [[-b *address(es)*]
RMAIL [[-f | -F] *filename*] [-x *debug*] -t

Availability

MS-DOS, Windows 3.x, OS/2, and Windows NT

Description

RMAIL is the Mail Delivery Agent (MDA) for UUPC/extended. That is, other programs such as MAIL and UUXQT pass it mail for delivery on standard input (STDIN), and RMAIL then handles actual writing to local mailboxes and/or queuing for remote systems. RMAIL is designed to be invoked from other programs, and as such, end-users should never have to invoke RMAIL directly. The following information is included primarily for those who need to invoke RMAIL from another program, such as an external news reader.

RMAIL operates in one of three modes:

As an RFC-822 parsing back-end to the MAIL user agent program
As a stand alone mailer for utility programs such as UUXQT
As substitute for the UNIX RMAIL program invoked by UUXQT for remote mail delivery.

These three operating modes are described below.

The general syntax of RMAIL is as follows:

RMAIL [[-f | -F] *filename*] [-x *debug*] *address(es)*
RMAIL [[-f | -F] *filename*] [-x *debug*] [-s *subject*] -w *address(es)* [-c *address(es)*] [[-b *address(es)*]
RMAIL [[-f | -F] *filename*] [-x *debug*] -t

Where:

-b *address(es)*	Specifies optional blind carbon copy *address(es)*. Must follow all other flags and addresses. Used only with -w flag.
-c *address(es)*	Specifies optional carbon copy *address(es)*. Must follow all other flags and addresses except for the -b flags and its associated addresses.
-f *filename*	Specifies *filename* is to be used in place of standard input
-F *filename*	Specifies *filename* is to be used in place of standard input. The file is deleted after use.
-s *subject*	Optional *subject.* Used with and implies the -w flag.
-t	Enables RFC-822 header parsing mode.
-w	Enable stand-alone mailer mode.

-x *debug* Display debug messages at or below level *debug*. The default value is 1. The option may be set to 0 for unattended production use, or as high 20 for detailed debugging.

address(es) One or more *addresses* the mail is to be delivered to. Not used with -t flag, but required for other modes.

RFC-822 Mode

In RFC-822 mode, RMAIL is invoked with the (**-t**) flag, which directs RMAIL to determine the addresses by parsing the mail's RFC-822 header. This mode acts as the back-end to a program such as MAIL which generates a the RFC-822 header and passes the mail to RMAIL for both local and remote delivery. RMAIL reads the header, validates the **From:** address, generates a a UUCP **From** line, RFC-822 **Message-ID:** and **Received:** lines, and delivers mail to each address included in the **To:**, **Cc:**, and **Bcc:** headers. The **Bcc:** header, if any, is read for its addresses but not copied; all other header lines are copied as-is.

Note: When an RFC-822 header prefixed by **Resent-** is found, only the **Resent-** headers are used; the original headers are copied but otherwise ignored. In this case, the preceding description applies to the **Resent-** headers; the original headers are copied without being examined.

The RFC-822 header read by RMAIL is subject to the following restrictions:

All **Resent-** headers, if any, must precede the original headers.

The **From:** header must precede the **To:** header.

Note: If the address in the **From:** header does not match the address of the user defined in the UUPC.RC and/or [*userid*].RC files, a **Sender:** line is generated with the correct address.

The **To:**, **Cc:**, and **Bcc:** headers must be together in the listed order. Each address in these headers must begin on a new line and be less than 512 bytes long.

Stand alone Mode

In stand alone mode, RMAIL is invoked with the (**-w**) flag to process mail without an existing RFC-822 header, bypassing the Mail User Agent (MAIL) for specialized applications such as mail generated by another program. This mail is subject to the following restrictions:

Mail is not logged in the user's outgoing mailbox

The user's signature file is not appended to the mail

User aliases are not expanded

All addresses plus the subject (if any) must fit on the MS-DOS or OS/2 command line

The following services are performed by RMAIL in stand alone mode:

A UUCP **From** line is generated.

A valid RFC-822 header is generated with **Received:**, **Date:**, **Message-ID:**, **From:**, and **To:** lines. In addition, **Subject:** and **Cc:** lines are generated as required if the Subject (**-s**) and/or Carbon Copy (**-c**) options are specified.

