

## Chapter 2

# *Setting up electronic mail*

---

This chapter explains how to set up electronic mail on your SCO UNIX system.

Electronic mail on the SCO UNIX system is handled by two utilities, the MAIL USER AGENT (MUA) and the MAIL TRANSPORT AGENT (MTA). The MUA is the program, such as **mail(C)**, that allows users to send, read, and manage mail messages. The MUA transfers the message to the MTA, the group of programs that actually route and deliver messages to their destinations. The MTA on the SCO UNIX operating system is MMDF (**M**ultichannel **M**emorandum **D**istribution **F**acility).

MMDF provides users with transparent access to the different networks and related mail transport PROTOCOLS, through CHANNELS, regardless of the MUA. (A channel is the method, such as UUCP, used to deliver messages; a protocol is a set of rules for communicating over a network and includes standards for mail message formats.)

In addition, MMDF provides the system administrator with tools to monitor and customize MMDF. Using these tools, the system administrator can tune MMDF dynamically, modifying the behavior of these programs even while MMDF is running.

With MMDF, users can send mail on the local network or over larger area networks across the interconnected group of networks known as the "Internet". The Internet is otherwise known as ArpaNet or DARPA (**D**efense **A**dvanced **R**esearch **P**rojects **A**gency) Internet.

The version of SCO MMDF provided with your operating system was derived from MMDF-II Release 43 from the University of Delaware. This version provides additional features, such as a more robust locking mechanism that allows you to specify how to lock user mailboxes, and support for **sendmail** *.forward* files. This version also differs in the format of the hashed table database built by **dbmbuild**(ADM), which is non-standard.

The system automatically configures MMDF for local (one system) mail delivery when you install your operating system. If you did not install the entire distribution, you should install the MAIL package now using the **custom**(ADM) utility. See your *Installation Guide* for information on installing packages with **custom**.

## Chapter overview

---

The first part of this chapter explains how to configure MMDF for most sites using the simple configuration utility provided with your distribution. First, the chapter gives a basic overview of how MMDF works and covers some background information necessary for using the configuration utility. With this information, you can then fill out the MMDF configuration checklist provided. Then, this chapter shows you how to run the configuration utility to set up MMDF to exchange mail with other computers.

The configuration utility does not work for all site configurations, therefore the second part of this chapter describes the different configuration files, how MMDF uses them, and how to modify them to work with your configuration. This section also covers how to test and maintain the MMDF system.

If you do not plan to exchange mail with other machines via UUCP or TCP/IP, you do not have to run the configuration utility; MMDF is already configured to send mail on the local machine. However, if you want to reroute mail sent to special system accounts, such as *root*, the configuration utility provides an easy way for you to set this up. For more information, see the sections on redirecting mail later in this chapter.

## How MMDF works

---

This section gives you an overview of how MMDF processes and delivers mail on your UNIX system. Most of the concepts relate to mail traveling in both directions; you should read both the outgoing and incoming sections with this assumption.

## Outgoing mail

---

Outgoing mail starts when the user invokes the MUA, such as **mail(C)**, to compose a mail message. The MUA requires that the user specify a “To:” header when creating the message. Then, the MUA adds two other headers, “Date” and “From”, when the user sends the message. These headers specify how MMDF sends the message through the system; the next section explains the format of these headers.

### Mail headers

For a mail header to be correct, it must include these three lines in the following format:

```
From: sender
To: recipient
Date: Weekday Mon DD hh:mm:ss year
```

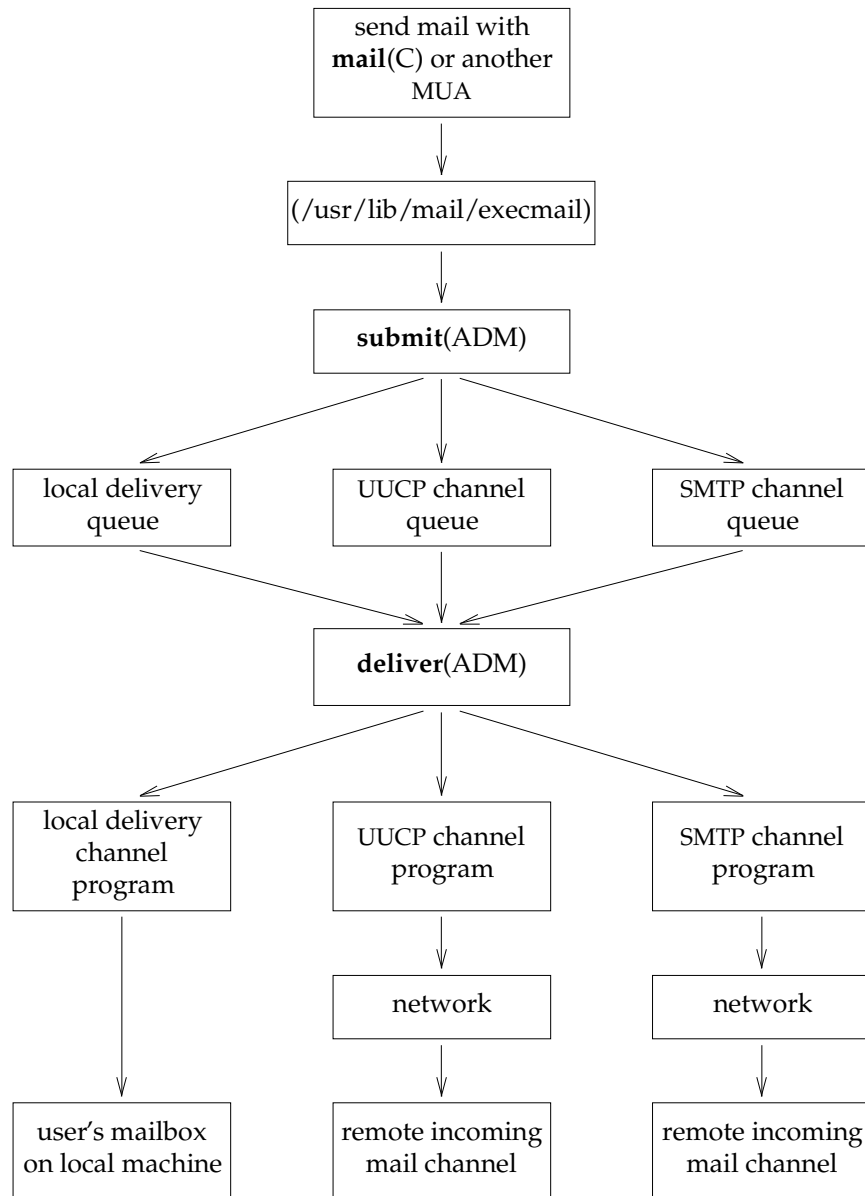
Here is an example:

```
From: fred@npr.COM (Fred Astaire)
To: ginger
Date: Wed Apr 3 12:21:23 PST 1991
```

In addition, most MUAs and mail submission programs, such as **mail(C)** and **execmail**, add extra header lines. For example, the MUA might add the “Message-Id” header. If the MUA does not add this header, MMDF adds it, as well as any “Received” headers. The user can also add other headers. For example, if the user specifies a carbon-copy recipient, the message header includes a “Cc:” line. MMDF allows these additional header lines, but does not use them; the recipient mail server and MUA handle these header lines.

In the example above, note the format of the address in the “From:” line. This is an example of a DOMAIN NAME. MMDF uses the domain name to determine how to route the message. Before you can configure MMDF with the configuration utility, you must understand how domain names work; the section on the configuration checklist later in this chapter covers domains and the different types of domain names, including fully-qualified domain names, in greater detail.

Figure 2-1 shows the path that the message takes through MMDF once you send your message with the MUA.



**Figure 2-1 Outgoing Mail Route**

The MUA allows the user to create, edit, and start a message on its journey. Then, the MUA transfers the message to **execmail**, which sends it to **submit**(ADM). The **submit** program uses the information in the "To:" line and the MMDF configuration files to build the fully-qualified domain name. From this, **submit** determines the channel to use (for example, UUCP) when sending the message and then places the message in the appropriate channel queue for processing later by the **deliver**(ADM) program. The next time **deliver** runs (by default, **deliver** runs every 10 minutes), it transfers the messages from the channel queue to the appropriate channel program.

For example, you send a message with the following "To:" line:

```
To: andrei@npr.com
```

The **submit** program looks in the appropriate MMDF configuration table for *npr* and builds the fully-qualified domain name; for example:

```
scribe.npr.com
```

Now, **submit** determines the appropriate channel to use to send the message. In this case, for example, all messages in the *npr.com* domain are sent to the outside world on the UUCP channel, so **submit** puts the message in the UUCP channel queue. Then **deliver** picks up the message and passes it to the channel program. The channel program transfers the message out of the MMDF mail system and into the UUCP subsystem where it is queued and sent via UUCP.

## Incoming mail

MMDF processes incoming mail in much the same way it processes outgoing mail; Figure 2-2 illustrates how MMDF handles incoming mail.

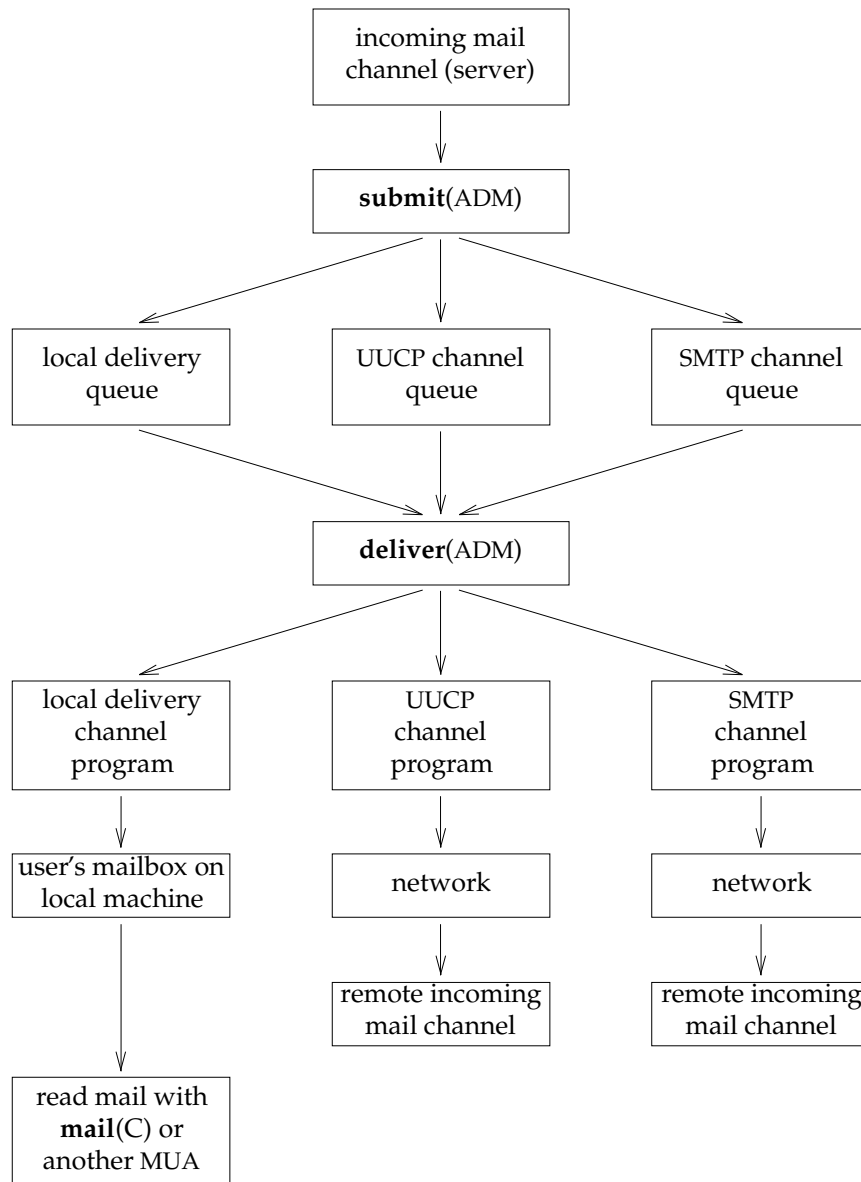


Figure 2-2 Incoming Mail

MMDF processes incoming mail exactly as it processes outgoing mail (the same as when the MUA calls **submit**(ADM)) except that MMDF adds a "Received" line.

MMDF receives incoming mail from a channel's input program (called a "server"), such as **rmail**(C) for the UUCP program. The **submit** program determines the destination of the message and then designates the channel to use to deliver the mail. At this point, MMDF generally selects channels such as **local**, **badusers**, and **badhosts**. Mail destined for the local machine goes to the **local** channel; when the machine is local, but the user is not, mail goes to **badusers**; and if it requires further processing, the message goes to the **badhosts** channel. If the message is destined for another machine, such as when the local machine is acting as a gateway between networks, MMDF selects another network channel, such as UUCP.

The **submit** program puts the message in the appropriate queue and the **deliver** program transfers the mail from each queue to the appropriate channel; eventually the message is delivered to the user.

## *The configuration utility*

---

The configuration utility facilitates MMDF configuration by building the appropriate configuration files at installation time. All sites should use the configuration utility to perform the initial MMDF configuration.

The configuration files that this utility builds are sufficient for most sites. If your site is one that the configuration utility does not handle, after building the initial configuration, you must edit the MMDF configuration files manually using the information in the sections "MMDF configuration files" and "Modifying MMDF configuration". (The section "Configuration utility limitations" later in this chapter explains the limitations of this utility.)

In addition, if your configuration changes (for example, you add another machine to your network), you must rebuild the configuration files either by rerunning the configuration utility or by editing these files manually.

## Configuring communications channels

---

Before you set up MMDF, you must configure all the communications channels that you plan to use to route mail. MMDF supports the following channels:

- UUCP (UNIX-to-UNIX Copy subsystem)
- SMTP (TCP/IP)\*

**NOTE** SMTP and TCP/IP channels form part of the TCP/IP product and are not supplied with the UNIX operating system.

The configuration utility configures for all the channels provided by SCO: UUCP, and SMTP. MMDF also supports other third-party mail products that provide channel programs; however, you cannot use the configuration utility to set up MMDF to use these products. See the section “Advanced MMDF configuration tasks” for information on configuring MMDF for use with other channels. For example, if you plan to exchange mail with other hosts via UUCP, you must configure UUCP on the local host before running the configuration utility. For information on configuring UUCP, see the “Building a remote network with UUCP” chapter in this guide.

## Configuration utility limitations

---

This section discusses the situations in which the configuration utility builds either an incorrect or incomplete MMDF configuration for your site. If the situation at your site matches one or more of the situations in this list, you must manually update the MMDF configuration files after building the initial configuration with the configuration utility.

- You use more than one communications channel to exchange mail with a particular host.

If your host communicates with another host using more than one channel, the configuration utility sets up MMDF to communicate with that host using the first configured channel in the following order:

1. TCP/IP
2. UUCP

If your host accesses another host using two or more channels, but you do not want to transfer mail using the first applicable channel, the configuration utility does not configure the MMDF correctly. See the “Modifying MMDF configuration” section for more information.

---

\* The configuration utility performs basic configuration for routing mail over TCP/IP; however, if this configuration is not complete or you change your MMDF configuration, you must refer to your *TCP/IP Administrator's Guide* for details on modifying MMDF configuration.



- Your host does not exchange mail with every host on each configured channel.

The configuration utility automatically sets up MMDF to exchange mail with every host on the configured channels. For information on removing hosts from a configured channel, see the sections on modifying the information for that channel in “Modifying MMDF configuration”.

- If you exchange mail with two hosts and you want to prevent one host from passing mail to the other host through the local host.

The configuration script sets up MMDF to allow one host to send mail through the local host to the other host. If you want to prevent this, you must set up authorization to prevent the transfer. For information on how to do this, see the section “Configuring MMDF authorization” later in this chapter.

- You have more than one channel of the same type.

The configuration utility lists all the hosts that your host accesses using a particular communications protocol in the same channel. To set up authorization to restrict a host’s access on a channel, see “Configuring MMDF authorization”.

