ANU News Setup and Site Maintenance Manual

January, 1995

This manual is a guide to the initial setup of ANU News for OpenVMS Systems.

Compiled By:

Charles Bailey (bailey@genetics.upenn.edu)

With contributions from:

The ANU News user community

Revision/Update Information: ANU News V6.1Beta10

Revision/Update Information: 23-Jan-1995

Contents

Preface xiii

Chapter 1 What is News Management?

1.1 What is News Management? 1–1
1.1.1 News Site Management 1–1
1.1.2 Newsgroup Moderator 1–2
1.1.3 Newsgroup Supervisor 1–2

Chapter 2 Finding and Following ANU News

2.1 Getting a Copy of ANU News 2–1
2.2 Unpacking the News Distribution 2–2
2.3 News Development and Support 2–3

Chapter 3 Basic considerations for setting up News at your site

3.1 Software required for News 3–1
3.2 Deciding which newsgroups to carry 3–3
3.3 Choosing a News database configuration 3–6
3.3.1 News reader/server using local database 3–7
3.3.2 Diskless NNTP client 3–7
3.3.3 Mixed client/local 3–7

Chapter 4 Setting Up News As An NNTP Client Only

4.1 Generating the required files 4–1
4.2 Setting up the News logical name environment 4–3
4.2.1 News_Address 4–3
4.2.2 News_GMT_Offset 4–3

4.2.3 News_HelpLib (optional) 4–3
4.2.4 News_Locked_Command 4–3
4.2.5 News_Locked_Wait_Minutes 4–4
4.2.6 News_Manager 4–4
4.2.7 News_Mgr_ID 4–4
4.2.8 News_NetPost (optional) 4–4
4.2.9 News_NNTP_Protocol 4–4
4.2.10 News_NNTP_Server 4–5
4.2.11 News_Node 4–5
4.2.12 News_NoNetPost (optional) 4–6
4.2.13 News_Organisation (or News_Organization) (optional) 4–6
4.2.14 News_Pathname (optional) 4–6
4.2.15 News_Post_CC (optional) 4–7
4.2.16 News_*remhost*_Task (optional) 4–7
4.2.17 News_Stop (optional) 4–7
4.2.18 News_Timezone 4–7
4.2.19 News_Usage_Anonymous (optional) 4–8
4.2.20 News_Usage_Log (optional) 4–8
4.3 Setting up optional configuration files 4–8
4.3.1 Editing the News.Aliases file 4–9
4.3.2 Creating the News.Classes file 4–9
4.3.3 Editing the News.Distributions file 4–10
4.3.4 Editing the News_Address.Cnf file 4–11
4.3.5 Editing the News_Post.CC file 4–12
4.3.6 Editing the News_Post.Defaults file 4–13
4.4 Installing News.Exe 4–14
4.5 Setting up a mail contact address for News matters 4–14
4.6 Providing a foreign command to run News 4–15
4.7 Moving News.Hlb 4–15
4.8 Optional steps to assist users 4–15
4.8.1 Documentation 4–15
4.8.2 System help library entry 4–16
4.8.3 NewsRC.Template file 4–16

Chapter 5 Setting Up News as a Local Reader/Server

5.0.1 Setting up disk space for News use 5–1
5.0.2 News images and support files 5–1
5.0.3 Index files for the local news database 5–2
5.0.4 Item files 5–2

5.0.5 Spool files . . . 5–3
5.0.5.1 NNTP over TCP/IP or DECnet . . . 5–3
5.0.5.2 Other methods of transport . . . 5–3
5.0.6 Scratch files . . . 5–4
5.1 Generating the required files . . . 5–4
5.1.1 Building local copies of News images . . . 5–4
5.1.2 Creating the News_Manager directory . . . 5–5
5.1.3 Installing News.Exe . . . 5–5
5.1.4 Moving News.Hlb . . . 5–6
5.2 Setting up Configuration Files . . . 5–6
5.2.1 Editing the MailPaths. file . . . 5–6
5.2.2 Editing the News.Aliases file (optional) . . . 5–7
5.2.3 Creating the News.Classes file (optional) . . . 5–8
5.2.4 Ignoring the News.Distribution file . . . 5–8
5.2.5 Editing the News.Distributions file . . . 5–9
5.2.6 Editing the News.Sys file . . . 5–10
5.2.7 Ignoring the News.SysD file . . . 5–13
5.2.8 Editing the News_Address.Cnf file . . . 5–14
5.2.9 Editing the News_Post.CC file (optional) . . . 5–16
5.2.10 Editing the News_Post.Defaults file (optional) . . . 5–16
5.2.11 Creating the NNTP message ID cache file (optional) . . . 5–17
5.2.12 Editing the NNTP_Access.News file (optional, but recommended) . . . 5–18
5.3 Creating the News manager account and identifier . . . 5–19
5.4 Setting up the News logical name environment . . . 5–20
5.4.1 News_Address . . . 5–20
5.4.2 News_Cache (optional) . . . 5–20
5.4.3 News_Device . . . 5–20
5.4.4 News_GMT_Offset . . . 5–21
5.4.5 News_GroupFile_BufferCount (optional) . . . 5–21
5.4.6 News_GroupFile_DFW (optional) . . . 5–21
5.4.7 News_GroupFile_FastDelete (optional) . . . 5–22
5.4.8 News_GroupFile_Global_BufferCount (optional) . . . 5–22
5.4.9 News_GroupFile_RTV (optional) . . . 5–22
5.4.10 News_HelpLib (optional) . . . 5–22
5.4.11 News_HistoryFile_BufferCount (optional) . . . 5–23
5.4.12 News_HistoryFile_DFW (optional) . . . 5–23
5.4.13 News_HistoryFile_FastDelete (optional) . . . 5–23
5.4.14 .hl 3 News_HistoryFile_RTV (optional) . . . 5–23
5.4.15 News_ItemFile_BufferCount (optional) . . . 5–23
5.4.16 News_ItemFile_DFW (optional) . . . 5–23
5.4.17 News_ItemFile_FastDelete (optional) . . . 5–23
5.4.18 News_ItemFile_Global_BufferCount (optional) . . . 5–24