The generated RFC-822 header also includes a **From:** user id derived from the environment variable **LOGNAME**[24], if defined; otherwise the default current user is used. When **LOGNAME** is defined the real name of the user will be taken from the UUPC/extended PASSWD file if available, or a dummy name will be used.[25]

Mail is queued for the addresses on the command line, including primary addresses, carbon copied addresses, and blind carbon coped addresses. As with RFC-822 mode, the output of RMAIL in stand alone mode does not include any reference to blind carbon copy users in the actual mail header.

UNIX RMAIL Emulation Mode

In UNIX RMAIL emulation mode, the default, the following processing takes place:

Mail is read in to a temporary file. The UUCP **From** line, which is the first line in the file, is parsed and stripped from the mail. No other mail headers are examined.

Mail is delivered to each local user on the command line with a UUCP **From** line generated from the system name(s) parsed from the incoming **From** line.

Mail is delivered to all other users on the command line with a UUCP **From** line generated from the system name(s) parsed from the incoming **From** line with the system name UUXQT received the mail from prefixed to the list unless it is already the first system in the list.

Note: UUXQT normally passes the incoming system name in the **UU_MACHINE** environment variable, if set. Otherwise, the incoming system name is taken from the "remote from system" portion of the **From** line. If this data is also missing or invalid, the incoming system name is generated as being from the local system and user id /dev/null.[26]

Bugs

All RFC-822 address data within quotes is ignored. It should be processed in accordance to RFC-822.

Input to the program RMAIL in local address mode (-t option) must be in a specific format. It should accept any valid RFC-822 input. When sending mail, headers cannot be edited once entered.

With the **bounce** option enabled, RMAIL only bounces mail back to the address it parsed from the information passed by UUXQT; it does not examine the RFC-822 headers for fields such as **Errors-To:**.

---

[24]Note that this is an operating system environment variable, **not** a UUPC.RC or [*userid*].RC variable.

[25]"Postmaster" is used for the user id defined as Postmaster in UUPC.RC and for the address POSTMASTER, "UNIX-to-UNIX Copy" for the userid UUCP. For all other undefined addresses, the real name is set to the same value as the userid itself.

[26]This behavior is based on the similar behavior of the BSD sendmail program. Why it does it is beyond the scope of this document (or most human understanding).

RNEWS

Synopsis

RNEWS [[-f | -F] *filename*] [-x *debug*]

Availability

MS-DOS, OS/2, and Windows NT

Description

RNEWS processes incoming news from other systems, and is normally never directly invoked by the user. It is invoked automatically by UUXQT upon the arrival of news at the local system, reading the news on standard input. RNEWS operates in one of two modes: regular mode and SNEWS mode, described below.

The syntax of RNEWS is as follows:

RNEWS [[-f|-F] *filename*] [-x *debug*]

Where:

-f *filename* Specifies *filename* is to be used in place of standard input

-F *filename* Specifies *filename* is to be used in place of standard input, and the file is to be deleted after use.

-x *debug* Display debug messages at or below level *debug*. The default value is 1, but the option may set to 0 for unattended production use or as high 20 for detailed debugging.

Note: For RNEWS to correctly process compressed news (news which has been compacted to take less time to be transmitted to the local system), a decompression program must be installed. Unless this program is named **uncompre** and resides in the path searched by **UUXQT**, the variable **Uncompress** must be defined in the UUPC.RC file. See the sample UUPC.RC and **The UUPC/extended UUPC.RC and [userid].RC files**, page 75, for details.

Note: The compress program is *not* supplied with UUPC/extended[27].

Regular Mode

The regular mode of RNEWS enabled by specifying or defaulting the Boolean option **nosnews** in the UUPC.RC configuration file. In this mode, RNEWS searches for the file ACTIVE in the UUPC/extended configuration directory, and reads this file for the local active news groups. Any news received for groups not included in the ACTIVE file is discarded without comment. A sample ACTIVE file is included with the UUPC/extended distribution.

---

[27]Either *compress* or *gzip* can be used to uncompress compressed news. Both can be found on many Internet archives. Check at wuarchive.wustl.edu for MS-DOS versions, hobbes.nmsu.edu for OS/2 versions. NT versions do exist, but not at any particular public archives at this point. Try rhino.microsoft.com, sunsite.unc.edu, or gowinnt.microsoft.com but no promises.