If your configuration changes for some reason after running the utility (for example, you might add a new UUCP host), you should re-examine this list. If the utility is still able to set up your site correctly, you can update your configuration by running the utility again; otherwise, you must update the configuration files manually.

## ***MMDF configuration checklist***

---

Once you configure the communications channels, use the checklist in this section to write down the information that you need to respond to prompts during the configuration procedure. Items in the checklist that are surrounded by boxes pertain to specific configurations (for example, if you are routing mail over UUCP); fill out these sections only if you have these configurations.

The sections following the checklist contain information on each of the items in the checklist.

## Configuration Checklist

1. Host name: \_\_\_\_\_
2. Domain name: \_\_\_\_\_
3. Fully-qualified host name: \_\_\_\_\_
4. Hide your host name? ☐ Yes  
☐ No
5. Redirect mail for *root*?: \_\_\_\_\_
6. Redirect mail for *mmdf*?: \_\_\_\_\_
7. *postmaster* address: \_\_\_\_\_

If you have UUCP installed and configured and you  
plan to use MMDF to route mail over UUCP:

- UUCP host name: \_\_\_\_\_
- UUCP hosts with full  
domain names? ☐ No  
☐ Yes; enter domain names:  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

If you have TCP/IP installed and configured and you plan  
to route mail over TCP/IP (SMTP):

- Configure a name server? ☐ Yes  
☐ No

8. Smart host (badhosts): \_\_\_\_\_  
(where to route mail to unrecognized machines)
9. Smart host (badusers): \_\_\_\_\_  
(where to route mail to unrecognized users)

Use the information in the following sections to complete the checklist.

## *Host name*

---

The HOST NAME (or machine name) is the name of the machine on which you are configuring MMDF. To determine the host name, enter this command at your UNIX system prompt:

**cat /etc/systemid**

An example of a host name is *scribe*.

## *Domain name*

---

A DOMAIN NAME is the section of a mail address that appears to the right of the at “@” character, for example, *npr.com*. The domain describes the site where your machine is located and generally includes the machine (host) name, a department (optionally), and the site’s organization or country. MMDF uses the domain to deliver the message to the appropriate location. Note that the domain name uniquely identifies a machine, but not the path by which messages reach that machine.

The following is the convention for specifying domains:

***hostname.subdomain.top-level***

If the domain includes a department, the convention is:

***hostname.local.subdomain.top-level***

Here is a description of each of the domain levels:

### **Top-Level Domain**

A top-level domain is an officially registered name that describes the purpose of a group of institutions or a code that is associated with a country.

You can only use registered top-level and subdomain names if you have registered your organization with Government Systems. For information on registering your domain, see the section “Registering domain names” later in this chapter. If you have not registered with Government Systems, use the UUCP top-level domain.

In the United States, the common top-level domains on the Internet are: \*

COM	commercial institutions
<i>code</i>	country code †
EDU	educational and research institutions
GOV	government institutions
MIL	military institutions
NET	network
ORG	organization (generic)
UUCP	an unregistered domain name where users transmit information between cooperating neighbor machines via UUCP.

#### **Subdomain**

An officially registered name that describes a company, department, or any subgroup under a top-level domain; *sco* is an example of a subdomain in the domain *COM*.

#### **Local Domain**

A name recognized only within an organization that has meaning only within that organization; a department name such as *enr* is an example of a local domain.

### ***Fully-qualified host name***

---

The complete domain name (the machine name and all other domain names) is known as the FULLY-QUALIFIED HOST NAME (or fully-qualified domain name). For example, if you have this mail address:

`andrei@scribe.npr.com`

the fully-qualified domain name is:

`scribe.npr.com`

In this case, *scribe* is the machine name, *npr* is the company, and *com* specifies that the machine belongs to a commercial organization.

---

\* Domain names can be either upper or lowercase; MMDF is case-insensitive when evaluating domain names.

† The International Standards Organization (ISO) standard 3166 defines the country codes. For example, US is the country code for the United States, AU for Australia, DK for Denmark, and JP for Japan.

The following table shows examples of other fully-qualified domain names:

slug.ucsc.EDU	<i>slug</i> is a machine at the University of California at Santa Cruz in the <i>EDU</i> domain
seismo.css.GOV	<i>seismo</i> is a machine at the Center for Seismographic Studies in the <i>GOV</i> domain
nessie.edinburgh.ac.UK	<i>nessie</i> is a machine at Edinburgh University in Scotland in the academic ( <i>ac</i> ) subdomain in the <i>UK</i> domain

To the configuration utility, the fully-qualified host name is the host name followed by the local (if appropriate), subdomain, and top-level domain name as shown in the previous examples. The domain name refers to just the subdomain and top-level domain name (without the host name). For example, *npr.com*.

### ***Registering domain names***

If you or any user at your site plan to receive or send mail outside of your organization, you should register a top-level domain or subdomain with the NIC (Network Information Center). Even if you are only using UUCP, registration with the NIC is your only guarantee that the name of your site is unique. To register a top-level domain and subdomain name call or write to the NIC standards organization at the following address:

Government Systems Inc  
ATTN: Network Information Center  
14200 Park Meadow Drive  
Suite 200  
Chantilly, VA 22021 USA

Help Desk Telephone Numbers:  
1-800-365-3642 (1-800-365-DNIC)  
1-703-802-4535

Help Desk Hours of Operation:  
7:00 am to 7:00 pm Eastern Time

Email: [hostmaster@nic.ddn.mil](mailto:hostmaster@nic.ddn.mil)  
[registrar@nic.ddn.mil](mailto:registrar@nic.ddn.mil)

Fax Number: 1-703-802-8376

The earlier your site enrolls a domain name and the NIC gives you an address, the less likely it is that you will have to alter a machine name or other site identifier later.

## Hide your host name?

---

If you have several machines at your site (domain) and you register your domain name with the NIC, you can “hide” your host name behind your domain name. When you configure MMDF to hide your host name, people outside your organization can send mail to people who receive mail on different machines within your organization without having to know the name of those specific machines. In this case, MMDF identifies all outgoing mail as coming from a single source (your registered domain name). In addition, MMDF routes mail sent to this source to its correct destination within your organization.

For example, the fully-qualified domain name for the host that you are configuring is:

`scribe.npr.com`

You can “hide” the hostname *scribe* behind the domain name *npr.com*. In this case, someone outside the *npr* organization can send mail to *andrei* on *scribe* without specifying the fully-qualified domain name. Instead, they can use *andrei@npr.com* as the address.

In addition, when *andrei* sends mail outside the organization, the message appears as if it came from *andrei@npr.com* instead of from *andrei@scribe.npr.com*.

If you want to hide the current host behind the registered domain name, press `<Return>` when the configuration utility prompts you.

**NOTE** When you join local machines under a single domain name, you create an administrative domain. Within an administrative domain, all user names must be unique so that mail can go to any person anywhere within the domain without a local machine name in the mail address.

If you decide to hide the host behind a domain name, you should create ALIASES to map each user in the domain to the machine where they receive mail. For information on how to do this, see the section “Creating aliases for users” later in this chapter.

## Redirect mail for root?

---

The system sends mail about any system problems to the *root* user. You can configure MMDF to redirect this mail to another person. For example, if you are the system administrator for a group of machines, you can redirect all mail sent to *root* on those machines to your own system mailbox. With this configuration, you do not need to log into each machine to read *root*’s mail.

If you want to redirect *root*'s mail, press `<Return>` when prompted by the configuration utility and then enter the address of the person to whom you want MMDF to deliver *root*'s mail. For example, if you are configuring MMDF on *scribe.npr.com* and you want the user *bob@talk.npr.com* to take care of mail to *root*, redirect *root*'s mail to *bob* by entering this at the prompt:

**bob@talk.npr.com**

(You can redirect *root*'s mail to a local user by entering just the user name at the prompt; you do not need to enter the fully-qualified domain name for a local user.)

### ***Redirect mail for mmdf?***

---

When problems occur with the mail system, the system sends mail to the *mmdf* user. This account is reserved for administering your mail system; unless you log in as *mmdf* regularly, you might not find out about problems with the system. For this reason, it is a good idea to redirect *mmdf*'s mail to another person. To do this, press `<Return>` when the configuration utility prompts you and then enter the address of the person to whom you want to deliver *mmdf*'s mail.

### ***Postmaster address***

---

On the Internet, people generally send any inquiries about user and host names to the "postmaster" address in that domain. RFC (Request for Comments) 821/822 requires that every host provide this reserved postmaster mailbox. For these reasons, you should designate a user on the local system or within the domain as the postmaster and define a *postmaster* alias. To set up the alias while running the configuration utility, simply enter the name of the postmaster user.

### ***Setting up MMDF for UUCP***

---

If you are setting up the system to exchange mail with another system using UUCP, the configuration utility prompts you for some information about your UUCP hosts.

#### ***UUCP host name***

The configuration utility prompts for the UUCP name for the host that you are currently configuring. The configuration utility gets this information from */etc/systemid* (or *uname*); when you install the operating system, the installation prompts you for the system name. In cases where the UUCP name is different from the host name, you should enter the correct name when prompted.

### ***UUCP hosts with full domain names***

The configuration utility asks if any of the UUCP hosts have full domain names. This might be the case if your host is not on the Internet but your site has an agreement with another machine on the Internet to transfer your mail. When people on the Internet send messages to you, they use the *user@machine.domain* address format (instead of the *machine!user* UUCP format). You then use UUCP to connect to that machine and pick up your spooled mail.

For example, your machine *scribe.npr.com* connects to *slug.ucsc.edu* on the Internet. The machine *slug.ucsc.edu* is a UUCP host with a full domain name. In this case, enter *y* when the configuration utility prompts you for UUCP hosts and enter the host name and then the fully-qualified domain name at the prompts.

### ***Configuring MMDF to use a name server***

---

The Versions of MMDF in this EFS includes NAME SERVER support. A name server is a program running on the network that provides a central database of information, such as Internet addresses and the names of hosts on which people receive mail.

If you want the configuration utility to set up MMDF to use the name server, you must set up the name server before running the utility. For details on setting up the name server see the chapter entitled "Name Server Operations Guide for BIND", in *Administering ODT-NET* in the *Administrator's Guide*.

**NOTE** The name server forms part of the TCP/IP product, not the UNIX operating system.

If you already have a name server running, the configuration utility sets up MMDF to use it automatically.

### ***Configuring smart hosts***

---

If the machine that you are configuring communicates directly with another machine that has more complete information about the entire mail network, you can set up MMDF to route any mail that it does not recognize to that machine. This machine is known as a "smart host". MMDF recognizes two kinds of smart host: hosts with information about machine names and hosts with information about the users on the network.

#### ***Badhosts channel***

If you specify a smart host for the machine names, MMDF routes mail destined for machines that it does not recognize to the **badhosts** channel. All mail directed to this channel is sent to the smart host that you specify via the normal channel (such as UUCP) that you use to communicate with that host.



For example, someone sends mail to *boris@kgb.gov* but the local host does not recognize the machine *kgb.gov*. However, your host uses UUCP to communicate with another machine, *rocky.npr.com*, that has more complete information about the network. If you configure *rocky.npr.com* as your smart host, MMDf puts the message to *boris@kgb.gov* in the **badhosts** channel and then uses UUCP to deliver it to *rocky.npr.com*. The mail system on *rocky.npr.com* then determines the correct route to send the message to its destination.

### ***Badusers channel***

You can also specify a smart host that contains complete information about all the users on a network. This is useful if you have a large number of people at your site. Instead of maintaining information on all the people at the site on each host, you can maintain this information on one central host (the smart host); each individual host maintains information about the local users only. In this case, any mail addressed to users that the local machine does not recognize is directed to the **badusers** channel. The mail routed on this channel is sent to the smart host via the channel (such as UUCP) that you usually use to communicate with that host.

For example, someone on the local host sends mail to *natasha* but the host does not recognize her as a local user. If you communicate with *moose.npr.com*, a smart host that contains complete user information, MMDf routes the message to this host. This does not mean that *natasha* actually receives her mail on *moose*, just that *moose* has more information about where *natasha* is located.

## ***Running the configuration utility***

---

Once you configure the communications channels (and name server, if appropriate) and complete the configuration checklist, you are ready to run the MMDf configuration utility.\*

**NOTE** The configuration utility provides common default values for many of the prompts; to accept these values, press <Return> at the prompt.

The configuration procedure varies slightly (from step 11 onwards) depending on whether TCP/IP is installed. The procedure detailed in this section assumes that TCP/IP is not installed. If TCP/IP is installed on your system, follow the instructions down to step 11 and then continue from the section headed "Running the configuration utility with TCP/IP installed" later in this chapter.

---

\* Remember, if you are only planning to route mail on the local machine, you do not have to run the configuration utility; MMDf is already set up for local mail delivery.

Use the following procedure to configure MMDF:

1. Change to the */usr/mmdf* directory and start the configuration utility by entering the following command as *root*:

**mkdev mmdf**

You see a screen with information about the limitations of the configuration utility.

2. At the bottom of the screen, you see this prompt:

```
Do you wish to continue the configuration process at this time? [y]
```

If you enter **n** at this prompt, you exit the configuration utility and return to the UNIX system prompt. Press **<Return>** or enter **y** to continue with the configuration.

The configuration utility displays a message about the current version of the software, for example:

```
This machine is running level 43NS MMDF.
```

3. If you have not already set the host name on the machine that you are configuring, you see the following prompt (if you have already set the host name, skip this step):

```
The name of this host has not been configured yet.  
What will you be calling this host?
```

Enter the host name at this prompt. You must set the name of the host before configuring MMDF.

4. Now, you see the following prompt where 'host.domain' is a fully-qualified host name like *scribe.npr.com*:

```
Is your fully qualified host name 'host.domain'? [y]
```

If the fully-qualified host name is correct, press **<Return>** and go to the next step. If the host name is incorrect, enter **n**. The utility displays this prompt:

```
What is the correct host name? [host]
```

Enter the correct hostname (for example, *pubsco*) or press `<Return>` to select the default hostname. You see the prompt for the domain name:

```
What is the correct domain name? [domain]
```

There are three possible responses to this prompt: you can press `<Return>` to accept the default domain name, you can enter a different domain name (for example, *eng*), or you can enter a domain and sub-domain name (for example, *techpubs.eng*).

Now, the utility prompts you to confirm the fully-qualified host name:

```
Is your fully qualified host name 'pubsco.techpubs.eng'? [y]
```

If the hostname is correct, press `<Return>` and go to the next step; otherwise enter **n** and repeat this step.

If you have specified a fully qualified host name which includes a sub-domain, for example, *pubsco.techpubs.eng*, the prompt shown in step 5 is displayed; if you have specified a fully qualified host name which does not include a sub-domain, for example, *pubsco.eng*, step 5 is bypassed.

5. Now, you see some information about “hiding” the host behind the domain, followed by this prompt:

```
At many sites, it is common for mail to be addressed as being from
"person@site" instead of from "person@machine.site". This allows people
to be moved between machines internally without requiring them to notify
all their external correspondents about the address change. This
configuration does, however, require a complete user alias table (see
below) containing mappings from user names to the host that they actually
plan to read their mail on. If you are not sure, then you should
probably answer "yes" if there are two or more machines in the domain
"techpubs.eng", and "no" otherwise.
```

```
Do you wish to hide 'pubsco.techpubs.eng' behind 'techpubs.eng'? [y]
```

To “hide” the host that you are configuring behind another host, press `<Return>` at this prompt. Otherwise, enter **n**.

## Setting up electronic mail

6. Now, the configuration utility displays some information about creating mail aliases for special accounts, *root*, *mmdf*, and *postmaster*, on your system.

First, the utility prompts you to create an alias for *root*:

```
Do you wish to have mail for root redirected to a real user? [y]
```

If you want mail addressed to *root* to go to *root*'s system mailbox, enter **n** and go to the next step.

If you want *root*'s mail to go to a different person, press **<Return>** and you see this prompt:

```
To whom should root's mail be sent?
```

Enter the login name of the person that you want to receive mail directed to *root*.