5.4.19 News_ItemFile_RTV (optional) 5–24
5.4.20 News_Locked_Command 5–24
5.4.21 News_Locked_Wait_Minutes 5–24
5.4.22 News_Manager 5–24
5.4.23 News_Manager_Dev 5–24
5.4.24 News_Mgr_ID 5–25
5.4.25 News_NetPost (optional) 5–25
5.4.26 News_NNTP_Log (optional) 5–25
5.4.27 News_NNTP_Protocol (optional) 5–25
5.4.28 News_NNTP_Server (optional) 5–26
5.4.29 .hl 3 News_NNTP_Server_Always_Spool (optional) 5–26
5.4.30 News_NNTP_Server_Batch_Flush_Disable (optional) 5–26
5.4.31 News_NNTP_Server_Batch_Size (optional) 5–27
5.4.32 News_NNTP_Server_Leave_Spool_Open (optional) 5–27
5.4.33 News_Node 5–27
5.4.34 News_NoNetPost (optional) 5–28
5.4.35 .hl 3 News_Organisation (or News_Organization) (optional) 5–28
5.4.36 News_Pathname (optional) 5–28
5.4.37 News_Post_CC (optional) 5–28
5.4.38 News_Root 5–28
5.4.39 News_*remhost*_Task (optional) 5–29
5.4.40 News_Stop (optional) 5–29
5.4.41 News_Timezone 5–29
5.4.42 News_Usage_Anonymous (optional) 5–30
5.4.43 News_Usage_Log (optional) 5–30
5.4.44 NNTP_Disk_Danger and NNTP_Disk (optional) 5–30
5.4.45 NNTP_More_To_Offer 5–30
5.4.46 NNTP_Scratch (optional) 5–30
5.5 Setting up required mail addresses 5–31
5.5.1 Administrative contact address for News matters 5–31
5.5.2 Addresses for incoming news feeds 5–31
5.6 Providing a foreign command to run News 5–31
5.7 Running News for the first time 5–32
5.8 Creating newsgroups on your system 5–32
5.9 Setting up the NNTP server 5–34
5.9.1 NNTP server running over DECnet 5–34
5.9.1.1 Setting up the single-threaded DECnet server 5–35
5.9.1.2 Setting up the multithreaded DECnet server 5–36
5.9.2 Setting up the CMU-IP NNTP Server 5–36
5.9.2.1 Setting up the single-threaded CMU-IP server 5–36
5.9.2.2 Setting up the multithreaded CMU-IP NNTP server 5–37
5.9.3 Setting up the Multinet TCP NNTP server 5–38

5.9.4 Setting up the TCPware TCP NNTP server 5–38
5.9.5 Setting up the UCX TCP NNTP server 5–39
5.9.5.1 Setting up the single-threaded UCX NNTP server 5–39
5.9.5.2 Setting up the multithreaded UCX NNTP server 5–40
5.9.6 Setting up the Wollongong TCP NNTP server 5–41
5.10 Setting up the maintenance jobs 5–41
5.10.1 Adding incoming items to the local News database 5–42
5.10.1.1 Batches received by your NNTP server 5–42
5.10.1.2 Batches received by direct file transfer 5–43
5.10.1.3 Items received via mail 5–43
5.10.2 Transferring enqueued items to downstream sites 5–45
5.10.3 Removing expired items from the News database 5–46
5.10.4 Obtaining exclusive access to the News database 5–47
5.11 Arranging to exchange news with other sites 5–48
5.11.1 Obtaining the information necessary to set up an exchange 5–49
5.11.2 Checklist for integrating a new upstream site into News processing 5–51
5.11.2.1 Set up appropriate network connections 5–51
5.11.2.2 Set up the spool area for incoming files 5–51
5.11.2.3 Include the new items in your News add processing 5–51
5.11.2.4 Test the incoming feed 5–51
5.11.3 Checklist for integrating a new downstream site into News processing 5–52
5.11.3.1 Add the News.Sys entry for the new site 5–52
5.11.3.2 Set up appropriate network connections 5–52
5.11.3.3 Set up transfer of items to the downstream site 5–52
5.11.3.4 Test the outgoing feed 5–53

Chapter 6 Setting up News in Mixed Local/Served Configuration

Chapter 7 News Support Utilities

7.1 Exchange of Items via NNTP Over DECnet or TCP 7–1
7.1.1 NNTP_Xfer 7–1
7.1.1.1 Installation 7–2
7.1.1.2 Command Syntax 7–2
7.1.1.3 Transfer Session Summary 7–5
7.1.1.4 Compilation constants 7–6
7.1.1.5 Examples 7–6