The format of the ACTIVE file defines one news group per line, with the format for each line as follows:

news.group first last post

Where:

news.group	The name of the news group.
first	The number of the first article currently on system. If no articles are on the system, the number should be 0.
last	The number of the last article on the system. If no articles are on the system, this number should be 1.
post	A single character describing whether or not local users can post to the news group. Y allows unrestricted posting, N disallows all posting, and M defines the group as moderated, and that posts are to be forwarded to the moderator.

Note: If the ACTIVE file does not exist or has gross format errors, RNEWS will abort and the incoming news will be lost.

In regular mode, news is processed as follows:

1. The ACTIVE file is loaded.

2. If the input file is compressed, the input is copied to a holding file and the user specified uncompress program is invoked to decompress the holding file. It is then treated as batched news.

3. If news is batched with multiple articles per file transferred, then the articles in the file are written to a temporary file one at a time. The temporary file is then processed as a single article.

4. For each article processed, the news groups line of the header is scanned. For each group listed in the header which is also in the ACTIVE file, the article is copied to the group's directory under the next available article number.

5. The ACTIVE file is written out with updated article numbers.

SNEWS mode

The SNEWS mode of RNEWS is enabled by specifying the Boolean option **snews** in the UUPC.RC configuration file. In this mode, RNEWS does not uncompress or unbatch news, nor does it distribute news to directories based on the news groups for which the news is intended. Rather, the files are written out in sequence in the UUPC/extended news directory with a file type of *.ART.

Note: In SNEWS mode, the ACTIVE file need not exist, and is ignored if it does exist.

Bugs

The invoked compress program may run out of memory if invoked on a memory constrained DOS system.

RNEWS does not translate slashes (/) in file names to back slashes (\) to conform to DOS command line conventions.

Updates to the ACTIVE file can be lost if RNEWS crashes while writing it. Given an old ACTIVE file, the article numbers can be brought back up to date by running EXPIRE.

See Also

EXPIRE

History

This is the 20th revision of this document and was last updated on October 1, 1993 by Katherine Derbyshire.

SU

Synopsis

SU *userid* [*command*]

Availability

MS-DOS and Windows NT (batch file)

OS/2 (REXX script)

Description

SU changes the environment variable UUPCUSRRC to change the user profile of the user using UUPC/extended. This allows multiple users to easily share the same PC for UUPC/extended.

> **Note:** Because SU is a batch file under DOS and Windows NT, it does **not** examine UUPC.RC for the name of the UUPC/extended configuration directory. Thus, under DOS and Windows NT, if the directory is not the default \uupc, the SU command file must be edited to point at the configuration directory.

The syntax of the SU command is:

SU *userid* [*command*]

Where:

userid	defines the user to use when setting the UUPCUSRRC variable. The file \UUPC\[*userid*].RC must exist.
command	defines the optional command to execute as this user, i.e. MAIL. The default is to run a second copy of COMMAND.COM. The default user id restored by typing EXIT to terminate the inferior command processor.

UUCICO

Synopsis

uucico [-r 1] [-s *system*] [-x *debug*] [-l *logname*] [-m *inmodem*] [-n] [-t]

uucico -r 0 [-w *user*] [-x *debug*] [-z *baudrate*] [-l *logname*] [-d *hhmm*] [-t]

Availability

MS-DOS, Windows 3.x, OS/2, and Windows NT

Description

UUCICO performs the actual exchange of files with other systems. Normally, UUCICO is invoked from either UUPOLL or UUIO, but the program can also be invoked directly. To make UUCICO call all connected systems to deliver and pickup remote mail and files:

```
UUCICO -s all
```

This directs UUCICO to poll all the systems listed in the SYSTEMS file.

To make UUCICO wait for an incoming call:

```
BREAK ON
UUCICO -r 0
```

UUCICO will wait for a successful telephone call, and exit upon completion. Entering Ctrl-Break from the keyboard will terminate UUCICO.[28]

Under OS/2 only, when UUCICO is invoked in passive mode to listen for a phone to ring, it will automatically surrender the port to a UUCICO calling out or the program UUPORT. If a calling out UUCICO suspends a passive UUCICO, it will automatically wake up the sleeping UUCICO to take the port back. If UUPORT is used to suspend a UUCICO, UUPORT must be invoked a second to reactivate the sleeping UUCICO.