7. Now, you see this prompt:

```
Do you wish to redirect mail addressed to mmdf? [y]
```

If you do not want to redirect mail to *mmdf*, enter **n** and go to the next step. If you want mail addressed to *mmdf* to go to another person (for example, the system administrator), press **<Return>**.

```
To whom should mmdf's mail be sent? [root]
```

The default for this prompt is the same as the selection made for step 6. For example, if *root* was selected in response to step 6 then *root* is displayed here.

To have *mmdf*'s mail go to *root*'s mailbox, simply press **<Return>**. If you want this mail directed to another person, enter the address of that person.

8. You see this prompt:

```
To whom should mail addressed to postmaster be sent? [mmdf]
```

Again, you can direct any mail sent to *postmaster* to another person, such as *mmdf*. Press **<Return>** to accept the default or enter a different address.

9. If UUCP is not installed on your system, you see a message like this:

```
UUCP not installed, skipping ...
```

10. Now, you see some information about associating login names for each user to the machines where the users actually read their mail by creating an alias file. See the section “Creating aliases for users” later in this chapter.

The configuration utility also displays some information about converting XENIX aliases to the correct MMDF format. For more information, see the section “Converting XENIX alias files” later in this chapter.

If you have UUCP installed on your system, you see this prompt:

```
Are you going to be using UUCP for mail? [y]
```

If you plan to use UUCP for transferring mail, press **<Return>**.

**NOTE** If you are running the configuration utility with TCP/IP installed on your system, go to the section “Running the configuration utility with TCP/IP installed.”

11. Now, you see the prompt for the UUCP host name:

```
Is this host known as 'host' for UUCP? [y]
```

If the host name is correct, press `<Return>` and go to the next step. Otherwise, enter `n` to display the following prompt:

```
What is this host's UUCP name? [host]
```

Enter the new name and press `<Return>`. You see this prompt again:

```
Is this host known as 'host' for UUCP? [y]
```

Press `<Return>` if the host name is correct, or press `n` to change it.

12. At this point, if the configuration utility detects a UUCP connection to the *sosco* machine at SCO, you see the following prompt; if not, go to the next step.

```
uname reports a connection to sosco.  
Does this link really exist? [n]
```

If your system has a UUCP connection for exchanging mail with *sosco*, enter `y` and go to the next step. If not, enter `n` to display the following message:

```
Deleting sosco from list of mail sites
```

If *sosco* was the only configured UUCP connection, the utility exits and displays the following message:

```
UUCP must be configured before mail
```

In this case, configure your UUCP connections before running **mkdev mmdf** again.

13. Now, the configuration utility prompts you for information about the hosts you communicate with using UUCP:

```
Because UUCP does not maintain information about domain names, it will be
necessary for you to provide the domain names of any of the hosts with
which you communicate via UUCP.
```

```
Do any of your UUCP hosts have full domain names? [n]
```

In most cases, UUCP hosts do not have full domain names, so you can simply press **<Return>** at this prompt.

If any of your UUCP hosts have full domain names, enter **y**. You see the prompts for the host names of your UUCP hosts:

```
Enter the UUCP site name (blank to terminate):
```

Enter the host name, for example, *pubsco* (not the fully qualified name) and press **<Return>**.

14. The following screen is displayed:

```
What is pubsco's fully qualified name? [pubsco]
```

Enter the fully qualified name, for example, *pubsco.techpubs.eng* and press **<Return>**.

15. You see the following messages:

```
Micnet not configured, skipping ...
```

```
TCP/IP not installed, skipping ...
```

```
Many sites do not have complete information about the entire mail network,
but rely on another "smarter" host to determine the correct route that
mail messages should follow to reach their destinations. Any mail that
the local machine is incapable of correctly handling is passed to the
smart host for further processing. In MMDF, this is called the "badhost"
channel.
```

```
Do you have such a "smart" host? [y]
```

If you enter **n**, go to the next step; if you enter **y** the following screen is displayed:

```
What is its name? (q if you have changed your mind)
```

16. The following screen is displayed:

```
Another option, which is often used on large sites is to have a central
machine which contains complete knowledge about all the users on the site,
and only maintaining local lists on each machine. MMDF provides the
facility to forward mail containing unrecognised local addresses to a
smarter host which will have a complete user data base (via the "baduser"
channel).
```

```
Do you have such a "smart" host? [y]
```

If you enter **n**, go to the next step; if you enter **y** the following screen is displayed:

```
What is its name? (q if you have changed your mind) [pubsco.techpubs.eng]
```

17. When you finish, the configuration utility displays information like the following as it creates and edits the MMDF configuration files (the messages might be different, depending on your configuration):

```
Creating the mmdftailor file: header, host name info, support
address, alias tables, local domain and channel, SMTP, root

Building the alias tables (mostly empty)
A Mailing list alias table (alias.list) already exists, skipping
A general user alias table (alias.user) already exists, skipping

Building channel files
local, list, SMTP, UUCP, badusers, badhosts

Building the domain tables
local, domain, root
done
building the database
```

The configuration is now complete.



## *Running the configuration utility with TCP/IP installed*

This section assumes that TCP/IP is installed on your system. It continues on from step 10 in the previous section.

11. You see the following messages:

```
Micnet not configured, skipping ...  
Are you going to be using SMTP for mail? [y]
```

If you enter **n** go to step 13; if you enter **y** go to step 12.

12. The configuration utility displays one of the following messages:

- If the name server is already configured on your system, you see the following screen:

```
A domain name server is running, mail will be configured to use it.
```

In this case, the configuration utility configures MMDF on the host that you are configuring to use the name server automatically.

- If the name server is not configured, you see:

```
The name server is not currently configured on this machine.  
Do you plan to configure a name server on the local network? [n]
```

If you do not want to configure a name server, press **<Return>** to continue with the configuration utility.

If you plan to configure a name server, you must do so before configuring MMDF. In this case, enter **y**. The configuration utility exits and displays this message:

```
Name server must be configured before mail.
```

Set up the name server and then run **mkdev mmdf** again. For information on configuring the name server tables, refer to Chapter 4, "Name Server Operations Guide for BIND", in your *TCP/IP Administrator's Guide*.

13. The following screen is displayed:

```
Many sites do not have complete information about the entire mail network,  
but rely on another "smarter" host to determine the correct route that  
mail messages should follow to reach their destinations. Any mail that  
the local machine is incapable of correctly handling is passed to the  
smart host for further processing. In MMDF, this is called the "badhost"  
channel.
```

```
Do you have such a "smart" host? [y]
```

If you enter **n**, go to the next step; if you enter **y** the following screen is displayed:

```
What is its name? (q if you have changed your mind) [ ]
```

14. The following screen is displayed:

```
Another option, which is often used on large sites is to have a central  
machine which contains complete knowledge about all the users on the site,  
and only maintaining local lists on each machine. MMDF provides the  
facility to forward mail containing unrecognised local addresses to a  
smarter host which will have a complete user data base (via the "baduser"  
channel).
```

```
Do you have such a "smart" host? [y]
```

If you enter **y** the following screen is displayed:

```
What is its name? (q if you have changed your mind) [ ]
```

15. When you finish, the configuration utility displays information like the following as it creates and edits the MMDF configuration files (the messages might be different, depending on your configuration):

```

Creating the mmdftailor file: header, host name info, support
address, alias tables, local domain and channel, SMTP, root

Building the alias tables (mostly empty)
A Mailing list alias table (alias.list) already exists, skipping
A general user alias table (alias.user) already exists, skipping

Building channel files
local, list, SMTP, UUUCP, badusers, badhosts

Building the domain tables
local, domain, root
done
building the database

```

The configuration is now complete.

## Preparing MMDF for use

Before you can use your mail system, you must perform steps to notify the UNIX system of the new configuration. The following sections contain information on restarting **deliver** and making this change permanent in the MMDF system startup file.

### Restarting the deliver daemon

When you enter multiuser mode, the system automatically starts a **deliver** process for the local channel only. Each time someone on the system sends or receives a message, the **deliver** daemon runs, placing the message in the correct channel.

With the MMDF configuration utility, you configured MMDF to use additional channels. However, the configuration utility does not affect the **deliver** daemons that are currently running. To make the changes to your MMDF configuration take effect, stop and restart any **deliver** daemons using these commands:

1. As MMDF enter the following command:

```
ps -ummdf | grep deliver
```

This command displays any **deliver** processes running on your system. For example:

```
285 ?      0:00 deliver
```

2. Stop the **deliver** process with **kill(C)**. For example:

```
kill 285
```

3. Now, restart **deliver** with this command:

```
/usr/mmdf/bin/deliver -b
```

This command restarts the **deliver** daemon so that it runs every 10 minutes. If you want **deliver** to run more often, use the **-T** option. We recommend that you set up **deliver** to run every 60 seconds using the following command:

```
/usr/mmdf/bin/deliver -b -T60
```

### *Modifying MMDF system startup*

Unless you want to run multiple **deliver**(ADM) daemons, you do not need to modify MMDF system startup. The MMDF system startup file `/etc/rc2.d/S86mmdf` already includes information for starting the **deliver**(ADM) daemon on all the configured channels. The system reads the `S86mmdf` file automatically when you enter **init** state 2 (multiuser mode).

By default, the line in this file looks like the following:

```
/bin/su mmdf -c "/usr/mmdf/bin/deliver -b"
```

To edit `S86mmdf`:

1. First, log in as *root*.
2. Edit `/etc/rc2.d/S86mmdf`.
3. If you want **deliver** to run more often than the default of every 10 minutes, add the **-T** option on this line. For example, set up your system so that **deliver** runs every 60 seconds using the following **deliver** startup line:

```
/bin/su mmdf -c "/usr/mmdf/bin/deliver -b -T60"
```

**NOTE** When you start the **deliver** program with the **-b** option only, *one deliver* process manages *all* the channels that you specified with the configuration utility. If you want different **deliver** processes to manage each of your channels or more than one **deliver** process for a single channel, refer to the section “Running multiple deliver daemons” later in this chapter.

## Other common configuration tasks

---

Depending on your configuration, you might need to perform some additional MMDF configuration tasks after running the configuration utility and restarting **deliver**. This section covers some common configuration tasks; for additional configuration information, refer to the sections “Modifying MMDF configuration” and “Advanced MMDF configuration tasks” later in this chapter.

### Creating aliases for users

If you chose to hide your local host name behind your domain name in step 6, you should add the login names of the users on the system and machine names where they receive mail to the `/usr/mmdf/table/alias.user` file. The configuration utility creates `alias.user` for you; add information to this file using the following format:

**user:** `user@machine.domain`

For example, to map *andrei* to the machine *scribe* in the *npr.com* domain, use this format:

**andrei:** `andrei@scribe.npr.com`

See the sections “Alias files” and “Maintaining user aliases” for more information on setting up aliases on your system.

### Converting XENIX alias files

If you are converting your mail system from XENIX to UNIX, you must use the **mmdfalias** utility to convert a XENIX-style `/usr/lib/mail/aliases` file to the correct format for MMDF. For information on how to do this, see the section “Converting from XENIX” later in this chapter.

### Changing logging levels

By default, the configuration utility sets the logging levels to log fatal errors only (FAT). If you are switching to MMDF from a **sendmail**-based system, you might want to change the **MMSGLOG** tunable parameter, in the `/usr/mmdf/mmdftailor` file, from FAT to FST to retain a logging level equivalent to “mqueue/syslog”. See the sections “Changing error logging levels” and “Changing MMDF parameters” for more information on changing logging parameters.

## Rebuilding the MMDF hashed database

---

The configuration utility rebuilds the MMDF hashed database automatically after modifying the configuration files. However, if you modify the alias files or `/usr/mmdf/mmdftailor` **after** running the MMDF configuration utility, you must rebuild this database manually.

To do this, enter the following commands as user *mmdf*:

```
cd /usr/mmdf/table
./dbmbuild
```

**NOTE** You must rebuild the MMDF database every time you make changes to the alias or routing files in the */usr/mmdf/table* directory.

# *MMDF configuration files*

The configuration utility provided with your MMDF distribution builds configuration files sufficient for most sites, but not for every site; see the section on the limitations of the configuration utility earlier in this chapter for more information. In cases where the files that the configuration utility builds are insufficient, you must edit these files manually. This section describes the different configuration files, how MMDF uses them, and how to modify them to work with your configuration. This section also explains how to test and maintain the MMDF system.

The material presented in this section assumes that you have already set up the channels over which you plan to route mail and that you built the initial MMDF configuration using the configuration utility.

This section begins with an introduction to the different configuration files and formats that MMDF uses to route mail.

Table 2.1 briefly describes the MMDF configuration files.

**Table 2-1** MMDF configuration files

File(s)	Description
mmdftailor	defines all the mail attributes for the local machine
alias files	defines aliases for mailing lists, programs, users, and system accounts
domain files	matches host names to fully qualified host names
channel files	expands fully qualified host names to specify the addressing information necessary to reach the host or domain

The following sections describe the format of these files in detail. The **tables(F)** manual page also contains information about the file formats.

## *The mmdftailor file*

---

The `/usr/mmdf/mmdftailor` file defines all the mail attributes for the local machine, such as its name, the domain, channel, and alias files to use, how to set up each channel, and how to perform logging.

By default, MMDF is distributed with a simple `mmdftailor` file that is configured for local mail only. When you perform the initial MMDF configuration (as described in the section “Running the configuration utility”), the configuration utility modifies the organization of the default `mmdftailor` file. This section describes the `mmdftailor` file generated by the configuration utility. Your `mmdftailor` file might be set up differently (for example, the names of your hosts are different and your `mmdftailor` file might not include all the channels in the example).

Table 2.2 briefly describes the keywords in the *mmdftailor* file; the sections that follow the table explain these keywords in more detail.

**Table 2-2 mmdftailor file keywords**

Keyword	Description
<b>MLDOMAIN</b>	defines the top-level domain name (for example, <i>COM</i> or <i>GOV</i> )
<b>MLNAME</b>	specifies the host name ( <i>scribe</i> ) or site name ( <i>npr</i> ) for hidden hosts
<b>MLOCMACHINE</b>	specifies the host name (for hidden hosts)
<b>UUnicode</b>	specifies the host name for UUCP
<b>MSUPPORT</b>	sets the address ( <i>postmaster</i> ) to send undeliverable mail and requests for information
<b>MTBL</b>	associates an abbreviated name with the more descriptive name for the alias, channel and domain files (also called “tables”). These abbreviated names are then used throughout the <i>mmdftailor</i> file as shorthand to refer to the table files.
<b>ALIAS</b>	defines the various sources for alias information, using the abbreviated names specified in the <b>MTBL</b> definitions
<b>MCHN</b>	defines the channels available to MMDF for mail transport*
<b>MDMN</b>	describes the domains known to MMDF
<b>MMSGLOG</b>	controls logging information from <b>deliver</b> (ADM) and <b>submit</b> (ADM)
<b>MCHANLOG</b>	controls logging that <b>MMSGLOG</b> does not control
<b>MLCKTYPE</b>	specifies the mailbox locking protocol

The **MTBL**, **ALIAS**, **MCHN**, and **MDMN** keywords accept specific parameters; Table 2.3 briefly describes the parameters in the default *mmdftailor* file. See the **mmdftailor**(F) manual page for complete information about each of these keywords and their associated parameters.

\* MMDF requires at least one channel (**local**), for delivering local mail. See the section “Local mail configuration” for more information.



**Table 2-3 mmdftailor file parameters**

Keyword	Parameter	Description
<b>MTBL</b>	name	specifies the shortname (table) for file
	file	describes the filename to associate with the shortname
	show	sets the description to display with programs, such as <b>checkup</b> (ADM)
<b>ALIAS</b>	table	specifies the name of the table to associate with the alias entry
	nobypass	prevents the <i>~address</i> alias bypass mechanism from working on the aliases in the specified table
	trusted	states that any alias in the described file is permitted to deliver mail to files or pipes using the permissions of any user on the system (including <i>root</i> )*
<b>MCHN</b>	show	sets the description to display with programs
	ap	specifies the type of address parsing used for the header of outgoing messages
	name	specifies the name of the channel
	mod	sets the delivery mode for the channel
	host	specifies the name of the host that is being contacted by this channel
	tbl	defines the file that lists the hosts that are accessible via the specific channel
	confstr	specifies channel-specific configuration parameters
<b>MDMN</b>	pgm	defines the channel program to invoke for this channel
	show	sets the description to display with programs
	table	specifies the name of the table containing information that describes the sites in this domain
	dmn	specifies the name of the domain

\* Normally, because this permits such accessibility, you should only set the *alias* file (used for administrative aliases) to **trusted**.