7.1.2 NNTP_Xmit 7–7
7.1.2.1 Installation 7–7
7.1.2.2 Command Syntax 7–8
7.1.2.3 Transfer Session Summary 7–9
7.1.2.4 Compilation constants 7–10
7.1.2.5 Examples 7–10
7.2 Testing NNTP connections over DECnet or TCP/IP networks 7–11
7.2.1 NNTP_Client 7–11
7.2.1.1 Installation 7–12
7.2.1.2 Command syntax 7–12
7.2.1.3 Session Summary 7–12
7.2.1.4 Examples 7–12
7.2.2 NNTP_TTY 7–13
7.2.2.1 Installation 7–13
7.2.2.2 Command syntax 7–13
7.2.2.3 Session summary 7–14
7.2.2.4 Example 7–14
7.3 Copying News index files 7–14
7.3.1 NewGrpFile 7–15
7.3.1.1 Installation 7–15
7.3.1.2 Command syntax 7–16
7.3.1.3 Session summary 7–16
7.3.1.4 Example 7–16
7.3.2 NewItmFile 7–17
7.3.2.1 Installation 7–17
7.3.2.2 Command syntax 7–17
7.3.2.3 Session summary 7–17
7.3.2.4 Example 7–18
7.3.3 Rebuild_Item_File 7–18
7.3.3.1 Installation 7–19
7.3.3.2 Command Syntax 7–19
7.3.3.3 Session Summary 7–19
7.3.3.4 Example 7–20
7.4 Miscellaneous maintenance utilities 7–20
7.4.1 BatchNews 7–21
7.4.1.1 Installation 7–21
7.4.1.2 Command syntax 7–21
7.4.1.3 Session Summary 7–22
7.4.1.4 Examples 7–22

7.4.2 Cachem 7–23
7.4.2.1 Installation 7–23
7.4.2.2 Command syntax 7–23
7.4.2.3 Session summary 7–24
7.4.2.4 Examples 7–25
7.4.3 FeedCheck 7–26
7.4.3.1 Installation 7–26
7.4.3.2 Command Syntax 7–26
7.4.3.3 Session Summary 7–27
7.4.3.4 Examples 7–28
7.4.4 NewsShutDown 7–31
7.4.4.1 Installation 7–31
7.4.4.2 Command Syntax 7–31
7.4.4.3 Session Summary 7–32
7.4.4.4 Examples 7–32

Appendix A Reporting Problems with ANU News

Appendix B RMS Tuning of the News Index Files

Examples

4–1 Sample News.Classes File 4–10
4–2 Sample Make_NewsRC_Tempate.Com Procedure 4–16
7–1 Using Cachem to Obtain Cache Usage Information 7–25
7–2 Using Cachem to find a Particular Message ID 7–25

Preface

Layout of this Document

This manual was written as part of the revised documentation to be released with version 6.1 of ANU News. Because it contains information which may be useful immediately, it has been included in earlier releases as an independent document. It may refer occasionally to other portions of the News documentation (e.g. detailed descriptions of News commands) not included in this release. In this case, you should use the corresponding sections of the old News documentation as a guide.

Document Conventions

Within this document, the following conventions are used:

- File specifications and logical names are written in mixed case to make them easier to read and remember (e.g. News_Root and News.Exe instead of NEWS_ROOT and NEWS.EXE). This is done only in the documentation; in actual practice, News follows the usual VMS conventions regarding case. For example, in order to define the logical name News_NNTP_Log, you could say

  ```
  $ Define/System/Exec News_NetPost News_Post_OK
  or
  $ define/system/exec news_netpost news_post_ok
  or
  $ DEFINE/SYSTEM/EXEC NEWS_NETPOST NEWS_POST_OK
  ```

  When the case of a string used as a command or configuration constant is significant, this is noted in the text.

- **Boldface** type is used to denote text which is displayed by News, or which should be entered by the user as shown.

- *Italic* type is used to denote expressions for which the user should substitute values appropriate to specific circumstances, as described in the text. (Note: This may appear as underlined text on terminal screens and some printers.)
- Within descriptions of commands and configuration files, items enclosed in square brackets ([ ]) are optional, and may be omitted if unneeded. This is not the case, of course, where square brackets are used to indicate the directory portion of a file specification.

Getting Help and Sugesstions for Improvement

If you run into any difficulties setting up or using ANU News, the best source for advice is the newsgroup news.software.anu-news, or the associated BITNET mailing list ANU_News@UBVM.BitNet. (There is a bidirectional gateway between the list and the newsgroup, so you need only report problems by one of these routes.)

If you have any suggestions for improvement of this document, please contact Dave Harrold (Harrold@MSOE.EDU).

Chapter 1
What is News Management?

1.1 What is News Management?

As you sit down to read this manual, you may be wondering just what is involved in managing news traffic. The common tasks can be divided into three areas: news site management (often called "news management"), newsgroup moderation, and newsgroup supervision. The first two of these categories can be applied to any site carrying network news, while the third is specific to ANU News.

This manual covers ANU News installation and configuration, as well as ongoing management of a site running ANU News, all of which fall under the first category listed above ("news management"). However, before we forge ahead with the details, we'll provide a description of each category. For more information on the function of newsgroup moderators and newsgroup supervisors, see the documentation for the News commands used by these individuals to exercise their privileges.

1.1.1 News Site Management

This term covers all of the tasks associated with setting up and maintaining news software on your system, getting your site hooked into the news distribution system, and serving as a contact person for your site in matters pertaining to news. If you plan to run only as an NNTP client, this involves just a little work - at a minimum, you need only build News.Exe and News.Hlb. If you're planning to set up a local database and exchange news with other sites, there's more configuration and site preparation to be done.

In general, the news manager at a site running ANU News will:

- Maintain the News software - this may involve building News from distributed source files, getting updates or patches as necessary, dealing with users' problems with News, and following discussion on the newsgroup news.software.anu-news to keep up with changes in News.

- Manage the site's configuration - this includes maintaining the configuration files used by News, creating and modifying newsgroups, and setting up the disk space News uses to store news items.
- manage news traffic - this covers making arrangements with neighboring sites to exchange news, monitoring traffic and correcting any problems which may develop, and keeping track of resources used by News, to insure that it doesn't become overloaded, run out of space, etc.

