UUPC/extended version 1.12b
Installation and User Reference

UUPC/extended§ 1.12b§ **Installation and**

October 4, 1993Kendra Electronic Wonderworks
P.O. Box 132
Arlington, MA 02174
(617) 641-3452

Table of Contents

Introduction

Introductiontc "Introduction"§

Copyrighttc "Copyright" \l 2§

Viewing on a Terminaltc "Viewing on a terminal" \l 2§

This document includes overstrikes for printing on a printer. To read this document on-line, extract the program NOVRSTRK from the UUPC/extended optional programs archive and issue the following command:

```
NOVRSTRK UUPCDOCS.PRN UUPCDOCS.TXT
```

You can then browse the terminal ready version of the document, UUPCDOCS.TXT. **However,** given that this document is well over 100 pages long, you'll want to either print it or have a good search function handy.

Revisiontc "Revision" \l 2§

This is the 3rd revision of this document, and applies to UUPC/extended version 1.12b. This document was last updated on September 24, 1993 by Katherine Derbyshire. Even if you have read previous versions of this document, **General Information** on page 7 and **Changes From Previous Versions** on page 133 should be reviewed for updates.

Installation

Licensing, Redistribution, and Warrantytc "Licensing, Redistribution, and Warranty"§

Terms and Conditions for Copying, Distribution, and Modificationtc "Terms and Conditions for Copying, Distribution, and Modification"\l2§

"UUPC/extended", below, refers to the entire UUPC/extended package, as distributed by Kendra Electronic Wonderworks. A "work based on UUPC/extended" means either UUPC/extended or any derivative work under copyright law: that is to say, a work containing UUPC/extended or a portion of it, either verbatim or with modifications and/or translated into another language. (Translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. UUPC/extended may be installed and run on as many systems as desired without restriction.

1. You may copy and distribute verbatim copies of UUPC/extended's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of UUPC/extended a copy of this License along with UUPC/extended.

a) You may not distribute versions of UUPC/extended that you know to be obsolete. You are specifically prohibited from knowingly distributing obsolete versions of UUPC/extended in order to support third party programs.
b) You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of UUPC/extended or any portion of it, thus forming a work based on UUPC/extended, and copy and distribute such modifications or works under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified files must carry prominent notices stating that you changed the files and the date of any change.
b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from UUPC/extended or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
c) You may not make malicious changes to UUPC/extended. You may not insert viruses, worms, or other code intended to damage or disrupt the end user's system and/or data.
d) You may not cripple UUPC/extended's functionality in any way. In particular, you may not make continued use of UUPC/extended contingent on payment of a registration fee.

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from UUPC/extended, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections. Specifically, unique works may interface to UUPC/extended via configuration, spooling, or other files using routines based on UUPC/extended source code without falling under the scope of this License.

It is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on UUPC/extended.

3. You may copy and distribute UUPC/extended (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for non-commercial distribution and only if you received UUPC/extended in object code or executable form with such an offer, in accord with Subsection b above.)
d) Distribution of multiple copies of the object code within a single organization is specifically exempted from this section.

The complete UUPC/extended source code means all the source code for all modules UUPC/extended contains plus the scripts used to control compilation and installation of the executable. However, UUPC/extended source code does not include any components of either the operating system under which UUPC/extended runs or the compiler and runtime libraries used to construct the executable.

4. You may not copy, modify, sublicense, or distribute UUPC/extended except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute UUPC/extended is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute UUPC/extended or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing UUPC/extended (or any work based on UUPC/extended), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying UUPC/extended or works based on it.

6. Each time you redistribute UUPC/extended (or any work based on UUPC/extended), the recipient automatically receives a license from the original licenser to copy, distribute or modify UUPC/extended subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

NO WARRANTYtc "NO WARRANTY"\l2§

7. Because UUPC/extended is licensed free of charge, there is no warranty for UUPC/extended,

to the extent permitted by applicable law. Except when otherwise stated in writing the copyright holders and/or other parties provide UUPC/extended "as is" without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the quality and performance of UUPC/extended is with you. Should UUPC/extended prove defective, you assume the cost of all necessary servicing, repair or correction.

8. In no event unless required by applicable law or agreed to in writing will any copyright holder, or any other party who may modify and/or redistribute UUPC/extended as permitted above, be liable to you for damages, including any general, special, incidental or consequential damages arising out of the use or inability to use UUPC/extended (including but not limited to loss of data or data being rendered inaccurate or losses sustained by you or third parties or a failure of UUPC/extended to operate with any other programs), even if such holder or other party has been advised of the possibility of such damages.

General Informationtc "General Information"§

Software Copyright and related creditstc "Software Copyright" \l 2§

Please read our complete license agreement in **Licensing, Redistribution, and Warranty**, page 3. Use or further distribution of this software will be interpreted as assent to the terms of the license. Briefly, this software may be used and copied freely so long as the applicable copyright notices are retained, but no money shall be charged for its distribution beyond reasonable handling costs, nor shall proprietary changes be made to this software so that it cannot be distributed freely. Whenever possible, the source must distributed with the executable files.

Portions of this software are copyrighted by the following persons:

Copyright (c) Richard H. Lamb 1985, 1986, 1987
Copyright (c) Samuel Lam 1987
Changes Copyright (c) Stuart Lynne 1987
Changes Copyright (c) Kendra Electronic Wonderworks 1989-1993
Changes Copyright (c) David M. Watt 1991-1993
Copyright (c) Dewey Coffman 1989
Copyright (c) Mitch Mitchell 1991
Changes Copyright (c) Jordan Brown 1991
Changes Copyright (c) Mike Lipsie 1991

Additional thanks go to Harry Broomhall, Harald Boegeholz, Robert B. Denny, John DuBois, Edward Keith, Phillip Meese, Dave Platt, Mike O'Carroll, Kai Uwe Rommel, Cliff Stanford, and Wolfgang Tremmel.

Honorable mention goes to Snuffles P. Bear, who says to please send chocolate. Warm thanks to Russell Nelson, who, among many other services rendered, did so.

The run-time libraries for Borland C++ are copyrighted by Borland International, and are used by permission. Likewise the run-time libraries for Microsoft C are copyrighted by Microsoft and are used by permission.

Borland C++, Microsoft C, Microsoft Word for Windows, Microsoft Windows, MS-DOS, OS/2, Windows NT, and IBM are trademarks of their respective owners. Other brand and product names are trademarks or registered trademarks of their respective holders and are capitalized accordingly.

Overviewtc "Overview" \l 2§

Greetings from Kendra Electronic Wonderworks!

UUPC/extended is a PC based version (and pun of) UUCP (UNIX-to-UNIX copy). UUPC/extended is a program which implements peer-to-peer networking using the UNIX UUCP protocols. MS-DOS, OS/2, Windows 3.x, or Windows NT-based personal computers using these protocols can exchange mail and files with a UNIX system, another UUPC/extended system, or other systems using UUCP over dial up or dedicated phone lines.

Introductiontc "Introduction" \l 2§

This document provides information on UUPC/extended version 1.12b, produced by Kendra Electronic Wonderworks with contributions from around the globe. UUPC/extended is based on the free widely distributed interim version of UUPC, version 1.05, written by Stuart Lynne, Richard H. Lamb, and Samuel Lam, in Vancouver, BC. The MS-DOS version of UUPC is no longer maintained in Vancouver; UUPC/extended 1.12b is the official current release of UUPC. The differences between UUPC/extended and UUPC version 1.05 include user aliases, routing to multiple hosts, dial in support, limited domain address parsing and routing, support for the OS/2, Windows 3.x, and Windows NT operating environments, and various improvements in the user interface. Because of these changes, much of what is contained in this document does not apply to the original UUPC 1.05 package.

UUPC/extended is written in C, and compiles with Borland C++ 3.1 under MS-DOS, with Microsoft C 6.0 under DOS, Windows 3.x, or OS/2, or with Microsoft C++ for NT under Windows NT (version 3.1), with one additional module in 8086 Assembler (MASM) for the MS-DOS versions. The documentation is written using Microsoft Word for Windows 2.0.

Minor revisions to UUPC/extended are denoted by changes to the letter suffix on the version number. Minor revisions include only those changes, no matter how great, that allow the user to return to a previous release of UUPC/extended by at most updating the configuration file. Major revisions to UUPC/extended are denoted by a change in the number of the version; a major revision has changes in it which introduce a permanent incompatibility with the previous release. Beginning with version 1.12, major revisions are those which are distributed to the registered user base, whether they introduce a permanent incompatibility or not. See How to Register UUPC/extended, page 127, for details on how to register your copy of UUPC/extended.

For a full description of the changes to UUPC/extended and how they affect upgrading from one release to the next, see **Changes From Previous Versions**, page 133, included in this package. It should be duly noted that there is no version 1.24b at this time. A user was having nightmares in which she didn't have version 1.24b and everyone else did . . .

Who Should Use This Programtc "Who Should Use UUPC/extended" \l 2§

As mentioned above, UUPC/extended is a peer-to-peer networking program: your local system and systems it communicates with are on an equal footing in that in general the capabilities and access of the systems are symmetrical. You have complete control over and must be responsible for your own system. Compare this to a client-server set-up such as when you dial-in to a central mainframe or Bulletin Board System on which the System Operator handles the administration (and restricts your access).

One application for UUPC/extended is to provide remote access to a private system or small network, for example allowing a laptop machine access a single remote PC or UNIX system. This requires very little overhead, and may be used for connections needed for just a few days.

A more common use for UUPC/extended is to provide dial-up e-mail access to public networks such as the UUCP network and/or the Internet. When UUPC/extended is used for this purpose then you, as the system administrator, will be responsible for keeping up your own system as a member of the global network community. Related duties include making sure your system calls other systems regularly (including when you are not around), monitoring your links to other systems, registering your system with a central authority such as the UUCP Project and/or the Internet Network Information Center, and other long term tasks.

Because of these tasks, not all people should use a program such as UUPC/extended for general e-mail access. A good rule of thumb is if you do not receive more than one e-mail message a week or do not require e-mail access for more than six months, then you should consider a commercial service such CompuServe, MCI Mail, ATT Mail, or others over setting up UUPC/extended. This reduces the set-up work required on your part and avoids the problems associated with registering and de-registering your system, leaving the system running during absences, and the like.

UUPC/extended should also be avoided if your primary interest is in a server which allows downloading files to other personal computers which do **not** have UUPC/extended or some other UUCP program. UUPC/extended is not intended for interactive logins, the UUCP protocols are not compatible with the protocols (Xmodem, Kermit, etc.) included with most PC terminal emulators, and UUPC/extended security is oriented towards a pre-defined list of known systems calling in regularly. Use a BBS program instead for generic download functions and/or interactive access.

Finally, UUPC/extended should be avoided if you lack the resources, financial or otherwise, to support the program. Installing the program does require some experience in setting up complex systems or a willingness to learn via associates or books. Trying to set up UUPC/extended without experience and resources (such as the Nutshell handbooks discussed below) can cause pain and suffering for a very important person--you.

System Requirementstc "System Requirements" \l 2§

To install UUPC/extended, you'll need the following:

- * A computer running MS-DOS, OS/2, or Windows NT. Under DOS, you need at least 512 KB of RAM and 2 megabytes of hard disk space (for executables, configuration files, documentation, and mail received). For OS/2, Windows 3.x, or Windows NT you need the minimum system configuration plus 2 megabytes of free hard disk space.
- * A modem.
- * The manual for the modem.
- * A text editor. The MS-DOS editor, Windows Notepad, or OS/2 editor will work

fine.

- * A friend with a machine that does UUCP. This can be a UNIX system, another UUPC/extended or other UUCP clone running on a PC, or a VMS system running DECUS UUCP.
- * A copy of PKUNZIP.EXE, version 2.04g or later, or a compatible utility. PKUNZIP is a shareware utility and can be found at almost all archive sites. Consult your local BBS, or get it from wuarchive.wustl.edu.

Required readingtc "Required Reading" \l 2§

Read **Installation**, page 15, now for installation and upgrade instructions. Users of previous versions of UUPC/extended should also read **Changes From Previous Versions**, page 133, for a summary of changes in the current release. Instructions on using the programs are in **UUPC/extended Command Reference**, page 35.

This document and its companions supplied with UUPC/extended only provide an overview of UUCP connections. The average person wishing to set up UUPC/extended will require a reference such as *Managing UUCP and Usenet*, part of the Nutshell Handbook series by O'Reilly and Associates.[1],[2] This book, while not specifically addressing UUPC/extended, includes detailed information on:

Format of the SYSTEMS file
Format of the PERMISSN (Permissions) file
Information on debugging login scripts
Information on registering your site with various networks

For using UUPC/extended, the Nutshell Handbook *Using UUCP and Usenet* will be helpful. It includes topics which are not covered in detail in this documentation.

Special Notestc "Special Notes" \l 2§

Previous versions of UUPC/extended also included a file called **simtel.inf** which documented a mail server for retrieving files via FTP at Princeton University. **This server is no longer available, and mail should not be sent to it.** Use listserv@kew.com instead.

Finally, previous versions of UUPC/extended documented an archive server on sun.soe.clarkson.edu. This server has been **replaced** by listserv@kew.com.

Mailing Listtc "Mailing List" \l 2§

A electronic mailing list, uupc-info@kew.com, is open to those interested in UUPC/extended.

[1] Kendra Electronic Wonderworks has no connection with O'Reilly and Associates. However, both of these books are on the bookshelf here.

[2] These are available from your local bookstore or software house, or call O'Reilly and Associates at 1-800-338-NUTS, or send them electronic mail at nuts@ora.com. They can also be ordered from Softpro Books. Call Softpro at (617) 273-2919, FAX them at (617) 273-2499, or send them mail at softpro@world.std.com. Kendra Electronic Wonderworks is a satisfied customer of Softpro, but we have no other connection with them, either.

To join it, send a command of the following format in the body of a e-mail message to listserv@kew.com:

```
subscribe uupc-info
```

The List Server also accepts the command:

```
help
```

To resign from the mailing list, send mail to the List Server of the following format:

```
signoff uupc-info
```

Note: **DO NOT** send requests to be added or deleted to uupc-info@kew.com. Mail sent to this address is automatically forwarded to all users on the list. Questions of an administrative nature which require human attention should be directed to uupc-info-request@kew.com.

Since undeliverable addresses cause mail to be bounced to the list administrator, the administrator reserves the right to drop from the list any user for whom mail is rejected. In many cases, the user will not be notified, since the mail to notify the user will bounce for the same reason the address was deleted in the first place.

Registeringtc "Registering" \l 2§

It is now possible to register UUPC/extended. See **How to Register UUPC/extended**, page 127, for the cost and benefits of registering.

Problemstc "Problems" \l 2§

Please report problems with this version of UUPC/extended via electronic mail to help@kew.com. If you are unable to reach Kendra Electronic Wonderworks via electronic mail, mail your problem report to:

Drew Derbyshire - UUPC/extended Help Desk
Kendra Electronic Wonderworks
Post Office Box 132
Arlington, MA 02174-0002 USA

Please include with your problem description listings of the files which apply, including:

UUPC.RC
[*userid*].RC
SYSTEMS
Directory of your UUPC configuration directory.
A log of the applicable program execution, if any. Both UUCICO and RMAIL automatically write logs into the UUPC/extended spool directory.

Note: Passwords and userids can and should be changed in the SYSTEMS file.

No claim is made that the problem will be corrected, or the person reporting the problem will be supplied a copy of the corrected code[1], but reasonable efforts will be made to correct the program. The Wonderworks does not provide telephone support to unregistered users.

PLEASE read the Nutshell handbooks before sending mail.

Questions and Answerstc "Frequently Asked Questions" \l 2§

The following is a list of general questions and answers about UUPC/extended. For additional questions related to installing UUPC/extended and general trouble shooting, also see the Question and Answer section in **Installation**, page 15.

Q. Does UUPC/extended support news?

A. In part. The RNEWS distributed with UUPC/extended now correctly decompresses and delivers local news to directories based on the news group name, but the news reader (a port of the UNIX program RN) is still in development.[2]

Q. UUPOLL takes over my entire system when it runs. Can I run it under DOS and still use the system?

A. No, but you can run it under MS-Windows 3.x in 386 enhanced mode, under OS/2, or under Windows NT. DESQView should also work. At the Wonderworks, kendra uses OS/2 2.1, and athena uses Windows 3.1.

Note: If you run UUPC/extended in the background, be sure to enable multitasking support by including "options = multitask" in your UUPC.RC file.

Q. It takes forever for the system to recognize a busy signal on the other end. What can I do?

A. Make sure your modem file includes the **NoConnect** variable to document the strings your modem uses to report failures in the dialing sequence. Consider enabling the option **CarrierDetect** in your modem file.

Q. Can UUPC/extended support multiple users on one system?

A. Yes. Each user should have an entry in the PASSWD file and a copy of [*userid*].RC under a unique name in the UUPC/extended configuration directory. A short command file similar to the sample SU.BAT file provided can be used to change the active user. UUPC/extended cannot prevent users from reading other users' mail or mail queued for another system. This is an MS-DOS restriction.

Q. UUPC/extended is wonderful, but it doesn't run on my (insert non-IBM or NT-compatible platform here).

A. While the original UUPC was targeted for both the Mac and Atari-ST in addition to the IBM, Kendra Electronic Wonderworks only supports the MS-DOS, Windows 3.x, OS/2, and Windows NT environments. Contact Dave Platt <dplatt@snulbug.mtview.ca.us>, on where to find his version of UUPC for the Macintosh. For Amiga systems, try anonymous FTP to wuarchive.wust1.edu. There are no known reliable sources for UUPC for the Atari-ST.

Q. Who is kendra?

[1] Registered users will be provided an update when a fix becomes available if they explicitly request it. See Error: Reference source not found § for details.

[2] The news support was originally promised for the end of 1991. We admit it, the schedule got blown.

A. Actually, that is correctly "What is kendra?" Kendra means "womanly knowledge" in Old English.

kendra originally was an 80286 based Epson Equity III+ running MS-DOS 5.0 and using a second 8088 system as a communications front end. During May of 1992, she was upgraded to a Northgate 386/40 running OS/2 2.1. Her sibling, our documentation system athena, is a Northgate 386/33.

Q. I want users to be able to login and read their mail on my system. Since I can't use COMMAND.COM, what shell should I specify for them?

A. Right idea, wrong question. UUPC/extended is not designed for remote user login. Rather, it is designed to allow routing mail among multiple peer sites, so set up each user as his/her own node. This gives each user all the tools of his/her own system when composing mail, and UUPC/extended keeps the time spent on-line to a minimum.

Q. Why does Snuffles want Chocolate?

A. Because Chocolate is happy food. Snuffles is a very happy bear. However, ever since someone sent her 18 pounds of chocolate, she now looks for donations to the Chocolate Ice Cream Fund instead. This leaves us more room in the freezer for more mundane groceries.

Note: Snuffles does still accept[1] Girl Scout Thin Mints.

[1] I LOVE Thin Mints!!!--SPB

Installationtc "Installation"§

New UUPC/extended installationstc "New Installations" \l 2§

Ingredientstc "Ingredients" \l 3§

The following assumes that you have never installed UUPC/extended before. If you are upgrading an existing UUPC/extended installation, skip to **Upgrading an existing UUPC/extended installation,** beginning on page 29.

> **Note:** If installing on Windows 3.x, first install the native DOS version and then add the Windows versions of the UUCICO and associated programs. This both allows debugging in a cleaner environment and provides the needed DOS foundation for running under Windows. See additional information on Windows 3.x on page 111.

Many thanks to David Watt and his sidekick Frederick Bear Watt for assisting with this section and providing the examples, and to Snuffles for reminding us to change the system names to protect the guilty.

To start sending and receiving mail on your system you'll need the following equipment and knowledge:

Equipment:

* A computer running MS-DOS, OS/2, or Windows NT. Under DOS, you need at least 512 KB of RAM and 2 megabytes of hard disk space (for executables, configuration files, documentation, and mail received). For OS/2, Windows 3.x, or Windows NT you need the minimum system configuration plus 2 megabytes of free hard disk space.
* A modem.
* The manual for the modem.
* A text editor. The MS-DOS editor, Windows Notepad, or OS/2 editor will work fine.
* A friend with a machine that does UUCP. This can be a UNIX system, another UUPC/extended or other UUCP clone running on a PC, or a VMS system running DECUS UUCP.
* A copy of PKUNZIP.EXE, version 2.04g or later, or a compatible utility. PKUNZIP is a shareware utility and can be found at almost all archive sites. Consult your local BBS, or get it from wuarchive.wustl.edu.

Knowledge:

* How to copy files.
* How to edit a text file.

Once you've got all of that, follow these steps to prepare for, install, configure, and test UUPC/extended.

Preparing for installation:

1. Review **General Information**, page 7, to make sure UUPC/extended is right

for you.

2. Read these instructions.
3. Name your system.
4. Find a UUCP neighbor.
5. Choose your user name.
6. Backup your system.

Installing UUPC/extended on your system:

1. Get the UUPC/extended archives needed for your operating system
2. Make any required directories
3. Copy the UUPC/extended files onto your system.
4. Create the on-line help files for MAIL.

Configuring UUPC/extended process mail and call other systems:

1. Find your modem among the .MDM files.
2. Configure several files:
    1. UUPC.RC
    2. *[userid]*.RC
    3. *[modem]*.MDM
    4. SYSTEMS
    5. PASSWD
    6. PERMISSN
    7. CONFIG.SYS
    8. AUTOEXEC.BAT
    9. *[userid]*.SIG
3. Reboot your system.

Testing the new installation:

1. Check your configuration via UUNAME.
2. Check your PERMISSNS file via UUXQT.
3. Send mail to yourself.
4. Check the PC <--> modem connection
5. Check the PC <--> modem <--> modem <--> mail server connection
6. Send remote UUCP mail.

Then you'll be up and running! Each of the steps is discussed in more detail below.

Before you begin installingtc "Before you begin" \l 3§

1. **General Information**, page 7, contains important information explaining what UUPC/extended will and will not do. It also discusses the responsibilities inherent in running a UUCP site. Please read it before proceeding further.

2. Read all of these instructions before doing anything. Trust us, you'll be glad you did.

3. Your machine needs a name. The name should be all lowercase, since many operating systems are not case-sensitive. If you plan to connect to the outside world, the first six characters of the name need to be unique over all of Usenet. This isn't precisely true, but it's close enough, and the truth is more complicated -- if you're interested in the details, buy a copy of *Managing UUCP and Usenet*, published by O'Reilly and Associates. Actually, go buy it anyway, and please read it before asking questions. **General Information**, page 7, has some suggestions on where to find this and other useful books. See **RFC1178.TXT** for

PA
some considerations to keep in mind when choosing a system name.

Example: Fred chose the name "toscis" for his machine, in honor of Snuffles' favorite ice cream.

4. Find a friend who has a machine which supports sending and receiving UUCP news and mail. Your friend must also be pals with the system administrator of the machine, or someone else who has the power to add accounts to the remote system.[1] They will have to add an account for your system and configure the remote UUCP to acknowledge its existence and permit your system to log in and exchange mail.

If you're looking for a UUCP system to connect your machine to, you might try local universities and local software companies for friendly volunteers. The connection at the Wonderworks, for instance, is provided courtesy of a machine belonging to the EE department at a local university. If you have Usenet access, you might peruse comp.mail.maps for the names of local systems and their administrators who might be willing to give you a feed.

If you are willing to pay money, there are a variety of commercial organizations willing to take it from you. One to try is Performance Systems International. Send mail to info@uu.psi.com.

Example: Fred uses a feed from Kendra Electronic Wonderworks, which is known in the UUCP world as kewgate. Snuffles assigned toscis an account on kewgate called "Utoscis". The name is conventional -- Ukewgate, Uflopsie, and Uzzyzyx are several accounts on Fred's machine, for some of his neighbors to log onto his system.

5. You need to choose a user name for yourself. This name does not have to be unique or anything, but you will need it for some of the configuration of UUPC/extended yet to come. It should be composed only of valid DOS file name characters and should have eight characters or less.

Example:Fred's user name on toscis is fbwatt. Our resident Plush Platinum Programming Polar Bear's[2] user name on kewgate is snuffles.

6. Backup your system. If you do not regularly backup your system, this is an excellent time to start. Again, trust us.

Copying UUPC/extended files onto your system.tc "Copying files" \l 3§

1. To install UUPC/extended, you need to get the specific archives needed for your system, as outlined in the table below.

UUPC/extended includes nearly two dozen executable files for each operating system. The files should be placed in their own directory to allow easy upgrades. This directory should then

[1] Specifically, someone has to have the power to add **legal** accounts to the remote system and to configure their mailer to recognize you. This is not the time for knowing people in low places.

[2] And I'm cute, too!! [S.P.B.]

be added to your MS-DOS, OS/2, or Windows NT PATH variable. The program will assume that you have used the following standard directories for the binaries:

Operating System	Standard directory name for executable files.	Archives Need To Install [1]
DOS	\UUPC\BIN	upc12bd1.zip upc12bd2.zip upc12bd3.zip upc12bad.zip
Windows 3.x	\UUPC\WINBIN	upc12bw1.zip upc12bw2.zip upc12bw3.zip upc12bad.zip
OS/2 16 bit (for OS/2 1.x)	\UUPC\OS2BIN	upc12b11.zip upc12b12.zip upc12b13.zip upc12bad.zip
OS/2 32 bit (for OS/2 2.x)	\UUPC\OS2BIN	upc12b21.zip upc12b22.zip upc12b23.zip upc12bad.zip
Windows NT	\UUPC\NTBIN	upc12bn1.zip upc12bn2.zip upc12bn3.zip upc12bad.zip

2. Make directories for the UUPC/extended files. You need to make the base directory (\UUPC)[2], a directory for the sample files (\UUPC\SAMPLES), if you have not unpacked them elsewhere already, a directory for the binary files, and a public directory (\UUPC\PUBLIC)[3].

