Advanced Installation and Configuration Topics

Changing How Mail is Addressed and Delivered Locally

An Overview of RMAIL

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:[51]

1. The user generates mail via the MAIL program.[52]
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.[53]
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.[54] If the "bounce" option is set, the mail is also returned to the sender.[55]

---

[51] See **Error! Bookmark not defined.** for the detailed syntax of the RMAIL command.

[52] For 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.

[53] By 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.

[54] If 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.

[55] If 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.

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 file

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[56]. 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 File

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.[57] 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 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:

---

[56] In UNIX systems which use SENDMAIL, the file is normally called **.forward**.

[57] This should not be confused with the SYSTEM aliases file, which has a fixed name as described on page 106.

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.[58] You can specify groups[59] 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>

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

---

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

[59] 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.

Implicit User Alias Processing, (or, What Your Polar Bear Never Told You about Aliases)

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 File

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**[60] 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 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.

---

[60] This file, which has a fixed name, should not be confused with the user alias file described above on page 104.

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 file

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

Overview of Mail Routing

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[61] is queued for that system.

* 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.

---

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

* 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.)

Routing Mail Via Non-Default Mail Servers

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 System

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 File

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 Hiding

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[62] 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
```

---

[62] 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?

```
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 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 Gateway

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.x

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.[63]

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.[64] 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,[65] 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.

Using UUPC/extended under OS/2 and Windows NT

UUPC/extended programs run as 16-bit[66] 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

---

[63]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.

[64]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.

[65]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.

[66]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.

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 LAN

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) names

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 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 Drivers

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 Environments

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 you

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 files

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 configuration

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 yours

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.[67]

Using FOSSIL communications drivers with UUPC/extended

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

---

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

· 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.

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 wrong

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.

Appendix 1: 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