This list may seem daunting at first, but it's actually not all that hard, expecially after you've finished setting up, and that's what this manual's for. Once you're up and running, News is pretty stable, so keeping an eye on routine operation doesn't require much effort.

1.1.2 Newsgroup Moderator

In order to keep the volume of off-topic traffic in a newsgroup to a minimum, postings to that group may be subject to the approval of a person who has been designated the newsgroup moderator. All items posted by anyone else into that newsgroup are mailed to the moderator, instead of being added to the newsgroup immediately. The moderator then decides whether the item is appropriate, and if so, adds it to the newsgroup with an Approved: header indicating that sites should accept it as an item in that newsgroup. If an item is posted to multiple newsgroups, it is not actually added to any of them until it has been approved by the moderators of all moderated newsgroups to which it has been posted. In addition, a newsgroup moderator may cancel any posting to that newsgroup, regardless of where it originated. There is only one moderator for a given newsgroup, who handles traffic originating from any site. In practical terms, this means that you don't have to worry about moderated newsgroups unless the moderator lives at your site.

ANU News understands this model of newsgroup moderation, and will handle moderated newsgroups correctly. If a newsgroup moderator is located at your site, he or she will use the Add File/Moderator command to add approved postings to the newsgroup. In addition, he or she will have all the privileges of newsgroup supervisor (see below) with respect to that group at your site. If the moderator is located at another site, News takes care of checking for approval, and mailing new postings to the moderator, automatically.

1.1.3 Newsgroup Supervisor

In addition to the moderated newsgroups, ANU News also supports restricted newsgroups, which allow you to control not only who can post to a newsgroup, but who can read it as well. Unlike moderated newsgroups, the designation of a newsgroup as restricted is local to your site; all other sites will treat it as a normal (moderated or unmoderated) newsgroup.

An individual may be granted limited privileges to control newsgroup access via the Add Member and Delete Member commands. This person is called the newsgroup supervisor, to distinguish their role from that of newsgroup moderator. In addition to controlling access, the supervisor has a limited ability to cancel messages: he or she can remove any message from the newsgroup which originated at your site. Supervisor privileges are turned on and off using the Set Supervisor command.

Chapter 2
Finding and Following ANU News

2.1 Getting a Copy of ANU News

If you're reading this document, you've probably got a copy of the News distribution already, or at least you know where to get it. If not, and you have ftp access, then you can pick up a copy of the latest release by anonymous ftp to one of the following sites:

North America

- kuhub.cc.ukans.edu - This is the official source site for ANU News at present, and is maintained by Bob Sloane (sloane@kuhub.cc.ukans.edu), who also coordinates News releases. The News distribution files are located in the directory [.ANU_V*version*], where *version* reflects the version of News available in that directory. For example, v6.1beta10 is located in [.ANU_V61b10]. UnZIP.Exe is located in this directory as well.
- genetics.upenn.edu - This site mirrors the regular release, and also contains binaries compiled with Wollongong or UCX TCP support (the binaries in the official release are compiled with Multinet TCP support), object files compiled with CMU-IP support, and a few extras. These files are all found in the directory [.ANU_News.Distribution]. UnZIP.Exe, along with the source code for ZIP and UnZIP, can be found in [.Util.VMS.Archive]. This site also houses the News patch archive (see below), in the directory [.ANU_News.Patches]. The contact person for questions regarding this site is Charles Bailey (bailey@genetics.upenn.edu).
- MRCserv - The mail server at Mount Royal College in Canada maintains a copy of the News distribution for public access via email. Since the News distribution is fairly large, you should try to use ftp in favor of this mail server if at all possible. For detailed information about retrieving the files, send a mail message to mrcserv@mtroyal.ab.ca with the word HELP as the body of the message. The contact for other questions is postmaster@mtroyal.ab.ca.

Europe

- ftp.arnes.si - The News distribution is located in the directory /software/VMS/anu-news, as is a copy of UnZIP.Exe. This site is maintained by the Academic and Research Network of Slovenia. Questions or problem reports may be directed to Mark Martinec (Mark.Martinec@ijs.si) or Benjamin Zwittnig (Benjamin.Zwittnig@arnes.si).

News is also periodically included on various DECUS tapes, so you may already have a copy sitting around unbeknownst to you.

The official News distribution consists of three files:

News_*version*_Src.Zip - Source code for all News images and the help library, as well as a few handy DCL procedures and some source patches used at a lot of sites. These files can be compiled without modification on a VAX system using DEC's VAXC or DECC or GNU's gcc, or on an AXP system using DECC.
News_*version*_Dist.Zip - VAX binaries built from the files in the Src ZIP file (with Multinet TCP support), as well as templates for most configuration files, and sample DCL procedures for many common tasks.
News_*version*_Doc.Zip - Documentation for News in various formats, notes on News development, and handy RFCs.

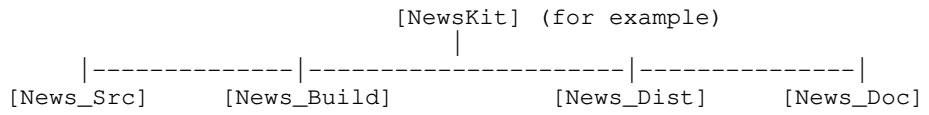
As of 23-Jan-1994, the current release of News is 6.1beta10, and the string *version* for this release is "V61-Beta10" (e.g. News_V61-Beta10_Src.Zip).