[1] The archives upc12a§d3.zip, upc12a§w3.zip, upc12a§13.zip, upc12a§23.zip, and/or upc12a§n3.zip can be omitted if you do not require the ability to receive and expire news from remote systems.

[2] You can actually use different directory names for the configuration and other directories through out this installation procedure, and not all the directories even need to be on same hard drive. However, this makes installation more complicated, and you can change it later after it works.

[3] The public directory is where your UUCP neighbors would send you files via the UUCP command. The directory must exist because UUCICO and UUXQT's initialization of internal

PA

*Example:*kewgate has both OS/2 (16 or 32 bit) and DOS versions of UUPC/extended installed. The DOS files are in E:\UUPC\BIN and the OS/2 files are in E:\UUPC\OS2BIN.

Example: Toscis runs the MS-DOS and Windows NT operating systems. Fred put the UUPC/extended DOS executables in C:\UUPC\BIN, the Windows NT executables in C:\UUPC\NTBIN and the documentation in C:\UUPC\SAMPLES. He built the directories at the C:\> prompt this way:

```
C:\> mkdir \uupc
C:\> mkdir \uupc\bin
C:\> mkdir \uupc\ntbin
C:\> mkdir \uupc\samples
C:\> mkdir \uupc\public
```

3. Unzip the archives. (Type "pkunzip" at the DOS prompt for instructions on how to unzip files.)
Example: To uncompress the documentation archive and program archives for MS-DOS and Windows NT, here's what Fred did:

```
C:\> pkunzip upc12bad.zip \uupc\samples *.*

C:\> pkunzip upc12bd1.zip \uupc\bin *.*
C:\> pkunzip upc12bd2.zip \uupc\bin *.*
C:\> pkunzip upc12bd3.zip \uupc\bin *.*
C:\> pkunzip upc12bn1.zip \uupc\ntbin *.*
C:\> pkunzip upc12bn2.zip \uupc\ntbin *.*
C:\> pkunzip upc12bn3.zip \uupc\ntbin *.*
```

4. Convert the printed formatted documentation for mail to on-line help files. This requires running the program NOVRSTRK against the MAIL.PRN and TILDE.PRN.
*Example:*Fred ran NOVRSTRK on both of the files:

```
C:\> \uupc\bin\novrstrk \uupc\samples\mail.prn \uupc\mail.hlp
C:\> \uupc\bin\novrstrk \uupc\samples\tilde.prn \uupc\tidle.hlp
```

All the files you need are now on your system.

Configuring UUPC/extended after installationtc "Configuring UUPC/extended" \l 3§

Here's where you earn your daily chocolate by configuring the system for basic local mail and dialing out to other systems.[1] The chart below summarize the files to be updated and their functions. All of the files must copied from the samples directory (\UUPC\SAMPLES) to their permanent happy home in the UUPC/extended configuration directory (\UUPC) except as noted.

security tables will otherwise print warning messages.

[1] UUPC/extended§ can also be configured to accept incoming calls, to process Usenet news, and to tap dance, but we're trying not to confuse the issue here. All large software systems that work start from small systems that work. See Error: Reference source not found §for further information on more complex configurations.

File name	Function
UUPC.RC	System UUPC/extended configuration file, it defines both required and optional information which is the same for all users of the local system, and default values for individual users. This includes the local system name, the name of its mail server, directory names (if you are not using the defaults), and selected performance parameters.
[userid].RC	User UUPC/extended configuration file, it defines both required and optional information for one user. A customized copy of this file exists under a unique name for each user of the system. This file is sometimes referred as the PERSONAL.RC file.
[modem].MDM	Defines the command strings used to initialize, dial, and answer with your modem. It also contains selected protocol information for UUCP connections made using the modem. The file is referenced by entries in the SYSTEMS file. **Note:** You may need more than one modem file if you use your modem to connect to more than one type of remote modem.
SYSTEMS	This file defines the system names, times to call, device (modem file) to call with, phone number, and login script to use when calling other systems.
PASSWD	This file defines the user ids, names, and home directories of the users of your system.
PERMISSN	This file defines the access remote systems have to your disk and what programs a remote systems may direct UUXQT to run on its behalf.
AUTOEXEC.BAT	Under DOS and Windows NT, this sets environment variables for your PATH (where to find programs), the location of the UUPC.RC file, the location of the *[userid]*.RC file, and the local time zone. **Note:** This file resides in the root directory of your hard drive.
CONFIG.SYS	Under DOS, this file is used to define how many files can be open at once; under OS/2, this is used in place of AUTOEXEC.BAT to define the environment information listed above. **Note:** This file resides in the root directory of your hard drive.

PA

[userid].SIG	This file is included automatically at the end of each piece of mail you send, providing return address information. **Note:** This file resides in your home directory, defined by the variable **Home** in your [*userid*].RC file. See step **2**, page 21, for further information.

1. The first file to copy and edit is UUPC.RC, the UUPC/extended system configuration file. This file defines the system name and associated parameters which make your system unique.

The fields are described in the sample UUPC.RC file itself, and additional optional fields are documented in a second sample file, UUPCBIG.RC.[1] Their uses are summarized below, along with the settings for toscis. You must set the fields described below. All other fields in the sample UUPC.RC file, such as the **options=** lines, can be left alone.

Field name	Description	Example
Nodename	The simple nodename you chose for your machine.	toscis
Domain	The fully qualified domain name for your machine. If you are just starting out, this will be your machine name, with the suffix ".UUCP".	toscis.cambridge.ma.us[2]
Mailserv	This is the name of the machine to which your outbound mail gets sent. This will be the name of the machine belonging to the friend you found up in step 2.	kewgate

Here is toscis's completed UUPC.RC file, provided as an example.

```
Nodename=toscis
Domain=toscis.cambridge.ma.us
postmaster=fbwatt
Mailserv=kewgate
MailExt=SPB
InModem=TB2500
options=nobang nodirectory multiqueue dot multitask
```

[1] A third sample configuration file, UUPC.KEW, is the live configuration file for kendra.kew.com.

[2] Fred registered his domain with the .US domain mapping project, which is discussed in Error: Reference source not found.

Note: toscis has some additional fields defined in its UUPC.RC. We left them in this example to remind you that UUPC/extended configuration can be made more complicated if you choose to.

2. The second file to edit is the *[userid]*.RC file, defining personal configuration options.

The *[userid]*.RC files are used to configure UUPC/extended for individual users. To configure your own *[userid]*.RC file, copy the file \UUPC\SAMPLES\PERSONAL.RC to \UUPC\ *[userid]*.RC, where [*userid*] is your user name.

Example: Fred's personal configuration file is called FBWATT.RC, and is located in C:/UUPC on toscis.

Edit the following fields in the *[userid]*.RC file:

Field name	Description	Example
Mailbox	Your user id.	Mailbox=fbwatt
Name	Your real name	Name=Frederick B. Watt
FileSent	Name of a file which all outgoing mail is to be logged.	FileSent=Outgoing
Signature	Name of a file which will be appended to each message sent.	Signature=fbwatt.sig
Home	The default directory UUPC/extended uses for files (including the FileSent and Signature Files) for your user id.	Home=C:\fbwatt

The descriptions in the sample PERSONAL.RC file explain these fields. If you include a Signature (*highly* recommended) and an Alias file, you will need to create these files as well. See step **9** on page 26 of this section for information on creating a signature file. See **The User Alias File**, page 104, for a discussion of alias files.

If multiple users are to share the same system, create multiple customized copies of the *[userid]*.RC in \UUPC, each with a unique name such as "TOM.RC", "DICK.RC" or "HARRY.RC".

Example: Fred's completed FBWATT.RC file looks like this:

```
PA
    editor=emacs %s
    FileSent=Outgoing
    Home=c:/fbwatt
    Mailbox=fbwatt
    Name=Frederick B. Watt
    options=autoedit
    Signature=fbwatt.sig
```

3. You must next decide which modem configuration file (*[modem]*.MDM file) best describes your modem, or write one. See MODEM.TXT for a complete list of supplied modem files.

Copy the modem file for your modem from the \UUPC\SAMPLES directory to the \UUPC directory.

> **Note:** If none of the supplied samples is appropriate, then you will have to start with one of them, or with the sample modem configuration file (SAMPLE.MDM), and customize it for your modem. We explain how to customize modem files in **Modem ([modem].MDM) Files**, page 87, and in **The Fine Art of Chat Scripts**, on page 29.

Once you copy the *[modem]*.MDM file, replace the port listed on the "Device=" with the communications port to which you connected your modem.

> **Note:** If using a modem on DOS and your modem is on port COM3 or COM4, you may need to run the COMM34 utility. Refer to the description of COMM34 in **UUPC/extended Command Reference**, page 36, for details.

4. You must copy and edit the sample SYSTEMS file, which defines the system names and when to call out to them. The file is described in detail in **The UUPC/extended SYSTEMS file**, page 96. It must be copied from \UUPC\SAMPLES directory to the \UUPC directory. Delete all the entries you do not need.

A typical line in the SYSTEMS file contains these fields, defined in more detail by the table:

```
system Time MODEM speed telephone protocol script
```

Field name	Description	Example
system	Name of the system to call. Any system you call or are called by must be listed at least once in this file. You may include the same system more than once if you have multiple phone numbers for it.	kewgate
Time	When to call. "Any" is what it says, "Night" and "Evening" refer to night and evening phone rates[1] respectively. "Never"	Any

[1] Specifically, the AT&T night rate is in effect from 11 PM to 8 AM every day, and from 11 PM Friday to Sunday at 5 PM. The UUCICO definition for Evening includes the times allowed for

	is used for a system which only calls you. **Note:** See the Nutshell handbooks for a further explanation.	
MODEM	Name of the modem file (without the .MDM extension) used to call this system.	TB2500
Telephone	Telephone number to call	1-617-641-4817
p (protocol)	Protocol to use when calling. If you don't know what these are, then leave the "g" alone if your mail server does not run UUPC/extended 1.11x or above, or change it to "v" if it does. The "g" protocol is widely supported, "v" is faster but only supported by newer UUPC/extended systems.	g
script	Login script for the system.	gin:--gin: Utoscis ssword:--ssword: AppleJuice

Example: Toscis's SYSTEMS file entry for kewgate looks like this:

```
kewgate Any TB2500 19200 1-617-641-4817 g gin:--gin: Utoscis ssword:--
ssword: AppleJuice
```

Copy this line into your SYSTEMS file, and do the following:

* Replace "kewgate" with the name of your mail server.
* Replace "Any" with the time to call.
* Replace "TB2500" with the name of the modem configuration file (.MDM file) you are using.
* Replace "1-617-641-4817" with the phone number of the mail server you arranged a feed from in step **4** of **Before you begin installing**, page 17.
* Replace "g" with the letter(s) of the UUCP protocols your mail server supports.
* Replace "Utoscis" with the name your friend in step **4** of **Before you begin installing**, page 17, gave you for use on their system. Replace "AppleJuice" with the password your friend gave you.

For simple connections, these should be all of the changes you need to make to this file. For more complex connections, refer to **The Fine Art of Chat Scripts**, page 29, below.

5. You must also create a \UUPC\PASSWD file, which defines the local users. This file is

Night plus 5 PM to 11 PM everyday.

PA
nearly identical in format to the /etc/passwd file used on UNIX systems.

The PASSWD file serves two purposes. If you have several people reading mail on your system, it is used to assign them default directories and mail files. Entries in the PASSWD file also control whether remote machines can log in to your system to send and receive mail themselves.

Note: The home directories and user names in the PASSWD file must match the same entries in the *[userid]*.RC files.

Copy the PASSWD file to the \UUPC directory, and then replace the sample users with your local users. This file defines the name and home directory of each local user, and the user id and password for remote users (systems) logging in using the following format:

mailbox:password:*special*:group:name:home:shell
More specifically:

Field name	Description	Examples
mailbox	User id to receive mail locally or login remotely. This should be from 1 to 8 valid DOS characters. For local users, this should be the same as the mailbox field(s) of the [*userid*].RC file(s).	fbwatt postmast
password	Ye ole security field. A * in this field means that no remote logins are allowed for that user. For instance, a local user is not able to login via phone lines.	* AppleJuice
special	This field contains the user id number on UNIX systems. UUPC/extended uses it to play music or otherwise make noise to announce incoming mail from remote systems if desired. The format of the field is tone1, length1, tone2, length2, and so on, where "tone" is the frequency in hertz and "length" is the time the tone is to sound in milliseconds. If the final length is omitted, the final tone will sound for 500 milliseconds. A special tone is 0, for silence.	*(empty)* 2000,250,1000,250
group	This field is the group id on UNIX systems, but is not used by UUPC/extended.	*(empty)*
name	The real human name of the user. For local users, this field is included in mail headers if no other name for the user is known.	Frederick B. Watt

home	Home directory for this user. Used only for local users. Default is /UUPC/PUBLIC, which is a lousy choice. **Note:** This field should match the **Home=** field for the user in his *[userid]*.RC file.	/u/fbwatt
shell	Login shell for remote users. The default is to perform normal UUCP processing. **Note:** Use of shells other than UUCICO is beyond the scope of this document.	*(empty)*

At first, you should need only two lines in your PASSWD file. The following example can serve as models.

Example: tosci's PASSWD file looks like this, with entries for Fred himself and the postmaster:

```
fbwatt:*:::Frederick Watt:c:/u/fbwatt
postmast:*:::Postmaster:c:/u/postmast
```

Replace fbwatt with your user name, and "Frederick Watt" with your name. Replace "c:/u/fbwatt" with the Home directory given in your *[userid]*.RC file. Leave the postmaster entry in the file (you can choose a new home directory), it exists to catch any failed mail.

6. The \UUPC\PERMISSN file is identical in format to a file called Permissions available on some UNIX UUCP systems. The file format is explained in detail in *Managing UUCP and Usenet*.

The PERMISSN file is used to control where remote systems are permitted to read and write files on your system. For a simple configuration where you only call a mailserver and no machines call your system, your PERMISSN file should look like this:

```
MACHINE=kewgate SENDFILES=yes REQUEST=yes
```

Replace "kewgate" with the name of your mail server, managed by your friend in step **4** on page 17.

For more complex configurations refer to the section **Permissions (PERMISSN) Files**, on page 31.

7. Under DOS, you need only update CONFIG.SYS if it does not exist, exists but does not have a FILES= line in it, or the FILES= is less than 20. In any of these cases, add or replace the following line to your CONFIG.SYS, or create a CONFIG.SYS in the root directory of your boot disk or diskette if you currently do not have one:

```
FILES=20
```

Note: If you already have a FILES= line in your CONFIG.SYS, it should read at least 10 and should be raised to 20. If the number is greater than 20, then do not lower it.

PA
If you are using OS/2, then you must update CONFIG.SYS in the same way you would update AUTOEXEC.BAT under DOS. Under Windows NT, you don't need to modify CONFIG.SYS at all.

8. AUTOEXEC.BAT is executed by MS-DOS whenever you reboot your system and by Windows NT at login. UUPC/extended needs three environment variables set, which is best done by adding some commands to your AUTOEXEC.BAT file. At the bottom of your AUTOEXEC.BAT file, add the following lines. If you are running OS/2, add the same lines to your CONFIG.SYS file.

```
set UUPCSYSRC=c:\uupc\uupc.rc
     set UUPCUSRRC=[userid].RC
```

But instead of writing [*userid*] on the second line, use your user name, the same as in the *[userid]*.RC file you wrote above. You'll also need to set TZ, your local time zone. Sample TZ variable SET commands are as follows:

SET TZ=CST6CDT	Central Standard Time, 6 hours behind GMT, with daylight savings
SET TZ=EST5EDT	Eastern Standard Time, 5 hours behind GMT, with daylight savings
SET TZ=EST5	Eastern Standard Time, but no daylight savings change
SET TZ=GMT0	Greenwich Mean Time, no offset, no daylight savings
SET TZ=MST-3	Moscow Standard Time, 3 hours ahead of GMT, no daylight savings

SET TZ=MST7MDT	US Mountain Standard Time, 7 hours behind GMT, with daylight savings
SET TZ=PST8PDT	US Pacific Standard Time, 8 hours behind GMT

Your mileage and time zone may vary. Note that time zones ahead of GMT use a minus sign in front of the hours offset.

You'll also need to edit your path variable. Fred, who put his UUPC/extended files in C:\UUPC\BIN, added the line:

```
PATH=%path%;C:\UUPC\BIN
```

at the bottom of the file. In DOS, you must make sure the path does not exceed 128 characters. Under OS/2, the file you need to edit is CONFIG.SYS. There is no path size limit under OS/2, but the path must be on one line and cannot use variables. Under OS/2, Fred would simply append ";C:\UUPC\OS2BIN" to his existing path.

> **Note:** Several of the programs, including UUCICO and UUXQT, change directories as required to search for spool files. Because of this, programs such as RMAIL and (if used) RNEWS must be in your path. Having them in the current directory is not sufficient.

9. For each user, customize the *[userid]*.SIG file, and copy it to the 'Home' directory defined in *[userid]*.RC for that user. This file is appended to all outgoing mail for the user to provide a boiler-plate signature with the user's reply address. If this file is not installed, then the line in the *[userid]*.RC file describing it must be commented out with a pound sign (#) in column 1.

Note: Especially for users of new systems, it is strongly advised that you do create a signature file and include in it your name, electronic mail address, snail mail address, and telephone number.   This extra information is needed for a person to contact you if your electronic mail address cannot be replied to.[1]

Example: Here's a typical (boring) signature:

Fred Watt -- N1HMB   Home: 617-555-9956

[1] Recently, help@kew.com was sent the same query multiple times by a user with an invalid return address. He failed to include his telephone number or other address in his note. Thus, neither his original query nor the follow-up letters asking why the original query was not answered could be replied to.

PA
fbwatt@toscis.cambridge.ma.us Work: 617-555-4330

10. Reboot your system. This allows your changes to your AUTOEXEC.BAT/CONFIG.SYS to take effect.

Congratulations! If you've done all of these things, you're ready to start using your UUPC/extended configuration.

Testing

1. Begin testing by running the UUNAME program.

Just type uuname at a system prompt to prove your configuration is valid. This will a) die gloriously or b) display a list of known systems (the remote systems you listed in your SYSTEMS file). If it dies, the error message you get should provide some clues. Check the necessary configuration files and try again.

2. Run UUXQT to check the PERMISSNS file information.

Type uuxqt at a system prompt. This will a) die or b) print a copyright message and exit. If it dies, there is a problem with your PERMISSNS file and the error message you get should provide some clues. See **Permissions (PERMISSN) Files**, page 31, for more information about PERMISSNS files.

3. Send mail to yourself.

Perform this test to convince yourself that UUPC/extended can send mail. At the system prompt, type:

```
C:\> mail -s "Test message" postmaster
   ? Test
   ? .
   Abort, Continue, Edit, List, or Send? Send
```

Note: The "." only works as a message terminator if the "dot" option is set in your UUPC.RC file. Otherwise, you have to use Ctrl-Z.

You should then receive the message:

```
    Delivering mail from [username] to postmaster
```

If you don't, the error messages should provide some clues as to why UUPC/extended was unable to deliver the message. Correct the necessary configuration files and try again.

3. Test the PC <-> modem connection.

Perform this test to convince yourself that your modem responds to commands. Using a telecommunications program such as Kermit, send some commands to the modem. Just sending AT (attention) to the modem, and getting "OK" back,[1] is good enough.

[1] This assumes your modem is a Hayes Smartmodem compatible. If it's not, you already knew this since you long ago discovered numerous programs only work with Hayes modems and clones, and know better than to type AT to wake up your modem.

4. Test the PC <-> modem <-> modem <-> mail server connection.

Perform this test to convince yourself that your computer can talk to your mail server, and that you can log in to the mail server using the user name and password your friend back in "**Before you begin installing**", step **4**, on page 17, gave you. Using a telecommunications program like Kermit, dial up the remote system and try to log in. The remote system should give you a message like "Shere=[*mailserv*]". If you succeed, hang up[1] and go on to the final step. If the remote system does not recognize your account for some reason, talk to the friend who gave you the account and find out why.

5. Try sending remote UUCP mail.

If you succeed at this, your system is able to properly receive and send electronic mail. At the system prompt, type the following:

```
C:\> mail -s "Test message" postmaster@[mailserv].UUCP
? This is my first message.  When you receive it, please reply.
? .
Abort, Continue, Edit, List, or Send? Send
Delivering mail from [userid] to postmaster via [mailserv]
```

but instead of typing "[*mailserv*]," on the first line, substitute the name of your mail server, exactly as you entered it on the "MailServ=" line in your UUPC.RC file.

Note: For full instructions on using the mail program, see the description of MAIL in **UUPC/extended Command Reference**, page 40.

If all goes well, you should be able to enter a message and send it off, with the UUPC/extended mail program giving you messages like those above.

Then, dial out to the remote system:

```
C:\> uuio -s all
```

And watch the connection. With luck, you will have just sent your first UUCP mail message. Welcome!

Upgrading an existing UUPC/extended installationtc "Upgrading UUPC/extended" \l 2§

Follow these steps to upgrade.

1. If upgrading from a release of UUPC/extended prior to 1.11a, you must clear your spool

[1] If you want to drive the Sysadmin nuts, type (with no spaces):

```
Ctrl-P S snuffles Ctrl-Shift-@ Enter
```

which will tell the remote system your system is snuffles (and make it hang up since it doesn't know you). Otherwise, just make the modem hangup via a terminal emulator command or the Big White Switch, since the remote system won't let you logout without answering the prompt.

PA

directories of files before installing the new release. In general, this is done by invoking UUIO for the previous release to deliver any queued files to other sites.

2. If upgrading from MS-DOS to Windows 3.x, OS/2, or Windows NT, create new directories for the new operating environment binaries.

3. Uncompress the new executables and documents. Be careful not to overwrite your customized configuration and batch files with the supplied sample files. See page 18 for the list of files to uncompress for your operating system.

4. Read **Changes From Previous Versions**, page 133, for changes related to the newest release. In many cases, **Changes From Previous Versions** includes documentation of new options and configuration file variables before any other documents.

5. Earlier versions of UUPC/extended have different user-customizable variables. Type "uuname -x 2" to see a list of missing or obsolete variables in your configuration files.[1] Update as needed. Look at **UUPC/extended Configuration Files**, page 133, to see what the new variables do.

6. If you are upgrading from a release of UUPC/extended previous to 1.10a, delete the old communications module, UUIO.EXE; this has been replaced by UUIO.BAT, UUCICO.EXE and UUXQT.EXE.

7. If you are converting from a release previous to 1.10a, you must update the SYSTEMS file to the newer format. You must also write a PERMISSN file. See the sample files for details.

8. If you are upgrading from a release of UUPC/extended prior to 1.11s, delete the old version of UUPOLL, UUPOLL.COM. This has been replaced by UUPOLL.EXE.

9. If you are upgrading from a release of UUPC/extended prior to 1.11p, delete the old version of UUXQT, UUXQT.COM. This has been replaced by UUXQT.EXE.

10. If upgrading from a release previous 1.12a, see the new section Modem (*[modem].MDM)* Files beginning on page 1 to determine if you should update your modem file(s).

11. If running Windows on top of DOS, you may optionally install the Windows version of UUPOLL, and UUCICO and UUPOLL. See 111 for additional Windows information.

12. Based on **Changes From Previous Versions**, page 133, and other documentation, update your configuration to exploit any new features of UUPC/extended you find useful. In particular, if using the OS/2 version of UUPC/extended, consider setting the **undelete** option in your UUPC.RC file.

The Fine Art of Chat Scriptstc "The Fine Art of Chat Scripts" \l 2§

Setting up the chat scripts used in the SYSTEMS and modem [*.MDM] files is the hardest part of setting up UUPC/extended. The easiest way to think of chat scripts is as a dialogue between

[1] We know 1.11v didn't work with this option. We fixed this, honest.

the local system running UUPC/extended and a modem or remote system. The script consists of a series of strings sent to the modem/remote system alternating with the expected response to those strings.

For example, a modem file needs to contain an initialization sequence, which will look something like this:

```
    "" "" "" \pATZ OK \dATX4&D2 OK
```

Since that line looks awfully intimidating at first, we'll break it down into its component parts.

"" An empty string. The first one is a flag marking the start of the script and showing that nothing is expected from the modem.

"" The second empty string sends nothing followed by an automatically appended carriage return so that the following statements will start on a new line.

"" The third empty string indicates that no response to the second one is expected.

\p Wait 0.4 seconds to allow the modem to catch up.

ATZ This command resets a Hayes-compatible modem to its stored profile. Again the carriage return is appended.

OK The response from the modem showing that it has been reset.

\d An escape sequence which produces a two second delay to give the modem time to recover from the previous input.

ATX4&D2 A modem initialization string. Contents of this string depend on your modem and desired options. As always, a carriage return is appended.

OK The modem responds that it has been initialized and is ready to dial.