The full list of options supported by UUCICO is as follows:

-d *hhmm* Duration. If specified, this flag specifies that UUCICO running in passive mode is terminate after *hh:mm* (hours and minutes). Note that this is a *duration*, not an end time.

Note: This flag replaces the execute period flag (-u) in previous versions of UUCICO.

-l *logname* Specifies the name of the log file used by UUCICO. The default extension is ".LOG" and the default log file is UUCICO.LOG.

---

[28]The BREAK ON command is required to insure that MS-DOS polls the keyboard for the Ctrl-Break sequence; otherwise, you may have to reboot your system to terminate UUCICO. BREAK ON is always enabled under OS/2, Windows 3.x, and Windows NT.

-m *inmodem* Specifies the name of the modem file to use for passive polling. This flag implies the -r 0 flag. The default is to use the value of the UUPC.RC variable **InModem** for passive polling.

-r 0 Slave Mode: initializes the modem and waits for the telephone to ring with an incoming call; presenting the caller with a UNIX style login prompt. This option is modified by the -d, -w, -x, and -z options.

-r 1 Master Mode: actively poll (call out) to the system defined by the -s option. This option is the default, and is modified by the -n and -s options.

-n Call now flag: when specified, UUCICO ignores the time fields defined in the SYSTEM file when determining if a system should be called, and treats all systems as if they were defined with a time of "any".

-s *sysname* System name to call. Default is to call "any", which is any system the local system has work queued for. *Sysname* may also be specified as "all", which requests all systems listed in the systems file be called, or as the name of any system listed the systems file. This option is ignored when -r 0 is specified.

-t Trace all data sent or received through communications port to file LINEDATA.LOG in the spool directory.

-w *who* Begin processing in slave mode as if user *who* had just logged in. This option is for use when another program has answered the modem and validated the user id. It implies the -r 0 option and is modified by the -z option.

-x *debug* Display debug messages at or below level *debug*. The default value is 1, but the option may set to 0 for unattended production use, or as high 20 for detailed debugging.

Note: Use of debug level 5 or higher for extended periods of time is not recommended as it will rapidly fill your hard disk with debugging output.

-z *bps* Set modem speed to *bps* bits per second when used with -w option. The default speed is the speed listed in the modem definition file defined in the UUPC.RC variable **InModem**. This option is ignored if -w is not specified.

Exit codes

UUCICO exits with a status of 0 if it manages to successfully connect to at least one remote machine while it was running.

Other exit codes and their meaning:

5	A normal exit, but UUCICO never managed to connect to any remote sites.
69	The program aborted due to a panic situation.
100	The user pressed Ctrl-C to terminate UUCICO.

Bugs

On UNIX systems, the default for UUCICO is -r 0 (to be called) rather than -r 1 (call out). UUCICO -r 1 is a more reasonable default for a PC, but might be unexpected for a UNIX expert.

DECUS UUCP 1.1 won't exchange protocol information with UUPC/extended. A fix for DECUS UUCP 1.1 was posted to the net, and the bug is also fixed in version 1.2.

UUPC/extended forces communications to No Parity, 8 data bits, 1 stop bit, no in-band flow control.

> **Note**: This is a requirement of the UUCP "g" protocol, which requires a transparent data line, and thus is a permanent restriction for "g" protocol connections.

The -w option doesn't work properly under OS/2 or Windows NT. Under OS/2, the handle of the open port should be passed.

The 32 bit OS/2 version of UUCICO does not support automatically suspending its access to a port it is listening on. This will be corrected in a future release.

If UUCICO is suspended, woken up, and resuspended immediately, the second suspend may fail because the UUCICO has not finished reinitializing the modem yet.

If for some reason UUCICO is not woken up after being suspended (if, for example, the suspending program crashes), the suspended UUCICO will do a Rip Van Winkle and sleep forever. It must be killed or manually woken up using UUPORT.

Named pipes under OS/2 are incredibly inefficient, with low throughput (~2K bytes per second on a loop back) and high CPU usage. It is unclear if this is a UUCICO bug or lossage in the operating system itself.