## ***MLDOMAIN***

The **MLDOMAIN** keyword describes the top-level domain used by the local host. If the local machine has a registered top-level or subdomain name, the top-level domain name for your site, such as *COM* or *EDU*, appears in the **MLDOMAIN** line. If your host does not have a top-level domain, use **UUCP**. (**UUCP** is a generic name that **MMDF** uses for **UUCP** connections.) The **MLDOMAIN** line in the following example shows *COM* as the top-level domain:

```
MLDOMAIN com
```

## ***MLNAME and MLOCMACHINE***

Generally, **MLNAME** describes the name of the local host. In the following example, *scribe* is the host name:

```
MLNAME scribe
```

However, if you are “hiding” the local machine behind a domain name, **MLNAME** describes the subdomain name of your site (for example, *npr*) and **MLOCMACHINE** describes the name of the local host. (**MLOCMACHINE** is only used if you are hiding the local host.)

For example, if you are hiding the local host, *scribe*, behind the *npr* subdomain, these lines look like the following:

```
MLNAME npr
MLOCMACHINE scribe
```

For more information about hiding the host name, see the section “Hide your host name?” earlier in this chapter.

## ***UUname***

The **UUname** line describes the name used with **UUCP**; this definition must exist for **UUCP** to work properly. In most cases, this name is the same as the host name (in either **MLNAME** or **MLOCMACHINE**), but can be different if required. For example:

```
UUname scribe
```

## ***MSUPPORT***

The **MSUPPORT** line describes the address to send mail delivery problem notifications. If your site is connected to the Internet, you must define **MSUPPORT** as *postmaster*. On the Internet, people use this address to send any inquiries about user and host names in the domain. In addition, RFC821/822 (Request for Comments, an Internet standard) requires that every host on the Internet provide the reserved *postmaster* mailbox. Use the following line:

```
MSUPPORT postmaster
```

Note that the address you specify with **MSUPPORT** must be legal; if it is not and MMDF cannot deliver the original undeliverable mail to the support address, MMDF creates a new piece of mail that is undeliverable, and so on until the machine runs out of processes.

You can either create an alias in the *alias.ali* file to redirect *postmaster*'s mail to a user on the system, or create a *postmaster* account. For more information, see the "Alias files" and "Changing the postmaster alias" sections later in this chapter.

### Alias configuration

The **MTBL** alias configuration keywords identify the filenames for the default alias files. By default, these **MTBL** lines look like the following:

```
MTBL name=alias,      file="alias.ali",    show="Administrative aliases"
MTBL name=lalias,     file="alias.list",  show="Mailing list aliases"
MTBL name=auser,      file="alias.user",    show="General user aliases"
```

Each **name** parameter defines the short name for the actual filename containing alias information (specified by **file**). Thus, the first **MTBL** entry identifies "alias" as the shortname for the *alias.ali* file. (MMDF uses this file to define the system administrative aliases for the local host.) See the "Alias files" section later in this chapter for more information about these files.

The **ALIAS** alias configuration keywords define additional information about the alias files in the **MTBL** lines. The following example shows the default **ALIAS** lines:

```
ALIAS  table=alias,      nobypass,    trusted
ALIAS  table=lalias,     nobypass
ALIAS  table=auser
```

**NOTE** The order of the alias files defined in this section determines the order that MMDF searches the files to find an alias.

For more information, see the "Alias files" section later in this chapter.

### Local mail configuration

As with alias configuration, the **MTBL** keywords identify the filenames that contain local mail configuration information. The following example shows these table definition lines:

```
MTBL name=local,      file="local.chn",    show="Local Host Aliases"
MTBL name=locdom,     file="local.dom",    show="Local Domain"
```

In this example, the first **MTBL** entry identifies "local" as the shortname for the *local.chn* file. (The *local.chn* file contains information about the **local** channel on the local machine.) See the section "Channel files" later in this chapter for more information about these files.

Below the **MTBL** definitions are the **MCHN** and **MDMN** definitions for local mail delivery. By default, the **MCHN** line for local delivery looks like the following:

```
MCHN    local,    show="Local Delivery",    ap=822,    mod=imm
```

The first parameter after **MCHN** is an arbitrary name that describes the channel (in this case, **local**). For a description of the **MCHN** parameters, see the **mmdftailor(F)** manual page. MMDF searches the channel tables in the order that the **MCHN** definitions appear in *mmdftailor*.

The local mail **MDMN** line looks like the following:

```
MDMN    "npr.COM",    show="Local domain",    table=locdom
```

The first parameter on the **MDMN** line specifies the name of the domain that the table lists (in this case, the local domain *npr.COM*).

### List processing configuration

The list processing configuration section of *mmdftailor* defines the table and channel that MMDF uses to process mailing lists. By default, these lines look like the following example:

```
MTBL    list,    file="list.chn",    show="List Channel"
MCHN    list,    show="List Processing",    ap=same,    mod=imm,
        host="scribe.npr.COM"
```

The **MTBL** line defines the shortname for the file *list.chn*. This file contains information about passing mail addressed to mailing lists to the **list-processor** program (see the “The alias.list file” section later in this chapter for more information).

### SMTP configuration

If you configured your system to route mail over TCP/IP, your *mmdftailor* file contains **MTBL** and **MCHN** entries like the following:

```
MTBL    smtpchn, file=smtp.chn, show="SCO SMTP Channel"
MCHN    smtp, show="SCO SMTP Delivery", ap=822, tbl=smtpchn, mod=imm,
        confstr="scribe.npr.COM"
```

**MTBL** describes the SMTP channel file, *smtp.chn*. **MCHN** gives more information about the SMTP channel; the **mmdftailor(F)** manual page covers these parameters in more detail.

## Local domain table configuration

The **MTBL** and **MDMN** entries in the local domain table configuration section describe the machines in the local domain. These lines look like the following:

```
MTBL domain,      file=domain.dom,      show="Local Ethernet "
MDMN "npr.COM",  show="Local Ethernet", table=domain
```

The first parameter on the **MDMN** line specifies the name of the domain that the table lists (in this case, the *npr.COM* domain). The *domain.dom* file describes the machines located in the local domain.

## UUCP configuration

If you configure MMDF to route mail over UUCP, the configuration utility adds parameters like the following that describe the UUCP configuration information:

```
MTBL uuchn,  file="uucp.chn",  show="NPR UUCP Channel"
MTBL uudom,  file="uucp.dom",  show="NPR UUCP Domain"

MCHN uucp,  show="NPR UUCP Delivery",  tbl=uuchn, ap=same
MDMN "UUCP", show="UUCP Domain",  table=uudom
```

The two **MTBL** entries define the UUCP channel (*uucp.chn*) and UUCP domain (*uucp.dom*) files. The **MCHN** line describes more information about the UUCP channel and **MDMN** defines the file that describes the machine accesses via UUCP.

## The badhosts channel

If you specified a “smart host” to redirect mail destined for machines that the local host does not recognize, the **MCHN badhosts** line looks like this:

```
MCHN badhosts, show="Last-Chance Routing", pgm=smtp, tbl=smtpchn,
ap=822, host="rocky.npr.COM", confstr=scribe.npr.COM
```

The **MCHN host** parameter describes the smart host (in this case, *rocky.npr.COM*). In this case, the local host accesses the smart host via SMTP. The **pgm** parameter specifies the filename of the channel program (*smtp*) in the */usr/mmdf/chans* directory and **tbl** specifies the file (*smtp.chn*) that describes the machines that the host accesses with SMTP.

The **badhosts** is not really a channel because it is not associated with its own transport program. In this example, the pseudo-channel uses the SMTP channel to relay mail to a more intelligent host. If the **badhosts** channel does not exist, MMDF returns mail to an unknown host to the sender.

### *The badusers channel*

If you specified a “smart host” to redirect mail to users that the local host does not recognize, the **MCHN badusers** line looks like this:

```
MCHN badusers, show="Last-Chance Routing", pgm=smtp, tbl=smtpchn,  
ap=822, host="moose.npr.COM", confstr=scribe.npr.COM
```

The **MCHN host** parameter describes the smart host (in this case, *moose.npr.COM*). See the previous section for information about the **pgm** and **tbl** parameters.

### *The root domain table*

The **MTBL** line defines the *root.dom* file which contains any domain information not named in other domain files.

```
MTBL rootdom, file="root.dom", show="Root Domain"  
MDMN "", show="Root Domain", table=rootdom
```

The **MDMN** entry defines the name of root domain. (The root domain definition has no name ("") because the root domain file (*root.dom*) can contain entries for many different domains.)

### *Logging levels*

By default, the configuration script sets **MMSGLOG** and **MCHANLOG** to the **FAT** logging level; this level logs fatal errors only. These entries look like the following:

```
MMSGLOG level=FAT  
MCHANLOG level=FAT
```

You can change these logging parameters to a higher level, such as **FST** to log full statistics, or add different logging controls, such as **AUTHLOG** to control authorization information. See the section on **MCHANLOG** in the **mmdftailor(F)** manual page for a list of levels. “Changing MMDF parameters” later in this chapter also contains information about these logging levels. For more information about the MMDF logging files, see the **logs(F)** manual page.

### *Mailbox locking style*

By default, MMDF uses the standard System V **fcntl()** kernel file locking protocol to lock users’ mailboxes. However, if users on the system use MUAs that do not use the default locking protocol, you can configure the locking type with the **MLCKTYPE** keyword. The default **MLCKTYPE** line looks like the following:

```
MLCKTYPE advisory
```

See the section “Changing MMDF parameters” later in this chapter for more information on setting locking protocols.

## Alias files

An ALIAS is an abbreviated name that MMDF translates into a larger string (a mail address or list of addresses). Aliases are useful for specifying a single name to represent a group of users. You might want to create an alias called *sales* to represent all the members of the Sales department in the company. To do this, you define an alias file in the */usr/mmdf/mmdftailor* file and then create the alias file in */usr/mmdf/table*. In the alias file, the *sales* alias entry might look like the following:<sup>1</sup>

```
sales: joe, jane, bob, mike, karen, ann
```

When users want to send mail to everybody in the Sales department, they can use the *sales* alias like this:

### mail sales

This sends a copy of the message to everyone in the Sales department.

You can name your alias files anything you like; however, you must define the file names in the */usr/mmdf/mmdftailor* file (see the “The mmdftailor file” section earlier in this chapter). By default, MMDF provides the files listed in Table 2.4 that you can use to specify aliases for user names:

**Table 2-4 MMDF alias files**

File	Description
alias.list	aliases for lists of users
alias.user	aliases mapping users to their “home” host machines (the machines on which they receive their mail) and nicknames for local users
alias.ali	local machine system administrative aliases, programs, or files. At the minimum, this file should identify the aliases for <i>mmdf</i> and <i>postmaster</i> .

These files are located in the */usr/mmdf/table* directory.

1. The colon “:” following the alias name is optional.

**NOTE** You should **not** create aliases for remote users (users not located at the local site) in the alias files. In other words, do not create an alias like the following, where *moocow.uucp* is not a local machine at your site:

```
david: david@moocow.uucp
```

In this case, if a remote user sends mail to *david@npr.com*, the message is delivered to *david@moocow.uucp*.

In general, users should set private remote-user aliases using their MUA (for example, use the **alias** command in **\$HOME/.mailrc**). For more information, about creating private aliases, see the chapter on electronic mail in the *User's Guide*.

### *The alias.list file*

Use the *alias.list* file to create multiple-user aliases. With list aliases, MMDf processes the mail using the **list** channel and it appears as being from the sender instead of from the first person in the "To:" line.

To use the **list** channel, you must specify the following three lines for each alias:

- the name of the list
- the "*name-outbound*" line that contains the logins that comprise the list
- the "*name-request*" line that provides a login of the list maintainer who makes additions and deletions to and from the list

The following example shows how to use these three lines:

```
writers:                writers-outbound@list-processor
writers-outbound:       hanna,dianna,george,laurie,meg,naomi,steve
writers-request:        hanna
sales:                  sales-outbound@list-processor
sales-outbound:         joe,jane,mike,karen,ann,uksales
sales-request:          joe
```

The **list-processor** keyword is a reserved word in MMDf and indicates access to the **list** channel for processing mailing lists. The *sales-request* alias provides a way to request additions or deletions to the *sales* alias.

When defining an alias that contains many user names, you can use a backslash character "**\**" as a line-continuation character. Use quotation marks ("**"**") to delimit a string containing spaces or punctuation. When using an alias to define another alias, be careful not to create an alias loop.



You can also use the output redirection symbol ">" with the pipe character "|" to do more complex processing such as redirecting messages to files. For example:

```
Loguucp:      "network//usr/spool/log/uucp"
Logmlog:      "network|cat -v >>/usr/spool/log/mlog"
printer2:     "network|/usr/bin/lp -dprinter2"
```

In this example, MMDF pipes mail addressed to *Logmlog* to the **cat**(C) command to log the mail in the *mlog* file. MMDF pipes mail addressed to *printer2* to the **lp**(C) command for printing. These redirection alias examples use the user and group IDs of the user *network*. Although *network* is appropriate in most cases, you can specify any user named in the */etc/passwd* file on your system.

If you have a long list of names for an alias, you might want to include them in another file (instead of listing them directly in the *alias.list* file). To do this, use the **:include** keyword. For example:

```
staff:        staff-outbound@list-processor
staff-outbound: ":include:/etc/alias/staff"
staff-request: ross
```

In this example, the **:include:** line specifies that you want to use the names listed in the */etc/alias/staff* file to define the alias. Note that you cannot use the backslash character "\" as a line-continuation character for lists of names in an **:include** alias list.

For details on setting up mailing lists, see the **list**(ADM) manual page.

### *The alias.user file*

To map users to specific machines, specify aliases in the *alias.user* file. Use this file when you want each person at your site to receive mail on a particular machine. The following example shows how to set up *alias.user*:

```
andrei: andrei@scribe
george: george@dera
hanna: hanna@scribe
karen: karen@guardian
```

### *The alias.ali file*

Use the *alias.ali* file for aliases that are not specific to *alias.list* or *alias.user*. This file usually contains aliases related to system administration, such as:

```
root:      david
mmdf:      david
postmaster: david
uucp:      david
```

In this example, all mail addressed to *root*, *mmdf*, *postmaster*, and *uucp* is redirected to *david* on the local machine.

In general, do **not** create aliases for remote users (users not located at the local site) in the *alias.ali* file.

### *How MMDF uses alias files*

When mail is addressed to *postmaster*, MMDF routes the mail by first searching the hashed alias table from *alias.ali* to expand the *postmaster* alias to the associated user name. For example, the *postmaster* entry in the *alias.ali* file:

```
postmaster:  david
```

Then, MMDF searches the *alias.user* file to find the local machine name associated with the user name. For example, the *alias.user* file contains an entry like this:

```
david: david@golem
```

MMDF then uses this information when searching the various *.dom* files, which map the local machine name to a fully qualified host name and the *.chn* files, which map the fully qualified host name to information on how to route the message. The section “How MMDF routes mail” explains the process for searching configuration files further.

## *Domain files*

---

The domain files are used to match a host name to its fully-qualified host name. Domain files serve two purposes: to convey information to MMDF about how machines are connected, and to specify special routing considerations for subdomains or top-level domains.

Domain files are named for the domain that they describe (except for *root.dom*, which contains domain information not named in other domain files); the filenames end in *.dom*. The general practice is to have a separate domain file for the domain in which the host machine resides.

The operating system distribution includes four domain files in the `/usr/mmdf/table` directory. Table 2.5 lists these files.