These files were packed with ZIP for VMS, and should be retrieved in binary mode. ZIP is a file archiving and compression tool which packages groups of files into archives somewhat like VMS Backup, though without as much error correction, and with better compression. The News distribution is packaged using a version of ZIP for VMS which preserves the RMS attributes of each file. In order to properly unpack these ZIP files, you should use UnZIP for VMS, version 5.0 or later. ZIP and UnZIP are free, and are available at many VMS ftp sites. In particular, you can get UnZIP.Exe from all of the ftp sites which carry the News distribution; it will either be in the same directory as the News ZIP files, or its location will be noted in a ReadMe file in that directory.

2.2 Unpacking the News Distribution

Once you've retrieved the necessary files, you should unpack them using UnZIP for VMS v5.0 or later, in order to preserve the VMS attributes of the files in each archive. In order for NewsBuild.Com to function properly, you should create a top-level directory for the News kit, and unpack the Src ZIP file into a subdirectory of this root named [.News_Src]. Similarly, the contents of the Dist ZIP file should be unpacked into a subdirectory named [.News_Dist], and the contents of the Doc ZIP file into a subdirectory named [.News_Doc].

Finally, you should create an empty subdirectory named [.News_Build], which will contain the object files created when building News. When you're done, then, you should have a directory tree which looks like this:



At this point, you should read the section of this manual which describes setting up News for the type of service you will use at your site, and follow the instructions found there.

2.3 News Development and Support

All development and maintenance of News is done by volunteers, and bug reports, questions, suggestions, and improvements are always welcome (especially when accompanied by code :-)). The clearinghouse for News support is the newsgroup news.software.anu-news, to which you should probably subscribe if you're in charge of maintaining News at your site. There is also a bidirectional gateway between this newsgroup and the BitNet mailing list ANU_NEWS, available at a LISTSERV near you. This newsgroup is fairly informal, and the regular inhabitants are quite helpful, so you shouldn't hesitate to post questions or problems (after checking the docs and the FAQ for the newsgroup, of course). You can improve your chances of a successful response by including a precise description of the problem, and appropriate details of your configuration (version of News, type of NNTP support, relevant logical name definitions or configuration files, etc.).

In addition to questions and discussion, source code patches containing bugfixes or improvements are also posted here regularly, so it's a good idea to follow this group even when things are running well at your site. Traffic is pretty low (<10 messages per day), so this shouldn't be a real hardship.

Ongoing development and releases are coordinated by Bob Sloane (sloane@kuhub.cc.ukans.edu(, who has given generously of his time and machines to fill in when Geoff Huston had to drop this role. If you're thinking about making any major revisions to News, it's probably a good idea to check with him first, to make sure someone else isn't already working on the same thing. Apart from this informal coordination, however, there is no structure to News development - anyone is welcome to post their suggestions and code to the newsgroup, where sites which so desire can pick it up and use it. Often, if it does good things and doesn't complicate News management unduly or have any nasty side effects, it will be incorporated into the next release as well.

In order to provide access to source code patches, an archive of all patches posted to the newsgroup since the first alpha release of version 6.1 is maintained at the site genetics.upenn.edu, where it is available via anonymous ftp. For more information on what patches are available, and how to use them, see the files $ReadMe.Txt and $Patches.List in the [.ANU_News.Patches] directory at that site. If you need patches and don't have access to ftp, send mail to bailey@genetics.upenn.edu, and I'll try to get you the patches you need.

One last note - News continues to exist because of the willingness of people to contribute their time and code to keeping it viable. It's a very nice package as it stands, but there's always a need for progress. If you've made any improvements to News, please share them with the rest of us; chances are there's someone out there (maybe lots of us) who can benefit from your work.

Also, when you run up against a thorny problem, please remember that all News support is done in the (often not so) spare time of people on the net, so the response to your postings may not be as fast as you might get for a commercial product. Nonetheless, most questions are answered within a day or so - not bad for an all-volunteer effort.

Chapter 3

Basic considerations for setting up News at your site

Before you can begin the process of setting up News, you will need to make several fundamental decisions about the way in which you want to make news available to your users. In making these decisions, you should consider the needs of your users, the amount of time you or someone else can devote to News management, the options open to you for exchanging news with other sites, and the system resources available for News maintenance. As is the case with most software, the more control you wish to exercise over news availability at your site, the more effort you will need to put into News configuration and maintenance. However, once News is up and running, maintenance usually requires only minor effort. This section describes briefly some of the issues and options you should consider when choosing a News configuration for your site.

Note for DECUS UUCP users

If you plan to integrate News with DECUS UUCP, you should also read the section on News setup in their installation manual, as many of the tasks described in this manual have been integrated into the UUCP setup procedures.

3.1 Software required for News

In addition to the News distribution, you need only basic VMS components to run the News images on your system. As News has been developed, there has been an effort to minimize the dependence of News on other software not included with VMS. However, some other tools may be necessary to maintain News on your system. In most cases, these tools are a common part of a well-run system, and you may want to obtain them, if you don't already have them, for their general utility.

If you wish to build the News images from source code rather than using distributed binaries, you will also need a C compiler. This is strongly recommended, not only because it's in general a bad practice to run programs on your system when you haven't been able to build copies you trust, but also because it's necessary for such things as site-specific modifications or incorporation of patches into News. (Actually, recompiling is necessary in any case if you

don't have TGV's Multinet TCP/IP stack, because the version of News.Exe distributed in the News_Dist Zip file is linked against the Multinet shareable socket library.) The C source code in News is written to be compiled by VAXC, DECC, or gcc without modification.

Note to gcc users

By default, NewsBuild.Com disables the '-fbuiltin' optimization, because of a bug in gcc versions before 2.6.3 which could cause corruption of memory when memcpy( ) was called. If you're using gcc 2.6.3 or later, you can remove the '-fno-builtin' from the /Cc1_Options value. Gcc is only available for VAX systems as of this writing, and can be obtained by anonymous ftp from ftp.gnu.ai.mit.edu, or other GNU ftp mirror sites. It takes a little work to install, but is a very nice compiler, and the price is right, especially if you can't stretch your budget to cover VAXC or DECC.