UUCICO uses operating system messages when possible for operating system API errors (such as communications port errors) under OS/2 and Windows NT. However, it does not perform parameter substitution, meaning percent signs (%) followed by a single numeric digit may show up in error message text.

Under OS/2, UUCICO runs at a raised priority to improve throughput. While this presents no noticeable impact on response time with the IBM serial port driver, the publicly available SIO.SYS driver may take sufficient CPU time to affect interactive response time. The **Priority** keyword in the [*modem*].MDM file can be used to adjust the UUCICO priority under OS/2.

Under Windows 3.1, UUCICO crashes UUPOLL when UUCICO exits.

See Also

UUPOLL, UUIO, UUXQT, RMAIL, UUPORT, and RNEWS

History

This is the 21st revision of this document and was last updated on October 4, 1993 by Katherine Derbyshire.

1993UUPC/extended

UUCLEAN

Synopsis

UUCLEAN [*spooldir* [*tempdir*]]

Availability

MS-DOS and Windows NT (batch file)

OS/2 (REXX script)

Description

UUCLEAN ages log files through five generations and clears old UUPC/extended temporary files from the spool and temporary directories. Under OS/2, UUCLEAN also executes UNDELETE to purge these files and other spool files from the OS/2 undelete cache. UUCLEAN must be executed when programs which create log files (UUCICO, UUXQT, and RMAIL) are not running.

Note: Under DOS and Windows NT, UUCLEAN does not clean log files created in the spool directory created with non-standard names via use of the UUCICO log file (-l) flag.

The syntax of the UUCLEAN command is:

UUCLEAN [*spooldir* [*tempdir*]]

Where:

spooldir Specifies the UUPC/extended spool directory to clean. The default is \uupc\spool on the current drive.

Note: Because UUCLEAN is a batch file under DOS and Windows NT, it does **not** examine UUPC.RC. Therefore, the file must be edited if you do not use the default directory structure and do use UUCLEAN.

tempdir Specifies the temporary directory to clean. The default is the value of the TEMP environment variable.

Note: tempdir is a positional argument. If you want to specify tempdir, you must also specify spooldir.

UUIO

UUIO is a batch file which executes UUSTAT -q and UUCICO followed by UUXQT. All UUIO command line arguments are passed to UUCICO. No user arguments are passed to UUSTAT or UUXQT.

> **Note:** The UUSTAT command can be deleted from the UUIO command file with no loss of function. It is included for the informational display only.

See **UUCICO** on page 55, above, for a description of the command line arguments.

UUCP

Synopsis

UUCP *srcfile system*!*destfile* or
UUCP *system*!*srcfile destfile*

Availability

MS-DOS, OS/2, and Windows NT

Description

UUCP queues binary or text files for transfer between two directly connected systems. The basic UUCP command syntax is:

UUCP *srcfile system*!*destfile* or
UUCP *system*!*srcfile destfile*

The first example copies a local file (*srcfile*) to a remote host (*system*) as *destfile*. The second example copies a file (srcfile) on a remote host (system) to the local file destfile. Filenames may be specified as an absolute path name, relative to a user's home directory (~user/file), or a path relative to the UUCP public directory (~/name).

Note: On most systems, access will be severely restricted. Check with a user or system administrator on the remote system before transferring files to possibly restricted locations.

For additional information on the UUCP command, see chapter 2 of *Using UUCP and Usenet.*

Bugs

Access checking is not as severely enforced as it would be on a true UNIX system, particularly for local originated UUCP commands.

Multi-hop transfers don't work.

UUNAME

Synopsis

UUNAME [-l | -d] [-x *debug*]

Availability

MS-DOS, Windows 3.x, OS/2, and Windows NT

Description

UUNAME reports names of the remote systems defined to UUPC/extended via the SYSTEMS file or the local system name (nodename) defined in UUPC.RC.

The syntax of the UUNAME command is:

UUNAME [-l | -d] [-x *debug*]

Where:

-d	specifies the local domain name is to be reported.
-l	specifies the local system name is to be reported.
-x *debug*	Display debug messages at or below level *debug*. The default value is 1. The option may set to 0 for unattended production use, or as high 20 for detailed debugging.