**Table 2-5 MMDF domain files**

Domain	Domain file	Describes
local	local.dom	local machine
domain	domain.dom	machines in the local domain
uucp	uucp.dom	machines in UUCP domain
root	root.dom*	domains not listed in other domain files

You can create new domain files for each domain. For example, create a domain file for the *npr.COM* domain and name it *npr.dom*. You do not need to use the *\*.dom* naming scheme; however, this filename extension makes it easier to determine the purpose of the file.

### Domain file format

Each domain file consists of two columns of information: the left column lists the host name and the right column lists the fully qualified name for that host. The domain names can be either upper- or lowercase. Use tabs, spaces, a colon, or a combination of these characters to separate the first column from the second.

The name of a domain file determines the domain names for which MMDF searches. For instance, the domain file for UUCP generally contains entries for names in the following form:

```
machine:      machine.UUCP
```

However, you can create an entry to map a specific UUCP address to another address. For example, to map *research.UUCP* to *research.jcn.com*, the entry looks like the following:

```
research:      research.jcn.com
```

The following four sections give examples of different domain files.

### The local.dom file

The *local.dom* file describes the local host. For example, the contents of the *local.dom* file on the host *scribe* in the domain *npr.COM* look like the following:

```
scribe: scribe.npr.COM
```

\* The *root.dom* file also contains information about how to access top-level domains, such as *MIL* and *GOV*. (The name *root* implies “top-level”, as in a hierarchy).

### *The domain.dom file*

The *domain.dom* file describes the machines in the local domain (independent of the channel that MMDF uses to reach each machine). For example, if there are four machines in the domain *npr.COM*, the *domain.dom* file looks like the following:

```
scribe: scribe.npr.COM
huey:   huey.npr.COM
dewey:  dewey.npr.COM
louie:  louie.npr.COM
```

Note that the *domain.dom* file maps each machine to the fully qualified host name in the *npr.COM* domain.

### *The uucp.dom file*

The *uucp.dom* file specifies the hosts in the UUCP domain. For example, if your host connects to the remote machine *cactus* via UUCP, the *uucp.dom* file looks like the following:

```
cactus: cactus.UUCP
palm:   palm.UUCP
```

In this case, MMDF directs any mail sent to the *cactus* or *palm* machines to the UUCP network.

### *The root.dom file*

The *root.dom* file defines the hosts and domains not defined in the other domain files. For example, if your host connects to the UUnet network system, you can set up MMDF to send all mail to specific domains to *uunet.UU.NET*. To do this, set up your *root.dom* file like the following:

```
COM:    uunet.UU.NET
EDU:    ucsc.EDU
MIL:    uunet.UU.NET
GOV:    star.GOV
NET:    uunet.UU.NET
```

In this example, all mail directed to the *COM*, *MIL* is sent out on UUnet; mail to *GOV* goes to *star.GOV*; and mail sent to the *EDU* domain goes to *ucsc.EDU*.

### *LAN considerations*

If you are configuring MMDF for use on a local area network (LAN), you can use the domain files to distribute the processing load on the machines, or you can designate a special mail server machine to route all the messages.

You have the following choices:

- MMDF Server — Designate one machine as the network server. (This machine may also have outer world access.)

Each machine's domain file only needs to include **badhosts** and **badusers** channels that contain the fully qualified host name of the server. The domain file can be the same for each machine in the network. The network can grow, and machines can be added and removed with no effect on the domain files of other computers. The disadvantage is that the server receives a great deal of traffic and should be dedicated to its task. In addition, if the server is down, so is all electronic mail between machines. The best policy is a system of machines grouped around a server that has knowledge of another server; each server has knowledge of yet another and so on.

- Distributed Processing — Give each machine's domain file knowledge of each other machine in the network.

The advantage is that networked machines can operate independently of each other. One machine's crash has no effect on the mail capability of the others (unless that machine is connected to another network via UUCP). The disadvantages are that system administration gets geometrically more difficult as you add or remove machines to or from the network. When a machine or user is added to or removed from the network, you must update all domain or alias files to recognize the change. Because a domain file can contain redundant information about the local machine, you can use one domain file on every machine. You should only use distributed processing for small networks.

You can greatly simplify the distributed processing configuration by running a name server program on your network. For details on setting up the name server, see the chapter entitled "Name Server Operations Guide for BIND", in *Administering ODT-NET* in the *Administrator's Guide*.

- Gateways — Use a gateway to connect a network to another Local Area or Wide Area network. (This setup is derived from the server setup.) In this case, the server machine is typically connected to more than one network.

In addition to containing the names of the local machines on the LAN, a gateway machine also contains the names of the other machines to reach over the other networks. This information is kept in the respective domain and channel files for the other networks and also in the *root.dom* file on the gateway machine. Other machines on the LAN use the **badhosts** channel to route non-local mail to the gateway machine, or a *root.dom* file that lists all the top-level domains as routing through the gateway. To avoid overloading the gateway machine, the other machines on the LAN use local domain files as described in the earlier "Distributed Processing" bullet.

## Channel files

MMDF uses the channel files to determine the channel to use for outgoing mail and the address of the host on that channel. Channel files map the fully qualified host name (as determined from the domain file entries) to channel-specific addressing information. For example, the UUCP channel file maps host names to UUCP paths (using exclamation points) specifying how to get to each host.

The operating system distribution includes four channel files in the */usr/mmdf/table* directory. Table 2.6 lists these files.

**Table 2-6** MMDF channel files

Channel	Channel file	Describes
local	local.chn	local machine
list	list.chn	list-processor
smtp	smtp.chn	machines accessed via SMTP
uucp	uucp.chn	machines accessed via UUCP

The **MCHN** definitions in the */usr/mmdf/mmdftailor* file direct MMDF to search the specified *.chn* files in the */usr/mmdf/table* directory for channel definitions.

Separate the left and right columns in the channel files by a space or tab, a colon character “:”, or both.

### The *local.chn* file

The *local.chn* file contains entries describing all the names local host is called, mapping them to the local host name. For example, if the local host is *scribe.npr.COM*, then the local host is known as *scribe* on the local machine. The *local.chn* file maps the local host name (on the right) to the different ways people might refer to *scribe*. This file looks like the following:

```
scribe:      scribe
npr.COM:    scribe
scribe.npr.COM: scribe
```

### The *list.chn* file

The *list.chn* file contains information about the **list-processor** program:

```
list-processor: list-processor
list-proc:    list-processor
```

The left column is a pseudo-host defined in a mailing list alias (see “The alias.list file” section in this chapter). These entries tell MMDF to pass mail addressed to a mailing list to the **list-processor** program.

## The *uucp.chn* file

The *uucp.chn* file contains entries describing the hosts that your host connects to using the UUCP channel and how to route mail to those hosts. For example, the format of this file looks like the following:

```
mcvax.UUCP:      uunet!mcvax!%s
sri-nic.ARPA:    uunet!sri-nic.arpa!%s
uunet.uu.NET:    uunet!%s
```

The left column contains the UUCP host name from the domain tables; the right column describes the UUCP address that MMDF uses to direct mail to that host. The “%s” at the end of the UUCP address means to use the rest of the address from this point on. In other words, when mail is addressed to the user *hillis* at *mcvax.UUCP*, the UUCP channel passes the mail to UUnet along with the rest of the UUCP address (*mcvax!hillis*). The second entry in this example shows how a domain name (*sri-nic.ARPA*) can be used within a UUCP path.

Channel file entries for the UUCP channel (in the *uucp.chn* file) when the destination machine is multiple hops away, appear as follows:

```
stooges.UUCP:    moe!curly!larry!stooges!%s
```

Specify the *address* of the host on the right-hand-side, where the *address* is a UUCP path.

## The *smtp.chn* file

The *smtp.chn* file describes the hosts that you connect to using TCP/IP and the IP addresses of those hosts. The format of this file looks like the following:

```
cocoa.npr.COM:      123.456.789.1
caramel.npr.COM:    123.456.789.2
taffy.npr.COM:       123.456.789.3
```

The left column contains the fully-qualified host names for the hosts that you connect to with TCP/IP and the right column contains the IP addresses. For more information about configuring MMDF to route mail over TCP/IP, see your *TCP/IP Administrator's Guide*.

Following the pattern of mapping the host name in the left column to the addressing information for delivering to that host in the right column, you can create channel files for each MCHN definition in *mmdftailor*. The exception to this is the **badhosts** pseudo-channel definition; the **badhosts** channel program is determined at configuration time automatically. However, you can set up your channel files to indicate the channel that you use to reach the “smart host”.

## How MMDF routes mail

---

This section describes how MMDF uses the information in the configuration files to route mail on your system. Note that MMDF never searches the alias, channel, and domain files directly. When you build the hashed database with **dbmbuild**(ADM), the contents of these files are stored in tables in the *dbm* database. MMDF then uses the information in this database to route mail.

Mail arrives at and leaves computers using one of several different methods (such as UUCP, or TCP/IP called “channels”. The MMDF **submit**(ADM) command accepts the incoming mail from a channel and determines the correct outgoing channel to use based on the destination host. The **submit** program uses the information in the domain tables to map the way the incoming mail describes the destination host to the way the host recognizes that destination host. Based on the host description, **submit** uses the channel tables to determine the outgoing channel to use to route the message and places the message in the appropriate queue.

Then, using the information from */usr/mmdf/mmdftailor*, **deliver**(ADM) moves the mail from the queue to the appropriate channel. For example, if **submit** places a message in the UUCP channel queue, **deliver** moves that message to the UUCP channel. The **deliver** command can also place mail in a channel and let another program (such as **uux**(C) for the UUCP channel) carry out additional steps to resolve the circumstances dictated by the type of channel. The channel program sends the mail across the network to the proper destination.

## Searching MMDF domain tables

---

The **submit**(ADM) command uses the domain tables for two purposes: to specify the fully-qualified host name and (optionally) to specify the route to a host by listing the fully-qualified host names of one or more intermediate hosts through which mail is to be routed.

First, **submit** separates the fully-qualified host name into two parts: the name of the domain table and the hostname to match on the left-hand side (LHS) of the entries in the domain table. For example, in the address *david@engr.canada.COM*, the name of the domain table to search is *canada.COM* and the hostname to search for is *engr*.

MMDF tests an address for matches against the domain names in the **MDMN** entries in *mmdftailor*. For example, the address *david@engr.canada.COM* matches the following **MDMN** entry:

```
MDMN    "canada.com", show="Canada Delivery", tbl=canadadom
```



Then, MMDF searches *mmdftailor* for the **MTBL** entry for *canadadom*:

```
MTBL canadadom, file="canada.dom", show "Canada Delivery"
```

MMDF uses the file associated with *canadadom* (*canada.dom*).

Thus, to route our message to *david@engr.canada.COM*, **submit** uses the following algorithm to search the domain tables for a match in the address:

1. Searches for the hostname (*engr*) in the LHS of the domain table.
2. Searches in the LHS of the domain table (in this case, *COM*), if it exists, for the domain name (*engr.canada*).
3. Searches the LHS of the *root* domain table for the fully-qualified host name (*eng.canada.COM*).
4. Searches the relevant tables that include *flags=route*<sup>2</sup> in the **MTBL** line in *mmdftailor* for substrings of the address. For example, search the *COM* table for *canada*; search the *root* table for *canada.COM* or *COM*.
5. Searches the tables that include *flags=partial*<sup>3</sup> in the **MTBL** line in *mmdftailor* for the input name, in this case, the fully qualified host name (*eng.canada.COM*), regardless of the domain name.

If **submit** finds no match at all, as a last resort, it uses the **badhosts** channel, if it exists. Because MMDF uses the first domain that has an exact match without looking for other matches in later tables, the order in which you list **MDMN** definitions is significant. Make sure the local domain table appears first and the *root* domain table is last in *mmdftailor*.

If the address provided is *username@host* (for example, *david@engr*), MMDF performs the same search as described above except that there is no *canada.COM* to match in the **MDMN** entries in *mmdftailor*. In this case, MMDF searches the *root* domain table and then the tables that include *flags=partial* in the **MDMN** line.

- 
2. The *flags=route* parameter in the **MTBL** entry enables that table entry to match addresses for an entire subdomain, acting as a gateway. Do not use *flags=route* on tables other than the *root* domain table unless you have internal subdomains.
  3. Use the *flags=partial* parameter on the *local* domain table so that users do not have to specify the full domain to send mail on the local machine.

## *Delivery channel programs*

---

A channel is a compiled program that permits a machine to talk to a single type of network communications protocol. Some simple channels only store mail for further processing. Channel programs reside in the */usr/mmdf/chans* directory.

Channel programs handle the communications protocol so that neither the operating system nor the rest of the MMDF system has to know about the intricacies of a particular communication protocol. This handling of the protocol is one of the advantages of MMDF; having specific protocol program modules permits a site to upgrade to other network types without having to rewrite the mail delivery system.

The different channels are:

- badhosts** Called when a specified machine is unknown to the local machine. The use of “bad” is really a misnomer; any mail to an unknown machine is sent on this channel. You can assign **badhosts** in the *mmdftailor* file to a channel type such as **uucp** or **smtp**.
- badusers** Called when mail arrives at the local machine, but the user does not have a login on this machine, nor is there an alias for the user on the local machine. Generally, MMDF queues up this mail for submission to another machine with a larger list of users (a “smart host”).
- list** Called to remail messages. This channel simply invokes **submit** and feeds the addresses and text back into the MMDF mail system. This is often used to avoid long address validation or to force the validation to occur in the background for very large mailing lists. This also ensures that MMDF sends any problem reports to the list maintainer.
- local** Called to deliver mail to mailboxes and processes on the local machine.
- smtp** Called to deliver or accept mail from a TCP/IP network connection. The **smtp** channel transfers messages by establishing a TCP/IP connection to a remote machine, and using the Simple Mail Transfer Protocol (SMTP) to send one or more mail messages. The Internet Protocol (IP) allows many local- or wide-area networks to be interconnected transparently. This permits the MMDF SMTP channel to exchange messages with any machine on any network to which it is connected. For example, if your machine connects to the Internet, you can exchange messages directly with any machine in the world that is also connected to the Internet.

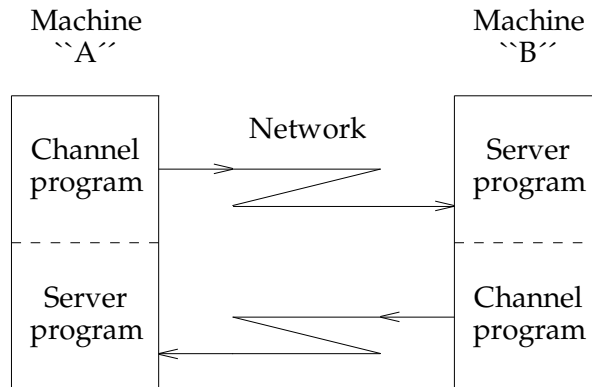
**uucp** Called to direct mail to UUCP delivery to another machine, or to accept mail from a UUCP connection from another machine. Incoming mail is converted into the format specified by the Internet technical bulletin, RFC822, available from the DDN Network Information Center, or NIC.\* Outgoing mail includes a "From<space>" line and the mail path arguments are separated by UUCP exclamation point characters "!".

For more information about UUCP, see the chapter entitled, "Building a remote network with UUCP", in *Administering ODT-NET* in the *Administrator's Guide*.

Channels act not only as protocol handlers, but in some cases actually initiate the communications to the network or to another machine as needed.

For each channel, MMDF provides two programs: an input program and an output program. A channel program is associated with outgoing mail; a server program is associated with incoming mail. For example, the UUCP channel has `/usr/bin/rmail` as its input program and `/usr/mmdf/chans/uucp` as its output program.

On the MMDF system, the server is a channel program that monitors the network for incoming mail. On systems other than MMDF, the function of a mail server is built into the mail delivery system. Figure 2-3 describes the relationship between a channel and server.



**Figure 2-3** Channel and server relationship

\* See the "Registering domain names" section earlier in this chapter for more information.

Even if MMDF is only being run on one side, the other side, while not correctly called a channel program, performs the same function.

## Modifying MMDF configuration

---

When you perform the initial MMDF configuration, the configuration utility sets up the appropriate *alias.\**, *\*.dom*, *\*.chn*, and *mmdftailor* files. This initial configuration is sufficient for most sites; if your site configuration changes, you can rerun the configuration utility to update the MMDF files.