In addition, if you plan to exchange news with other sites, you'll need some means of communicating with these sites. If you only expect to exchange a small amount of information, you can do it via mail, though this rapidly becomes inefficient as news traffic increases. For higher volume exchanges between systems with intermittent or low bandwidth connections, you may want to use UUCP (a VMS version is available from DECUS) or a similar file exchange protocol. In the end News doesn't much care about the means by which items are transported, so if you want, you can communicate with other sites by mailing them tapes, though this is rather slow.

If you have continuous high-speed links to the sites with which you exchange news, the News distribution includes tools to handle this exchange via DECnet or TCP connections, using the Network News Transfer Protocol (NNTP), and standard for exchanging news between sites. If you're interested in the details, NNTP is described in RFC977, which is provided in the News_Doc ZIP file, but you don't need to know the details to run News. For most sites, TCP connections are preferable to DECnet connections, since they involve less overhead, but some sites may want to take advantage of capabilities which DECnet provides but TCP/IP lacks (e.g. user authentication). These configuration choices are described in more detail in the later sections of this manual. As of this writing, News includes support for the following TCP/IP packages:

- CMU-IP - available by anonymous ftp to ftp.csus.edu, in /pub/cmuip
- DEC's TCP/IP Services for VMS (previously known as UCX)
- Exos TCP
- Process Software's TCPware for VMS
- TGV Multinet
- The Wollongong Group's PathWay for VMS or WIN/TCP for VMS

News may work with other packages as well, if their API matches one of the packages described here. If you are running News over some other TCP/IP stack, please let us know, so we can add that information to the manual.

3.2 Deciding which newsgroups to carry

If you are planning to set up a local news database and exchange news with other sites, you will need to decide what portions of the newsgroup hierarchies available to you will be carried at your site. If you choose to run News as an nntp client only, this is not really an issue, since the NNTP server will determine which groups it makes available to clients connecting from your site.

In making this decision, you should bear several factors in mind. On the one hand, in order to best serve your users, you may want to carry a broad spectrum of newsgroups, covering topics of personal as well as professional interest. On the other hand, as the size of your feed increases, so does the amount of space necessary to hold items, the CPU and I/O load associated with maintenance batch jobs, the cost of transmitting and receiving items, etc. At the time this was first written (July 1994), a full feed of all hierarchies available to a typical site in the US includes over 5000 newsgroups, requires up to 7 Gbytes of disk space, and uses over an hour per day of (off hours) CPU time for maintenance. For example, here're a few quotes one site manager recently posted to the net:

```
From: sloane@kuhub.cc.ukans.edu (Bob Sloane)
Newsgroups: news.software.anu-news
Subject: Re: newsskim efficiency
Date: 20 Sep 93 16:12:35 GMT
Organization: University of Kansas Academic Computing Services

Others have commented on various performance enhancements that you
might try. I am running NEWS on a VAX 9210 which is pretty overloaded
with scientific computing. Over the course of a month, there is NEVER
any idle cpu time. NEWS processing accounts for about 8-10 percent of
CPU utilization. I have NEWS_MANAGER, NEWS_ROOT and NEWS_DEVICE all
on separate drives. I have six different incoming feeds of more than
2000 newsgroups, and I provide about ten outgoing full feeds, as well
as twelve partial feeds of varying sizes and NNTP service for our
campus. A normal SKIM command (done every night) takes about 4-6
hours, and a full SKIM (done once a week) take 8-12 hours depending on
other load. I have no problems keeping up with the news flow, so it is
possible.

From: sloane@kuhub.cc.ukans.edu (Bob Sloane)
Subject: Re: disk init params
Date: 25 Jul 94 14:09:20 CDT
Organization: University of Kansas Academic Computing Services

In article <1994Jul20.191110.5023@govonca.gov.on.ca>,
 newsmgr@[192.197.192.2] writes:
> 1.  If I wanted all groups for about 30 days, what size disk would do the
>     job.
```

```
I think about 6-7 Gigabytes of total space would be enough. You would
need about 150 Megabytes per day to store the articles, plus 2 blocks
overhead for each article (INDEXF.SYS entry + NEWS.ITEMS entry), plus
whatever was wasted due to the cluster factor of the disk. With a
cluster factor of 3, we are currently getting about 50,000 articles
per day, so about 300,000 for the articles, plus 100,000 for overhead,
plus 75,000 for cluster factor totals 475,000 blocks per day that you
want to keep should be enough, or about 4-5 days per GB.  Of course,
news volume is increasing daily, so by the time you get this the
reqirements may be higher. :-)
```

Your decision, then, must be based in part on local resources and policy regarding access to and priority for usage of those resources.

As you may already know from experience, newsgroups are organized into large groups called 'hierarchies' according to topic; names of groups within a particular hierarchy generally begin in the same way. Several hierarchies are commonly available on the net, including

- Usenet - this is actually a collection of seven hierarchies, in which sites have agreed to adhere to common policies for creation and control of newsgroups. If you're interested in the details, you should look in the newsgroup news.announce.newusers for a start. The individual hierarchies are:

comp.*	(516 newsgroups)	topics related to computing, including hardware, software, operating systems, networks, theory, etc.
misc.*	(44 newsgroups)	topics which don't fall into another hierarchy
news.*	(22 newsgroups)	topics related to news software and administration
rec.*	(298 newsgroups)	topics related to recreational activities
sci.*	(88 newsgroups)	topics of scientific interest
soc.*	(96 newsgroups)	topics related to sociology and culture
talk.*	(23 newsgroups)	open discussions on various issues

  In addition, Usenet includes the small trial_.* hierarchy, a place for new newsgroups to live temporarily for trial runs before they are submitted for a formal vote on their permanent creation.