The default is report all the defined remote systems instead.

Bugs

Too stupid to have any.

UUPOLL

Synopsis

UUPOLL [-r 0 | 1] [-f *firsttime*] [-i *hhmm*] [-d *hhmm*] [-e *hhmm*] [-c *hhmm*] [-l *logfile*] [-m *modem*] [-B *batchfile*] [-C *command*]

Availability

MS-DOS, Windows 3.x, OS/2, and Windows NT

Description

UUPOLL allows unattended operation of the PC, automatically running UUCICO on a timed basis. Each time UUPOLL invokes UUCICO, it also automatically runs UUXQT to process any files received by UUCICO.

To use UUPOLL to have UUCICO call out on a regular basis:

```
UUPOLL -f 0240 -i 0600
```

This will cause UUCICO to call out at 2:40 A.M. and every six hours thereafter until the user presses Ctrl-Break. Both flags are specified as hhmm (hours and minutes).

To use UUPOLL to have UUCICO call out on a regular basis, and automatically answer the telephone between outgoing calls:

```
UUPOLL -f 0240 -i 0600 -r 0
```

This will cause UUCICO to call out at 2:40 A.M. and every six hours thereafter until the user presses Ctrl-Break, and in addition UUCICO will be invoked in passive mode to answer the telephone between outgoing calls.

The full list of arguments allowed by UUPOLL are as follows:

-a *hhmm* Automatically actively poll system "any" after each successful incoming poll if *hhmm* seconds have passed since last active poll. This allows mail delivered by incoming systems to be automatically forwarded to systems the local system is allowed to call out to. The delay time may be specified as 0, in which case UUCICO will actively poll system "any" after every successful incoming telephone call. This option has no effect if -r 0 is not specified.

-B *command* Batch command to execute before each active call at -i *hhmm* intervals. Default is to execute no commands before calling out.

-c *hhmm* Time to execute UUCLEAN to clean the spool directory in hours and minutes. The default is to not run UUCLEAN.

-C *command* Command to run to clean spool directory at time specified by -c flag. The default is the run UUCLEAN command with no operands.

-d *hhmm* Duration of polling in hours and minutes. If the exit (-e) flag is not specified, then after this period UUPOLL exits. If neither -d or -e is specified, the default is poll until the user presses Ctrl-Break.

-e *hhmm* Time of day that UUPOLL is to exit. If neither -d or -e is specified, the default is poll until the user presses Ctrl-Break. If the specified time has already passed and the duration (-d) flag is not specified, then UUPOLL immediately exits. If the duration flag is specified, then UUPOLL will poll if and only if the exit time (-e *hhmm*) would be reached before the duration (-d *hhmm*) expires. In any case, UUPOLL will exit no later than the time specified by the exit time.

Example: UUPOLL -e 2400 -d 2400 will always run until the next midnight, but UUPOLL -e 300 -d 1200 will exit immediately if it is more than 12 hours before 3 AM (that is, the current time is before 3 PM).

-f *hhmm* First time to poll in hours and minutes. Default is to not actively poll unless the user specifies an interval via -i, in which case the default is the current time plus the interval.

Note: UUPOLL automatically determines when it is later than the specified first poll time and selects the available next time to poll. If it desired for UUPOLL to poll 24 hours a day, then the first poll time should be specified as close to the previous midnight as possible.

If, for example, the system is to poll at 6:13 AM, 2:13 PM, and 10:13 PM, UUPOLL must be invoked with:

```
UUPOLL -i 0800 -f 0613
```

Even if it is after 6:13 AM.

-i *hhmm* Interval to poll in hours and minutes. Default is 0400 (4 hours) if -f is specified.

-l *log* log file name to pass to UUCICO.

-m *modem* modem name to pass to UUCICO for inbound calls.

-r 0 Directs UUCICO to answer telephone between active polls. Default is to not answer the telephone.

-r 1 Directs UUCICO not to answer the telephone, but to actively poll after the interval specified with -i.

-s *system* System name to poll. Default is "all" for the first poll, followed by "any" for succeeding polls, which cannot be explicitly specified.

-x *debug* Debug level passed to UUCICO and UUXQT. Default is 1.