However, if this initial configuration does not take care of your site configuration needs, you must edit the MMDF configuration files manually after running the configuration utility. You might also need to edit your configuration files manually if your site configuration changes so that you can no longer configure MMDF with the configuration utility. For more information, see the section “Configuration utility limitations” earlier in this chapter.

## Guidelines for manual configuration

---

Keep in mind the following guidelines when you modify your MMDF configuration files:

- Always perform configuration as *mmdf*.
- Always rebuild the hashed database with **dbmbuild**(ADM) after modifying files. See the section “Rebuilding the MMDF hashed database” for more information.
- If you make any changes to the channel configuration (for example, to add or remove a channel), restart the **deliver**(ADM) daemon.

## Changing the postmaster alias

---

Because RFC821/822 requires that every host provide the special postmaster mailbox, you should never change the **MSUPPORT** line in */usr/mmdf/mmdftailor* to anything other than *postmaster*. However, if the person responsible for mail administration at your site changes, you should change the *postmaster* alias in the */usr/mmdf/table/alias.ali* file.

For example, to change the *postmaster* alias to *hanna*, change the line in *alias.ali* to look like this:

```
postmaster:      hanna
```

**NOTE** If you do not define an alias (in the *alias.ali* file) that maps *postmaster* to a real user on the system, you must create a *postmaster* account (using the **sysadmsh**(ADM) command) to direct mail inquiries and problems.

You can designate anyone to receive undeliverable mail, but a local user is best because the address is simpler and, therefore, more likely to be a valid address. Also, with a local user, delivery of the message does not depend on a network connection that might be malfunctioning and therefore responsible for the original mail problem.

## Maintaining user aliases

As you add new users to your system, you should update your `/usr/mmdf/table/alias.user` file to map the new users to their home machines. For example, if you add *laurie* to the machine *poet* at your site, add the following line to *alias.user*:

```
laurie: laurie@poet
```

You can change the aliases in this file for other reasons, such as when you switch a user to another machine or the user leaves the company. For example, if a user leaves the company, you can redirect their mail to another person, such as their manager. To do this, place a line like the following in *alias.table*:

```
matt: mark@scribe
```

If you create multiple-user aliases (mailing lists), you might need to update the information in your `/usr/mmdf/table/alias.list` file periodically. For example, if you add a new alias called *docstyle*, add the following lines to *alias.list*:

```
docstyle: docstyle-outbound@list-processor
docstyle-outbound: joan,kelly,hanna,laurie,tammy,meg,teresa
docstyle-request: joan
```

## Changing the host name

When you install your operating system, the installation procedure prompts you to provide a name for your machine. This machine name is also called the HOST NAME or SYSTEM NAME.

If you decide to change your host name after installing the operating system, use the following procedure:

1. Log in as *root*.
2. Change the host name using the **uname(C)** utility. Use the following command, where *new\_name* is the new name of your host:

```
uname -S new_name
```

For example, to change a machine name to *elgrande*, enter the following command:

```
uname -S elgrande
```

For more information on the **uname** command, see the **uname(C)** manual page.

3. Now, log in as *mmdf* and verify that you are in the */usr/mmdf* directory. If not, change to the */usr/mmdf* directory with this command:

```
cd /usr/mmdf
```

4. Edit the *mmdftailor* file and change all occurrences of the old host name to *new\_name*.
5. Change to the */usr/mmdf/table* directory and edit any *\*.dom* and *\*.chn* files, changing all instances of the old host name to *new\_name*.
6. Now, verify that you are in the */usr/mmdf/table* directory and rebuild the hashed database using the following command:

```
./dbmbuild
```

7. Now, log in as *root*, and bring the system down to single-user (system maintenance) mode with the following command:

```
init 1
```

8. At the prompt, enter **<Ctrl>d** to bring the system back up to multiuser mode.
9. If you are routing mail over TCP/IP, verify that the */etc/hosts* file contains the new host name.

Now, MMDF uses the new host name.

## ***Hiding your host name***

---

If, after running the initial MMDF configuration with the configuration utility, you register your top-level domain or subdomain with the NIC (Network Information Center), you might decide to hide the host name behind the domain name. In this case, people outside your organization can send mail to people at your site without knowing the host name where they receive their mail.

To hide your host name, you must edit the */usr/mmdf/mmdftailor* file and change the **MLDOMAIN**, **MLNAME**, and **MLOCMACHINE** keywords. For example, if you are not currently hiding the host name, these lines look like the following:

```
MLDOMAIN      npr.COM
MLNAME        scribe
```

To change the configuration to hide *scribe* behind the *npr* domain name, add **MLOCMACHINE** and change these parameters so they look like this:

```
MLDOMAIN com
MLNAME npr
MLOCMACHINE scribe
```

Now, people can send mail to *andrei@scribe.npr.com* using the *andrei@npr.com* address (they no longer have to specify the host name, *scribe*).

## Changing your smart host

If your configuration changes so that you no longer rely on a smart host for complete information about the network, or your smart host changes, you must modify the **MCHN** lines for **badhosts** and **badusers** in *mmdftailor*. These lines look something like the following:

```
MCHN badhosts, show="Last-Chance Routing", pgm=smtp, tbl=smtpchn,
ap=822, host="rocky.npr.COM", confstr=scribe.npr.COM
MCHN badusers, show="Last-Chance Routing", pgm=smtp, tbl=smtpchn,
ap=822, host="moose.npr.COM", confstr=scribe.npr.COM
```

In this case, change the value of the **host** parameter (in this example, *rocky.npr.COM* and *moose.npr.COM*) in each **MCHN** definition to the name of the new smart host. For more information about smart hosts, see the "Hide your host name?" section earlier in this chapter.

## Changing error logging levels

By default, the MMDF configuration script sets logging levels with the **MMSGLOG** and **MCHANLOG** keywords in the *mmdftailor* file. These keywords are set to the **FAT** logging level (to log fatal errors only) and look like the following:

```
MMSGLOG level=FAT
MCHANLOG level=FAT
```

You can change these logging levels to a higher level, such as **FST**. In addition, you can use the **AUTHLOG** keyword to control authorization information. For a complete list of available logging levels, see the **MCHANLOG** section in *mmdftailor*(F).

## Converting from XENIX

If you are converting from a XENIX to UNIX system or you have a mixed UNIX/XENIX system network, you must convert your XENIX aliases file and your UUCP routing files (as appropriate). (If you install your UUCP configuration files before running the MMDF configuration utility, you only have to convert the alias files.) This section explains how to convert these files.

Before converting your XENIX files, you must install and configure UUCP. You must install and configure these packages so that the files that you use to create MMDF alias, channel, and domain files contain the correct information.

If you did not install the entire distribution, you should install the UUCP and MAIL packages using the **custom**(ADM) utility. See your *Installation Guide* for information on installing packages with **custom**. For information on configuring UUCP, see the “Building a remote network with UUCP” chapter in *Administering ODT-NET* in the *Administrator’s Guide*.

MMDF provides the utilities listed in Table 2.7 in the `/usr/mmdf/table/tools` directory for converting files from XENIX to MMDF.

**Table 2-7 MMDF Conversion utilities**

Utility	Converts this XENIX file:	To these MMDF files:
mmdfalias	<code>/usr/lib/mail/aliases</code>	<code>/usr/mmdf/table/alias.list</code> <code>/usr/mmdf/table/alias.user</code>
uulist	<code>/usr/lib/uucp/Systems</code>	<code>/usr/mmdf/table/uucp.chn</code> <code>/usr/mmdf/table/uucp.dom</code>

For more information, see the **mmdfalias**(ADM) and **uulist**(ADM) manual pages. Note that **uulist** uses **uuname**(C) to retrieve the list of accessible sites.

Use the following procedure to convert the XENIX files:

1. First, use the following command to make sure *aliases* and *top* are readable by *mmdf*:

```
ls -l /usr/lib/mail/aliases /usr/lib/mail/top
```

If the *mmdf* user does not have permission to read these files, use **chmod**(C) to change the permissions. Log in as *root* and enter the following command:

```
chmod +r /usr/lib/mail/aliases /usr/lib/mail/top
```

2. Now, log in as *mmdf* and enter the following command:

```
cd /usr/mmdf/table
```

3. Use the following commands to convert the routing or alias files, as appropriate:

- To convert alias files, enter:

```
tools/mmdfalias
```

The **mmdfalias** utility creates the files *alias.list* and *alias.user* in the `/usr/mmdf/table` directory.



- To convert UUCP routing files, enter:

**tools/uulist**

The **uulist** utility creates the files *uucp.chn* and *uucp.dom* in the */usr/mmdf/table* directory.

If any of the MMDF output files already exist, the MMDF utility backs up the original *filename* to *filename-*.

4. Now, make sure that the permissions on each of the output files are set to 644, and that the owner and group are *mmdf*. Use the following commands:

```
cd /usr/mmdf/table  
ls -l alias.list alias.user uucp.chn uucp.dom
```

The permissions should look something like the following:

```
-rw-r--r--  1 mmdf      mmdf          150 Jun 11 1990 alias.list  
-rw-r--r--  1 mmdf      mmdf          120 Jun 11 1990 alias.user  
-rw-r--r--  1 mmdf      mmdf           86 Apr 17 14:23 uucp.chn  
-rw-r--r--  1 mmdf      mmdf           61 Apr 17 14:23 uucp.dom
```

If the permissions for any of these files are wrong, use **chmod(C)** to modify them.

Now, all the XENIX routing and alias files are converted to MMDF.

If you are converting to MMDF from XENIX or another UNIX system, the system mailboxes for the users on the system are in the XENIX-style (older UNIX system) format. XENIX format uses the "From<space>" lines to delimit between messages in mail folders; MMDF uses <Ctrl>a characters. You should convert the system mailboxes to use the new MMDF format using the **cnvtmbox(ADM)** utility.

Use the following procedure to do this:

1. First, verify that users on the system are not accessing the system mailboxes.
2. Log in as *root*. (You must have read permission on the mailbox that you are converting.)
3. Now, change to the system mail spool directory (generally, */usr/spool/mail*).
4. For each system mailbox in the mail directory, run the following commands:

```
/usr/mmdf/bin/cnvtmbox old_mailbox new_mailbox  
mv new_mailbox old_mailbox
```

By default, the **mailx**(C) program maintains the current format of every mail folder that users read with **mailx**. For example, if a mail folder is in MMDF format (with <Ctrl>a delimiters), **mailx** uses this format when adding messages to the folder. If a user runs **mailx** with a folder that has not been converted to MMDF format, it prompts the user to convert the folder.

If any users on the system use an MUA other than **mailx**, advise these users to run **cnvtnbox** on each of their mail folders. As long as users have read permission on a mail folder and write permission on the output, they can convert it to the MMDF format with **cnvtnbox**.

## Modifying UUCP host configuration

---

If you change your UUCP host configuration, use the information in this section to update your MMDF configuration files.

### Specifying options to **uux**

If you use UUCP to route mail, you can specify the options that **uux**(C) should invoke when sending files and executing commands on the remote machine. By default, MMDF uses the following options to **uux**:

```
uux - -r
```

This setting specifies that UUCP queue the job until the next time **uucico**(ADM) runs rather than transfer it immediately (see **uucico**(ADM) for more information).

If you want to change this, add the **UUXSTR** keyword to the */usr/mmdf/mmdftailor* file. For example, if you want **uucico** to run immediately, the following **UUXSTR** setting is correct:

```
UUXSTR "uux -"
```

In general, we recommend that you use one of the following **UUXSTR** settings:

```
UUXSTR "uux -"
```

```
UUXSTR "uux - -r"
```

```
UUXSTR "uux - -r -gA"
```

For more information on these options, see the **uux**(C) manual page.

## Adding or removing a UUCP host

To add a host that you access via UUCP to your system, use the following procedure:

1. Add the UUCP path to route mail to the host to the `/usr/mmdf/table/uucp.chn` file. For example, to add the UUCP host *dudley*, add the following line to *uucp.chn* on the local host:

**dudley.UUCP: dudley!%s**

The `%s` specifies to use the remainder of the address indicated in the "To:" mail header from this point.

2. Add the UUCP host name to the *uucp.dom* file. For example:

**dudley: dudley.UUCP**

3. Now, rebuild the hashed database with **dbmbuild**.

To remove a UUCP host, remove the entries for that host from the *uucp.chn* and *uucp.dom* files and rebuild the hashed database.

## Adding or removing an SMTP host

Use the following procedure to add an SMTP host to your MMDF configuration:

1. For each new host in `/etc/hosts`, add the fully qualified host name and IP address to the `/usr/mmdf/table/smtp.chn` file. For example, to add two new hosts *chocolate* and *truffle*, enter:

**chocolate.npr.COM: 123.456.789.4**  
**truffle.npr.COM: 123.456.789.5**

2. Now, edit the `/usr/mmdf/table/smtp.dom` and match the unqualified SMTP host name to the fully qualified host name. For example, enter:

**chocolate: chocolate.npr.COM**  
**truffle: truffle.npr.COM**

3. Now, use **dbmbuild(ADM)** to rebuild the MMDF hashed database.

Use the information in the section "Testing MMDF configuration" later in this chapter to check the new SMTP configuration. For more information on routing mail over TCP/IP, see *Administering ODT-NET* in the *Administrator's Guide*.

## Running multiple deliver daemons

---

If you configure MMDF to route mail on channels in addition to the local channel, you must restart the **deliver** daemon and modify the `/etc/rc2.d/S86mmdf` system startup file to tell **deliver** about the new channels.<sup>4</sup> The **deliver** daemon periodically checks each channel's queue for mail to deliver. This is known as "sweeping the queues".

The "Modifying MMDF system startup" section explains how to start one **deliver** process for all the channels that you have configured. When you have a single **deliver** program managing a number of channels, **deliver** goes through the channels individually and tries to deliver all the messages in a channel's queue before going on to the next channel. You can configure your system to start multiple **deliver** programs, each servicing a single channel, instead. In this case, the **deliver** daemons work in parallel. This is a desirable configuration for a mail gateway machine because it increases the overall mail bandwidth of the machine.

**NOTE** For each **deliver** process that you initiate, system overhead increases.

To start separate **deliver** processes to manage each channel, use the following information to add **deliver** startup lines for each channel to the `/etc/rc2.d/S86mmdf` file.

If you removed the **-clocal** from the **deliver** startup line in your `S86mmdf` file, the **deliver** startup line looks like the following:

```
/bin/su mmdf -c "/usr/mmdf/bin/deliver -b"
```

To create separate **deliver** processes, use the **-c** option to add lines for each channel. The following example shows separate **deliver** processes for each of the local, UUCP, and SMTP channels:

```
/bin/su mmdf -c "/usr/mmdf/bin/deliver -b -clocal"  
/bin/su mmdf -c "/usr/mmdf/bin/deliver -b -cuucp"  
/bin/su mmdf -c "/usr/mmdf/bin/deliver -b -csmtp"
```

---

4. See the section "Modifying MMDF system startup" earlier in this chapter for information on how to do this.

If you have a lot of mail traffic on a single channel, you can start more than one **deliver** process running on that channel. To do this, simply add more than one **deliver** startup line for the channel in */etc/rc2.d/S86mmdf*. For example, to start three **deliver** processes to sweep the queue for the SMTP channel, add these lines:

```
/bin/su mmdf -c "/usr/mmdf/bin/deliver -b -csmt"
/bin/su mmdf -c "/usr/mmdf/bin/deliver -b -csmt"
/bin/su mmdf -c "/usr/mmdf/bin/deliver -b -csmt"
```

Keep in mind that each **deliver** process that you start increases the load on the system.

### *Specifying different time intervals for deliver*

The **MSLEEP** parameter in the *mmdftailor* file determines the default time interval for the **deliver** daemon to sweep through the queues for all the channels. The default setting for **MSLEEP** is 600, meaning that **deliver** sweeps the queues every 600 seconds, or 10 minutes. We recommend that you set up **deliver** to run every 60 seconds. To use this interval for all the channels, change the value of **MSLEEP** in *mmdftailor* to 60.