- 'alt' - the 'alternative' hierarchy, so called because it began as an alternative to Usenet, is a free-wheeling collection of newsgroups on almost any topic. The hallmark of this hierarchy is its lack of organization; anyone can create a newsgroup, and many of the groups are short-lived responses to fads. Some groups are serious discussions with too small or specialized a following to support creation of a Usenet group, while others deal with 'fringe' topics not acceptable to some sites which carry Usenet. Newsgroups in this hierarchy begin with 'alt.'; at this writing, there are 990 of them (though that will probably change within the hour).

- professional hierarchies - several smaller newsgroup collections covering topics of interest to a particular profession, or centered around a particular organization, exist. Some have well-defined policies for creation and control of newsgroups, while other are less organized. This category includes, among others,

bionet.*	(46 newsgroups)	topics of interest to biologists and biomedical scientists. Administrative matters are handled via the newsgroup bionet.announce.
biz.*	(40 newsgroups)	topics related to business
gnu.*	(28 newsgroups)	topics related to the Free Software Foundation's GNU project and the software they distribute
ieee.*	(6 newsgroups)	topics related to the IEEE
k12.*	(39 newsgroups)	topics related to elementary and secondary education

- bit.* - A large number of BitNet LISTSERV mailing lists are gatewayed into newsgroups for more efficient distribution and reading than is provided by individual email. These are grouped into the bit.* hierarchy, which presently contains 227 newsgroups. (Many newsgroups in other hierarchies are also gatewayed into mailing lists; FAQ postings to those lists and groups will usually mention this.)
- VMSnet - This hierarchy runs under the aegis of the VMSnet Working Group of the VMS MIF of DECUS' US chapter, and contains groups of interest to VMSnet readers. Not surprisingly, most groups deal with VMS - technical topics, networking, administration, software, jobs, etc. The hierarchy was established because most Usenet sites run Unix, and many were reluctant to carry newsgroups dealing with VMS-specific issues. While there is much more crossover between VMSnet and Usenet now, VMSnet continues to flourish as a community centered on VMS, operating under looser rules than the larger Usenet community. There are currently 36 groups in this hierarchy, whose names all begin with vmsnet.*; administrative issues are handled in the group vmsnet.admin.
- Commercial hierarchies - Some businesses sell information services which are distributed as newsgroup hierarchies which sites pay to receive. For example, ClariNet is a collection of around 250 groups produced by a news (in the traditional sense - wire services, print journalism, etc.) abstracting service. Control of groups in these hierarchies is usually by the producer, and arrangements for feeds are made with them directly.
- Regional hierarchies - in many areas there are collections of newsgroups defined by their distribution, carrying topics of interest to all sites within a municipality, state or province, nation, system of universities, etc. The administrative policies governing these hierarchies may vary from place to place, and details are best obtained from a nearby site.
- Institutional hierarchies - newsgroups provide a convenient conferencing tool and medium for disseminating information within an organization, so you may find (or want to create) a hierarchy of groups for internal use.
- to.* - at many sites, it is customary to create a newsgroup, named to.*site*, for each site with which news is exchanged (including yourself). Each of these groups is distributed only to the local system and the site which appears in the newsgroup name. (You do this when configuring News by specifying the newsgroup name in the *subscriptions* field of the News.Sys entry for the appropriate site only.) This provides you with a channel through which to post test and administrative messages intended only for a specific site to the appropriate to.*site* newsgroup, and the News distribution mechanism will forward

it only to that site. For instance, if your system is named *plato*, and you exchange news with sites *socrates* and *aristotle*, you would create newsgroups on your system named 'to.plato','to.socrates', and 'to.aristotle'. The group 'to.plato' would not be forwarded to any other sites, but would contain incoming messages intended for your site only. The group 'to.socrates' would be distributed to site *socrates* only, and the group 'to.plato' to site *plato* only, so you could easily send items to one site and not the other.

You'll probably want to choose a few of the possible hierarchies to start with, and then expand or set up local hierarchies as you become more familiar with News configuration and usage patterns at your site.

3.3 Choosing a News database configuration

If the machines on which you plan to install News are connected by high speed DECnet or TCP links, you can configure News to run in either of two ways: as a local reader/server, in which case News uses a database of items on a disk available to your node, or as a remote NNTP client, in which case News retrieves items as they are needed from another server, and does not maintain any local database. When functioning as a news reader/server, News can also operate in a hybrid configuration, in which items for some groups are maintained locally, while for others items are retrieved on demand and either discarded immediately or cached for a short time on the local node, in the expectation that other users will want to read them in the near future. Which configuration you choose will depend on the importance to you of the following factors (among others):

- number and size of the newsgroups you wish to make available
- disk space available to store news items
- speed with which News starts up and retrieves items
- CPU capacity available to process news feeds
- network capacity available to transfer news batches and carry requests from clients and responses from servers
- number and type of machines running News
- control over access to groups

These two configurations, as well as a mixed client/local configuration, are discussed in more detail below, and detailed instructions are given for setting up News in the configuration you choose. If you choose to run News as an NNTP client only, then you should follow the instructions in the section of this manual titled "Setting Up News As An NNTP Client Only". If you plan to have a local database, follow the instructions in the section of this manual titled "Setting Up News as a Local Reader/Server", even if you *also* plan to use News as an NNTP client some of the time. You're welcome to read the entire manual - you'll find a fair amount of material duplicated between sections, but where there are differences, follow the descriptions in the section that applies to your configuration.