Note: Either -r, -i, or -f must be specified.

Note: For additional information on the -l, -m, -s, -x, and -r options, see the description of UUCICO on page 55, above.

Note: If you specify both the -r 0 and either the -f or -i options, the effect is to have UUCICO invoked to answer the telephone between the active polls defined by the -f or -i options.

Exit codes

UUPOLL exits with a status of 0 if it ran, carried out its work, and exited without encountering any problems.

Other exit codes and their meaning:

69	The program aborted due to a panic situation.
100	The user pressed Ctrl-C to terminate UUCICO.

Bugs

Polls after 2 AM on the day when daylight savings time begins or ends in the US will be off by one hour. This is due to the way that UUPOLL handles (or fails to handle) days that are not 24 hours long. UUPOLL will not attempt to poll during the 25th hour of such days, instead waiting for the first poll of the next day.

UUPORT

Communicate with background UUCICO to surrender communications port.

Synopsis

UUPORT [-s | -r] port

Availability

OS/2 only (DOS under OS/2 yields interesting results. See below)

Description

UUPORT contacts an OS/2 background UUCICO via a named pipe and requests the UUCICO to surrender or resume use of the communications port specified. This allows another program to use the port without canceling the UUCICO waiting for the and restarting it by hand.

The syntax of the UUPORT command is:

UUPORT [-s | -r] port

where:

-s specifies the port is be surrendered.

-r use of the port is be resumed.

Note: If both the -s and -r flags are omitted, the current status is queried but left unchanged.

port The port name to process.

Files

\PIPE\UUCICO\ZZ*port*	named pipe for contacting local background UUCICO
servername\PIPE\UUCICO\ZZ*port*	named pipe for contacting networked background UUCICO

Bugs

The DOS version of the program can be built with no source changes, but the results of attempting to run the program are unpredictable and usually catastrophic to either the DOS box or the system as a whole.

See UUCICO for additional problems and caveats.

See Also

UUCICO, UUPOLL

Author

Kai Uwe Rommel, with the usual additional hacking by Kendra Electronic Wonderworks.

History

This is the 21st revision of this document and was last updated on October 4, 1993 by Katherine Derbyshire.

UUSTAT

Synopsis

UUSTAT [-a] [-m] [-q] [-k *jobid*] [-r *jobid*] [-x *debug*]
UUSTAT [-s *system*] [-u *user*] [-x *debug*]
UUSTAT [-P *system*] [-x *debug*]

Availability

MS-DOS, Windows 3.x, OS/2, and Windows NT

Description

UUSTAT will display the status of, or cancel, previously specified uucp commands, or provide general status on uucp connections to other systems. Only one of the following options can be specified with UUSTAT per command execution:

-a List all jobs in queue.

-m Report the status of accessibility of all machines.

-q List the jobs queued for each machine. If a status file entry exists for the machine, its date, time and status information are reported. In addition, if a number appears in parentheses next to the number of C or X files, it is the age in days of the oldest C/X file for that system.

Note: For systems with a large number of outstanding jobs, this could take 30 seconds or more of real-time to execute.

-k *jobid* Kill the uucp request whose job identification is *jobid*.

-r *jobid* Rejuvenate *jobid*. The files associated with *jobid* are touched so that their modification time is set to the current time.

Note: The refresh option hasn't yet been fully implemented, but that's okay because job expiration doesn't happen either.

-P *system* Create a poll work file. This empty call file causes *system* to be called every time UUCICO is invoked with the "-s any" option until the system is contacted.

-P all Create poll work files for all systems.

Any or all of the following options can be specified with UUSTAT:

-s *sys* Report the status of all uucp requests for remote system *sys*.

-u *user* Report the status of all user requests issued by *user*.

-x *debug* Display debug messages at or below level *debug*. The default value is 1, but the option may set to 0 for unattended production use, or as high 20 for detailed debugging.

When no options are given, UUSTAT outputs the status of all uucp requests issued by the current user.

Note: Poll files created by the current user with the -P option are not displayed unless the -s or -a options are used.

Returns

exits with a status of 0 upon successful completion of its task.

Other exit status and their meaning:

1	Bad usage.
2	stat(), open(), access(), or utime() call failed.
3	fopen() failed or file pointer was NULL .
4	An internal problem, usually with file access.
69	The program aborted due to a panic situation.
100	The user typed Ctrl-C.