If you create a separate **deliver** daemon for each channel, you can set up a different time interval for each **deliver** process. To do this, add the **-T** option to the appropriate **deliver** startup line in */etc/rc2.d/S86mmdf*. For example, to run **deliver** on the local channel every 30 seconds and on the UUCP channel every 5 minutes, put the following lines in *S86mmdf*:

```
/bin/su mmdf -c "/usr/mmdf/bin/deliver -b -clocal -T30"
/bin/su mmdf -c "/usr/mmdf/bin/deliver -b -cuucp -T300"
```

2

## *Rebuilding the MMDF hashed database*

The hashed database gives MMDF quick access to alias and routing information. You must rebuild this database whenever you change *mmdftailor* or any of the alias, channel or domain tables.

To rebuild the database, log in as *mmdf* and run the following commands:

```
cd /usr/mmdf/table
./dbmbuild
```

If any of the files in */usr/mmdf/table* are missing when you rebuild the database, **dbmbuild** reports that they are missing. See the **dbmbuild**(ADM) manual page for full details.

## Testing MMDF configuration

---

Once you've modified your MMDF configuration and rebuilt the hashed database, use the information in this section to test the new configuration.

### Checking for MMDF problems

---

Use the **checkup**(ADM) command to examine the full MMDF system on the local machine and report any inconsistencies. Normally, **checkup** reports all problems, including correct states; **checkup** marks problems with two asterisk characters **\*\*** and encloses advisory information in square brackets. You can tell **checkup** to report only the problems using the **-p** option. For example, enter the following command:

```
/usr/mmdf/bin/checkup | more
```

The following example shows the output from **checkup**:

```
** Asterisks indicate potentially serious anomalies.

Tailor file           : /usr/mmdf/mmdftailor
**   Unknown tailor   : 'MLCKTYPE advisor'
```

(In this case, **checkup** does not recognize the **MLCKTYPE** parameter **advisor** because it is misspelled (the correct parameter name is **advisory**).

If the permissions on any of the MMDF configuration files are incorrect, **checkup** lists this information.

### Testing mail addresses

---

Use the **checkaddr**(ADM) command to test an individual address for validity. For example, to test the validity of the address, *andrei@scribe*, enter the following command:

```
/usr/mmdf/bin/checkaddr andrei@scribe
```

The **checkaddr** checks the address and displays the following message:

```
andrei@scribe: OK
```

This command is particularly useful for testing aliases after you have installed a new alias file. For more information, see the **checkaddr**(ADM) manual page.

## Maintaining the MMDF system

---

Once you configure MMDF, the MMDF system requires minimal system administration. In addition to maintaining the alias files (as described in the “Maintaining user aliases” section earlier in this chapter) and adding new machines to your network (see the “Modifying MMDF configuration” section), you must perform the following tasks:

- Monitor space in the `/usr/spool/mail` directory where mail is received, and in the `/usr/spool/mmdf` directory where queued mail is stored.
- Clean outdated mail from the mail queues.
- Remove log files periodically from the `/usr/mmdf/log` directory.

The following sections describe the utilities you can use to maintain your MMDF system.

### Checking the status of mail queues

---

Use the **checkqueue**(ADM) program to check the status of the mail queues; this utility reports on the number of messages waiting for delivery. For example, enter the following command:

```
/usr/mmdf/bin/checkqueue
```

Here is an example of output from **checkqueue**:

```
Thu May  9 16:07:  2 queued msgs / 80 byte queue directory
                  4 Kbytes in msg dir

      2 msgs      4 Kb (local  ) local      : Local Delivery
                                deliver start : Thu May  9 15:15
                                deliver message : Thu May  9 15:15
                                deliver end   : Thu May  9 15:15 / 0 hours
      *** WAITING ** First message      : Wed Apr 24 16:03
```

If a queue is backed up with waiting mail, you can manually force **deliver**(ADM) to deliver the mail using a command like the following:

```
deliver -w -clist,uucp
```

The **-c** option specifies the channels to be processed and **-w** forces **deliver** and the channel programs to output informative messages as they try to deliver the mail. You can review the output for abnormalities, such as a rejected sender or recipient. For a complete description of **checkqueue**, see the **checkqueue**(ADM) manual page.

---

## Removing old mail from the queues

---

Use the **cleanque**(ADM) program to remove outdated files from the mail queues. Use **crontab**(C) as *mmdf* to run **cleanque** daily. You might want to run **cleanque** more often, depending on your mail volume. You can also run **cleanque** using **cron**(C). To do this, create a *crontabs* file for *mmdf* in the */usr/spool/cron/crontabs* directory that includes the following line:

```
0 0 * * 0-6 /usr/mmdf/bin/cleanque
```

You can also run **cleanque** by hand whenever you suspect a problem with mail delivery. The **cleanque** manual page provides a complete description of this program.

---

## Monitoring the size of log files

---

You should periodically check the size of log files in */usr/mmdf/log*. To limit the amount of log data that accumulates in a log file, move it to another filename, for example *chan.log-*.

If you move aside a log file, make sure that *mmdf* owns the new empty file. To back up the log file and create a new empty file, enter the following commands as *mmdf*:

```
cd /usr/mmdf/log
cp chan.log chan.log-
cat /dev/null > chan.log
```

---

## Advanced MMDF configuration tasks

---

This section describes some additional configuration tasks that you can use to customize your MMDF system. In general, most sites do not need to perform any of the configuration procedures described in this section.

---

## Changing the location of system mailboxes

---

By default, MMDF delivers mail to a file named with the user's login in the */usr/spool/mail* directory. If you want MMDF to deliver mail to a file or directory other than the default file, you can add lines (anywhere in the *mmdftailor* file) using the following **MDLVRDIR**, **MMBXNAME**, and **MMBXPROT** keywords.



Only one of **MDLVRDIR** and **MMBXNAME** can be non-null. If **MDLVRDIR** is null, MMDF delivers to the user's home directory; if **MMBXNAME** is null, MMDF uses the user's login as the name of the mailbox file. You can also change the location of the spool directory by setting **MDLVRDIR** to be non-null.

**MMBXPROT** sets the protection mode on mailbox files using the same set of octal numbers that you use to change access permissions on files with **chmod(C)**. For example:

```
MDLVRDIR ""
MMBXNAME ".mailbox"
MMBXPROT 0600
```

In this example, MMDF delivers to a *.mailbox* file in the user's home directory and sets the file protection so that only the owner can read or write to the file.

## *Specifying the MMDF "signature"*

---

You can change the signature that MMDF uses when notifying senders of mail delivery problems by setting the **MSIG** keyword in the */usr/mmdf/mmdftailor* file. This message should indicate which mail routing system was responsible. For example:

```
MSIG      "MMDF Mail System"
```

With this setting, when users receive "Failed" mail messages from MMDF, the message appears to be from "MMDF Mail System".

## *Configuring MMDF authorization*

---

The MMDF authorization feature allows you to control the flow of mail through your host. For example, if your site is on the Internet (you registered your domain name with Government Systems), you might want to allow your employees (but no one else) to send mail from their home machines through your system to the Internet. In this case, you would want to set up MMDF to authorize mail from those users to pass through your machine, but prevent mail from other users from passing through.

With MMDF, you can control mail transferred through your host on a per-channel or per-user basis (or both). In other words, you can control the flow of mail on specific channels and from specific users.

To control authorization on a particular host, use the **auth** parameter to set an authorization level on the **MCHN** line in the `/usr/mmdf/mmdftailor` file. MMDF provides the seven levels of authorization listed in Table 2.8.

**Table 2-8** MMDF authorization levels

Level	Description
free	performs no authorization checks (default)
inlog	logs incoming authorized and unauthorized access, but allows mail to pass
outlog	logs outgoing authorized and unauthorized access, but allows mail to pass
inwarn	logs incoming unauthorized access, transfers the message, and sends a warning to the originator
outwarn	logs outgoing unauthorized access, transfers the message, and sends a warning to the originator
inblock	logs the incoming unauthorized access attempt and bounces the mail
outblock	logs the outgoing unauthorized access attempt and bounces the mail

The default channel authorization level is **free**; any channel that you do not set authorization for is set to **free**.

The following sections explain more about using these authorization levels to set up host- and user-based authorization.

### *Setting authorization for hosts*

You can control the flow of mail both to and from your host and network by setting up host-based authorization controls. The most common reason that sites use host-based authorization is to make sure that hosts which are not part of a private network do not use that network to send mail to other parts of the world. For example, you can use host-based authorization to allow mail to enter your domain, but restrict the mail passing through it.

By default, MMDF performs all authorization on the next “hop” host in the address; if someone specifies a route through your host, MMDF on your host can authorize the next hop in that route. In other words, MMDF does not perform authorization based on the final destination unless the final destination is the next hop. However, you can perform authorization on the entire route to the destination or from the source using the **dho** flag with the **auth** parameter to specify routing-based authorization controls. See the “Routing-based authorization” section later in this chapter for more information.

You can control access to a particular channel based on whether the mail is inbound or outbound on the channel and whether the source or the destination of the mail is authorized. To do this, use one (or more) of the four tables listed in Table 2.9.

**Table 2-9 Host-based authorization tables**

Table	Description
insrc	when mail arrives from a particular host (the source host), this table verifies that that host is allowed to send mail to the destination host by authorizing either the destination host or the channels used to access the destination host
outsrc	when mail arrives from the source host via a particular channel, this table determines if that source host or channel is allowed to send mail to the destination host
indest	when mail is sent to a particular host or via a particular channel, this table determines whether the source host is allowed to send mail to the destination host or via that channel
outdest	when mail is sent to a particular host, this table verifies that the source host or channel is allowed to send mail to the destination host

For example, you might want to set up MMDF so that only hosts that belong to a certain network can send mail out on a specific channel. This could be the case if your site has limited resources and you do not want to allow other sites to pass mail through your site.

Use the following procedure to set up the host-based authorization controls:

1. First, log in as *mmdf*. (You must perform all MMDF configuration as *mmdf*.)
2. Use the **MTBL** keyword to define a table in */usr/mmdf/mmdftailor*. For example:

**MTBL "world-auth", file="authinfo/world"**

The *authinfo/world* table specifies the privileges for all the hosts that do not belong to the network. This table is stored in the file *world* in the directory */usr/mmdf/table/authinfo*; you can specify locations like this for all the tables.

3. Now, use **MCHN** to define two channels in *mmdftailor*: one channel for the hosts that belong to the network, one for the hosts that do not belong.

For example, define a channel for the *sconet* network:

```
MCHN sconet, auth=free, show="SCONET Delivery",
      ap=822, mod=imm
```

(The “auth=free” authorization setting is the default; you do not have to specify it explicitly as in this example.)

Now, define a channel for all the hosts (**world**) not in the *sconet* network (this appears as one line in *mmdftailor*):

```
MCHN world, auth=inblock, indest="world-auth",
      show="WORLD Delivery", ap=822, mod=imm
```

In this case, anyone can send mail out on the **world** channel, but MMDF checks the *world-auth* table to authorize the destination of mail arriving on this channel.

4. Now, create the channel tables (\*.chn files in the */usr/mmdf/table* directory) for the channels that you define in *mmdftailor* and include descriptions of each host accessed via that channel. (See the “Channel files” section in this chapter for more information.)
5. Finally, create the *world-auth* table and include lines like the following:

```
world:
local:
sconet: moocow.uucp
```

The entries on the LHS of the authorization table specify that if the destination for a message is on either the **world** or **local** channels, MMDF authorizes anyone using the **world** channel as an input channel to send mail.

The entry on the RHS of the **sconet** channel entry specifies the hosts and channels that are authorized to send outgoing mail using the **sconet** channel. In this case, *moocow.uucp* is the only machine allowed to pass mail into the *sconet* network.

6. Now, rebuild the hashed database using the information in the section “Rebuilding the MMDF hashed database”.

### *Routing-based authorization*

You might want to set up authorization to keep mail that enters your domain from leaving your domain. In other words, people can send mail from inside the domain to people outside the domain as well as to other people in the domain, but people outside the domain cannot send mail through your domain to another destination.

For example, if your domain includes machines in different cities with links to different outside computers, people from outside the domain might use these links to send mail from one city to another, through your domain. In this case, you can set up MMDF authorization to prevent people from using your system to transfer mail.

Because MMDF performs authorization only on the next “hop” in the address, you cannot use the procedure described in the previous section to set up authorization control. To do this, you must use the **auth** parameter with the **dho** flag to specify routing-based authorization controls.

The following procedure explains how to set up routing-based authorization for the hosts that are not in the *nprnet* domain:

1. Log in as *mmdf*.
2. Define a table in */usr/mmdf/mmdftailor*. For example:

**MTBL "world-auth", file="authinfo/world"**

This defines the *authinfo/world* file that contains the authorization information for the **world** channel.

3. Now, specify a channel for your domain. For example, if the domain name is *npr.com*, create an **MCHN** entry like the following:

**MCHN nprnet, auth=free, show="NPRNET Delivery",  
ap=822, mod=imm**

4. Now, define a channel for the rest of the hosts that are not in the local domain (again, this appears as one line in *mmdftailor*):

**MCHN world, auth=inblock, auth=dho, indest="world-auth",  
show="WORLD Delivery", ap=822, mod=imm**

The “auth=indest” parameter specifies that when **world** is the input channel, MMDF checks the *authinfo/world* file to verify that the inbound host is authorized to send mail to the destination.

When you specify the “auth=dho” parameter on a channel, MMDF replaces the “host” (in host-based authorization) used to check authorization with a route. The route is either from the source or to the destination, depending on which flag (from Table 2.9) that you specify. MMDF replaces the local section of the route (the user’s name) with the string “username”. Then, MMDF compares this route to the route specified in the message, to determine if the message is authorized or not.

5. Create the channel tables in the */usr/mmdf/table* directory for the channels that you define in *mmdftailor*. (See the “Channel files” section in this chapter for more information.)

6. Finally, create the *authinfo/world*, and include entries like the following:

```
world:
username@npr.com:
username@larry.npr.com:
username@moe.npr.com:
username@curly.npr.com:
```

This table authorizes MMDF to deliver any mail addressed to people in the *npr.com* domain arriving or leaving on the **world** channel. This does not allow mail to pass through the **nprnet** channel to a destination outside the *npr.com* domain.

7. Now, rebuild the hashed database with **dbmbuild**.

### Setting authorization for users

To configure MMDF authorization on a per-user basis, you must first set up the authorization level on the channels that you want to restrict as described in the previous section. Once you set up channel authorization levels, you must create a table in the */usr/mmdf/table* directory that maps user names to authorization levels and then declare this table in the */usr/mmdf/mmdftailor* file.

The following is the format for setting up the per-user authorization table:

```
username: keyword channel [ , channel]
```

The *username* can be either a local or remote user name, *keyword* is one of the keywords for the actions that you can authorize users to perform, and *channel* is the channel name from the **MCHN** line.

Table 2.10 lists the action authorization keywords:

**Table 2-10** MMDF authorization keywords

Keyword	Description
both	allows user to send and receive mail
send	allows user to send mail only
recv	allows user to receive mail only
expire	expires access privileges for the user (and includes this information in any error mail)

MMDF treats any action not listed in the table above as **expire**, except that MMDF sends the text of the action instead of “expire” to the user in error mail.

Use these steps to set up authorizations for specific users:

1. First, log in as *mmdf*.

2. Use the information in the section “Setting authorization for hosts” to set up authorization for any channels that you want to restrict.
3. Declare the user authorization table in the `/usr/mmdf/mmdftailor` file. For example, if the name of the user authorization table is `auth.user`, the **MTBL** declaration looks like this:

```
MTBL auth, file="auth.user", show "Per-user authorization"
```

Note that you must call the per-user authorization table “auth”; MMDF treats any table called “auth” as the per-user authorization table.

4. Now, create the user authorization table in the `/usr/mmdf/table` directory. Edit the file and include one line for each user to whom you want to grant mail access. Any users not listed in the user authorization table are not authorized to use any channel except the ones set to the **free**, **inlog**, or **outlog** authorization levels.