The complete list of escape sequences supported by UUPC/extended follows. The empty string ("") is used on input. All other strings are recognized on output only. Quotation marks may be used on input or output to enclose a string which includes whitespace. The list includes:

""	Expect a NULL string. May also enclose a string which includes whitespace.
BREAK	Send BREAK
\b \B	Insert Backspace

PA

\d \D	Delay two seconds.
\c \C	Omit carriage return at end of string
\m \M \r \R	Insert carriage return
\n	Insert newline
\p \P	Delay four tenths of a second
\t \T	Insert tab character
\s \S	Insert space character
\Znnnn	Set serial port speed to nnnn in the range 300 to 38400[1]
\nnn	Convert the numeric octal string nnn to a single character and transmit it
\\	Insert a backslash (\)

[1] the maximum port speed is 57600 under Windows NT and 32 bit OS/2.

One other special syntax exists. To allow for alternate output when an expected string is not received, hyphens in an expect string delimit alternate output to be sent, such as:

```
"" "" ogin:--ogin:--ogin:
```

If a login prompt (ogin:) is not received by the standard time-out period (generally, 30 seconds, but configurable in the modem file), the string between the hyphens (nothing, followed by the appended carriage return) is sent; if no response is received, the sequence is repeated one more time.

If your chat script doesn't work, check and make sure that you have allowed delays between transmissions to allow the modem to recover. Failure to include needed delays can cause dropped characters and is one of the most common mistakes made by new chat script writers.

Permissions (PERMISSN) Filestc "Permissions Files" \l 2§

Each system you contact must be defined in a PERMISSN file located in the UUPC/extended configuration directory.   In the permissions file, systems can be classified by whether they are called out to, dial in to the local system, or both. Permissions files define which directories and commands are accessible to the remote system. They are discussed in more detail in *Managing UUCP and USENET.*

For your own protection, think carefully before giving a remote system read or write access outside of the default (\uupc\public, \uupc\spool) directories. Think *very* carefully before giving write access to anonymous logins.

Systems you call out to must have a MACHINE entry, such as

```
MACHINE=pandora
```

> This defines the existence of the machine "pandora", and grants it default permissions. These permissions include the ability to execute RMAIL and RNEWS, and the ability to send files to the spool directory. If the additional keyword REQUEST=yes is added to the MACHINE statement, then the remote system may also read and write files in your public directory, which defaults to \uupc\public or can be redefined by the variable **PubDir** in the UUPC.RC file.

Systems which dial into you must have a LOGNAME entry, such as:

```
LOGNAME=userid VALIDATE=system SENDFILES=YES
```

This allows host "system" to login as user id "userid" with the same default permissions as described above. It further allows your system to transmit files to the other system even though it called you (SENDFILES=YES)[1]. However, to run UUXQT they must also have a MACHINE entry, because UUXQT does not look at LOGIN entries.[2] Thus, to handle the general case of a system calling you to deliver mail, both of the above entries must exist in the permissions file,

[1] The default, carried over from UNIX, is that files are only transmitted when you are the calling system. This means that SENDFILES=YES is not needed on a MACHINE statement.

PA
but they can be combined:

```
     LOGNAME=userid VALIDATE=system MACHINE=system SENDFILES=YES
```

The remote system can be granted access to additional directories and programs through the use of additional parameters on the LOGNAME and MACHINE statements; these are documented in *Managing UUCP and Usenet.*

If you allow anonymous logins, an entry should also be placed in the PERMISSN file for the system "*anonymous". Such an entry might look like this:

```
     LOGNAME=nuucp MACHINE=*anonymous REQUEST=yes READ=~/ NOWRITE=~/
```

This allows anonymous UUCP systems to login as nuucp (provided nuucp has an entry in your PASSWD file), and to read but not write to your **PubDir.**

> **Note:** One difference between the PERMISSIONS file as defined in the Nutshell Handbooks and as implemented in UUPC/extended is that only one user id may be specified per LOGNAME entry in UUPC/extended. This restriction is a security feature to prevent one system from logging in as another.

PERMISSN files can be difficult to set up and use, so don't be alarmed if you have some trouble at first.

When Things Go Wrongtc "When Things Go Wrong" \l 2§

If you have difficulty getting a connection to your mail feed working the first time, don't worry. Almost everyone has to spend a bit of time ironing out the crinkles in a new connection. You may wind up spending time on the phone with your mail server system operator trying to get things to work. Don't worry.[1] UUCP is just like that sometimes. There are several places in UUPC/extended where problems can crop up.

Here is a list of some symptoms you might see, and where to look for solutions:

Q. UUCP on your remote host keeps sending you messages about RNEWS exiting normally, and you want the messages to go away.
A. The sending host should set the -n flag for the UUX command which generates the RNEWS command for you.
Q. A mail message created in the editor is sent, but the signature is not appended.
A. Your editor is appending a Ctrl-Z to the file before the signature is appended. Disable that "feature," or use a different editor.
Q. When you start a UUPC/extended application, it displays a message like "environment variable UUPCSYSRC must be specified," or "User configuration parameter "mailbox" must be set, "then exits.
A. You need to set the UUPCSYSRC and UUPCUSRRC environment variables. See step **8**, page 25, under **Configuring** UUPC/extended **after installation**, above.

2This is because UUXQT runs and processes eXecute files after UUCICO has terminated. As UUCICO has terminated, there is no way to tell if the file were delivered by an outbound or inbound telephone call, so UUXQT treats the file as if it were from an outbound call.

1Well, actually, you can worry, but try not to panic. "Calm down, it's only 1's and 0's".

Q. When you start up UUCICO, it displays a message like: "Invalid host id in c:/uupc/permissn, MACHINE=[name]; Unable to initialize security," then exits.

A. The PERMISSN file and your SYSTEMS file are inconsistent. There is an entry in your PERMISSN file which does not match a corresponding system in the SYSTEMS file, a userid which does not appear in PASSWD, or a directory does not exist. See steps **4**, **5**, and **6** in **Configuring** UUPC/extended **after installation**, page 19, above.

Q. When you start up UUCICO, it displays a message like: wanted "OK" got ??? "ERROR????" then exits.

A. You have a problem with your modem file. Check your UUPC.RC file (for incoming calls) or SYSTEMS file (for outgoing calls) to make sure you're using the .MDM file you think you are, then check the lines in the file to find the one that gives your modem problems. Try "uucico -x 4" to see more information as UUCICO processes the script. Also, try using a FOSSIL driver to reduce the chance of lost connections. See also **The Fine Art of Chat Scripts**, page 29.

Q. When you try to connect to the remote machine, it refuses the connection, even though you were able to dial in with another telecommunications program.

A. This could be one of several things, most of which could be due to errors in the entry in your SYSTEMS file for your mail server. See part **4** in **Configuring** UUPC/extended **after installation**, page 22. Or, you could have given your system a different name in your UUPC.RC file than the mail server expected. See part **1.** in **Configuring** UUPC/extended **after installation**, page 20.

Q. UUPC/extended will connect but will not exchange data with another system, What's wrong?

A. Any number of things, which is why you should seek additional information from either a UNIX guru or the Nutshell Handbooks. However, one hint is that any programs communicating via the UUCP "g" protocol must have a clean eight bit connection. A seven bit even parity connection or a connection with software (X-ON/X-OFF) flow control will ruin your whole day. If you can login as a normal user to the remote system, you can use the STTY command to determine the parity and flow control settings. Also, make sure the **variablepacket** option does **not** appear in your modem file.

Q. After UUPC/extended picks up my mail, UUXQT tries to deliver it and fails with a message "PERMISSION DENIED". What's wrong?

A. A good chance is that UUXQT cannot find RMAIL. Any program invoked by UUXQT must be in the path. Placing a command in the directory UUXQT is invoked from is not enough, because UUXQT changes directories as it runs.

Q. When UUPC/extended logins the remote system, the remote delivers **Shere** as the first message and then **RLOGIN** as the second message. Why doesn't UUPC/extended like this second message?

A. The full answer (or at least the full list of the possible error responses to the second message sent to the remote host) is listed in *Managing UUCP and USENET***.** Suffice to say the **RLOGIN** means that the remote system does not know your system. This is caused by your system being missing from the remote's **USERFILE**, **L.sys**, **Systems**, or **Permissions**[1] files.

Q. I get dropped characters in the login script when running the DOS version of UUPC/extended under OS/2. How can I fix that?

A. Run the OS/2 version or use an OS/2 FOSSIL driver. The DOS version doesn't

[1] A single system will not have all these files. For example, a UUPC/extended§ system or a system running the BNU version of UUCP will have **Permissions** and **Systems**.

PA

handle it well when a multi-tasking operating system steals its time slice. The problem also occurs when running the DOS versions under Windows, and can be corrected by using the Windows version of UUPC/extended instead.

UUPC/extended Command Referencetc "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 Descriptionstc "Command Descriptions" \l 2§

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

COMM34tc "COMM34" \l 3§

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.

EXPIREtc "EXPIRE" \l 3§

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.

FMTtc "FMT" \l 3§

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
```

GENSIGtc "GENSIG" \l 3§

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.

MAILtc "MAIL" \l 3§

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 24th revision of this document and was last updated on October 4, 1993 by Katherine Derbyshire.

NOVRSTRKtc "NOVRSTRK" \l 3§

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.

RMAILtc "RMAIL" \l 3§

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**[1], 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.[2]

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.[3]

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.

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

[2] "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.

[3] 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).

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:**.

RNEWStc "RNEWS" \l 3§

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[1].

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.

[1] 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 24th revision of this document and was last updated on October 4, 1993 by Katherine Derbyshire.

SUtc "SU" \l 3§

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.

UUCICOtc "UUCICO" \l 3§

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.[1]

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.

-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.

[1] 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.

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 24th revision of this document and was last updated on October 4, 1993 by Katherine Derbyshire.

1993UUPC/extended

UUCLEANtc "UUCLEAN" \l 3§

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.

UUIOtc "UUIO" \l 3§

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.

UUCPtc "UUCP" \l 3§

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.

UUNAMEtc "UUNAME" \l 3§

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.

UUPOLLtc "UUPOLL" \l 3§

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.

UUPORTtc "UUPORT" \l 3§

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 23rd revision of this document and was last updated on October 4, 1993 by Katherine Derbyshire.

UUSTATtc "UUSTAT" \l 3§

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.

UUSUBtc "UUSUB" \l 3§

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.

UUXtc "UUX" \l 3§

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.

UUXQTtc "UUXQT" \l 3§

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.

WAITINGtc "WAITING" \l 3§

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.

Changes From Previous Versions

UUPC/extended Configuration Filestc "UUPC/extended Configuration Files"§

Overview

This section describes the UUPC/extended configuration files. 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*.

The UUPC/extended UUPC.RC and *[userid]*.RC filestc "The UUPC/extended UUPC.RC and *[userid]*.RC files" \l 2§

Introduction

The UUPC.RC file provides the basic configuration information required to initialize the various UUPC/extended programs. It provides information which applies to the entire local system. It can also provide defaults for individual users. These defaults can then be overridden by the values in each user's *[userid]*.RC file[1].

Blank lines and lines beginning with a pound sign (#) are ignored. Entries in the file are of the form:

prefix.**keyword**=value

If multiple lines exist with the same keyword , the last line is used. An exception to this rule is the **options=** line; if multiple **options=** lines exist, the options are processed individually and the last occurrence of each option is used.

The *prefix* is optional, and if present must be DOS, 32BITOS2 (for OS/2 2.x), 16BITOS2 (for OS/2 1.x), WIN16(for Windows 3.x), or WIN32(for Windows NT) and must be followed by a period. Case is not significant for the *prefix* or **keyword**. If the *prefix* is present, the line is only used if the current environment matches the keyword. This allows multiple lines in the same configuration file for different environments, such as:

```
DOS.Editor=edit %s
OS2.Editor=epm %s
WIN16.Editor=notepad %s
WIN32.Editor=notepad %s
```

This allows the use of three different editors in four different environments. This could also be specified as:

```
Editor=notepad %s
DOS.Editor=edit %s
OS2.Editor=epm %s
```

[1] Note that text fields which are supplied in the UUPC.RC file cannot be cleared in the *[userid]*.RC file. For example, a user cannot remove the organization field from her mail's headers if the UUPC.RC supplied a default one. However, the user could change the organization name by adding an "Organization=[name]" line to her *[userid]*.RC file.

In the latter example, the program notepad is the editor specified for all environments, but it is superseded for both the DOS and OS/2 environments.

> **Note:** If the DOS versions of UUPC/extended programs are invoked under OS/2 or Windows NT, they will use the DOS environment keywords rather than attempting to determine the native environment.

The fields are described in the tables below as being of several types, most of which are self-explanatory. Of special note, however, are fields described as **tokens**, **path names** or **file names**. Tokens are single words with no spaces. Any tab or space terminates the value. As for path and file names:

- Path and file names must be valid names under the host operating system.

- Path and file names without drive letters are presumed to reside on the same drive as the UUPC/extended configuration directory.

- Slashes (/) and back slashes (\) are considered equivalent in paths. Back slashes are translated to slashes internally and translated back when opening files and for external commands such as editors.[1]

- Relative path and file names (except for those listed in the next paragraph) are presumed to be relative to the UUPC/extended configuration directory.

- The files defined by **Aliases**, **AltSignature**, and **FileSent** fields are assumed to be relative to the user's home directory. In addition, **only** these files may use the ~userid convention defined for files in the section on MAIL in **UUPC/extended Command Reference**, see page 40.

Keywords valid in individual or system configuration files

The following are the valid fields for the UUPC.RC or *[userid]*.RC file. They are normally placed in the UUPC.RC if applicable to all users of local system, and otherwise put in each *[userid]*.RC file as needed.

Keyword	**Type**	**Description**	**Default**[2]	**Example**[3]
Aliases	File name	**User** aliases (nicknames) file. **Note:** If the path is omitted, this file is presumed to reside in the current user's home directory.	**None**.	aliases.txt

[1] Forward slashes are not translated to backslashes for the compress program invoked from RNEWS, which is technically a bug but not a problem in practice since most compress programs are derived from UNIX programs which don't care about the difference.

[2] Required fields must appear in one of the two files, but need not appear in both.

[3] For *omitted* entries, the entire entry would be omitted to allow the default to be in effect.

Alt-Signature	File name	Name of alternative file included in response to ~A command when sending mail.	**None.**	*omitted*
BackupExt	File Extension	Extension used for files renamed by backup processing. **Note:** See **backup** option, below.	BAK	*omitted*
Editor	String	Command string including the name of the editor used for editing outgoing mail. **Note:** %s must appear in the string to define where the file name should be substituted.	**None.**	edit %s
FileSent	File name	Name of file in which all mail sent by local user is saved.	**None.** **Note:** If this parameter is omitted, no copy is saved.	outgoing
Folders	Path name	Included for use by MUSH (Mail User's Shell). Not used by UUPC/extended.	N/A	*omitted*
Home	Path name	Home directory of current user. This directory is the directory in which FORWARD files are searched for, and the default directory for the files defined by **FileSent**, **Aliases**, and **Signature** files.	**None.** **Note:** This field is required.	C:\u\snuffles
MailBox	User id	Name (userid) of the currently active user. Should be 1-8 characters and only contain valid alphanumerics.	**Note:** This field is required.	snuffles
Name	String	Full name of the user	**Note:** This field is required.	Snuffles P. Bear
Organization	String	Name of the organization to be listed in the Organization field	**None.**	Itty-Bitty Machines

		in headers of outgoing mail.		Corporation
Pager	String	Command string including the name of the pager used for displaying incoming mail. **Note:** %s must appear in the string to define where the file name should be substituted.	**None.**	list %s
ReplyTo	Token	Address to which replies to outgoing mail is to be sent.	Replies are send the originator of the mail.	*omitted*
Signature	File name	Name of signature file included in response to ~a command or by the **autosign** option when sending mail.	**None.**	*omitted*
Version	Token	UUPC/extended version file was last updated for. **Note:** This field is for documentation purposes only.	**None.**	1.12b

Keywords valid only in system configuration files

The follow keywords are **only** valid in the UUPC.RC configuration file, and always apply to all users of the local system.

Keyword	**Type**	**Description**	**Default**	**Example**[1]
AnonymousLogin	Token	Specifies times (using same format as time field in **SYSTEMS** file, see page 96) when anonymous login is allowed.	**None.** (No anonymous logins are allowed.)	Any Night Evening
ArchiveDir	Path name	Directory to which the **EXPIRE** command moves archived news articles. **Note:** Must be on same disk partition as the **NewsDir**	*[ConfDir]*/ARCHIVE	*omitted*
BackupExt	File Extension	Extension used for files renamed by backup processing. **Note:** See **backup** option, below.	BAK	*omitted*

[1] For *omitted* entries, the entire entry would be omitted to allow the default to be in effect.

Banner	File name	File displayed before login prompt by UUCICO.	**None.**	
CharSet	String	Valid characters allowed in file names and used for mapping UNIX names to local names.	a-z, 0-9, !#$%&'()-@^_`{}~	*omitted*
ConfDir	Path name	Base UUPC/extended configuration directory. Location of various files (SYSTEMS, PERMISSNS, PASSWD) and default parent directory for **ArchiveDir**, **MailDir**, **NewsDir**, **PubDir**, **SpoolDir**, and **TempDir**.	Taken from path component of UUPCSYSRC environment variable.	C:\UUPC
Domain	Token	Fully-Qualified Domain name of the local system, included in headers of all mail sent.	**None.** **Note**: This parameter is required.	kew.com nodename.UUCP
From-Domain	Token	Fully-qualified Domain name of system which overrides the **Domain** field for the RFC-822 **From:** field. Used in site hiding. See **The Ever So English Sport of Site Hiding**, page 110. **Note:** This option should *only* be used when site hiding. Omit it otherwise.	**None.**	*omitted*
InModem	Simple file name	Simple name (without path or extension) of *[modem]*.MDM file to use for when UUCICO is invoked with the -r 0 (accepting incoming calls) option.	**None.** **Note:** This field is required for using the UUCICO -r 0 option.	HAYES24
Internal-Commands	List of commands	List of commands processed by UUXQT as internal commands.	break cd chdir copy ctty date del dir echo erase for md mkdir rd rem ren rename rmdir	*omitted*

			time ver verify vol	
LocalDomain	Token	Domain name which is automatically appended to simple system names being searched for during host table look-ups. See also the sections on mail routing in **Advanced Installation and Configuration Topics**, page 103.	If the **Domain** has one or two parts (x.y or UUCP), the entire domain name. Otherwise, all but the left most word of the domain name (y.z out of x.y.z.)	*omitted*
MailDir	Path name	Directory to which RMAIL delivers mail.	*[ConfDir]*/MAIL	*omitted*
MailExt	Token	Extension used for mailbox files.	**None.**	SPB[1]
MailServ	Host name	Simple host name of system to handle default mail forwarding. **Note:** This system must be listed in your **SYSTEMS** file. The systems file is discussed in **The UUPC/extended SYSTEMS file**, page 96.	**None.** **Note:** This parameter is required.	myserv
MaximumHops	Integer	Maximum number of Received: lines allowed in remote mail before it is rejected by RMAIL because of looping.	20	*omitted*
MOTD	File name	File name with text to be displayed after successful login of **non-**UUCICO shells.	**None.**	*omitted*
MushDir	Path name	Included for use by MUSH (Mail User's Shell). Not used by UUPC/extended.	N/A	*omitted*
NewsDir	Path name	Directory into which RNEWS writes news.	*[ConfDir]*/NEWS	*omitted*

[1] Snuffles will feel hurt if you don't use this.

		Note: Must be on same disk partition as the **ArchiveDir**		
NewsServ	Host name	Host to which posted news is sent for distribution to the outside world. **Note:** Posting news is currently not supported. This field is included for future expansion only.	Value of **MailServ**.	*omitted*
NodeName	Host name	Simple UUCP host name of the local system.	**None.**	toscis
Options	List of tokens	List of Boolean options from list below.	All options off.	askcc autoedit autoinclude autosign backup doskey expert pager purge suppresscopyright verbose undelete multitask multiqueue symmetricgrades
Passwd	File name	File name of PASSWD file containing name and directory information for local users and passwords for remote systems logging in.	*[ConfDir]*/PASSWD	*omitted*
Path	Path String	Semi-colon delimited list of directories for UUXQT to search for external programs.	Value of PATH variable when UUXQT is invoked.	*omitted*
Postmaster	User id	Name (userid) of the user to receive bounced mail and all other mail addressed to local user **Postmaster**. Should be 1-8 characters and only contain valid alphanumerics.	**None.** **Note:** This field is required.	postmast
PubDir	Path name	Default file upload directory for files sent to the local system via remote UUCP commands.	*[ConfDir]*/PUBLIC	*omitted*

SpoolDir	Path name	Directory used for log files, files queued for other systems, and incoming mail/news files.	*[ConfDir]*/SPOOL	*omitted*
Systems	File name	File name of SYSTEMS file containing names and dialing information for all remote systems.	*[ConfDir]*/SYSTEMS	*omitted*
TempDir	Path name	Path of directory for temporary files generated by MAIL, UUXQT, and other programs. The drive this directory resides on must have space for **twice** the largest mail or news file you expect to receive. **Note:** If possible, this directory should be located on a RAM drive or a drive with a lazy disk cache.[1]	*[ConfDir]*/TMP	*omitted*
Uncompress	String	Command string including the name of the program used for uncompressing incoming news. **Note:** %s must appear in the string to define where the input file name should be substituted. A second %s must appear to define the output file name if the default output file name is not the input name minus the extension,	uncompre %s	compress -d %s gzip -d %s

Boolean options in either UUPC.RC or *[userid]*.RC

The following are Boolean options, which can be set by the user in the UUPC.RC file, their *[userid]*.RC file, or via the MAIL **set** command (See **UUPC/extended Command Reference**, page 40.):

Note: All options default to False.

Option	Default operation	Operation if set

[1] On the other hand, if you don't know what a RAM disk or lazy disk cache is, you don't want to use them.

name		
askcc	The user must enter all addresses when invoking **mail** or **forward**. Additional addresses cannot be added to mail generated by **reply**.	After composing, reply to, or forwarding mail, the user is prompted for carbon copy (Cc:) addresses.
autoedit	When sending mail, the user is presented with the line oriented send mail interface. To use a full screen editor, the user must use the **~e** (edit) command or exit line oriented data entry to invoke his editor.	If the user has defined an editor, when sending mail, the editor is immediately invoked for the user to enter mail with.
autoinclude	Do not automatically include the text of mail being replied to.	When the reply command is invoked, automatically include the text of the mail being replied to as if the **~m** command was issued.
autosign	When sending mail, the user's signature file is not appended to outgoing mail automatically. If defined, the user may still include the file via the **~a** (include autograph) command.	If defined in the user's configuration file, the signature file is automatically appended to all outgoing mail.
backup	Do not create a backup of the mailbox being processed before rewriting or deleting it.	Before updating or deleting the current file when exiting mail, rename the existing file to the same file name with the extension defined by the UUPC.RC variable **BackupExt**. **Note:** If no extension is defined, a default extension of "BAK" is used.
doskey	Under DOS, interactive input is read from the console using standard DOS services (INT 21H function 0AH). DOSKEY, if installed, is ignored. Empty input lines are processed normally.	If installed, the MS-DOS 5.0 DOSKEY program is used to read input from the console; normal DOSKEY functions, including the ability to edit and scroll input, along with macro expansion, is available. Empty lines (which can be generated by DOSKEY macro processing) are ignored at the command prompt unless the **expert** option is also set. **Note:** If DOSKEY is not installed, when the user is first prompted for

		input the **doskey** option is reset to **nodoskey** with a warning message.
dot	When sending mail, interactive input must be terminated by the DOS end of file character (Ctrl-Z).	When sending mail, interactive input can be terminated by a single period (.) in column one.
expert	All informational messages are displayed, and if the **doskey** option is also set, empty input lines are ignored in response to the command prompt.	Boiler plate messages, such as the initial help prompt, are suppressed, and empty input lines are not ignored when the **doskey** option is set.
forward-save	Mail sent via the **forward** command is not saved in the user's outgoing mailbox.	If the user has defined an outgoing mailbox, then mail sent via the **forward** command is saved in the same fashion as other outgoing mail.
fromsep	Items in the mailbox must be separated by a line of binary ones (1). (If you use the DOS **type** command to look at your mailbox, these look like smiley faces.) **Note:** This option is provided for compatibility with other RMAIL versions.	Items in the mailbox can be separated only by UUCP format **From** lines. The line of binary ones (1) required between items by the default operation of **nofromsep** is still generated when updating mailboxes. **Note:** This option must be set in the UUPC.RC or [*userid*].RC file, because it is only examined at start up.
pager	**help**, **print** and **type** use the external pager if one is defined, and **Print** and **Type** use the internal pager.	**help**, **print** and **type** use the internal pager, and **Print** and **Type** use the external pager if one is defined
purge	Empty mailboxes are left in place when the user exits mail.	Empty mailboxes are deleted when the user exits mail.
save	Mail is left in the user's system mailbox after reading.	If mail is not deleted from the user's system mailbox after reading, it is automatically saved in his **~/mbox** file when the user exits mail.
speedover-memory	Strings are allocated favoring low memory usage. Specifically, when allocating fixed permanent strings, strings already allocated are examined to determine if a pointer to existing duplicate string can be used.	Strings are allocated favoring speed over memory usage. Specifically, permanent strings are allocated without examining if a copy of the string already exists.

suppress-copyright	Display the UUPC/extended copyright notice at program startup.	Suppress the copyright notice at program startup. Of course, this does not suppress the associated responsibility. **Note:** This option must be set in a system or user configuration file, because it is only examined at start up.
undelete	*OS/2 only*. The environment variable **DELDIR** is reset to a null string by **UUPOLL**, causing files deleted by its children (**UUCICO**, **UUPOLL**, and **UUCLEAN**) to not be archived by the OS/2 operating system for later recovery.	The **DELDIR** environment variable is left alone, causing OS/2 to copy all deleted files to a hidden directory. This causes a performance and free disk space impact by saving files which the user should never have to access, much less recover. **Note:** This option must be set in the UUPC.RC or [*userid*].RC file, because it is only examined at **UUPOLL** start up.
verbose	When invoked from mail, **RMAIL** only displays error messages.	When invoked from mail, RMAIL displays both error messages and nominal status messages, including the addresses mail was delivered to. **Note:** This option must be set in a system or user configuration file, because it is only examined at start up.

Options only used in the UUPC.RC file

The following system-oriented options must apply to all users on a system and can **only** be set in the UUPC.RC file:

Option name	Default operation	Operation if set
bang	RFC-822 headers are generated with a "User name" <user@node> format.	RFC-822 headers are generated with UUCP style (User name) node!user format.
bounce	Undeliverable mail is delivered directly to the local postmaster mail with no additional header or text	Undeliverable mail is sent to the originator of the mail (as determined by the UUXQT requester

	generated.	information) and to the local postmaster via a newly generated mail message from UUCP which explains why the mail failed.
collect	The size of mail is not reported by RMAIL.	The RMAIL delivery and spooling messages include the size of the files delivered.
directory	Mail for "user" is delivered to the file *user* in the directory specified by the **maildir=** line in the system configuration file.	Mail for "user" is delivered to the file **newmail** in the *user* sub-directory of the directory specified by the **maildir=** line in the UUPC.RC configuration file.
escape	Only Ctrl-Break can be used to exit from UUPOLL or UUCICO.	If the Esc (Escape) key is pressed, then the next time UUPOLL or UUCICO polls the type-ahead buffer the program will act as if Ctrl-Break was pressed. **Note:** Enabling this option effectively disables type-ahead, since all characters except Esc are discarded with an error message.
history	No news history is maintained.	**Note:** This option is currently ignored if enabled.
honordebug	When UUCICO is called by another system, it ignores any debugging level transmitted by the caller.	When UUCICO is called by another system, it uses any debugging level transmitted by the remote system to set its own debugging level.
kanji	No translation of characters takes place during RMAIL processing.	Mail which originates locally is translated from a 2 byte Kanji code called Shift-JIS (Japanese ideogram) to Kanji in a 7-bit subset of ISO 2022 which can be transmitted via SMTP. Mail from remote systems which is delivered locally is translated from JIS 7bit back to Shift-JIS.
longname	Under OS/2 and Windows NT, file names for incoming files are made to conform to the DOS 11 character name limit even if the file system supports longer names.	Under OS/2 and Windows NT, file names are not made to conform to DOS name 8 + 3 naming conventions.

monocase	Job sequence ids are generated in base 62 using numerics and upper and lower case alphabetics.	Job sequence ids are generated in base 36 using numerics and upper case alphabetics.
multiqueue	When queuing mail for other hosts, each addressee is delivered separately.	When queuing mail for other hosts, a single file delivered via the remote UUCP can have multiple addressees.
multitask	Processing is optimized for speed over system integrity. **Note:** We strongly recommend that you enable this option in all environments except DOS.	Additional processing is performed to insure system integrity. This additional processing includes creating lock files to prevent concurrent access to system spool directories, writing program logs to temporary files and then appending the file to the permanent log file at program termination, and moving new mail from the user's system mailbox into the user's home directory when **MAIL** is first run after the new mail's arrival.
senddebug	When UUCICO calls another system, it does not report its debugging level to the remote system.	When UUCICO calls another system, its debugging level is transmitted to the remote system.
snews	Incoming news is written to separate files in directories based on the news group names listed in the ACTIVE file located in the **NewsDir** directory.	Incoming news is written to a file in the **NewsDir** directory without examining the ACTIVE file.
symmetricgrades	When actively polling, **UUCICO** does not transmit the maximum grade for file transfers allowed by the **SYSTEMS** file to the remote system at startup.	**UUCICO** transmits the maximum grade for file transfers allowed by the **SYSTEMS** file to the remote system at startup.
syslog	No record is made of files transferred.	The file SYSLOG is written in the spool directory with a record for each file transferred to or from the local system.

Modem (*[modem].MDM)* Filestc "Modem Files" \l 2§

Introductiontc Introduction \l 3§

Modem files (.MDM) define the strings used to command a modem when dialing out. They allow commands always used with a particular modem to be written once rather than being placed everywhere they are used, and also allow different modems to be used by changing only the modem file (or changing the reference to the modem file in the SYSTEMS or UUPC.RC files).

All strings defined in a modem file are standard scripts as described in **The Fine Art of Chat Scripts**, page 29, with the exception of the dial prefix and dial suffix strings. The dial prefix and suffix strings are combined with the phone number listed in the SYSTEMS file and sent as one string to the modem when dialing out. Most modems have behavior similar to one of the sample modems, so minor changes to one of these files should get you up and running.

> **Note:** If you make changes to a modem file, copy it to a new name to avoid confusion with the distributed version.

The syntax of a modem file is the same as the UUPC.RC and *[userid]*.RC files described in **The UUPC/extended UUPC.RC and [userid].RC files**, page 75. The keywords accepted in modem files are described below.

Modem File Guidelines tc "Modem File Guidelines" \l 3§

There are several things to keep mind when writing a modem file, especially for an error correcting modem. Not all of the following apply to all modems, but as a general rule:

1. Initialize the modem from factory defaults as opposed to simply resetting the modem. Starting with factory defaults allows the modem configuration to be shared between modems that may not have been initialized the same way.

   Note: Do **not** have the modem file rewrite the modem defaults to non-volatile RAM (often done with a AT&W command). This may confuse other programs which use the modem.

2. Initialize the speaker to off. (Typically ATM0)[1]

3. Have the modem hang-up and reset to factory defaults when Data Set Ready (DTR) is lowered.

4. Write a unique modem file for dialing out with error correction only, rather than taking a modem default which accepts error or non-error correcting connections. Rejecting a remote connection if the expected speed or protocol is not available insures that a long distance connection does not run at a sub-par rate.

5. Likewise, write a modem file with error correction disabled, to bypass attempts at modem-

[1] You may want the speaker enabled for initial testing, but UUCICO is designed to run unattended, and waking house mates up to hear a modem dial at 3 AM will not make you popular.

to-modem protocol negotiations with modems which do not support them.[1]

6. Disable auto-answer unless you actually want to have the modem ready to answer the phone.

7. Disable software (XON/XOFF) flow control, and enable hardware (CTS/RTS) flow control.[2]

8. Enable the modem inactivity time-out to automatically hang up the modem after two minutes if no data is sent or received. If **no** data is send by UUCICO in two minutes with the modem off hook, the program and/or operating system has crashed.

9. Some modems have a time delay to insure all data buffered for the remote system is sent. Set this delay to zero (no delay), as UUCICO will have insured data delivery via software protocols.

10. Disable remote loop back testing.

Valid Fields in Modem Files tc "Valid Fields in Modem Files" \l 3§

The following fields are valid in a modem file.

Keyword	**Type**	**Description**	**Default**	**Example**[3],[4]
Answer	Script	Script to pick up phone (if **Ring** script did not enable auto-answer) and determine an incoming call has been answered.	**None**. This field is required for answering the phone.	CONNECT
Answer-Timeout	Integer	Time in seconds for last response expected by **Answer** script.	30 seconds	*omitted*
BigGPacketSize	Integer	Maximum number of bytes the remote host may send the local host per packet when using the	512 bytes	*omitted*

[1] Phone calls to the system *nbstime listed in the sample systems file will fail if error correction is not disabled, because otherwise an expected string is lost from the remote system while the modem is still trying enable error correction.

[2] Sample in older releases of UUPC/extended§ explicitly turned off hardware flow control. Enabling it is a change from the previous suggestions, and is required for many high speed modems.

[3] Examples are for a ZOOM VFX v32bis modem. Your mileage will vary.

[4] For *omitted* entries, the entire entry would be omitted (blank) to allow the default to be in effect.

		"G" protocol. Must be a power of 2 between 32 and 512.		
bigGWindowSize	Integer	Number of packets active at a time when using the "G" protocol. Must in range 1 to 7.	7	*omitted*
CharDelay	Integer	Delay in milliseconds between characters sent to the modem when processing dialing scripts.	0	*omitted*
Connect	Script	Script used to determine if modem has connected to remote host.	**None**. This field is required for dialing the modem.	CONNECT
Description	String	One line description of the modem file. This field is for documentation purposes, and is extracted to build the list of modems on page 93.	**None.**	Zoom v32bis with v.42 and v.42bis enabled
Device	Token	Communications Port Name This field is ignored by the TCP/IP and named pipe suites.	**None.** This field is required.	COM1
DialPrefix	String	String prepended to the phone number in SYSTEMS file to dial modem.	**None**. This field is required for dialing the modem.	ATDT
DialSuffix	String	String appended to the phone number in SYSTEMS file to dial modem.	**None.** Normally not needed.	*omitted*
DialTimeout	Integer	Time in seconds last string in **Connect** script is allowed to wait for a respond.	40	20 (local calls) 60 (long distance)
fPacketTimeout	Integer	Time in seconds allowed for responses by the remote host under "f" protocol.	20	*omitted*

fPacketSize	Integer	Bytes processed at a time by the "f" protocol.[1]	512	*omitted*
gPacketSize	Integer	Maximum number of bytes the remote host may send the local host per packet when using "g" protocol. Must be a power of 2 between 32 and 512. **Note:** Due to kernel buffer sizes, most UNIX systems cannot handle a **gpacketSize** larger than 128.	64 bytes	*omitted*
gPacketTimeout	Integer	Time in seconds allowed for responses by the remote host under "g", "G" , and "v" protocols.	10	*omitted*
gWindowSize	Integer	Number of packets active at a time when using "g" protocol. Must in range 1 to 7.	7	*omitted*
TransferBuffer	Integer	Size in bytes of internal disk I/O buffer in bytes. **Note:** If the specified value is less than the default, the default is used.	The larger value of 512 or four times the largest packet size in use.	*omitted*
Hangup	Script	Script to hang-up modem. **Note:** Before this script is executed, the modem Dataset Ready (DTR) line is dropped for 0.5 seconds to request a modem hang-up.	**None.**	ATH OK
Initialize	Script	Script to initialize modem.	**None.** This field is required.	AT &F OK ATEM OK
InSpeed	Integer	Speed at which modem is initialized for remote login.	**None.** This field is required for	2400 38400

[1] This parameter is used only for an internal buffer size and does not affect the transmitted stream of data.

			answering the phone.	
Maximum-Errors	Integer	For all protocols except "t", the number of consecutive errors allowed before the connection is dropped. **Note:** Consecutive errors are defined as errors occurring with no data successfully transmitted between the errors.	10	*omitted*
Modem-Timeout	Integer	Number of seconds allowed for modem command responses in scripts **except** for the **last** responses to the **Answer** and **Connect.**	3	*omitted*
NoConnect	One or more strings	Modem responses, which, if given, indicate that dialing or answering has failed and the call should be given up on.	**None.**	"NO DIALTONE" "NO CARRIER" ERROR
Options	List of Boolean options	See below for individual options.	nocarrierdetect nodirect nofixedspeed novariablepacket	carrierdetect fixedspeed
Priority	Integer	Magic number to set OS/2 and Windows NT execution priority for UUCICO. **Note:** This field is ignored under DOS and currently not supported under Windows NT. **Note:** Under OS/2, may be 1-4.	Under OS/2, 4 (PRTYC_FOREGROUND-SERVER)	*omitted*
PriorityDelta	Integer	Fine tuning value for UUCICO priority. **Note:** This field is ignored under DOS and currently not supported under Windows NT. **Note:** Under, OS/2, 0 - 30 lowers priority, 32 - 62 raise the priority.	Under OS/2, 31 (no change from **Priority**)	*omitted*

Ring	Script	Script to initialize modem for answering and to detect ring.	**None**. This field is required for answering the phone.	ATS0=1 OK "" RING
Script-Timeout	Integer	Time allowed for responses to scripts specified in SYSTEMS file.	30	*omitted*
Suite	Token	Name of communications software driver suite. All systems can use the default, **internal**, which drives the serial port using internal and Operating System support. DOS systems can also use the **fossil**, or **ArtiComm** suites. Windows 3.x and Windows NT systems can also use the **TCP/IP** suite if they have TCP/IP support installed.[1] OS/2 2.x systems can also use the **NamedPipes** suite, although this support is primarily of use for testing on a single system.	internal	*omitted*
tPacket-Timeout	Integer	Time in seconds allowed for responses by the remote host under the 't' protocol.	60	*omitted*
Version	Token	Program version number the file was last updated for. This field is for documentation purposes and is extracted to build the list on page 93.	**None.**	1.12b
vPacket-Size	Integer	Bytes sent per packet when using "v" protocol. Must be a power of 2 between 32 and 4096.	512 bytes	*omitted*

[1] Windows 3.1 systems must have the **WINSOCK.DLL** installed to use TCP/IP support. Contact your TCP/IP vendor to see if they this library for their product. Windows NT natively supports TCP/IP for many ethernet cards, but does not support dial-up TCP/IP.

vWindow-Size	Integer	Number of packets active at a time when using "v" protocol. Must in range 1 to 7.	7	*omitted*

Boolean Options In Modem Files tc "Boolean Options in Modem files" \l 3§

The following option flags can be set on the **options** line of the modem file.

Option name	**Default operation**	**Operation if set**
Carrier-Detect	UUCICO processing ignores the state of the modem status lines.	If the modem Terminal Ready (DTR) or Carrier Detect (DCD) lines drop after a connection is established, the connection is aborted.
FixedSpeed	After the **Connect** or **Answer** script executes, UUCICO reads the serial port for the connect speed of the modem and changes the port speed to match this reported speed.	The modem speed is fixed at the value specified by the **InModem** keyword or **Systems** file speed.
Variable-Packet	Data Packets sent by the "g" and "G" protocol are always padded to the full packet length negotiated at startup.	Data Packets sent by the "g" and "G" protocols are padded only to the nearest power of 32 (32, 64, 128 ...) and these short packets are sent. **Note:** Most UNIX systems **cannot** handle variable length packets, and connections to such will fail in mysterious ways if this option is enabled.[1]
Direct	The modem Dataset Ready (DSR) line is used for hardware flow control: When the modem lowers this signal, UUCICO stops transmitting data until the signal is raised again.	The modem Dataset Ready line is ignored.

[1] Systems running Ian Taylor's Taylor UUCP are a notable exception, but virtually all UUCP programs shipped with various UNIX systems will die if fed short packets. Just because UNIX developers invented the UUCP 'g' protocol doesn't mean they implemented it properly.

Supplied Modem files tc "Supplied Modem Files" \l 3§

The follow table lists the sample modem files supplied with UUPC/extended. As these files have been collected from a number of sources previous to the above guidelines being written, the modes and setups vary. Most of the files do not exactly conform to the above guidelines.

> **Note**: Please mail new and updated modem files to **modems@kew.com**. Please verify the file is updated to conform with the guidelines on page 87, supply **Description=** and **Version=** fields, and include instructions as to whether the file modifies an existing modem file or should be added to the collection as a new file.

File name	Version	Description
CODEX		CODEX Modem with MNP Protocol enabled
DIR		Direct connection, between two systems connected by a null modem cable.
GVCV32		GVC 9600 V.32/V.42bis with smart options, serial port speed locked
GVCV32G		GVC V.32/V.42bis with autobaud, for answering calls from 2400 baud modems
HAYES12		Hayes Smartmodem 1200
HAYES24		Generic Hayes Smartmodem at 2400 baud, disabling command echo

HAYES241		Generic Hayes Smartmodem at 2400 baud
HAYES24D		Generic Hayes Smartmodem at 2400 baud
INSATFAX		Intel SatisFAXtion modem
INTEL24E		Intel 2400ex modem
PPI-V32B		PPI 14400FX V.32bis/V.42bis modem
QBLAZER		Telebit QBlazer Modem - v32 mode with port locked at 9600
QBLAZER2		Telebit QBlazer Modem - v32 mode with speed shifting
SAMPLE	1.12a	Generic sample modem file with verbose comments

SUPRA		Supra V.32bis FAX modem with MNP 2, 5, and 10
SUPRA12		Supra V.32bis FAX modem with MNP disabled, 1200 baud only
SUPRA24		Supra V.32bis FAX modem with MNP disabled
SX1200		Microcom SX1200 MNP/4 modem with MNP 4 enabled
SX1200D		Microcom SX1200 MNP/4 modem with MNP disabled
TB1000		Telebit Trailblazer T1000
TB2500		Telebit Trailblazer T2500 (V.32/V.42bis) with 7.x PROM upgrade
TB2500A		Telebit Trailblazer T2500 (V.32/V.42bis) with 7.x PROM upgrade
TBPLUS		Telebit Trailblazer Plus with 7.x PROM upgrade

TBPMNP5		Telebit Trailblazer Plus with 4.x PROMs -- MNP5, but no V.42bis
TBPPEP		Telebit Trailblazer Plus locked into PEP mode
TBW56K		Telebit WorldBlazer (V.32bis/V.42bis) modem, at 57.6 Kbaud
TBWORLD		Telebit WorldBlazer (V.32bis/V.42bis) modem
TELEPATH		Gateway 2000 Telepath modem
USRCV32B		U. S. Robotics V.32bis/V.42bis Courier Dual Standard
USRSPORT		U. S. Robotics Sportster V.32/V.42bis
USRSPRT2		U. S. Robotics Sportster V.32/V.42bis

WBDUMB		Telebit WorldBlazer modem with MNP/5 and V.42bis disabled
ZOOM2400		Zoom 2400/V.42bis, with V.42bis disabled
ZOOM4BIS		Zoom 2400/V.42bis with v.42/v.42bis options enabled
ZOOMDUMB	1.12a	Zoom VFX 14400 with V.42bis and MNP5 disabled
ZOOMVFX	1.12a	Zoom VFX 14400 modem locked into V.32/V.42bis mode
ZOOMVFXB	1.12a	Zoom VFX 14400 modem with smart options enabled
ZOOMVFXT		Zoom Turbo V.32/V.42bis modem
ZYXEV32B		ZyXEL U-1496E+ V.32bis/V.42bis modem

The UUPC/extended SYSTEMS filetc "The UUPC/extended SYSTEMS file" \l 2§

Introductiontc "Introduction" \l 3§

The SYSTEMS file contains the names of your UUCP neighbors, and describes how and when to communicate with them.

The SYSTEMS file contains comments and system descriptions. As in the other system configuration files, any line beginning with "#" is treated as a comment. The system descriptions lines look like the following:

hostname Time MODEM speed phone protocol expect-string send-string expect-string send-string...

Example: Toscis's SYSTEMS file entry for kewgate looks like this:

kewgate Any TB2500 19200 1-617-641-4817 g gin:--gin: Utoscis ssword:--ssword: AppleJuice

The SYSTEMS file may contain several entries for the same remote system. If UUPC/extended can't connect to a system using the first entry in the SYSTEMS file, it tries the others. See **Multiple entries in the SYSTEMS file**, page 100, for more details.

Each of the fields in the SYSTEMS file is described in the table below.

Field name	Description	Example
hostname	Name of the system to call. Any system you call or are called by must be listed at least once in this file. You may include the same system more than once if you have multiple phone numbers for it.	vanilla
Time and grade	When calls may be made to this system. **Any** allows calls 24 hours a day, **Night** and **Evening** refer to night and evening phone rates[1] respectively. **Never** is used for a system which you never dial, and usually used for systems which only call you. **Note:** See below under "Time" for a further explanation.	Any
MODEM	Name of the modem file (without the .MDM extension) used to call this	TB2500

[1] Specifically, the AT&T night rate is in effect from 11 PM to 8 AM every day, and from 11 PM Friday to Sunday at 5 PM. The UUCICO definition for Evening includes the times allowed for Night plus 5 PM to 11 PM everyday.

	system.	
Telephone, host name, or pipe name	Telephone number to call, host name to connect to over the network (when using **TCP/IP** suite). or named pipe to connect to (when using **NamedPipes** suite).[1]	1-617-641-4817 mailserver.domain.name \\servername\pipe\uucp
p (protocol)	Protocol to use when calling. If you don't know what protocols available, then leave the "g" alone. **Note:** See below under "Protocols" for a full list of available protocols.	g
script	Login script for the system. **Note:** See below under "Scripts" for a further explanation.	gin:--gin: Utoscis ssword:--ssword: AppleJuice

Timetc "Time" \l 3§

The Time field allows you to restrict the hours that your machine calls your neighbors. You can combine several different times to call into the Time field. When you do so, the times your machine is permitted to call is the or-combination of all of the times entered.

Note: Time fields are ignored (treated as if you specify **Any**) if the -n option is used on the UUCICO command line.

The entries in the time field are combinations of labels and times. The labels represent days of the week, or groups of days, or combinations of days and times that match U. S. telephone rate schedules, like **Evening** or **Night**. The times are on a 24 hour clock, and must use four digits. The specified times are logically and-ed together with the times for the labels listed in the table below.

Example: Mo0800-1700,Night

In this example, the machine would be permitted to call out on Mondays between 8 AM and 5 PM, or at any time that night rates are in effect. (The exact time is in the table below.)

Example: ROA0800-1700

[1] When UUCICO listens on a named pipe, it always listens to the hard coded name **\pipe\uucp**.

In this example, the times listed below for ROA are and-ed with the times listed in the table, so in practice, the system will only be permitted to call out between 8 AM and 5 PM on Saturdays.

If you don't care what time your machine calls your neighbors, put "Any" in the Time field. If you never want to call your neighbor (if they want to call you instead, for example), then put "Never" in the Time field. The other possible entries are:

Keyword in Systems File	**Days of the week dialing can occur**	**Time of day dialing can occur**
Any	Any time at all	24 hours a day
Never	Never	
Wk	Weekdays only	24 hours a day
Mo	Mondays only	24 hours a day
Tu	Tuesdays only	24 hours a day
We	Wednesdays only	24 hours a day
Th	Thursdays only	24 hours a day

Fr	Fridays only	24 hours a day
Sa	Saturdays only	24 hours a day
Su	Sundays only	24 hours a day
Evening	Monday through Friday Weekends	5 PM until 8 AM 24 hours a day
Night	Monday through Friday Saturday Sunday	11 PM until 8 AM 24 hours a day Midnight until 5 PM and after 11 PM
NonPeak[1]	Weekdays Weekends	6 PM until 7 AM 24 hours a day
ROA[2]	Weekdays Saturday	10 PM until 8 AM 24 hours a day

[1] These are the permitted hours for non-peak use of the Telenet PC-Pursuit network as of summer, 1989. Your mileage and network vendor may vary.

[2] These are the permitted hours for use of AT&T's "Reach Out America" program. Some program features exist mainly because the program's author has a specific use for them.

	Sunday	midnight until 5 PM and after 10 PM

Call grades tc "Call Grades" \l 3§

Each job which is queued for UUCICO to transfer has a call grade associated with it. This is simply the first letter of the job name after the system name as shown by the UUSTAT command. Grade **0** has highest priority, and **z** has the lowest. Overall, the priorities from highest to lowest are **0**-**9**, **A**-**Z**, and **a**-**z**, in that order. RMAIL queues jobs at class **C** by default, and UUCP queues jobs at class **n** by default.

Normally these classes are of no concern to the end user, but if the local system must call long distance or has other reasons to restrict traffic processed at particular hours, then each time field in the SYSTEMS file may modified by appending a slash and the lowest call grade to processed at that time. For example, to process only mail during the day and all other files to a system at night, the SYSTEMS file time entry would like this:

```
Any/C,Night
```

> **Note:**   The processing of call grades is affected by both the grades used by the remote system, the use of the **symmetricgrades** option in the UUPC.RC file as described in **The UUPC/extended UUPC.RC and [userid].RC files**, page 75, and use of the -g option on the UUCICO command line.

Protocolstc Protocols \l 3§

Two machines which talk to one another using UUCP can use one of several protocols to do so. Some of these protocols are more efficient than others, depending on the how fast and reliable the modem connection is, and whether or not 7- or 8- data bits are used in transmitting messages and files between them. UUPC/extended supports several different protocols, any of which might be used for talking to a remote system, depending on the circumstances.

In addition, the protocols have tunable parameters, set in the [*modem*].MDM file, which can improve their performance, depending on the circumstances. The protocols, the names of their tunable parameters, and the circumstances under which you might choose a particular protocol are described in the table below.

Protocol	**Tunable parameters**	**When to use the protocol**
*	**none**	This flag denotes the entry is not for a UUCP system at all, but rather for calling the U.S. National Institute for

		Standards and Technology (NIST) atomic clock to set the local system time.[1] **Note:** The NIST clock can be reached at 1-303-494-4774 at 1200 characters per second with no error correction.
f	fPacketSize fPacketTimeout MaximumErrors	This is intended for use over reliable 7-bit links such as X.25, where XON/XOFF (software) flow control is used instead of RTS/CTS (hardware). It is also only a 7-bit protocol, which means that it will be inefficient on binary data.
G	BigGPacketSize BigGWindowSize GPacketTimeout MaximumErrors	For supported systems, G protocol allows for higher performance on high-speed modems by increasing the number of bytes between packet checksums and acknowledgments. Its drawback is that the UNIX implementation of the G protocol is only available on newer implementations (System V Release 4), and is hard to configure.
g	gPacketSize gWindowSize GPacketTimeout MaximumErrors	When attempting to make the FIRST connection to another system, use g protocol with the default packet size of 64 bytes. It is widely supported. The 'g' protocol is it is slow because its default configuration requires six bytes of overhead data to be transmitted for every sixty-four bytes of user data. The 'g' protocol also requires an acknowledgment after sending only 448 bytes of data, which is smaller than the internal buffer of many error correcting modems. This protocol should be also used with Telebit Modems and other modems which spoof the UUCP 'g' protocol for additional performance boosts. Spoofing modems generally only support 'g' protocol but have better performance compared to non-spoofing modems.

[1] The NIST was formerly called the National Bureau of Standards (NBS). The sample SYSTEMS file includes an entry for calling NIST under the entry *nbstime. (A rose by any other name . . .)

t	tPacketTimeout	This protocol is for use over reliable network connections, such as TCP/IP or named pipes. With this protocol, files are transferred with no checksums to detect errors, which improves transfer speeds. Because no error correction is performed, this protocol is **not** available for modem connections.[1] **Note:** If a network level error occurs, this protocol aborts with no retries.
v	vPacketSize vWindowSize MaximumErrors	When two UUPC/extended systems are connecting over high-speed, use v protocol. The 'v' protocol supports larger packet sizes, while more flexible and easier to configure than 'g' or 'G' protocols. The main disadvantage to the 'v' protocol is that only UUPC/extended supports it.[2]

Scriptstc Scripts \l 3§

Finally, the last few fields of the system description are the strings that UUPC/extended expects, and will send, when it has successfully connected to the remote system, while it is trying to log in. The example above is a typical simple example. If you need something more complicated, then read **The Fine Art of Chat Scripts**, page 29.

Multiple entries in the SYSTEMS filetc "Multiple entries in the SYSTEMS file" \l 3§

It is possible to have more than one entry in the SYSTEMS file for the same remote system. (See the example at the beginning of this section.) You might want to do this if one of your UUCP neighbors has several phone lines, or several ways of logging in.

If the SYSTEMS file has multiple lines for one remote system in it, UUPC/extended uses the entries for the system in the order they appear until it gets connects to the remote system and completes the call. The call will not complete if:

- The time field specifies an entry should not be used

[1] To prevent use of the 't' protocol over a modem connection, UUPC/extended§ UUCICO will not present it as an available protocol when another system logs in.

[2] If you use or connect to a system running Taylor UUCP, send a note to Ian Taylor (taylor@airs.com) asking him to add 'v' protocol support to his package.

- The modem fails to initialize
- Your system doesn't connect to the remote system because of a busy signal or other dialing error
- The connection fails, because of trouble with the log-in script or an excessive number of bad packets.

In other words, the later lines in the SYSTEMS file for the system will only be used if the first connection is not *completely* successful.

Example: A system has SYSTEMS file entries for kewgate that looks like this:

```
kewgate Any TB2500 19200 1-617-555-4817 g gin:--gin: Utoscis ssword:--ssword: AppleJuice
kewgate Any TB2500 19200 1-617-555-4817 g gin:--gin: Utoscis ssword:--ssword: AppleJuice
kewgate Night HAYES24 2400 1-617-555-4817 v gin:--gin: Utoscis ssword:--ssword: AppleJuice
```

In the example, kewgate will be dialed up to three times until the call completes, twice using the TB2500 modem file and once using the HAYES24 modem file. Both TB2500 attempts are made at any hour, but the HAYES24 is only attempted at night.

Changes From Previous Versions

Advanced Installation and Configuration Topicstc "Advanced Installation and Configuration Topics"§

Changing How Mail is Addressed and Delivered Locallytc "Changing how mail is addressed and delivered" \l 2§

An Overview of RMAIL tc "Overview of RMAIL" \l 3§

Much of the advanced configuration you can do with UUPC/extended involves changing how RMAIL delivers mail. Before explaining how to change it, a short summary of the default processing for mail sent from a local user is in order:[1]

1. The user generates mail via the MAIL program.[2]

2. The user presses "s" to send the mail, invoking RMAIL.

3. RMAIL reads the headers for the newly created message, and decides to whom the message is addressed.

4. RMAIL invokes a subroutine called deliver() for each address.

5. deliver() determines if the address to be delivered to is local, remote, or handled by a gateway program.[3]

6. For remote mail addresses, the mail is queued for UUCICO to transmit at a later time.

7. For local mail addresses, a check is made for the address in the system alias file. If the alias exists, it is expanded without checking the PASSWD file. The addresses in the expansion are processed via a recursive call to deliver(), **except** that addresses read directly from the system alias file are not checked to be system aliases again.

8. If the local address is not defined as a system alias, then it is searched for in the PASSWD file. If the address is not a local user id, then the mail is bounced (delivered) to the local postmaster.[4] If the "bounce" option is set, the mail is also returned to the sender.[5]

1See UUPC/extended Command Referencetc "UUPC/extended Command Reference"§ §for the detailed syntax of the RMAIL command.

2For remote mail, the MAIL program is bypassed, with UUXQT given the job of invoking RMAIL for mail which has arrived on the local system via UUCICO.

3By default, mail destined for the nodename or domain name specified in the UUPC.RC configuration file is assumed to be local, and all other mail is remote. This behavior is warped by entries in the optional HOSTPATH file, if it exists.

4If the user id defined as the postmaster does not exist, the mail is bounced a second time to the postmaster with validation disabled. This causes the mail to be delivered to the postmaster even if the user id doesn't exist in the PASSWD file.

9. If the local address is valid, then the user's home directory is checked for a file named **FORWARD**. If this file exists, the file is processed for delivery instructions, otherwise the mail is delivered to the user's system mailbox in the UUPC/extended mail directory.

There are a number of ways to customize this processing, described below.

Using Alias Files, Forward Files, and the HOSTPATH filetc "Using Alias Files, Forward Files, and the HOSTPATH file" \l3§

Aliases in electronic mail serve two purposes. First, they allow a individual user to type a familiar nickname (fred) rather than a long full address (FFlintstone@dino.bedrock.bc) to address mail. The alias file thus works like an e-mail address book. Secondly, aliases allow groups of users (such as mailing lists) to be addressed by standard addresses. Under UUPC/extended, aliases are handled by two different types of optional alias files: the user alias file and the system alias file (**ALIASES)**. The user alias file is used **only** for mail sent by a local user using the MAIL front end. The system alias file is used for all local addresses processed by the RMAIL command.

In addition, many mail systems (including UUPC/extended) support the concept of a user forwarding his/her own mail via an optional **FORWARD** file[1]. The processing performed by the **FORWARD** file overlaps that of the system alias file. The primary difference between the system aliases file and an individual's forward file is that the system aliases file is generally maintained by a system administrator for all special addresses in a system or network, while the **FORWARD** file is maintained by the end user for just his own mail.

In addition to the files described above and detailed further below, UUPC/extended supports aliasing and re-routing entire systems via an optional **HOSTPATH** file. This file allows mail for entire systems to be treated as local mail, rerouted by a non-default path, or directed to an arbitrary program for further processing.

The User Alias Filetc "The User Alias File" \l3§

MAIL uses the user alias file to translate a short nickname to the longer full name and address needed to send mail to a remote site. The expanded name and address are actually written as part of the mail header generated by MAIL.

The user alias file may have any name. It is located by the **Aliases=** variable in the UUPC.RC or *[userid]*.RC file. If this variable is not defined in either configuration file, no user alias file is used.[2] Blank lines and lines which begin with the pound sign (#) are ignored. Multiple users may share a user alias file by defining the same name on the **Aliases=** line in the UUPC.RC file or their *[userid]*.RC file, but only one alias file is included by MAIL. If the **Aliases=** line

5If for some reason the postmaster is not a valid user, the mail bounces again to the postmaster, but with the validation bypassed to avoid an endless loop. If validation is bypassed, the mail is written to the local mail directory under the Postmaster's user id.

1In UNIX systems which use SENDMAIL, the file is normally called **.forward**.

2This should not be confused with the SYSTEM aliases file, which has a fixed name as described on page Error: Reference source not found.

appears in both the UUPC.RC and file the *[userid]*.RC file, only the alias file in the *[userid]*.RC file is used. A sample user alias file, ALIASES.TXT is included with the distribution file.

Note: The addresses supplied in the ALIASES.TXT file are valid addresses. They are provided as examples and to assist you in getting more information about UUPC/extended, communications software in general, USENET, and the Internet. However, they are *not* for random testing. Abuses such as the user who sent mail to all the addresses in the file as a "hello world" test would force discarding of the live information for future releases.

User aliases are defined in one of two formats. One is used to a define a nickname for a single addressee, and the other defines a list of users. The first format, defining one addressee's real name and associated e-mail address, is done by putting the information on one line in the alias file with the following syntax:

nickname "*Fullname*" <*address*>

Where *nickname* is the name you wish to use for the person when sending mail to them, *"Fullname"* is the full name of the person, and <*address*> is the electronic mail address for the person. The three fields must be separated by spaces or tabs. Nicknames are not case sensitive, addresses are case sensitive for some older mailers, and the full name is not case sensitive but its owner may be. Thus, for our friend Fred, the alias would be:

```
fred            "Fredrick Flintstone" <FFlintstone@dino.bedrock.bc>
```

When sending mail, the command:

```
mail fred
```

Will produce an addressee line with "Fredrick Flintstone" <FFlintstone@dino.bedrock.bc>.

You can also define lists of addresses by defining a nickname with one or more nicknames for the addressee, such as:

```
groupname       nickname1 nickname2 nickname3 ...
```

You must **not** put multiple full names/address pairs on one line, as the results will be unpredictable.[1] You can specify groups[2] within other groups. As an example, to define all of Fred's family and send to them as group, the aliases could look like:

WRONG:

```
parents   "Fred" <FFlintstone@dino.bedrock.bc>  "Wilma"
<WFlintstone@dino.bedrock.bc>
```

RIGHT:

```
fred            "Fredrick Flintstone" <FFlintstone@dino.bedrock.bc>
wilma     "Wilma Flintstone" <WFlintstone@dino.bedrock.bc>
pebbles   "Pebbles Flintstone" <Pebbles@dino.bedrock.bc>
```

[1] and will also be unpleasant. Note that such undefined behavior can be changed by the Wonderworks at will.

[2] You cannot directly or indirectly include a group within itself. If you violate this rule MAIL will cheerfully recursively expand the alias until it runs out of memory or disk space, leaving you cursing.

```
    family      pebbles parents
    parents     fred wilma
    flintstones family
```

Note that the alias for **flintstones** is totally optional, it merely allows you have an additional choice to send mail to the users defined by the **family** alias. This allows sending to the entire first family of Bedrock by typing either:

```
mail flintstones
OR

    mail family
```

Implicit User Alias Processing, (or, What Your Polar Bear Never Told You about Aliases)tc "Implicit User Alias Processing" \l3§

If MAIL cannot find a nickname for an address, it scans the alias table seeking a match on the user id and host of the addressee. The entries from both the user alias file and the **PASSWD** file are scanned, in that order. If a match is found, then the associated information for the address and user name is used just as if the nickname was entered. This behavior insures that a "standard" address is used for replies to mail.

The System Alias Filetc "The System Alias File" \l3§

Beginning with release 1.11x, UUPC/extended also supports a system wide alias file. The system alias file can be used to reroute mail for local addresses to local or remote users or groups, or to pipe mail into a program (such as an automated response program). RMAIL checks all local addresses that it processes against the system alias file. If an alias is found, the mail is redirected but the mail header is never changed.

To be used, the system wide alias file must be called **ALIASES**[1] and reside in the UUPC/extended configuration directory. While the user alias file performs a simple text substitution, the system alias file interacts more subtly with the rest of mail delivery.

As in the user alias file, blank lines and lines beginning with a pound sign (#) are ignored. Unlike the user alias file, aliases may span more than one line, and one or more blank lines **must** be inserted between alias. Also, the system alias file **does not** include the full (human) name of the aliased address. The general format of an alias in the system alias file is:

```
    alias:      address1
        address2
        address3 ...

    or

    alias:
        address1
        address2
        address3
```

At least one address must be specified, and only one address may appear per line. The

[1] This file, which has a fixed name, should not be confused with the user alias file described above on page Error: Reference source not found.

address may take one of the following formats:

address Simple address, subject to normal forwarding rules on the local system.

\address An address prefixed by a backslash. Remote addresses are delivered normally, but local addresses are delivered without user id verification or forwarding.

|command An MS-DOS, OS/2, or Windows NT command prefixed by an "or bar" (|). The mail, including the header, is piped into the specified command.

pathname The absolute path name of a file. To be recognized as a file, the file must begin with a slash (/), a tilde (~), or a driver letter/colon sequence (x:). The mail is appended at the end of the specified file normally, with a line of binary ones separating mail items just as if it were a mailbox.

Note: For file names beginning with tilde (~), the file name is expanded as described in **UUPC/extended Command Reference**, page 40.

:include: pathname The indicated file is read for additional addresses. The entire included file is read. Blank lines are ignored in the included file, but unlike the system aliases file blank lines **do not** indicate the alias is ended.

One special rule applies to the above. If a simple local address appears in the system aliases file, then when forwarding rules are applied it will **not** be looked up in the system aliases file a second time. If the same address appears in either a file included by the system aliases file or in a **FORWARD** file the system alias **will** be used. This rule allows a group of systems to all use the same system alias file to forward mail to users spread across the systems but at the same time allows the use of system aliases in mailing lists or personal **FORWARD** files. For example, given the following systems alias file on the systems mine.woods.com, cottage.woods.com, and atlarge.woods.com:

```
dopey:    dopey@mine.woods.com

sneezy:   sneezy@mine.woods.com

doc:              doc@mine.woods.com

swhite:   swhite@cottage.woods.com

  prince: prince@atlarge.woods.com
```

If mail is sent from any of these systems to the user **swhite**, the system aliases file will cause it to be rerouted to swhite@cottage.woods.com. However, once cottage.woods.com receives the mail it will process the alias **once** and then deliver it normally.

The FORWARD filetc "The FORWARD File" \l 3§

As noted above, a user can also use a file to direct how mail inbound to the user is handled. This file must be named **FORWARD** and reside in the user's home directory as defined in the **PASSWD** file or *[userid]*.RC file. The file format is simply one or more addresses on separate lines:

```
address1
```

```
address2
address3
```

These addresses are in the same format as the system alias file addresses described in The System Alias File, above. Blank lines and comment lines with a pound sign (#) in the first column are ignored but do **not** terminate reading of the file.

As an example, the following file, if saved in the user's home directory under the name **FORWARD**, would cause the mail to be formatted to the console via the UUPC/extended FMT program (See **FMT**, page 38.) to be forwarded to the postmaster normally, to be saved in the mailbox for user BOGUS without checking to see if BOGUS exists or has forwarding enabled, and finally to be saved in the user's home directory in the file BKUPMAIL.TXT:

```
|FMT
postmaster
\bogus
~/BKUPMAIL.TXT
```

Controlling Routing for Entire Systems and Subdomains tc "Controlling routing for Entire Systems and Subdomains" \l 2§

Overview of Mail Routingtc "Overview of Mail Routing" \l3§

By default, RMAIL and MAIL combine to deliver mail in the following fashion:

* Mail to a user id without a host name is delivered locally, as described in **An Overview of RMAIL** on page 103.

* Mail to the current system's host name (as defined by the UUPC.RC **NodeName=** line) is delivered locally.

* If the **FromDomain=** line is not defined in UUPC.RC, then mail to the current system's domain name (as defined by the UUPC.RC **Domain=** line) is delivered locally.

  Note: The use of the **FromDomain=** keyword is documented below, as part of **The Ever So English Sport of Site Hiding**, page 110.

* If the **FromDomain=** line is defined in UUPC.RC, then mail to the current system's domain name (as defined by the UUPC.RC **Domain=** line) is queued as mail for the system listed on the UUPC.RC **MailServ=** line.

* Mail to a simple system name listed in the **SYSTEMS** file is queued for that system.

* Mail to a system of the form *system*.UUCP where *system* is listed in the **SYSTEMS** file is queued for that system.

* Mail to a system with a system name listed in the **SYSTEMS** file and the local domain name[1] is queued for that system.

[1] The local domain name is taken from the **LocalDomain=** line in the UUPC.RC file if it exists.

- All other mail is queued for the system defined as the mail server (as defined by the UUPC.RC **MailServ=** line).

You may sometimes need to override the default rules. For example, a system which is not directly connected may still be routed via a system other than the default mail server, or the local system's mail may be passed to a gateway program for delivery on a LAN. UUPC/extended handles such overrides via a **HOSTPATH** file. If used, the file **must** be called **HOSTPATH** and must reside in the UUPC/extended configuration directory.

Within the file, blank lines and lines which begin with a pound sign (#) are ignored. Other lines have one of three formats, with common guidelines:

- The operands must be separated by white space.
- All operands must fit on one line.
- The *victim* (the system for which rerouting is desired) may be specified as a simple host name (dino), a fully qualified domain name (dino.bedrock.bc), or a sub-domain prefixed by an asterisk (*.bedrock.bc) to denote all hosts within a sub-domain.
- A *canonical-name* may be a simple host name or a fully qualified domain name, but it **cannot** be a generic sub-domain definition.

For a entry defining the routing of a system or sub-domain, the syntax is simply:

```
victim      canonical-name
```

Where *victim* is the system or sub-domain to be routed via *canonical-name*. For an entry defining a system or sub-domain as an alias of another system, the syntax is similar but adds a literal equals sign (=) between the two names:

```
victim      =       canonical-name
```

Where *victim* is the system or sub-domain that is to be considered equivalent to the system *canonical-name.* Lastly, when mail for a system or sub-domain is to be piped into a program, the equals sign is replaced by a pipe (|) symbol:

```
victim      |       program-name
```

The use of these entries is described below in separate sections.

Note: When the **HOSTPATH** file is used to alias systems or change default routes, the original system information is **not** changed in the mail header or in commands passed to other systems. This means that the systems defined by *canonical-name* must have at least the same amount of information about routing as the local UUPC/extended system has. (The reason is simple: If you screw with something long enough you will break it, and mail headers are no exception.)

Otherwise, if the **Domain=** value ends in UUCP, UUCP is defined as the local domain, else two level domain names (x.y) are used as-is, and all longer domain names are stripped of the first name (for example, kendra would be stripped from kendra.kew.com) and the rest is used as the local domain.

Routing Mail Via Non-Default Mail Serverstc "Routing Mail via Non-Default Mail Servers" \l 3§

As noted above, mail for other than the local system or directly connected systems is routed to the default mail server. This is desirable when a system only calls a mail server and/or a few other systems, as the local system will automatically route to each of these. However, if any directly connected systems other than the mail server call additional systems, then UUPC/extended must be explicitly told about them. Once these new routes are defined, other systems could be routed via these newly defined systems, and so on.

For example, if the local system calls system fee in addition to its regular mail server, and fee provides the cheapest route to systems fie, foe, and foo, then the following entries would be required in the **HOSTPATH** file:

```
fie         fee
foe         fee
foo         fee
```

Once fie, foe, and foo are defined, additional systems can be routed via them, up to one hundred systems deep:

```
giant               fie
bean.sales.com      foe
*.bean.stalk.com    foo
```

Note the sub-domain reference for *.bean.stalk.com; this defines that all systems that may exist in that domain (low.bean.stalk.com, middle.bean.stalk.com, way.up.there.bean.stalk.com) will **all** be routed via fee, foo, and then on to systems in the bean.stalk.com domain.

Note: If the canonical system name is not known to the local system, it is presumed to be routed via the default mail server. This effectively negates the routing entry in **HOSTPATH**.

Preventing Mail For The Local Domain From Leaving The Local Systemtc "Preventing Mail For the Local Domain From Leaving the Local System" \l3§

A special case of routing is when a route is specified via the local system. If an otherwise unknown system or domain is defined to be routed via the local system, mail for that system is rejected as unreachable and bounced. This allows the gateway for a domain to prevent mail from leaving the domain via a **HOSTPATH** entry like so:

```
*.my.domain.com               localhost
```

If the local system is *localhost*, then mail for otherwise unknown systems within the my.domain.com domain will bounce to the mail server. (Mail for the mail server will itself will be delivered normally, because the mail server is defined in the **SYSTEMS** file.)

Aliasing Systems Via The HOSTPATH Filetc "Aliasing Systems Via the HOSTPATH File" \l 3§

At times, it is not enough to route mail via a particular system. For example, if a system is renamed or has a different domain name than the local domain, then the two names for the system must be made equivalent via the **HOSTPATH** file. For example, to define bull as an alias of cow:

```
bull      = cow
```

If a system is both aliased and explicitly routed, then the route will override the alias.
Aliases for systems are taken into consideration when performing implicit user alias lookups. Thus, if the system bull is aliased as above and a user alias exists for a user at system cow, mail sent to the user at system bull will use the user alias for user at system cow. For a full discussion of implicit user alias processing, see **Implicit User Alias Processing, (or, What Your Polar Bear Never Told You about Aliases)**, page 106.

The Ever So English Sport of Site Hidingtc "The Ever So English Sport of Site Hiding" \l3§

One special combination of routing and aliasing is so common as to not always even require a **HOSTPATH** entry at all. Site hiding, the ability to allow multiple systems to share a single domain name[1] to the outside world, is enabled by the **FromDomain=** keyword in the UUPC.RC file. This keyword overrides the **Domain=** keyword in selected mail headers, and automatically aliases the value of the **Domain=** header to the mail server. For example, if four systems, all in the "forest.oz" domain, share the domain name used by their mail server, then each hidden system would have UUPC.RC entries like:

```
    Domain=lions.forest.oz
    FromDomain=forest.oz
    MailServ=toto

    Domain=tigers.forest.oz
    FromDomain=forest.oz
    MailServ=toto

    Domain=bears.forest.oz
    FromDomain=forest.oz
    MailServ=toto

    Domain=ohmy.forest.oz
    FromDomain=forest.oz
    MailServ=toto
```

Mail for each of the systems would appear to come from forest.oz, and internally

[1] Across the puddle, the rocket scientists controlling domain names in Great Britain in 1992 were charging for each system in a domain hierarchy, which led numerous small sites to cluster themselves under one name. Never mind, that even UUPC/extended§ will handle sub-domains correctly and thus the authorities need not concern themselves with sites within a domain, but what do we Yanks know?

UUPC/extended would automatically alias toto to forest.oz. Note that mail to a simple user id would be expanded to include the **FromDomain** name, which would mean it would also be routed to toto.

Setting up the central server requires a small amount of additional work. Each user on each hidden system must have a corresponding entry in the system alias file on the central server (See **The System Alias File**, page 106.).   You must use the full system name (lions.forest.oz, etc.) or simple system name (lions, tigers, bears, ohmy ...) in the system aliases file, **not** the **FromDomain** which is the same for all systems. Also, the mail server can be defined to be hidden as well, with UUPC.RC entries like:

```
NodeName=toto
MailServ=wizard
Domain=toto.forest.oz
FromDomain=forest.oz
```

However, in this case, you must define forest.oz to be an alias of the local system because otherwise the default would be to forward mail for forest.oz to its mail server, in this example wizard. The alias would take the form of a single **HOSTPATH** entry like so:

```
forest.oz    =         toto
```

Using UUPC/extended As A Mail Gatewaytc "Using UUPC/extended As A Mail Gateway" \l3§

The final use of the **HOSTPATH** file is to direct mail for a system or domain to an external program for processing. This processing is invoked via the or bar (|) separating the *victim* from the canonical name. In this case, the canonical name is actually the program to run. The following arguments are passed to the program:

The *victim* for whom routing is being done.

The actual host name of the addressee.

The user id of the addressee.

The mail to be delivered is read by the gateway program from standard input. For example, to route all mail for the funky.lan.com domain to a gateway program called foogate, all the following to the local HOSTPATH file:

```
*.funky.lan.com   |          foogate
```

Using UUPC/extended Under Windows 3.xtc "Using UUPC/extended Under Windows 3.x" \l 2§

Warning: As this document went to press for release 1.12b of UUPC/extended, work was still being finished on the initial Windows 3.x support for UUPC/extended. The programs included should be viewed as in the final testing phase and not formally released to production. Please report any problems with the Windows version of UUPC/extended to **help@kew.com**.

See **General Advice on Multitasking Environments** on page 114 for various considerations to keep in mind when running multiple copies of UUPC/extended programs on a single system.

Unlike the DOS, Windows NT, and OS/2 versions of UUPC/extended, the Windows version is not self-sufficient; the Windows version primarily exists to support UUCICO in the Windows environment, which alleviates performance problems with the DOS version of the UUCICO program under Windows. In addition, most of the utilities (including UUSUB, UUCP, etc.) are included.

However, the Windows version of MAIL is a straight port of the DOS MAIL program (it does not support pull down menus or dialog boxes), and both MAIL and UUXQT operate more efficiently under DOS than Windows. Thus, binaries for these programs and their support modules RMAIL and RNEWS are not included for Windows.[1]

It is **specifically recommended** that the DOS versions of MAIL, UUXQT, and RMAIL be used in conjunction with the Windows version of UUPOLL, UUCICO, UUSTAT, and UUNAME.[2] To do so, install the Windows supplied programs in the \UUPC\WINBIN directory and when running Windows place the \UUPC\WINBIN directory before the \UUPC\BIN directory in your PATH. This may require altering your path when you enter and exit Windows.

If you use the Windows version of UUXQT or MAIL, you **must** also use the Windows versions of RMAIL and RNEWS. This is because a) UUPC/extended Windows programs cannot detect when DOS programs exit,[3] and b) a special flag (-f) is passed to the RMAIL and RNEWS programs to take the place of file redirection, which is not available under Windows.

[1] Because we **strongly** discourage using the Windows versions of UUXQT, MAIL, RMAIL and RNEWS, they must be built from the available source files using Turbo C++ 3.1 or Borland C++ 3.1.

[2] The gory details: Several UUPC/extended programs such as MAIL and UUXQT invoke child programs such as RMAIL and RNEWS to process data for them. When running RMAIL and/or RNEWS, the invoking program usually redirects the input and output streams (called stdin and stdout), so that instead of reading from the keyboard, the child programs data read and write disk files. Unfortunately, stdin and stdout *cannot* be redirected under Windows, and as a consequence, programs that read from a redirected stdin and stdout under DOS must be restructured for Windows.

[3] An obvious technical failure on our part, but Windows 3.x is not our strong suite. Give us DOS or a 32-bit operating system any day. We're open to suggestions, of course, on how to fix this.

Using UUPC/extended under OS/2 and Windows NTtc "Using UUPC/extended Under OS/2 and Windows NT" \l 2§

UUPC/extended programs run as 16-bit[1] native text mode applications under OS/2 1.3 and as 32-bit native text mode applications under OS/2 2.x and Windows NT. Given enough system resources, either OS/2 or Windows NT can support multiple UUCICO/UUPOLL processes in background without affecting foreground performance. DOS versions of all programs except UUCICO can also be used in the respective DOS environments. The DOS UUCICO (and by extension UUPOLL) should not be used in a DOS box because overhead from simulating the DOS serial port environment can impact transfer speeds and can cause transmission errors (resulting in resent data, also impacting transfer speeds).

> **Note:** Under OS/2, DOS performance problems can be somewhat alleviated by using an OS/2 specific FOSSIL driver (such as the one supplied with the SIO.SYS driver), which uses blocked I/O.

See **General Advice on Multitasking Environments**, page 114, for various considerations when running multiple UUPC/extended programs at once under OS/2 and Windows NT.

Using UUPC/extended on a LANtc "Using UUPC/extended on a LAN" \l 2§

UUPC/extended can be used in a LAN environment via a shared network drive, or using Universal Naming Convention (UNC) share names. In addition, UUCICO has limited direct network support (see **Specialized Communications Drivers** on page 113, below).

However, it is not possible to protect the UUPC configuration directory from the network nodes (MAIL and RMAIL require access to the **SEQF**, **PASSWD**, and **SYSTEMS** files), and thus local users can learn the passwords for remote systems and, in some configurations, read other users' mail. **Network administrators must determine if this is a security exposure**.

> **Note:** Effective with release 1.12a, the SYSTEMS file and PASSWD files can be renamed using the UUPC.RC variables **Passwd** and **Systems**. Thus, two similar UUPC.RC files pointing at different sets of SYSTEMS and PASSWD files can be used; general users can access world readable files with fake password data, and a specially authorized user can access the live SYSTEMS and PASSWD files used by UUCICO to make and accept calls.

Configuring using Universal Naming Convention (UNC) namestc "Configuring using Universal Naming Convention (UNC) names" \l 2§

UUPC/extended can be configured using Universal Naming Convention (UNC) names instead of conventional file names to refer to its configuration files. Different from conventional file names, UNC names look like this:

\\machinename\sharename\path\filename.ext

Where *machinename* is the name of the remote system where the share point is located, *sharename* is the name of the remote share point, and *path* and *filename.ext* are just like normal

[1] 16-bit support for OS/2 1.x will be dropped and 32-bit support phased in late in 1993. Please contact help@kew.com if you have a continued need for 16-bit OS/2 support.

paths and file names. Many different PC network operating systems, including Novell, Lantastic, Windows for Workgroups, and Windows NT, allow you to use UNC names when opening files over the network.

Note: If you use UNC names in your UUPC.RC or *[userid]*.RC files, you must make sure that RMAIL.EXE and RNEWS.EXE can be found in the path, or that the path= line is specified in UUPC.RC to point to the UNC share point and directory where RMAIL and RNEWS can be found.

Example: Some typical lines that might be in a UUPC.RC configured to use UNC names might be:

```
Path=\\dino\uupc\dosbin
ConfDir=\\dino\uupc
NewsDir=\\dino\uupc\news
```

Specialized Communications Driverstc "Specialized Communications drivers" \l 2§

Specialized communications drivers are enabled in a modem file via the **Suite=** keyword as described in the modem file keyword table on page 93 in **Modem ([modem].MDM) Files**

Under MS-DOS, UUCICO supports connections driven by:

1. The internal communications driver (COMMFIFO), which directly controls to the Universal Asynchronous Receiver/Transmitter (UART) driving the serial port.

2. A generic FOSSIL interface which supports any FOSSIL (INT 14) driver which conforms to Version 5 of the FOSSIL specification.

3. An Artisoft Lantastic specific driver for use with the ArtiComm product. This Lantastic utility allows modems on an a LAN to be shared by workstations.

Under Windows, Windows NT, and OS/2, UUCICO will use any serial device supported by the system. In addition, TCP/IP connections over a LAN be made if such support is installed under Windows or Windows NT. Under OS/2 2.x, named pipe connections over a LAN be made if such support is installed.

For connections which UUCICO does not directly support via LAN protocols (i.e. IPX or NETBIOS), RMAIL can be used as gateway to programs that do. See **Using UUPC/extended As A Mail Gateway**, page 111, for details.

General Advice on Multitasking Environmentstc "General Advice on Multitasking Environments" \l 2§

UUPC/extended includes specific support for environments where more than one program may be accessing UUPC/extended files at a time. Some of this support, such as checking the time stamp of a mailbox before overwriting it when exiting MAIL, requires little additional time and is always active. Other support requires special processing to perform and must be enabled via the addition of the **multitask** option to the **options=** line in the UUPC.RC. Under DOS, you should also load the SHARE TSR program.

> **Note:** Under all environments except DOS, the **multitask** option should **always** be enabled in the UUPC.RC file, and may enabled under DOS as desired.

Specifically, the **multitask** option enables:

- If a permission denied error is returned when opening files, UUPC/extended retries for up to 20 seconds waiting for the file to become available.
- Log files are written to temporary files in the spool directory and then appended to the permanent log when the program exits.
- If the **syslog** option is enabled, the **SYSLOG** file is opened and closed for each file transmitted or received.
- UUCICO creates a lock file when connecting to a system to insure only one conversation can go on with that system at time.
- UUXQT creates a lock file when processing commands from a system to insure only one UUXQT is processing commands from that system at a time.
- When mail is read from the system mailbox for the user, it is copied to the user's home directory and the mailbox is cleared to allow additional deliveries while the user is reading mail.

This additional processing prevents conflicts such as multiple programs writing to log files or mailboxes, and attempts to insure programs do not abort when files such as the **SEQF** file are being updated and are thus inaccessible. The only major remaining exposure is that a program can explicitly write to a mail file in a user's directory, but this is uncommon and MAIL will detect when the file size or date changes and warn the user before losing the other program's changes.

Passive polling: making other systems do the work by calling youtc "Passive polling" \l 2§

What if you want to have other systems dial in to yours to route mail and news, rather than calling them? To configure UUPC/extended to allow other systems to call yours, you will need to modify several of the UUPC configuration files.

In addition, the owners of the other systems will have to modify configuration files on their machines. Describing how to do that is beyond the scope of this documentation, but descriptions of how to configure many other systems can be found in *Managing UUCP and Usenet*.

To allow a remote system to dial in, you'll need to do the following:

A. Modify your SYSTEMS, PASSWD, and PERMISSN files.

B. Test your machine's new configuration.

C. Test having their machine call yours.

Example: Let's assume that one of your UUCP neighbors, whose machine is named "bosox," decides that she wants to be able to dial in to your machine, called "snuffles."

Modifying your SYSTEMS, PASSWD, and PERMISSNS filestc "Modifying your SYSTEMS, PASSWD, and PERMISSNS files" \l 3§

We'll go through the changes that need to be made to each of these files one at a time.

1. SYSTEMS

In your SYSTEMS file, you will need to add a line for the remote system.

If you already successfully call the other system, then that line in your SYSTEMS file is good enough: You don't need to make further changes.

If you don't already call the remote system, then add a line to your SYSTEMS file that looks like this:

Example:

```
bosox Never TBWORLD 38400 999-9999 gG
```

Replace bosox with the name of their machine.

As described in **4**, page 22, the second entry on this line describes how often the remote system is being called. "Never" means that you never call the remote system. Because you never call them, the modem type, outgoing baud rate, phone number, and protocol fields are not used. Despite that, they still have to be filled in.

2. PASSWD

You will also need to add a line to your PASSWD file for the remote machine. For more information about the PASSWD file format, see the documentation in the file itself, or in step **5** of **Installation**, page 24.

As documented in the PASSWD file itself, the PASSWD line you'll want to add will look something like this:

Example:

```
Ubosox:Apple#Juice:::bosox
```

Replace Ubosox with their dial-in user name, Apple#Juice with their password, and bosox with the name of their machine.

3. PERMISSN

For a simple configuration, add a couple of lines like the following to your PERMISSN file:

Example:

```
LOGNAME=Ubosox VALIDATE=bosox MACHINE=bosox \
 SENDFILES=yes REQUEST=yes COMMANDS=rmail:rnews
```

Replace Ubosox with their dial-in user name from your PASSWD file, and bosox with their machine name.

Test your machine's new configurationtc "Test your machine's new configuration"\l3§

To test your new configuration, run UUNAME, and then UUXQT. If they do not print out any warnings, then you have probably configured your PASSWD, SYSTEMS, and PERMISSN files correctly.

Test having their machine call yourstc "Test having their machine call yours"\l3§

Once your neighbor has configured their machine, they can try calling in. If they are using UUPC/extended, they need only run UUCICO or UUIO:

UUCICO -s *yoursystem*   or
UUIO -s yoursystem

If they are not running UUPC/extended, they should consult their system's documentation.[1]

Using FOSSIL communications drivers with UUPC/extendedtc "UUPC/extended and FOSSIL Drivers" \l2§

As of version 1.11z, the DOS version of UUPC/extended supports using a special type of communications driver called a FOSSIL (Fido-Opus-Seadog Standard Interface Layer). A number of drivers exist supporting the FOSSIL standard, and these drivers are used by a wide variety of PC bulletin board programs, particularly in the FidoNet electronic mail and bulletin-board world. Because of this large installed base, FOSSIL drivers are robust and have been tested on a variety of systems.

While the internal COMMFIFO driver handles most serial ports with no trouble, FOSSIL drivers allow UUCICO to run on an even wider variety of systems. FOSSIL drivers can be used with UUCICO to support any of the following situations:

- FOSSIL already is installed for other programs
- Special serial port configurations such as non-standard interrupts used for COM3 or COM4
- UUCICO using COMMFIFO cannot initialize the modem
- Running DOS UUCICO under OS/2

Note: Kendra Electronic Wonderworks does not endorse or support any particular FOSSIL driver.

Here's how you configure UUPC/extended versions 1.11z and later for use with FOSSIL drivers.

[1] As usual, *Managing UUCP and Usenet* has many useful suggestions on various system behaviors.

Materials needed:

- A FOSSIL driver
- A text editor
- The latest version of PKUNZIP

Steps to perform:

1) Get a FOSSIL driver.

2) Install the FOSSIL driver.

3) Edit your .MDM file, adding Suite=FOSSIL.

4) Reboot your system if CONFIG.SYS was edited.

5) Test the results.

Instructions:

1) Get a FOSSIL driver. These are available from many sources, including by anonymous UUCP from kewgate, and by anonymous FTP from any Simtel-20 mirror site, including wuarchive.wustl.edu and ftp.uu.net. Three drivers tested with UUPC/extended are:

BNU170.ZIP
OCOM_531.ZIP
X00V124.ZIP

Other FOSSIL drivers should work as well.

2) Install the FOSSIL driver.

The FOSSIL drivers all come with installation instructions. You will have to tune them for use with UUPC/extended, however. If you have a high-speed modem (9600 baud or greater), and you use large window-size protocols like G or v, you will need to install your FOSSIL driver with larger transmit and receive buffers than the defaults. The largest size you should need is 4 KB.

See the FOSSIL driver's instructions for further details.

Examples:

For a high-speed configuration, here are the settings our friend Fredrick Watt used for installing each of the three tested FOSSIL drivers on tosci's:

BNU170: Nothing was added to CONFIG.SYS. In AUTOEXEC.BAT, Fred added the BNU directory to the path and then added:

```
BNU /T:4096 /R:4096 /F- /Z0
```

X00V124: Nothing was added to AUTOEXEC.BAT. In CONFIG.SYS, Fred added:

```
DEVICE=C:\X00\X00.SYS E F=15 R=4096 T=4096
```

OCOM_531: Nothing was added to CONFIG.SYS. In AUTOEXEC.BAT, Fred added the directory with the driver to the path and then added:

```
OPUSCOMM
OCOM_CFG C S2=4096,4096
```

Note: S2=4096,4096 set the comm buffers for COM2 *only* -- your setting may differ depending on your communications port. Read the instructions for the driver.

3) In tosci's .MDM file, add the line **Suite=fossil**.

Example: In tosci's TBWORLD.MDM file, Fred added the line **DOS.Suite=FOSSIL**. Since toscis runs more than one operating system, Fred needed **DOS.Suite** so that OS/2 and Windows NT still use the internal serial port drivers. If you don't use additional operating systems, there is no need for the **DOS.** prefix.

4) Reboot your system.

If everything was configured properly, you'll get a sign-on message from the FOSSIL driver you installed.

5) Test the results.

Call one of your neighbors with the new driver, and enjoy.

When things go wrongtc "When things go wrong"\l3§

Check the buffer sizes the driver is installed with. UUCICO, which can queue 4K of data at once, can easily overrun the standard buffer sizes used by FOSSIL drivers.
The FOSSIL driver documentation is often helpful. In addition, several of the FOSSIL driver packages come with diagnostic programs that display the status of various RS-232 lines, so you can see whether data is actually getting sent.

Changes From Previous Versions

Appendix 1: How to Get UUPC/extendedtc "How to Get UUPC/extended"§

Introduction

UUPC/extended is a PC based version (and pun of) UUCP (UNIX-to-UNIX copy). UUPC/extended is a program which implements peer-to-peer networking for using the UNIX UUCP protocols.   MS-DOS or OS/2 based Personal Computers using these protocols can exchange mail and files with a UNIX system or other UUPC/extended systems over dial up or dedicated phone lines.

UUPC/extended 1.12b is available via:

Automated mail server running at Kendra Electronic Wonderworks.
Clarkson University BBS
Anonymous UUCP from kewgate, Kendra Electronic Wonderworks's file server
BIX
CompuServe
Ordering floppies by mail from Crynwr Software
Registering your current copy

Note: Because of time and disk space restrictions, we do not manually e-mail copies of the source or executables, provide custom .ARC or .ZIP files with only a limited set of the files, or provide other custom distribution services-- please don't ask.

Instructions

Anonymous FTP

UUPC/extended is available from the following anonymous FTP site:

ftp.clarkson.edu, IP address 128.153.4.2, directory pub/uupc

FTP programs vary from system to system; contact your local system support staff for instructions on anonymous FTP.

Kendra Electronic Wonderworks archive server

Send a mail message containing the commands "index" and/or "help", one command per line, in the body of the message to listserv@kew.com for the current file list and/or instructions.

Clarkson University Bulletin Board

The Clarkson University Bulletin Board is reached 24 hours a day at (315)268-6667, 8N1, 1200/2400 Baud. Follow the instructions after logon to download files.

Anonymous Kermit

Alas, because of conflicts between UUCICO and C-Kermit under OS/2, kendra's anonymous

Kermit server has gone to the great lily pad in the sky. Use anonymous UUCP, below, instead.

Anonymous UUCP

The UUPC/extended files are archived in the directory "~nuucp" on kewgate and can be downloaded via anonymous UUCP. To use anonymous UUCP, define entries for the system kewgate in your SYSTEMS file and PERMISSN file. The SYSTEMS file entry should define the phone number as 1-617-641-4817, the user id as "nuucp", and the password as "nuucp". kewgate uses a Zoom Telephonics V.32bis modem, so connect speed may be up to 14400 baud, depending on your modem. kewgate presently accepts anonymous UUCP phone calls 24 hours a day, seven days a week, except for interruptions for system maintenance. We reserve the right to change these hours without notice.

> **Note:** The sample systems file in the documentation archive includes an entry for kewgate.

> **Note:** You cannot *send* mail or files to kewgate via the anonymous dial-in. Send mail to help@kew.com, our normal email address.

The files are located in the home directory for nuucp, so files should be requested via the UUCP command with the syntax ~nuucp/filename. For example, to retrieve the index for the archive to your local /tmp directory, issue the command:

```
uucp  kewgate!~nuucp/index  /tmp/
```

The UUCICO program must then be invoked to actually process the request. Users should retrieve the file index first, and then issue additional requests for the actual files wanted based on the current directory listing.

> **Note:** Do **not** use a debug level greater than 3 when calling kewgate. To avoid filling the hard drive on kewgate with useless debugging information, the system will automatically hang up on you.

BIX

Douglas Hamilton of Hamilton Laboratories posts UUPC/extended to the ibm.os2/listings area on BIX. For additional information, contact him (userid hamilton) on BIX. Tell him Snuffles sent you.

CompuServe

The archives are loaded to CompuServe's UNIX Forum (GO CIS:UNIXFORUM). The files can be searched for within the forum using the keyword UUPC. As not every release is loaded to CompuServe, contact the UUPC/extended Help desk (from CompuServe, >Internet: help@kew.com) before downloading files from CIS which are more than two months old.

Because of CompuServe's six character limit on file names, the files uploaded to CompuServe do not follow the normal UUPC/extended naming convention described below. Rather, the files are named UUPCx.nnn, where 'x' is the archive suffix as described below and 'nnn' is the version number. Files should be downloaded as UUPCnnnx.ZIP to restore their normal DOS names.

Registering your current copy

To receive a copy of the next major release when it becomes available, register your current copy according to the instructions in **How to Register UUPC/extended**. You can also order the current release from us using the form in **How to Register UUPC/extended**.

About the files

The available UUPC/extended files include:

Name	Description
index	The current directory listing for the UUPC/extended archive. (Only available on kewgate and the various Clarkson systems)
upc12bad.zip	Formatted documentation files for UUPC/extended
upc12baw.zip	Word for Windows documentation source files for UUPC/extended
upc12bap.zip	Documentation files formatted for a PostScript printer
upc12bd1.zip	Executable files for MS-DOS, 1 of 3
upc12bd2.zip	Executable files for MS-DOS, 2 of 3.

upc12bd3.zip	Executable files for MS-DOS, 3 of 3.
upc12bw1.zip	Executable files for Windows 3.x, 1 of 3.
upc12bw2.zip	Executable files for Windows 3.x, 2 of 3.
upc12bw3.zip	Executable files for Windows 3.x, 3 of 3.
upc12b11.zip	Executable files for 16 bit OS/2 (version 1.x), 1 of 3.
upc12b12.zip	Executable files for 16 bit OS/2 (version 1.x), 2 of 3.
upc12b13.zip	Executable files for 16 bit OS/2 (version 1.x), 2 of 3.
upc12b21.zip	Executable files for 32 bit OS/2 (version 2.x), 1 of 3.
upc12b22.zip	Executable files for 32 bit OS/2 (version 2.x), 2 of 3.

upc12b23.zip	Executable files for 32 bit OS/2 (version 2.x), 3 of 3.
upc12bn1.zip	Executable files for Windows NT for Intel x86, 1 of 3.
upc12bn2.zip	Executable files for Windows NT for Intel x86, 2 of 3.
upc12bn3.zip	Executable files for Windows NT for Intel x86, 3 of 3.
upc12bs1.zip	Source files for UUPC/extended, 1 of 2.
upc12bs2.zip	Source files for UUPC/extended, 2 of 2.
howtoget.txt	This file. (Only available on kewgate and various Clarkson systems)
announce.txt	A summary of the newest release. This file is generally an abridged version of the current CHANGES.PRN file included in the documentation archive. This file is not created if the current release has limited changes. For all releases, CHANGES.PRN is the definitive summary of changes. (Only available on kewgate and various Clarkson systems)

Note: MS-DOS users will want upc12bd1.zip, upc12bd2.zip, upc12bd3.zip, and upc12bad.zip.

Note: OS/2 users will want upc12b21.zip, upc12b22.zip, upc12b23.zip, and upc12bad.zip.

Note: Windows NT users will want upc12bn1.zip, upc12bn2.zip, upc12bn3.zip, and upc12bad.zip.

The files names change with each release to include the current release number in the file name. For example, the source archive for version 1.24b would be upc24bs1.zip.

From time to time, test releases are loaded to kewgate and/or the various Clarkson systems. These releases are placed in a sub-directory on the Clarkson file server and in the same directory as the production directory on kewgate.

Formatted Documentation Archive Contents

The documentation archive includes both the formatted documents describing UUPC/extended and various configuration files. Most of the configuration files are samples and must be modified to be used. The files in the documentation archive are as follows:

README.PRN provides an overview of UUPC/extended.

REGISTER.PRN describes how to register UUPC/extended to encourage us to keep putting out this insanity.

Note: You do **not** have to register to UUPC/extended to use it or give it to others.

INSTALL.PRN tells how to install or upgrade UUPC/extended once you have read README.PRN.

COMMANDS.PRN summarizes the commands supplied with UUPC/extended, information on how to use most of them, and instructions on where to find information on the other commands.

HOWTOGET.PRN (this file) describes how and where to get a new version of UUPC/extended.

CHANGES.PRN is a list of recent changes to UUPC/extended (fixed bugs and enhancements).

MAIL.PRN is the documentation file for the MAIL command.

TILDE.PRN is used to generate a text file printed by the help (~?) command when sending mail.

UUSTAT.PRN is the documentation for the UUSTAT program.

BUGS.PRN is a list of known bugs in UUPC/extended.

UUPC.RC defines the system information related to UUPC/extended, such as the system name and work directories.   Only one UUPC.RC file should exist per PC.

PERSONAL.RC defines the user information related to UUCP, such the user id, (mailbox) of the user, the user's full name, and the directory to store mail in. If several people share the same PC, then multiple PERSONAL.RC files may exist under different names.

PERSONAL.SIG is a sample signature file, which is appended to each mail file you send. Use of this file is optional, but highly recommended.

SYSTEMS defines the telephone numbers and login information for other computers the local PC can call.

PASSWD defines both remote systems that can call the local PC and all local users.

PERMISSN defines access for remote systems accessing your system via UUPC/extended. It is based on the format of the UUCP PERMISSIONS file.

SAMPLE.MDM is a heavily commented modem configuration file which explains the use of the various fields in modem configuration files. Other modem files are also included.

HOSTPATH defines routing for hosts that are not directly connected to the PC.

ALIASES.TXT defines short aliases for common addresses, allowing a short nickname to be used in place of the full name and mail address of a person.

Program Archive 1 of 3

The following programs are included in the UPCxxxD1, UPCxxxW1, UPCxxx21 and UPCxxxN1archives:

MAIL.EXE is the mail user agent (MUA); it is used for both sending and receiving mail.

UUCICO.EXE acts the mail transport agent by exchanging files with other systems via the serial port (modem). It can both call out (active polling) and wait for another system to call (passive polling).

UUXQT.EXE processes requests received via UUCICO, invoking RMAIL.EXE or other programs to process data. It should be run after each UUCICO.EXE invocation.

RMAIL.EXE is called by both MAIL.EXE and UUXQT.EXE to deliver mail; as such, it must be called RMAIL.EXE.

Note: There are no .COM files in the OS/2 or Windows NT archives; all the OS/2 and Windows NT files have .EXE extensions.

Program Archive 2 of 3

The following programs are included in the UPCxxxD2, UPCxxxW2, UPCxxx22 and UPCxxxN2 archives:

FMT.COM is a simple program to format text by wrapping it after 72 columns.

GENSIG.COM randomly appends quotes or other user selected text to the end of a fixed signature file.

NOVRSTRK.COM is a program which removes overstrikes from files for viewing on a terminal. It's use is described above, and is optional.

UUPOLL.EXE serves as a driver for UUCICO and UUXQT, invoking UUCICO to answer the phone and/or dial out on regular basis and then automatically invoking UUXQT to process any files received; this permits unattended operation of UUPC/extended, allowing it, for example, to call another system in the middle of the night.

UUCP.COM copies binary files between two systems without going through mail.

UUSTAT.COM is a utility to report on and update the status of jobs queued for other systems.

UUSUB.COM reports and optionally clears statistics on the amount of data transferred between systems.

SU.BAT is a sample command file for changing the active user.

UUCLEAN.BAT is a sample command file which deletes old logs and temporary files from the UUPC/extended spool and temporary directories.

UUIO.BAT is a sample command file for running UUCICO and UUXQT in sequence.

Note: There are no .COM or .BAT files in the OS/2 and Windows NT archives. All the OS/2 and Windows NT files have .EXE or .CMD extensions.

Program Archive 3 of 3

The following programs are included in both the UPCxxxD3, UPCxxxW3, UPCxxx23 and UPCxxxN3 archives:

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.

EXPIRE archives and/or deletes articles created by RNEWS in normal mode after a specified period of days.

Source Archives

The source archives include all of the files needed to build UUPC/extended under DOS using Borland C++ 3.1. The program can also be built under DOS or OS/2 using Microsoft C 6.0. Microsoft Macro Assembler is required if building the DOS modules with the Microsoft C compiler.

> **Note:** The upc12bs.zip file should be PKUNZIPed with the unpack directory (-d) option.

Unformatted Documentation Archive

The unformatted documentation archive includes all of the Word for Windows (2.0) files and templates used to create the formatted documents in the regular documentation archive.

Other Trivia

Special Thanks

Russ Nelson and Keith Petersen have aided us in getting many of the past releases out to the public. Thanks Clarkson University's Educational Resources Center, with special attention to Brad Clements and Bob Barringer, for providing upload and anonymous FTP facilities.

A final thanks to Shawn McKay and Alan Steele of the Massachusetts Institute of Technology and eddie.mit.edu for handling kendra's Internet mail forwarding (including all those questions to help@kew.com).

Questions and Comments

Please direct all questions on retrieving UUPC/extended to help@kew.com. Within the United States, we will help you get a current copy of this document if you call us at our voice number, (617) 641-3452.

Appendix 2: How to Register UUPC/extendedtc "How to Register UUPC/extended"§

Why You Should Register UUPC/extended

Kendra Electronic Wonderworks now offers the opportunity for you to register your copy of UUPC/extended. Here's why:

> Three years of work have gone into it here at the Wonderworks. This is a non-trivial amount.
>
> Most of the e-mail received outside the KEW.COM domain is now related to UUPC/extended. This, combined with the major intra-domain e-mail requirements[1] for KEW.COM having gone away, threatens to make UUPC/extended its own perpetual motion machine: we maintain UUPC/extended to stay in touch with the outside world, and the outside world sends us mail about maintaining UUPC/extended.
>
> It is an accepted fact that the original UUPC was not written by Kendra Electronic Wonderworks, and that others have written numerous features. What is not as clear is the work done here which goes into integrating and polishing these features. Because of work done here, people can take the work done all over the world and use it by unpacking a few files.
>
> Ryder rents trucks, Avis tries harder, MIT wins the Yale/Harvard game, and Kendra Electronic Wonderworks hunts bugs. Lots of bugs: bugs from the dawn of UUPC, bugs from other people's code, and even bugs we introduced (OOPS). A second system is maintained here primarily for providing downloading services to others via anonymous UUCP and for hunting bugs. Only RAID kills more bugs. We also integrate bug fixes from other people, so those few files we give to others to unpack not only have more features but also bug fixes.
>
> Bug hunting can be a bitch, and writing documentation is worse. Kendra Electronic Wonderworks has done both gratis for two years. Beginning with release 1.11m, the documentation has been expanded, and will be further enhanced to make the package easier to use. This does not benefit the internal users; we know how it works, and don't need the documents.
>
> Because if you register, we can buy more chocolate ice cream for Snuffles.
>
> The short version of the above is you have the package; what you can optionally pay for is additional enhancements and support by the creatures most knowledgeable about UUPC/extended. **However,** we would rather you used an unregistered copy of our package than used someone else's inferior package.

[1] The owner of kendra.kew.com married the owner of athena.kew.com in October, 1992. The transcontinental UUPC/extended link previously used by the happy couple has been replaced by an Ethernet. Send congratulations to binkley@kew.com, our resident ether bunny.

And as an added bonus

Send us anything and Snuffles will think you're wonderful.

Send us $20 and you can call us for telephone support at the UUPC/extended support telephone number [(617) 641-3452] for 18 months. We'll even return your phone calls in the United States.

Send us $40 and you can use the support number for 18 months. When available, we'll also mail the next major release of UUPC/extended on floppies.

Send us $50 and you can use support number for 18 months, we'll mail the next major release of UUPC/extended when available, and we'll throw in a hard copy of the documentation.

Send us $100 and you can use the support number for 18 months, we'll mail the next major release of UUPC/extended, throw in a hard copy of the documentation, and have you committed to the sanitarium of your choice.

Corporations cannot be committed to sanitariums, but we will take your money. Use the above schedule for each contact person, and add $10 for each additional node supported. For example, XYZ Systems Corp. has one contact person who calls the Wonderworks for support and needs the next release and hard copy documents ($50), and one backup contact at a secondary location ($20). They want the docs so that they can distribute them to 10 different UUPC nodes scattered around the company ($10 each). The primary contact person figures out how to get a contribution to Snuffles' Chocolate Ice Cream Fund ($1) past accounting. XYZ Systems would thus send us a check for $171.

We will invoice corporations on request. Corporations wishing to register 100 or more nodes should contact us for large site pricing.

Disclaimer of warranty

Please note that **none** of the above contradicts the No Warranty clause of our license agreement. Should UUPC/extended prove defective or unsuitable for your purpose, we will gladly refund your registration fee. We will not, however, be liable for any incidental or consequential damages resulting from your use of the package. Please read our license agreement (**Licensing, Redistribution, and Warranty**, available upon request) for further details.

The Form

Please fill in **both pages** of the form, and enclose a check or money order in US dollars on a US bank payable to Kendra Electronic Wonderworks**.** Do not enclose cash. We do not accept credit card orders. Mail the completed form to:

Kendra Electronic Wonderworks
Post Office Box 132
Arlington, MA 02174-0002 USA

User Information

We can't provide support services without this information, and we promise not to sell or give it to anyone else. Please give us your:

Name: ____

Company: ____

Street: ____

City: ____

State/Province: ____

Country: ____

Postal Code: _____________ Phone: ______________________

E-mail address: ____

Description	Amount	Quantity	Total
(A) Basic support services	$20.00		
(B) Next release on diskette (includes A)	$40.00		
(C) Documentation (includes A and B)	$50.00		

(D) Additional nodes	$10.00	
(E) Current release on diskette (does not include A)	$20.00	
Overseas postage (per item B or C above)	$5.00	
Snuffles' Chocolate Ice Cream Fund	$1.00	
Total enclosed		

Binkley's Use Only

System Information

We need this information to keep track of the environments UUPC/extended is running in so we'll know what enhancements might be useful. We also can't send you a floppy if we don't know what size or operating system.

Operating System

(check all that apply)

OS/2 version ___ MS-DOS version ___

Microsoft Windows version ___ DR-DOS version ___

Microsoft Windows NT version_____ DESQView version ___

Other ________________

Diskette Size

(Check preferred size. For multiple copies, give number of each size. No check mark gets potluck!)

__ 360K (Low Density 5 1/4 inch)

__ 720K (High Density 3.5 inch)

__ 1.44M (High Density 3.5 inch)

Processor

__ 8088/8086 __ 80286 __ Other 16 bit processor

__ 80386SX __ 80386DX __ Other 386 family

__ 80486SX __ 80486DX __ Other 32 bit processor

Modem

Brand: ____

Speed: __ 1200 BPS or less __ 2400 BPS
__ 9600 BPS ___14400 BPS or higher

Protocols supported: ____

Network

Brand: ____

Protocols supported: ____

Comments:

____

____

____

____

Changes From Previous Versions

Appendix 3: Changes From Previous Versionstc "Changes From Previous Versions"§

This file summarizes most changes made to UUPC/extended since release 1.11q. Please contact Drew Derbyshire, help@kew.com, for information on changes previous to the scope of this document or other questions.

Versions 1.12a through 1.12b Revision Summary

David M. Watt <dmwatt@smersh.cambridge.ma.us> tested many of the new functions in the release and provided numerous bug fixes, including all the Windows NT fixes. Kai Uwe Rommel <rommel@ars.muc.de> also provided many bug fixes, and keeps threatening us with more, especially for RNEWS. Special credit for 1.12a also goes to Robert Denny <denny@alisa.com> for the Windows support discussed below.

Bug Fixes

The mail routing functions cheerfully routed mail to a non-local system by writing mail in an invalid spool directory if routing overrides existed for the target system. Corrected various interactions in file ADDRESS.C to clear up the improper routing and an associated problem with the non-detection of aliased systems.

RMAIL sometimes throws an internal error (reporting not enough mail delivered) after calling Bounce() to bounce mail. Corrected by insuring Bounce()'s deliveries are counted.

RMAIL would fail to bounce mail properly if the program had been renamed to other than RMAIL. Corrected by saving the program name for recursive invocations by Bounce().

The MAIL short help (?) command incorrectly reported that the minus sign (-) was an alias for the non-existent "previous" command when in fact it is an alias for the alive and well "up" command.

When running MAIL, users defined in the PASSWD file with no name would be entered into the local nickname table anyway, causing their name to appear as "????" when replying to them. Changed LoadAlias() to suppress such addresses.

If the UUPC/extended configuration directory was the root directory of a drive, such as a network drive set aside for UUPC/extended, then derived directory names were incorrectly made relative directories by omitting the first slash (/) in the name. Corrected normalize() to not drop the required slash.

If the run time library _fullpath() function failed to return the full path of a file, normalize() returned a NULL even though its callers did not check for this special value. Corrected normalize() to call panic() if the full path of a file could not be determined.

In version 1.11z **only**, if a file was transferred to other than the spool directory (normally using the UUCP command on the local or remote system), UUCICO changed directories for the transfer and then did not restore the original working directory. This resulted in files for mail and news being delivered to the Public directory rather than the Spool directory. Corrected in expand_path() to pop all directory pushes.

If a file to be opened for reading was temporarily busy and the **multitask** flag was active, FOPEN() failed to wait for the file to become free. Corrected to spin for the same length of time as a file to be opened for writing.

Temporary files names were always generated starting with a sequence number of zero. This guaranteed extra overhead when multitasking because of duplicate file names. Corrected mktempname() to begin sequencing file names with the program's process id obtained from getpid().[1]

If the sequence file for the spool directory job identifier was unavailable, the job id generated was always 1; this could cause duplicate jobs to be generated in a multitasking system. Corrected getseq() to use the program's process id obtained from getpid() for the next sequence id if the sequence file was unavailable.

A known host in the local domain was not recognized if the case of the host's domain did not match the local domain case. Made host comparison case insensitive in checkname().

Under Windows NT, the host status file was generated with a different format than under DOS. Corrected via MS C #pragma statement in HOSTABLE.H to pack data structures.

Under Windows NT, attempts to abort any UUPC/extended program which was trapping Ctrl-C, prompting for permission to abort, and which had its standard input redirected met with dismal failure. Corrected in SAFEIO.C and SAFEOUT.C by explicitly opening the console to ask permission to abort.

If a program could not open a log file, it reported this interesting fact with a NULL pointer for the file name. Corrected to use the real file name.

Under OS/2, over length invalid mailbox names could be generated if the **directory** option was set. Corrected by limiting the mail box name to eight characters in mkmailbox().

Non-existent directories listed in the PERMISSN file caused security initialization to abort for UUCICO or UUXQT, even when those directories would be automatically created by UUCICO. Changed messages in InitDir() for such missing directories from errors to warnings.

The creation time of files was incorrectly reported by stater(), affecting when news files were expired after a restore of a directory under OS/2. Corrected stater() to return the last update time.

UUCICO prioritized outgoing packets over incoming acknowledgments, resulting in a possible backup of requests to resend data. gsendpkt() is corrected to always look for at least one acknowledgment. (Fix by Robert Denny.)

If invoked without a time to terminate, UUCICO printed a nonsense time for when it would exit. Corrected message in callin() to indicate UUCICO would run until explicitly killed.

[1] This call actually works out to be the address of the program under DOS. However, this at least allows programs invoking one another (RMAIL from MAIL, for example) to use different sequnce numbers.

UUSTAT loaded the host status file before processing its command line options, making it difficult to debug problems detected during the loading. Moved the host status load to after the options are parsed.

The OS/2 version of UUCLEAN.CMD failed to age files if one or more generations of the file was under the length to age files. Corrected to merge larger files with the under length files.

The login script processor wrote to the communications driver layer one character at time. While this was as efficient as any other method with the DOS COMMFIFO driver, it requires additional system calls for blocked I/O drivers such as FOSSIL, OS/2 and Windows NT.   It was highly inefficient for network communications drivers such as TCP/IP or named pipes. The script processor has been modified to buffer writes unless the user has requested a delay between each character.

Errors caused by multiple programs trying concurrently open the same log file were corrected.

Applied patch to Borland C++ 3.1 library to cause system stat() call to release memory after use. Also added new debugging information to ndir.c to assist tracing network related failures of UUXQT.

UUX failed to format commands properly for execution on remote UNIX systems. Corrected by Richard H. Gumpertz, <rhg@cps.com>.

> **Note:** For another UUPC/extended system to execute most UUX commands, UUXQT must perform additional processing which is not yet implemented. Thus, the UUX command remains unsupported.

Enhancements

Katherine Derbyshire, Dave Watt, and Drew Derbyshire rewrote and expanded the documentation into a large manual.   The simplified installation is documented with additional examples, various configuration file formats are documented in a new section, a new section on advanced configuration topics has been added, and the entire manual has been reorganized. Please send comments on the documentation to docs@kew.com.

Kai-Uwe Rommel supplied code to cause a 16 Bit OS/2 UUCICO listening on port to give up the port to either another UUCICO processing dialing out. The port can also be released and restored through a newly supplied utility program, UUPORT.

Dave Watt wrote TCP/IP support for Windows 3.x and Windows NT.

Ported OS/2 version to 32 bit API's using Borland C++ for OS/2. The 16 bit version continues to be available.

Added "t" protocol for TCP/IP connections.

Added OS/2 named pipe support.

Added support for Windows 3.x via Borland C++ 3.1 EasyWin environment provided by Robert

C. Denny <denny@alisa.com>. Changes include native Windows communications driver for UUCICO, Windows external program invocation support, Windows directory search functions, Windows delay functions, and numerous small changes in support of these enhancements. Additional changes including required MAKEFILE support, isolating external program invocation, support for **Win16.** prefix in configuration files, and other minor clean up performed by Drew Derbyshire. Also added new Boolean option **windows**, which when used in Windows environment causes RMAIL and RNEWS to be invoked with file input (-f) flag to allow invoking without file redirection.

Added support for Universal Naming Convention (UNC) path names and Lantastic Networks. Specifically, normalize() is modified to allow double leading slashes indicating a UNC name and to pre-pend the logical UNC name as required to file names. In addition, various file opens are modified to revert slashes to backslashes for system and/or routines which cannot handle forward slashes in names.

Added new -B command option to UUPOLL. When used, the specified command is executed before each active poll out. If this command exits with a non-zero return code, UUPOLL itself immediately exits.

UUPC/extended distribution archives are now being packed with the public domain ZIP 1.9 program. Previously, the first generation PKZIP was used. Due to the explosion in the number of versions, the archives have also been renamed as documented in the revised HOWTOGET to include a letter identifying the environment each archive is for.

A new documentation archive with the documents pre-formatted for a PostScript printer now exists.

Long names are now better supported under OS/2 and Windows NT HPFS and NTFS file systems.

Added (actually further modified undocumented) supported for Universal Naming Convention (UNC) names.

Version 1.11z Revision Summary

Bug Fixes

When UUSUB was invoked with the clear (-c) option, it wrote the cleared status file to the **SpoolDir** rather than the new directory for the file, the **ConfDir**. Corrected this oversight in hostrset.c. Also corrected reset of start of the collection time.

RNEWS **snews** mode processing failed to process input files in binary mode. Corrected by forcing **all** input to be read in binary mode.

Corrected expand_path() so that path names beginning with tilde (~) need not end with a slash (/). This mostly affects UUCP file transfers.

Corrected scrsize() to sanity check BIOS values for the screen size on systems running versions of DOS previous to DOS 4.0. Previously, invalid screen sizes were not trapped on such older systems, causing the internal pager to require input after every line.

Corrected address parser to include the percent sign (%) in an address as an indicator of a non-local address. Previously, such addresses were handled as invalid local users.

Corrected *bounce()* processing to not recursively deliver mail to an invalid address such as the local postmaster.

Corrected system alias processing to use configuration directory as current directory rather than a possibly invalid user home directory pointer. Also inserted altered system alias processing to call expand_path() to allow using ~user syntax in system alias include files.

Correct RMAIL to not use a debug level less than zero when initializing, which caused configuration failure messages to be suppressed (resulting in RMAIL beeping and dying without offering a clue as to why).

Corrected bug in UUPOLL which caused the program to spin for one second spitting out messages about waiting for zero seconds before running the daily clean up command.

Bill Plummer provided fix to **COMMFIFO.ASM** serial port driver to provide delay after selected modem command operations to allow slow modems to respond.

Corrected bug in **syslog** processing which caused the file to be prematurely closed and never reopened if **multitask** was not set.

When UUX generated a command for a remote system, it requested the command file be sent before any data the command needed. Corrected this to sent the data first.

Enhancements

Added -t option UUCICO to cause tracing of data received or sent to various communications drivers to the file LINEDATA.LOG in the Spool directory. Previously, an undocumented feature would cause this tracing to be performed if the file already existed.

Added multiple parallel communications drivers support to **UUCICO**, and added new drivers for via FOSSIL and ArtiComm controlled ports under DOS. The FOSSIL support is enabled by installing a FOSSIL port driver (not supplied) and adding the keyword:

> **suite=fossil**
>
> To the modem (*.MDM) file for the affected port. Likewise, an ARTICOMM supported port is enabled by providing a **suite=articomm** keyword in the modem file for the port.

Rewrote **syslog** processing to display requester of a file, the remote system involved, and host (UNIX format) name. The local (DOS) name is dropped from the log file. [Many thanks to Mark W. Schumann <mark@whizbang.wariat.org>, who provided the articomm driver.]

Added optional global keywords:

> **Systems=** and
> **Passwd=**

to UUPC.RC file. These keywords define non-default locations of **SYSTEMS** and **PASSWD** files. This allows unique (dummy) copies of these files to be used for client workstations without passwords on shared network environments. The default location for these files remain in the UUPC/extended configuration directory.

Added new option keywords **senddebug** and **honordebug**. In their default state, UUCICO neither transmits the current debug level to the remote system when calling, nor does it use the debug level sent by the remote system called.

> **Note:** The previous behavior was to not send the current debugging level but to honor the remote system's debug level.

Renamed **hpfs** option to **longname**, which more accurately describes its (as yet unimplemented) processing in support of longer file names on any file system which supports them.

Added new UUPC.RC keywords **banner=** and **motd=**, which define files displayed to remote users by UUCICO. If defined, the file named as the **banner** is displayed before the login prompt. Likewise, if defined the file named as the **motd** (message of the day) is displayed after a successful login **only** if the login shell is **not** UUCICO.

Modified loading of the **PASSWD** file to use the configuration directory drive as the default when determining the drive letter for home directories. This allows omitting the drive letter within the **PASSWD** file.

Added new OS/2 command file, **GETUUPC.CMD**, which is called by the other OS/2 scripts to determine UUPC/extended configuration information.

Made post-login startup message time out configurable.

Version 1.11y Revision Summary

Bless Snuffles for she has sinned. She failed to keep track of all the contributors of bug fixes and enhancements for the 1.11y release. But you know who you are, bless you all ...

Bug Fixes

Dropped the beta test port of RN 4.3. We're working on porting the newer RN 4.4, but neither source nor object for this is included in 1.11y.

Corrected bug in configuration processor which assumed all tables (i.e. the modem configuration table) were as long as the master configuration table.

Added specific error message for when local host name was improperly added to SYSTEMS file. Formerly, UUCICO died gloriously with a (incorrect) message about a memory shortage in dcpsys.c.

Corrected configuration processing to load UUCP spool related variables for RNEWS. Previously, RNEWS failed to load such information and thus ignored overrides of the default character set.

Modified directory search functions to support longer file names used by non-FAT (non-DOS) file systems and to return file time and size information to the caller. This implicitly fixes problems with programs such as EXPIRE running out of memory when running on network drives due to bugs in the Borland C++ stat() call, which no longer has to be used as much.

Modified UUCICO status file processing to lock the status file to avoid overlapping updates. Previously, race conditions could cause the file to be reset by mistake.

Corrected printmsg() to use unique buffer for time stamp information in log file. Formerly, the default static buffer used by various subroutines would be overwritten with the current time when logging.

Made return codes from RMAIL more unique to allow better debugging of failed mail.

In RNEWS, redid processing performed by SNEWS option to make it even dumber.

Rewrote RNEWS batched news processing to correctly process files under 512 bytes. Formerly, such articles and all articles following were dropped.

Upped maximum length of news group names in RNEWS to keep it from dying when brain dead lusers create news groups with excessive length names.

Corrected OS/2 WAITING command to load needed subroutine packages before using them.

Corrected UUPOLL bug which caused it to exit with a return code 0 no matter how it was terminated.

Corrected UUPOLL bug which caused it to not terminate at the correct time if both the -e and -f flags were specified.

Added fix to COMMFIFO.ASM from William Plummer to fix incorrect assembly of references to COMMFIFO error counts.

Modified UUCICO 'g' protocol processor to use FAR memory for packet buffers to reduce number of out of memory errors under 16 bit DOS and OS/2 compilers.

Modified UUCICO 'g' protocol processor to correctly negotiate packet sizes with remote systems, thus allowing the remote system to send the local system larger packet than the local system can send the remote. (Error pointed out by Ian Taylor.)

Corrected sample modem files to NOT enable VariablePacket processing by default.

Corrected UUCICO to not update host status if host information was not initialized during a passive poll.

Corrected **syslog** option to only set buffering once for the log file if not multi-tasking. This bug caused 1.11x to repeatedly report that SYSLOG file was not found when, in fact, it was fine.

Corrected UUCICO modem processing to flag that graceful shutdown was needed for OS/2 versions of UUCICO. Previously, it exited immediately without closing the serial port.

Corrected UUCICO modem processing to invoke carrier detect processing whenever the modem connects. This was previously missed for hot logins invoked using the UUCICO -w flag.

Corrected UUCICO carrier detection processing for OS/2 communications driver. It previously OR'ed modem status information rather than AND'ing it, allowing a half-ready modem to appear ready.

Modified UUCICO to trap unique error code returned by SIO serial port driver under OS/2 when interrupted.   Interruptions of the DosRead() system call are not really errors, and the default error message alarmed Snuffles.

Corrected UUSTAT to not de-reference NULL pointers when killing queued jobs.

Enhancements

Dave Watt <dmwatt@smersh.cambridge.ma.us>, with assistance from Evan Ross <evan@valentine.com>, made numerous additional changes to round out the Windows NT support. One visible change is that lines in UUPC.RC and [*userid*].RC can be prefixed with WIN32. to cause the line to take effect only under Windows NT.

The **ConfDir** variable is now optional. If the **ConfDir** is omitted from UUPC.RC, its value is taken from the directory for the UUPC.RC file specified in the UUPCSYSRC environment variable.

The directories (except the user's home directory) in the UUPC.RC and *[userid]*.RC file can now be omitted or made relative to the **ConfDir**. If omitted, they are assumed be sub-directories of the **ConfDir** with default names. The user's home directory may be relative to the **ConfDir**, but it must not be omitted.

Added new configuration variables **OS2.priority** and **OS2.prioritydelta**. These specify the OS/2 priority of UUCICO when connected in the range 1 through 4, and a sub-priority of 0 through 62.

Added new Boolean option **bounce.** When enabled, this causes undeliverable mail to be returned to the sender (as determined by UUXQT and/or by the UUCP From line) with a copy to the local Postmaster.

William Plummer <plummer@theme.com> made various additional changes to the test driver for DOS COMMFIFO package.

Per request of Katherine Derbyshire <docs@kew.com>, modified GENSIG to print out a more useful message about the quote selected, specifically including the output file name.

Versions 1.11w through 1.11x Revision Summary

Bug Fixes

Corrected bug in local delivery routine which failed to detect a letter followed by a colon in a FORWARD file was the beginning of a path name. This fix was supplied by Russ Nelson.

Modified internal pager to not scroll top line off the screen when the display is aborted.

Corrected **RMAIL** usage message to display all the operands allowed by the command's rich syntax. Fix supplied by Russ Nelson.

Modified chdir() to attempt directory change before invoking mkdir() to build the directory to change to. This should be faster, since a directory will always exist after the first chdir() to it.

import_path() mangled valid paths under certain conditions. Modified the routine to properly convert path names with a single character in the path and/or terminated by a slash.

Corrected import_newsgroup() to not pad the numbers generated as part of names of DOS news articles. Fix provided by Kai-Uwe Rommel.

Corrected via spool directory search routines to correctly handle interrupted searches when a connection to a host is lost. Previously, if a search was not determined to be interrupted, export_path() would catch the bad file name and abort the program ... usually.

Corrected included strlwr() function to lower case the string, not upper case it. This problem only affected users using unsupported compilers which lacked this function in their run time library.

Modified RNEWS to not shadow or otherwise process an empty compressed input file. Also corrected the file name in various error messages related to uncompress processing.

Modified RNEWS to report when it has to delete an article, and to summarize number of articles dropped.

Modified RNEWS to report specific error messages via printerr() when it is unable to process a file.

Modified RNEWS to write news in the proper news directory when the **snews** flag is set. Fix supplied by Kai-Uwe Rommel.

Applied numerous minor fixes to RNEWS supplied by Kai-Uwe Rommel.

Corrected EXPIRE to properly expire one news group via the -n flag. Fix supplied by Kai-Uwe Rommel.

UUPOLL would be confused by 25 hour days caused by the end of Daylight Savings Time and attempt to poll continuously during the last hour of such a day. Corrected UUPOLL to realize not all days are created equal, and to correctly compute the next poll as the first poll of the following day.

Altered numerous files to be unbuffered, saving on memory and reducing memory to memory copying of data.

Deleting extra lock files was a major performance hit under OS/2 with UNDELETE processing enabled. Corrected UUCICO and UUXQT lock processing to not lock systems without reason.

Suppressed closing UUCICO warning message about unable to connect to remote when the program was only answering the phone.

Made UUCICO 'G' protocol use packets of fixed size to be more compatible with System V Release 4 UUCP.

Made unique entries in the modem (*.MDM) file for entries for the "g", "G", and "v" protocol options. Also added **largepacket** modem file option for "g" protocol to enable large packet processing under this protocol.

Added new UUCICO 'v' protocol, which is 'g' protocol with optionally large (512) byte variable length packets. This protocol is only supported by other UUPC/extended systems.

Corrected two problem with 'g' protocol start-up, from problem descriptions provided by Ian Lance Taylor, author of Taylor UUCP. Both fixes provide for insuring one system does not get ahead of the other, that is one system thinks it has initialized the connection when the other has not.

Corrected UUCICO to not abort when fields are missing from the SYSTEMS file. This was based on an original fix by Eugene Nesterenko, Moscow, Russia.

Corrected various UUCICO start-up error messages to display at debug level 0, not 1. Thanks to Bill Plummer for pointing this out.

UUCICO would previously reject anonymous UUCP connections if the local system was running a high debug level. Corrected UUCICO to not confuse local and remote debug levels.

Modified UUCICO to use a single data transfer buffer for data input and output as well as a command buffer, thus allowing this buffer to be bigger.

Modified UUNAME to accept the debug level (-x) option. This was embarrassing since the upgrade instructions specifically said to use this option to determine obsolete keywords in the UUPC.RC file.

Corrected UUXQT to send mail to the originator of a UUX job, not the local postmaster. Also corrected UUXQT to not mangle the dummy input file name NUL and then be unable to open the mangled name. Fixes supplied by Charles R. (C. R.) Oldham of Arizona State University.

Corrected UUSUB to reset the starting time of statistics collection when clearing all totals.

Corrected UUXQT to report **why** RMAIL dies, not just that it failed.

Corrected RMAIL to not close its own log file, instead leaving this task to the common logging functions.

Enhancements

Dave Watt ported the entire source to Windows NT. Note this platform has been not been tested, and executables are not supplied at this time. Problems are specifically known to exist in creation of the HOSTSTATUS file.

Dave Watt also wrote DESQView support. UUPOLL and UUCICO give up the CPU when idle.

Added a number of new sample configuration sample files, including a sample PERMISSN file and copies of several files used on kendra.

Modified configuration file routines and expand_path() to use run time library supplied functions to normalize directory paths. This allows determining the path on drives other than the current drive.

> **Note:** Because of this change, placing floppy diskette drive letters in the PERMISSN file can cause the system to hang when it attempts to determine the current directory on the drive.

Rewrote the OS/2 versions of UUIO, UUCLEAN and WAITING commands into REXX, made enhancements including using the UUPC/extended environment variables and aging files based on size.

Added MAILCHEK command from Evan Champion <evanc@uuisis.isis.org> to provide running report of mail waiting in an OS/2 window using Visual REXX. (This command does not exist under DOS.)

Modified FOPEN() to automatically retry failed file opens for 20 seconds if the error is a sharing violation. Also modified openline() and lock_system() to **not** use this retry processing.

Moved **HOSTATUS** file from spool directory to configuration directory, and moved news **ACTIVE** file from news directory to configuration directory. This allows easily backing up these files without backing up the news or spool directories.

Modified majority of permanent string allocations to use newly written routine strpool**()**, which allocates memory in larger blocks, thus bypassing malloc() space overhead. Routine also searches duplicates in strings to avoid allocating the same string twice. If running at debug level 2 or above, a summary message of the memory used (and saved) is printed. Also, added new boolean option **SpeedOverMemory** to bypass the duplicate string scan on slower systems.

Wrote support for a system wide user aliases file. This file, used only by RMAIL, allows forwarding for users and/or mailing lists to be defined without adding the local address to the PASSWD file, resulting in improved efficiency for other programs which use the PASSWD file such as MAIL and UUCICO. Use of the system wide alias file also eliminates the need for dummy home directories containing forward files, resulting in 4 kilobytes disk savings per forwarded user. This support was based on a prototype supplied by Russ Nelson.

Modified **MAIL** to print the numbers of items deleted by the **delete** command on one line to keep the screen from scrolling madly. Also, items previously deleted are not listed a second time.

Added number items in the mailbox to the **MAIL status** command.

Added "Path:", "Lines:", and "References:" to the list of items ignored by the internal mail pager used by the **print** command.

Under OS/2 and DOS 4.0 or above, the internal pager now uses the true screen size to determine the number of lines to scroll. This allows exploiting 50 line VGA screens or odd-sized OS/2 windows.

Modified loading of ACTIVE file by news processing to not build directories for empty news groups. So long as the target directories are built by the routine posting news (i.e. RNEWS uses fopen()), then the directories will be created on demand.

Enhanced EXPIRE to report bytes and articles processed, including totals for data purged, moved, or left eating disk space. Running EXPIRE at debug level 2 (-x 2) will report this information on a per group level.

Modified UUPOLL to report when the UUCLEAN command is scheduled to next run at startup.

Modified UUCICO processing enabled by the **syslog** option to work with the **multitask** option. Specifically, the SYSLOG is only opened when needed in multitasking mode and closed immediately.

Modified OS/2 communications support to show communications port errors in English. Also added a ShowModem() function to OS/2 communications support to show modem status in English.

Under OS/2, increased modem recover time from 0.5 seconds to 2.0 seconds after hanging up the telephone.

Versions 1.11u through 1.11v Revision Summary

Bug Fixes

Modified ImportNewsGroup() import articles by number rather than name.

Corrected internal copyright statement displayed at program startup. Also changed protected mode program start up message to report protected mode rather than include Big Blue's registered trademark.

Modified RMAIL to better hide sites which put their full domain address on their UUCP From line.

Modified RMAIL to use the domain name and not the UUCP name for requester name when generating requests to remote systems.

Corrected standard **FOPEN** processing to use correct sharing flags. Previously, it allowed multiple writes to the same file.

Corrected default choice of the **LocalDomain** variable to use at least two levels in the domain, that is if the system domain name is "kew.com", don't use "com" for the **LocalDomain.**

Changed protected mode linkage editor from LINK386 to LINK, as some people reported problems running on older versions of OS/2, and DOS systems don't have a LINK386.

Enhancements

Modified option **multitask** to also enable locking of systems for **UUCICO** and **UUXQT**, preventing multiple calls to the same system at once, and to log into temporary files, merging these files into the master logs at the end of program execution.

Mike McLagan <mmclagan@invlogic.com> contributed call grading support for UUCICO via Systems file and calling grading option (-g). See *Managing UUCP and Usenet* for use of call grading. The changes made by the Wonderworks have not (alas) been fully tested. Also added option **symmetricgrades** to request grade specified on command line be transmitted to remote system when actively polling, an option which is not in *Managing UUCP and Usenet*.

Changed **UUCICO** to accept a time to run (-d hhmm) rather than the older poll interval (-u) option. This allowed shortening the code required to compute the time to answer the telephone and made it more reliable.

Modified **EXPIRE** to report number of files and bytes purged.

Modified **MAIL** alias command to report routing of addresses fed it.

Translated OS/2 error number for sharing violation to English "Port in use message" in OS/2 communications library.

Added **noundelete** option to disable OS/2 highly inefficient undelete processing for programs running under UUPOLL.

Version 1.11t Revision Summary

Bug Fixes

Added new global option **multitask**. This option enables the following:

> When MAIL is invoked to read the system mailbox for a user, the mail is automatically merged into a private mailbox for the user in the user's home directory. This reduces the possibility of updates to the system mailbox while the user is reading mail.
>
> UUCICO updates the status file used by UUSTAT whenever the status of a call is updated.

Changed standard file open function FOPEN() to use shared open under both DOS and OS/2. All read only files (such as the systems file) can be read by multiple programs this way.

Modified expand_path() to support expanding relative paths on non-default drives via standard library calls.

Modified mailbox routines to truncate mailbox names to eight characters to prevent file open errors under OS/2.

Corrected bug in OS/2 directory search routines to not double append slashes to directory names.

Modified printmsg() to use standard output if logfile stream is closed for an unexpected reason.

Corrected bug in **ValidateCommand** to correctly validate when a user is allowed access to all commands on a host.

Modified ddelay() to force enabling interrupts under DOS; this is a work around for a bug not in UUPC/extended, but rather in the DOS **mirror** command which sometimes appears to not enable interrupts after saving the location of a deleted file.

Added traps in MAIL for over length addresses, which formerly caused UUXQT and/or RMAIL to fail in unexpected ways.

Modified alias functions to build local aliases via the local **FromDomain** name used in return addresses, not the local real domain address.

Corrected "Purging news message" to correctly report RNEWS was purging news, not archiving it.

Added trap to checktime() to check over length time fields in the Systems file.

Dropped attempts by 'g' protocol module in UUCICO to close and reopen port if no activity occurs on port for specified time. (This action had caused COMMFIFO to drop carrier.)

Corrected UUCICO login prompt to display the OS/2 version number as "2.0" and not "20.0".

Modified rmsg() to not zap the 59th byte of a 50 byte buffer; this caused logins to fail under OS/2 with 1.11s. Instead, the actual buffer length is now passed in to the program.

Corrected UUCICO transfer engine to use internal temporary file name generation rather than standard library routine, which sometimes generated files in the root directory.

Enhancements

Added support DOS. and OS2. prefixes to options in the configuration files. Thus, editors and other options can be set differently for DOS and OS/2 environments on same system.

Added ability to define obsolete keywords in the UUPC.RC file, which keeps the configuration routines from complaining when testing new releases. Obsolete keywords are reported as obsolete at debug level 2 and higher.

Added additional status flags to hostable and associated status messages for UUSUB and UUSTAT. The full list of possible status reported by UUSTAT are now:

(invalid - entry not properly initialized)
(local host system)
(host for gateway purposes only)
Never called
Dialing now
Invalid device or speed in SYSTEMS file
Device not available
Conversation start-up failed
Talking

Callback required
Modem initialization script failed
Dial failed
Script failed
Max retry reached
Retry time not reached
Call succeeded
Wrong machine name
Unknown host
Failed
Wrong time to call
(call succeeded, entry not reset)

Modified UUCICO to not prompt for confirmation of shutdown if it is not doing anything. (I.e., it is waiting for phone to ring.)

Added global option **collect**, which causes RMAIL to report size of mail items delivered.

Altered RNEWS default archive period of seven days to zero days. News is thus completely purged after seven days.

Added new variable to configuration files, **Uncompress**, to define program to run to decompress news. Normal usage would be:

```
Uncompress=compress -d %s
```

Which runs a standard compress program in uncompress mode against the file name specified.

Note: Compress is not included with UUPC/extended.

Corrected name of file that compressed news reports to have been copied to in RNEWS.

Added kludge to RNEWS to allow copying of ALL compressed news received to a second system. By setting the DOS or OS/2 environment variable **UUPCSHADOWS** to a list of locally known systems, the compressed input to RNEWS is queued for the second system as well.

Note: This "feature" will be removed in the next real release of UUPC/extended when real news support is added.

Modified GENSIG to maintain a list of recently used quotes and not repeat them until 90% of the available quotes have been used.

Added new alternative log file name flag (**-l logname**) to change name of log file used by UUCICO. Simple file names are written in the UUPC spool directory, and files with path names are written as specified.

Note: The special name NUL will suppress logging to a file. Because the spool directory contains sub-directories for each remote system, the alternative log file name MUST NOT be simply the name of a remote system.

Added new alternative input device flag (**-m modemname**) to override name of modem file used by UUCICO to initialize passive polling.

Also added alternative input device flag (**-m modemname**) and alternative log file name flag (**-l logname**) to UUPOLL. These flags are passed to UUCICO as-is.

Added support for automatic invocation of UUCLEAN by UUPOLL to purge news and spool log files. Automatic clean time (**-c hhmm**) specifies the time for UUCLEAN to run daily; the default is do not run UUCLEAN. The automatic flag command name flag (**-C command**) specifies the command to run daily; the default is UUCLEAN. Note that the command is run once every 24 hours.

Moved debugging trace for login and initial messages from various modules to rmsg() and wmsg(). Specific errors are still reported in various modules.

Modified rmsg() to support backspace and delete characters when entering user id and password during login.

Version 1.11s Revision Summary

Bug Fixes

Previously, the MS C version of UUCICO would not reset stolen interrupts if canceled. Rewrote Ctrl-Break handler in UUCICO to be compatible with both Borland C and MS C; the handler will now prompt before aborting, and will also attempt to gracefully close any active connection. In addition, added new boolean option **escape**, which enables the ESC key as abort key during processing. Added same Ctrl-Break handler to RMAIL, and a modified version (without the prompt) to UUPOLL.

Altered definition of a successful call by UUCICO to a call which processes all the way through a graceful shutdown of UUCICO. This means a call which is interrupted during file transfer will be retried if the host appears more than once in the SYSTEMS file. Also deleted didcall() and calledhost() subroutines, which were made redundant by the status flags used for the HOSTATUS file.

Modified 'g' protocol module to dynamically allocate buffers on demand rather than using fixed buffers. This reduces memory requirements when less than seven buffers are used.

Added line pacing to ulib.c to prevent buffer overruns by the UUCP 'f' protocol. The 'f' protocol now works ... sometimes.

Corrected various connection abort routines in dcpxfer.c to reset the **xfer_stream** variable after closing the file being transferred.

Deleted parse of user name on incoming UUCICO command line sent from other systems. Excessive name lengths caused unpredictable storage overlays in dcpxfer.c.

Deleted redundant debugging messages from dcpxfer.c.

Moved default setting of speed for hot logins to UUCICO (using the **-w** and **-z** options).

Formerly, the modem speed could be set to zero, causing a divide exception in the communications package.

Corrected GetGPacketSize() to only return valid package sizes (which are all powers of 2).

Clarified various error messages in UUCP. Also reformatted source of uucp.c to personal taste.

Added detection of invalid or duplicate system names on the UUSTAT command line.

Modified UUSTAT to always generate job 000 as the dummy job for the poll (**-P**) option.

Corrected various error messages in UUX to give the user some idea of why the program died.

Modified dcpsys.c to generate a dynamic list of available protocols rather than using a fixed list which was prone to fall out of date as protocols were added.

Altered timeout processing for scripts to not reset the timeout period when a character is received. Formerly, the time out was always reset after each character received, resulting the modem not timing out until after the modem had been silent for the timeout period (usually after it had already hung up and reported so.)

Restructured definitions of program flags in lib.h to avoid multiple programs using the same flags when calling configuration file processing.

Clarified the error message issued by ExportPath() after a name conversion failure.

Deleted original support for site-hiding via the HOSTPATH file. Sites wishing to hide behind another host *must* use the **Domain** and **FromDomain** keywords in the UUPC.RC file.

Altered UUXQT to delete command files after a **-1** (spawn failed) return code.

Modified conversion of UNIX to DOS names as follows:

> Broke out the conversion of a simple name to DOS into a new routine ImportName()
> Modified ImportPath() to convert the entire path passed to a valid DOS path, rather than just the trailing name and extension.
> Also corrected mapping of a UNIX name with multiple periods.
> Rather than mapping all invalid characters in name with a path to the underscore (_), invalid characters are now mapped to a semi-unique character in the valid character.
>
> > **Note:** Because valid characters in names with paths are not remapped, collisions can occur.

Fixed bug in ValidDOSName() to use default character set if none was provided.

Corrected ValidDOSName() to not reject valid DOS names with lengths greater than 72.

Barry Schwartz <trashman@crud.hawkmoon.mn.org> rewrote the MS C version of getdta() function as in-line assembler; it now properly loads the ES segment register. Failure to do so previously caused various improper behavior.

Disabled stack checking in most of the programs to reduce program size and improve speed. Stack checking is explicitly enabled for printmsg() under both MS C and Borland C++, so this central routine should catch any serious problems. We hope.

Corrected banner routine to report time stamp of program when output is to the console, as opposed to the previous test for the input being from the console. This means that RMAIL's banner will now display when run from MAIL.

Added Mitch Mitchell's quicky UUNAME program to make it look like we've been busy.

Deleted UUPC.RC configuration variables **rmail** and **rnews**. These programs are now located via the path.

> **Note:** **T**he current directory is changed during the execution of UUXQT, so placing these programs in the current directory is not sufficient.

Modified address_at_node() parser to process explicit RFC-822 paths in a more robust fashion.

Modified RMAIL to use more unique characters when creating spooling file names for other hosts.

Altered MAIL command line parser to reject such syntax as a subject with no addressee.

Corrected prompt in MAIL sub-shells to only show the message "Enter EXIT to return to MAIL" once.

The documented but disabled process and delete file option (**-F**) in RMAIL was re-enabled. Also added check to UUXQT to insure a remote system does not use this option to delete COMMAND.COM.

Enhancements

Numerous changes were made to the COMMFIFO.ASM routines by William W. Plummer (Plummer@wang.com); as a result, the program now provides a superset of the previous COMM.ASM functions for both FIFO and non-FIFO UARTS, and is part of the default UUCICO.EXE.

> **Note:** For a detailed list of changes to COMMFIFO.ASM, see the top of the source file.

> **Note:** Because the changes to COMMFIFO moved various buffers out of the C data area, larger **TransferBuffer** sizes are now possible.

Modified dcpgpkt.c to report at debug level 2 and higher the amount of available storage after data packet storage has been allocated.

Added support to UUXQT to detect and process internal MS-DOS commands. As part of this support, added new variable **InternalCommands** to define what commands are internal to the MS-DOS. The default list of internal commands includes:

break **cd** **chdir** **copy**
ctty **date** **del** **dir**

echo **erase** **for** **md**
mkdir **rd** **rem** **ren**
rename **rmdir** **time** **ver**
verify **vol**

In the better late than never department, added display of a copyright notice to configuration file processing. Also added new boolean option **suppresscopyright** to turn off the message (but not the responsibility).

Enhanced 'g' protocol processing to support long (512 bytes) and short packets (32 bytes). Also modified 'g' protocol module to dynamically allocate buffers on demand rather than using fixed buffers.

Note: This support is not exploited unless either the protocol in the systems file is specified as upper-case G, or if the new boolean option **variablepacket** is enabled in the modem configuration file.

Note: The standard 'g' procotol should be used for Telebit TrailBlazers and the like in PEP mode, for most unmodified UNIX systems.

Altered script processing to not print expected strings at debug level one unless an error occurrs.

Modified ulib.c to display modem status in English (or at least words) in addition to hexadecimal.

Added support to UUPOLL to exit at a specific time via the new **-e hhmm** option.

Modified UUSUB to allow clearing stats for one or more systems without losing call status or host data. Change contributed (I hope) by Mitch Mitchell.

Modified dcpsys.c to disallow large debug levels with anonymous UUCP.

Added new modem boolean option **carrierdetect** to enable checking of the modem Dataset Ready (DSR) and Data Carrier Detect (DCD) lines. With **carrierdetect** enabled, if DSR is lost at any time, or DCD is lost after the modem reports connected to the remote host, then the connection is immediately aborted to the remote host.

Added new modem variable **NoConnect**, which defines various strings which denote the modem has failed to connect; when any of these strings are received during login script processing. A typical example for a Hayes modem would be:

NoConnect="NO DIALTONE" "BUSY" "NO CARRIER" "NO ANSWER"

Modified configuration file processing to accept selected variables from the DOS environment. All such variables will be overriden by the same variables in the configuration file. These DOS variables, and the UUPC/extended variables they set, are:

DOS variable	**UUPC/extended variable**
EDITOR	**EDITOR**

MAILBOX	**MAILBOX**
NAME	**NAME**
TEMP	**TEMPDIR**
TMP	**TEMPDIR**

Modified host table processing to support rerouting the local host to a gateway.

Moved port and packet related variables from the main configuration file to the modem configuration files, and in addition defined unique "f" protocol time out variables. The new modem variables include:

fpacketsize
fpackettimeout
gpacketsize
gpackettimeout
gwindowsize
porttimeout
transferbuffer

Added a magic word display to the MAIL status command as suggested by KEW.

Added mailbox name to summary of header lines in MAIL.

The display of aliases by the **alias** command now shows the routing for the address.

Added new RNEWS program contributed by Mike Lipsie, and wrote a new news EXPIRE program based on Mike's EXPIRE program.

Modified UUSTAT to generate dummy jobs for all valid systems if requested via **-Pall**.

Version 1.11r Revision Summary

Bug Fixes

The various configuration file variables which controlled error processing and other tuning parameters were global to the system rather than to a single device or system. Corrected by moving the following variables from the master configuration table to the modem configuration table:

> **MaxmimumErrors**
> **PacketTimeout**
> **PortTimeOut**
> **TransferBuffer**
>
> With the security provided by the PERMISSIONS file and the function provided by new UUXQT program provided by Mitch Mitchell and the associated **PATH=** variable, there is no need to specify the names of the RMAIL and RNEWS programs; these can be located via the default or explicit PATH.

Blank lines (as opposed to empty lines) were flagged as errors in the **UUPC.RC** and *[userid].RC* files. Corrected parsing of configuration files to delete leading blanks from all lines.

Several people pointed out the need to support the "-" option to have UUX read files from stdin. getopt() could not process a minus sign (-) as a stand-alone flag; modified to do so by Mitch Mitchell for UUX.

If the local host supported both unvalidated and validated logins, a host with a validated user id could also be logged on a host for access was not validated. This provided a security exposure for systems which publicized anonymous login access. Corrected by not allowing hosts which appear in a **VALIDATE=** clause in the **PERMISSN** file to login on unvalidated users id..

MAIL still had hard-coded assert() statements which failed the program without explaining the error in question. Modified MAIL to use printerr() and panic() rather than assert(). Also modified printerr() to print the location of the failure if debugging is enabled and to print the extended MS-DOS error information when using Borland C++ under DOS.

Bill Plummer's (plummer@wang.com) enhanced communication driver (COMMFIFO.ASM) had various problems due to minor bugs. Corrected by Bill.

The routine ssleep() didn't work properly in some intermediate levels of 1.11r; corrected by doing ddelay() for maximum allowable time.

If an explicit RFC-822 route of was imbedded in an UUCP style address, such as:

> node1!node2!@node.domain:user
>
> then the address parser was confused. Corrected User_at_node() to examine UUCP addresses for such constructs, although due to problems with the order of the host names the resulting address may or may not be correct.

David Friedlander (df@iochom.UUCP)provided changes to **UUX** to enable forwarding; two fixed obvious bugs and are implemented.

Andrew Lindh (lindh@uhasun.hartford.edu) discovered the bug in **UUX** that corrupted up binary files read from stdin. A call to fgets() should have been a call to fread(), and a call to fputs() should have been a call to fwrite(); both are fixed now.

Tubby Turtle (tubby@harlie.lonestar.org) fixed a bug in **UUXQT** that caused problems with redirecting output of commands executed locally. A subtle difference in meaning was attributed to flag xflag[X_OUTPUT] when it was read than was intended when it was set.

Mitch Mitchell (mitch@harlie.lonestar.org) supervised and applied all the fixes to **UUXQT** and **UUX**, and has been busy trying to avoid being called "Tubby Mitch".[1]

UUPOLL always had the auto-forward (**-a**) option enabled by default. Corrected initialization of variable to **FALSE**.

Changed various assert() statements in **UUPOLL** to panic() statements to provide better diagnostics.

When UUCP connections failed for various reasons, various status programs still listed the last call as successful. Modified **UUSUB** and **UUSTAT** to report a new status, "FAILED", when a call failed after connecting to the remote system.

If an excessively long user name was included in the command line in a call (C.*) file, the file transfer would fail because of storage overlays. Modified both in-bound and out-bound command parsing in dcpxfer.c to ignore the user id.

Purged this revision summary for changes previous to release 1.11g.

Enhancements

Directories listed in the access lists for hosts in the **PERMISSN** file are now verified to exist.

> **Note:** Some people may not view this as a feature.

Added percent of mailbox read message to the status update displayed while reading the user mailbox in **MAIL**.

Added support to **UUCICO** for long packets of up to 1024 bytes, including changing various communications buffers to be dynamically allocated for required lengths, lengthened outbound buffers in COMM.ASM and COMMFIFO.ASM to 4096 bytes. Also added support for new modem file keywords **GWindowSize** and **GPacketSize**.

> **Note**: For compatibility with previous releases and UUCP programs which do not correctly long packets, the default packet size is the "standard" UUCP packet size of 64 unless short packets are enabled via the **SmartPacket** option (changed to **VariablePacket** in version 1.11t).

Also added support for packets shorter than the negotiated maximum for faster processing of short data streams. This support is enabled by the new modem file boolean option **SmartPacket** (changed to **VariablePacket** in version 1.11t).

[1] Broadcasting a name a person doesn't want to be called is counter-productive. We document all the wrong things around here!

Note: Not all UNIX systems support packets shorter than the negotiated maximum; DECUS UUCP and FSUUCP do.