Files

/uupc/spool/system/C/*	Command Files for host "system"
/uupc/spool/system/D/*	Data files
/uupc/hostatus	System status summary file

Bugs

The Retry and Count features shown by UNIX systems are not implemented since UUPC/extended doesn't implement STST.system files where some of the needed information would come from.

Spooled eXecute (X*) files are not listed by UUSTAT. This can be corrected without too much trouble, but since the local system doesn't generate the names, they could have been truncated (meaning UUSTAT will blow up during DOS to UNIX name translation).

Work poll files are created with class Z, which is what SunOS 4.1.3 uses for poll files. This may not be desirable if the local system exploits call grading to process only classes above Z.

UUSTAT does not actually refresh times with the -r option.

The time reported by UUSTAT for each system is the time the local system last connected to the remote system, not the time of the last attempt as in UNIX versions of UUSTAT. This is both for historical reasons and because the author thought it was more useful. UUSUB can be used to view the time of the last attempt.

Poll work files (generated with the -P flag) are not shown in a per user display.

UUSUB

Synopsis

UUSUB [-x debuglevel] [-c] [-s system]

Availability

MS-DOS, Windows 3.x, OS/2, and Windows NT

Description

UUSUB reports statistics on the data transmitted between the local and remote systems since the last time the file HOSTATUS in the UUPC/extended spooling directory was created. UUSUB is invoked with no operands to report these statistics:

```
UUSUB
```

The syntax of UUSUB is:

UUSUB [-x debuglevel] [-c] [-s system]

Where:

-c	Clears (resets) the totals.
-s sysname	System name to report on or to clear. The default is report on or clear all known systems.
-x *debug*	Display debug messages at or below level *debug*. The default value is 1. The option may be set to 0 for unattended production use, or as high 20 for detailed debugging.

UUX

Availability

Usage discouraged under all operating environments. (See below.)

Description

The UUX command queues commands for execution on remote systems. It used by other facilities, such as news functions and the UUCP command, to handle processing more complex than simple file transfers.

Bugs

The UUX command is still under development. Its use is discouraged at this time.

Specific problems are believed to exist with gathering files from multiple systems to process, and multi-hop commands.

UUXQT

Synopsis

UUXQT [-x *debuglevel*] [-s *system*]

Availability

MS-DOS, Windows 3.x, OS/2, and Windows NT

Description

UUXQT must be executed to process remote files after UUCICO has received these files from a remote host. It normally is invoked with no operands:

```
UUXQT
```

This will automatically process all eXecute files in the local spool queues with the default debugging level in effect.

UUXQT supports the following command line options:

-s *sysname*	Process work only for *sysname*. The default is system "all", which processes work for all known systems.
-x *debug*	Display debug messages at or below level *debug*. The default value is 1. The option may be set to 0 for unattended production use, or as high 20 for detailed debugging.

To automatically execute UUXQT every time UUCICO is run, use UUPOLL or UUIO.

Bugs

Under DOS, if a large number of eXecute files are processed at once by UUXQT under DOS, the DOS environment becomes corrupted and called programs (usually RNEWS or RMAIL) crash because of the bad environment. This is hard to debug because the redirected output causes the program error to not be displayed on the console.

WAITING

Synopsis

WAITING [*userid*]

Availability

MS-DOS and Windows NT (batch file)

OS/2 (REXX script)

Description

The WAITING command reports on which users have mail waiting in the UUPC/extended mail directory. It exits without comment if no mail is waiting.

Note: Because WAITING is a batch file under DOS and Windows NT, it assumes the mail directory is \uupc\mail on the current drive. The file *must be edited by hand* if this is not the directory for system mailboxes.

Note: This command is more effective if the **multitask** option is specified in the UUPC.RC file, as then mail is moved out of the system mail directory as soon the user invokes the MAIL command after new mail arrives.

The syntax of the WAITING command is:

WAITING [*userid*]

Where:

userid defines the *userid* to check for waiting mail. The default is all users.

Bugs

WAITING assumes the extension of mailbox files is SPB.

WAITING does not support the **directory** option, which requires look in subdirectories for the waiting mail.