In our example, MMDF allows users who are not listed in the user authorization table to pass mail in and out through the UUCP channel and in through the TCP/IP channel.

5. To set up access authorizations for a local user, specify the unqualified user name. In the following example, the local user *andrei* can both send and receive mail on the TCP/IP and UUCP channels:

```
andrei: both smtp,uucp
```

However, if you set up host-based authorization to restrict access to one of these channels, for example UUCP, *andrei* might not be authorized to send or receive mail on that channel.

Because mail on the local channel is not restricted, *andrei* can pass mail through this channel even though the user authorization list does not include “local” in the list of channels.

6. To set up access authorization for a remote user, specify the fully-qualified address of that user. For example, to allow *natasha* on the machine *kgb.GOV* to send mail through TCP/IP on this host, add a line like the following to the authorization table:

```
natasha@kgb.GOV: send smtp
```

If mail arrives for *natasha* through UUCP, or if she tries to send mail through the UUCP channel, MMDF rejects the mail.

If you want to expire a particular user’s access and tell MMDF to send an error message, add the message to the user authorization table line for that user. For example, to expire *rocky@squirrel.COM*’s access and send the text “No more mail for you!”, include a line like this one:

```
rocky@squirrel.COM: "No more mail for you!" uucp
```

7. Rebuild the hashed database with **dbmbuild**.

## Setting host and user authorization controls

The default authorization algorithm allows MMDF to deliver a message if either the user-based or host-based authorization tables allow it. Thus, if a user table authorizes a user to send mail over a channel, but the host table does not, that user can still use the channel (and vice versa).

If you want to require that both the user and host tables authorize access to a channel, use the **hau** keyword with the **auth** parameter in the **MCHN** channel declaration in *mmdftailor*.

## Understanding authorization log files

If you set the **level** parameter to the **AUTHLOG** keyword to an authorization level of **FST** (to give full statistics) or higher<sup>5</sup>, MMDF maintains a complete log of authorization attempts and reasons for failure or success.

For example, to set the authorization logging level to save all diagnostic messages to the */usr/mmdf/log/mmdfauth.log* file, set **AUTHLOG** in the *mmdftailor* file like this:

```
AUTHLOG /usr/mmdf/log/mmdfauth.log level=FTR
```

The format of authorization messages in the log file is similar to other MMDF log files. Each message includes the date, time, message source, and message ID, followed by the log-specific information. For example:<sup>6</sup>

```
4/29 9:44:54 AU-0000: msg.a000561: i='local' o='ucsc'
a='lisa@rsre.AC.UK' r='CH' hi='' ho='username@rsre.ac.uk'
4/29 9:44:54 AU-0000: msg.a000561: i='local' o='ucsc'
a='jane@rsre.AC.UK' r='CH' hi='' ho='username@rsre.ac.uk'
4/29 9:44:55 AU-0000: msg.a000561: END size='2102', sender='robert'
```

5. See the section on **MCHANLOG** in the *mmdftailor*(F) manual page for a complete list of logging levels.

6. The authorization messages appear on one line; the examples in this section split the lines for readability.



Table 2.11 describes the authorization message keys.

**Table 2-11 Authorization keys**

Key	Description
i	input channel
o	output channel
a	destination address
r	reason for authorization
hi	inbound host
ho	outbound host

In addition, the authorization message includes an “end of processing” message that describes the message sender and size. (In the example above, the third line is the “end of processing” message.)

Each authorization message can have either one or two reasons for authorizing a particular message. If the authorization line includes one reason, the “r” key specifies a single authorization code which describes both the inbound and the outbound authorization when you use host-based authorization. These codes are listed Table 2.12.

**Table 2-12 Single-reason authorization codes**

Code	Description
OH	outbound host/route
HC	outbound host/route and inbound channel
HH	inbound host/route and outbound host/route
CC	inbound channel and outbound channel
CH	inbound channel and outbound host/route
IH	inbound host/route

The following example uses the **CH** authorization code:

```
4/29 9:44:54 AU-0000: msg.a000561: i='local' o='peaks'
a='bob@rsre.AC.UK' r='CH' hi='' ho='username@rsre.ac.uk'
4/29 9:44:54 AU-0000: msg.a000561: i='local' o='peaks'
a='mike@rsre.AC.UK' r='CH' hi='' ho='username@rsre.ac.uk'
4/29 9:44:55 AU-0000: msg.a000561: END size='2102', sender='cooper'
```

In this example, the authorized message has two recipients (*bob* and *mike*). The first authorization message shows that the inbound channel (“i”) is the local channel and the outbound channel is **peaks**. The “a” key indicates that the recipient’s address is *bob@rsre.AC.UK*.

The reason (“r”) given for authorizing the message is **CH**; in other words, the inbound channel (**local**) has authorization to send mail to the given outbound host or route (specified by “ho”), in this case *username@rsre.ac.uk*.

The two-reason authorization codes describe the reason for authorization in terms of user-based authorization. Table 2.13 lists these codes.

**Table 2-13 Two-reason authorization codes**

Code	Description
IL	inbound channel, outbound = <b>LIST</b>
OL	outbound channel, dest = <b>LIST</b>
IS	inbound channel by sender
OS	outbound channel by sender
IR	inbound channel by receiver
OR	outbound channel by receiver
I	inbound channel, log unauthorized access*
O	outbound channel, log unauthorized access*

The authorization message in the following example uses two-reason authorization (if no authorization is required for a channel, MMDf leaves the reason field (“r”) empty):

```
4/29 9:53:09 AU-0000: msg.a000653: i='local' o='npr'
a='john@edxa.ac.uk' r='' r='OS'
4/29 9:53:10 AU-0000: msg.a000653: END size='197', sender='david'
```

In this example, the message arrived (with no authorization required) on the local channel and is authorized to leave on the **npr** channel because the sender (*david*) is authorized to use it (**OS**).

## Adding a new alias file

It is much easier to add entries to an existing alias file than to create one; see the sections “Alias files” and “Maintaining user aliases” earlier in this chapter for more information. If you want to add a new alias file, use the following procedure:

1. Log in as *mmdf*.
2. Create a file in the */usr/mmdf/table* directory.

\* MMDf only uses these authorization codes when you set “auth=inlog” or “auth=outlog”.

3. Verify that the new alias file is owned and group-owned by *mmdf* and set the permissions to 644.

**NOTE** Do not give write permissions on alias files to other users. With write permissions to alias files, a user can forward another user's mail to another location. For this reason, you should designate a trusted user to make changes to the alias files as requested.

4. Add alias information to the new alias file. See the "Alias files" section earlier in this chapter for information on the format of the various alias files.
5. Now, define the alias file in the */usr/mmdf/mmdftailor* file. Create an **MTBL** line to associate an abbreviated name and a more descriptive name with the new alias file. For example, the **MTBL** line in *mmdftailor* defines the alias file *alias.user* as:

```
MTBL auser, file="alias.user", show="User Aliases"
```

Now, you can refer to the file */usr/mmdf/table/alias.user* as "auser" throughout the rest of *mmdftailor*.

6. Add an **ALIAS** entry to *mmdftailor* to define information about the new alias file. For example, here is the **ALIAS** entry for *alias.user*:

```
ALIAS table=auser
```

The default parameters in the *mmdftailor* file are adequate for most requirements; however you can specify characteristics for the alias file using the following parameters:

**nobypass** specifies that MMDF search the alias files first for all mail addresses on outgoing or incoming mail. In MMDF, placing a tilde "~" before an address causes the address to be interpreted as destined for its literal meaning. The **nobypass** parameter means that normal aliasing cannot be bypassed. For example, suppose the alias file contains this entry for George and **nobypass** is not set:

```
george: george@vegan.edu
```

If you enter:

```
mail george
```

then the mail goes to the *george* login on the *astoria* machine. If you then enter:

```
mail "~george"
```

then aliasing is disabled and mail is sent to a George on the local machine. If **nobypass** is set, entering the tilde character "~" has no effect and every mail request is translated by the alias files.

- |                |  |
|----------------|--|
| <b>public</b>  | allows remote sites to expand aliases from this table using the SMTP <b>EXPN</b> command; you can also use <b>malias(C)</b> to display this information.   |
| <b>trusted</b> | directs mail to be delivered to any file or process using the permissions of any user on the system (including <i>root</i> ); only the super user should have access to modify a trusted alias file. |

Note that MMDf searches the alias tables in the order that you list them in *mmdftailor*.

For more information, see the **mmdftailor(F)** manual page.

## *Adding a new domain file*

---

Use the following procedure to create a new domain file:

1. Log in as *mmdf*.
2. Create a file in the */usr/mmdf/table* directory. Generally, you should name domain files for the domain that they describe. For example, the domain file for the *npr.COM* domain is called *npr.dom*.
3. Verify that the new domain file has the correct permissions (see step 3 in the previous section).
4. Add the information for the domain to the new domain file. See the "Domain file format" section earlier in this chapter for more information.

**NOTE** The entries on the left-hand side in this file must contain no leading whitespace.

5. Create an **MTBL** entry in *mmdftailor* for the new domain file. For example:

```
MTBL nprdom, file="npr.dom", show="NPR Domain"
```

See the **mmdftailor(F)** manual page for more information about additional parameters that you can use to specify additional properties about the table that you are defining with **MTBL**.

6. Add an **MDMN** entry for the new file:

```
MDMN "npr.COM", show="NPR Domain", table=nprdom
```

The first argument is the name of the domain.

## Adding a third-party channel program

If you add a new channel program from an SCO add-on package, you must define and add a new channel file. A channel is the mechanism for delivering mail either to a mailbox on the local machine or across the network to a remote machine. The **MCHN** entries in *mmdftailor* define the channels available to MMDF for mail transport.

You can define additional channels for the network protocols configured on your system by specifying them in the *mmdftailor* file. Channel definitions look like this:

```
MCHN    uucp, show="UUCP Delivery", que=uucp,
        tbl=uuchn, ap=822, pgm=uucp, mod=imm
```

The first argument on the **MCHN** line is the name of the channel.

Use these steps to add a new channel:

1. Create an **MCHN** entry in */usr/mmdf/mmdftailor* and specify the parameters to use in your channel definition. For a complete list of the **MCHN** parameters, see the **mmdftailor(F)** manual page.
2. Create a channel file (\*.chn) in */usr/mmdf/table* and include entries for each host that you access via the new channel and how to route mail to those hosts. For more information, see the section “Channel files” earlier in this chapter.

## Changing MMDF parameters

The MMDF parameters in the */usr/mmdf/mmdftailor* file allow you to redefine the certain MMDF variables. This section covers some of the more common MMDF parameters; for a complete list, see the **mmdftailor(F)** manual page.

**AUTHLOG** controls the authorization information. The following example shows the format of this parameter:

```
AUTHLOG /tmp/mmdf/mmdfauth.log, level=FST, size=40, stat=some
```

The **AUTHLOG** level must be at least FST, or MMDF does not save any authorization logging information. For more information, see the section on **MCHANLOG** in the **mmdftailor(F)** manual page.

**MFAILTIME** is the time (in hours) a message can remain in a queue before MMDF sends a failed mail message to the sender and purges the message from the queue.

```
MFAILTIME    168
```

**MLCKTYPE** allows you to specify the locking protocol for MMDF to use when locking users' mailboxes. This is useful if the users on the system use third-party MUA's that use a lock file that is different from the standard System V lock file. Set the **MLCKTYPE** parameter to one or more of the keywords in Table 2.14.

**Table 2-14** MMDF locking keywords

Keyword	Lock file
advisory	System V <code>fcntl()</code> kernel locking protocols
v7	Version 7 and System V Release 3, and earlier locking protocols
xenix	XENIX locking protocols
all	all above locking protocols (default)

The default locking protocol is **all**; however the MMDF configuration utility sets **MLCKTYPE** to **advisory**.

With the **v7** keyword, MMDF creates a file called *username.lock* in */usr/spool/mail*. The **xenix** keyword specifies that MMDF create a file called */tmp/username.mlk*. In both cases, *username* is the name of the user's mailbox.

You can specify more than one locking protocol on the **MLCKTYPE** line. If you do, MMDF must successfully lock the user's mailbox using all the locking protocols before the mailbox is considered locked. For example, to use **advisory** and **xenix**, set **MLCKTYPE** like the following:

```
MLCKTYPE advisory, xenix
```

In this case, MMDF must lock the mailbox, using the `fcntl()` kernel file locking protocol *and* create a file called */usr/spool/mail/username.lock*. If it fails to perform both of these locks, MMDF must release the successful lock and try again later. Thus, if MMDF failed to perform the `fcntl` lock, it must unlink */usr/spool/mail/username.lock*.

**MMAXHOPS** specifies the maximum number of "Received:" or "Via:" lines a message can contain before the MMDF considers that the message is looping and rejects it.

```
MMAXHOPS          20
```

**MWARNTIME** specifies the time (in hours) that a message can remain in a queue before MMDF sends a warning message about delayed delivery to the sender.

```
MWARNTIME         72
```

**MMSGLOG** controls the logging information produced by the **deliver**(ADM) and **submit**(ADM) programs. The following example shows the format of this parameter:

```
MMSGLOG /tmp/mmdf/mmdfmsg.log, level=FST, size=40, stat=some
```

For more information, see the **logs**(F) manual page and the section on **MCHANLOG** in the **mmdftailor**(F) manual page.

**MCHANLOG** controls MMDF logging, except for information controlled by **AUTHLOG** and **MMSGLOG**. See the **mmdftailor**(F) manual page for details.

**MSLEEP** determines the length of time (in seconds) that the **deliver** daemon sleeps between scanning the queues. By default, **MSLEEP** is set to 600 (10 minutes). We recommend that you set **MSLEEP** to 60 seconds:

```
MSLEEP      60
```

## *Getting more information*

---

If you need more information to understand how the various components of MMDF interact, refer to the manual pages listed in Table 2.15.

**Table 2-15** MMDf manual pages

Manual Page	Description
<b>checkaddr</b> (ADM)	checks a mail address for validity
<b>checkque</b> (ADM)	reports on MMDf queue status
<b>cleanup</b> (ADM)	checks MMDf system configuration
<b>cleanque</b> (ADM)	sends warnings and returns expired mail
<b>cnvtmbox</b> (ADM)	converts XENIX-style mailboxes to MMDf format
<b>dbmbuild</b> (ADM)	builds the MMDf database
<b>dbmedit</b> (ADM)	edits the MMDf database
<b>deliver</b> (ADM)	delivers mail to MMDf channels
<b>list</b> (ADM)	handles mailing lists
<b>mmdf</b> (ADM)	routes mail locally or over network
<b>mmdfalias</b> (ADM)	converts XENIX-style aliases file to MMDf
<b>rmail</b> (ADM)	submits remote mail received via MMDf
<b>submit</b> (ADM)	accepts mail to be handled by MMDf
<b>uucico</b> (ADM)	transfers UUCP files
<b>uulist</b> (ADM)	converts UUCP file to MMDf format
<b>checkmail</b> (C)	checks for submitted but not delivered mail
<b>mail</b> (C)	mails messages between users
<b>uucp</b> (C)	sends information between UNIX machines over serial lines
<b>rcvalert</b> (C)	alerts user when mail is received
<b>rcvfile</b> (C)	stores mail in a file automatically
<b>rcvprint</b> (C)	prints received mail automatically
<b>rcvtrip</b> (C)	informs sender that user is away
<b>uname</b> (C)	prints the name of the current system
<b>uux</b> (C)	transfers UUCP files and executes commands on remote system
<b>logs</b> (F)	logs MMDf messages
<b>maildelivery</b> (F)	describes the user delivery specification file
<b>mmdftailor</b> (F)	provides run-time tailoring
<b>queue</b> (F)	queues files for storing mail in transit
<b>systems</b> (F)	UUCP Systems file format
<b>tables</b> (F)	describes alias, domain, and host tables
<b>llog</b> (S)	performs standardized information logging for programs
<b>ml_send</b> (S)	provides simple interface for mail submission
<b>mmdf</b> (S)	simplifies MMDf MUA mail submission and pickup
<b>phs</b> (S)	notes the MMDf transmission phase
<b>tai</b> (S)	gets site tailoring information



*Getting more information*

2