3.3.1 News reader/server using local database

In this configuration, News maintains a collection of text files for all items available to readers, as well as a set of index files used for tracking articles and providing ready access to summary information. Since everything News needs to act as a reader is located on your system, you have the maximum amount of control over the newsgroups you carry, the reliability and redundancy with which news is passed to your site, the retention time for items, and the access a given user is allowed to each group. In addition to receiving distributed newsgroups, you can maintain a collection of local groups which serve as in-house conferences, and you can set up gateways between newsgroups and mailing lists. Finally, your node may act as a server for remote clients, making your news items available to users lacking the capacity to maintain an independent database. The tradeoffs for this flexibility are increased complexity in configuring News, the need for sufficient disk space to store all available articles, and the overhead incurred in exchanging news with other sites, adding new items to the local database, and removing expired items. If you serve as a central computing or information resource for your site, need to maintain fine control over news access, or don't have a connection to another news server readily available, you should probably choose to set up a local database. On the other hand, if your CPU or disk capacity is limited, or you have relatively few users who need basic access to news, or you have access to a central news server, you may wish to run News in the simpler NNTP client configuration.

3.3.2 Diskless NNTP client

In this configuration, News acts as the user interface to a remote news server, from which it retrieves group and item statistics and individual articles as they are needed by an individual user. This is the simplest configuration to use - you just define a few logical names which News uses to determine how to access the server and how to mark local postings, and go. However, you must depend on the news administrator for your server to maintain the newsgroups you want, and you have no control (other than by arrangement with this news administrator) over newsgroup configuration, access, or distribution. In addition, newsgroup moderators will be unable to exercise their roles via an NNTP client (except through some unpleasant workarounds).

Finally, running as an NNTP client is somewhat slower than using a local database, with the magnitude of the change depending on number and size of newsgroups and network capacity and load.

3.3.3 Mixed client/local

This configuration combines many of the advantages of the previous two options, relying principally on a local database of newsitems, but allowing the local news administrator to designate specific newsgroups as served by a remote node. This can relieve a disk space crunch or offload processing overhead for large newsgroups to a host with more capacity, or it can allow hosts on a LAN to distribute the task of maintaining large newsgroups. As currently implemented, however, this option is efficient only if used for a few newsgroups per site. Further, while users on the local node can read served newsgroups, the local nntp server will not be able to send item text to its clients (though it will provide item count and header information, often causing confusion among users as they see 'phantom' newsgroups).

For instance, if on node *cicero* the newsgroup comp.os.vms is set up as a remote newsgroup served by node *scipio*, users on *cicero* will have access to the group, but nntp clients on node *pliny* using *cicero* as a server will see only the item directory display, and will not be able to access the items themselves.

Chapter 4
Setting Up News As An NNTP Client Only

If you plan to run News as an NNTP client only, you will only need to set up a small subset of the News package, as most of the configuration details are subsumed by the server.

4.1 Generating the required files

Once you have unpacked the News distribution files as described above, read the comments in NewsBuild.Com describing how to use this procedure. In addition, you should read through the file NewsSite.H. In most cases, the options taken in the distributed version of NewsSite.H will work at your site. In some cases, however, you may need to make changes to accommodate unusual aspects of your situation. These settings fall into three categories:

- Parameters which may be changed freely to meet the needs of your particular site. This group includes DISPLAY_BY_NEWSGROUPS, FAST_LOAD, MENU_SUPPORT, NEWGROUP_REG, NOALLOW_USER_FROM, SILENT_CLIENT_STARTUP, STARTLINE, STRIP_ERR_LINES, USENET, and VMSV4/VMSV5.
- Parameters which should not be changed without a good understanding of the way in which they are interpreted, and the effects changes may have on News and related supporting code. This group includes ARBITRON, DEBUG, MAXUNK, MEM_CACHEMAX, MEM_CHECK_QUOTA, MEM_DYNAMIC_QUOTA, MEM_RESERVE_SIZE, NEWS_INIT, NNTP_USE_XGNOTICE, NNTP_USE_XGTITLE, NNTP_USE_XHDR, NNTP_USE_XOVER, NNTP_USE_XUPDGROUP, NOTE_RANGE_MAX, RECHECK_TIMER, SECLNM_MODE, SECLNM_TABLE, SECURE_LOGICALS, and UWO.
- Parameters which identify the version of News you are running. These should not be changed under normal circumstances. This group includes NEWS_DDATE, NEWS_VERSION, NEWS_VDATE, and nntp_version.

Those parameters not listed here do not usually affect operation of News as a NNTP client.

If possible, you should build copies of all necessary binaries on your own system, in order to insure that the proper header files are used, and that you are linked to the proper RTLs. The C source code in the News distribution can be compiled using VAXC, DECC, or gcc without modification.

If you don't have access to a C compiler, consider obtaining and installing a free compiler such as GNU's gcc. It's only available for VAX systems as of this writing, and can be obtained by anonymous ftp from ftp.gnu.ai.mit.edu, or other GNU ftp mirror sites. It takes a little work to install, but is a very nice compiler, and the price is right, especially if you can't stretch your budget to cover VAXC or DECC.

Note to gcc users

By default, NewsBuild.Com disables the '-fbuiltin' optimization, because of a bug in gcc versions before 2.6.3 which could cause corruption of memory when memcpy( ) was called. If you're using gcc 2.6.3 or later, you can remove the '-fno-builtin' from the /Cc1_Options value.

If you cannot do this, or wish to get News running immediately, copies of binaries built with Multinet TCP support are provided in the News_Dist ZIP file, and copies of binaries built with Wollongong or UCX support are available for anonymous ftp at genetics.upenn.edu, in the directory Anon_Root:[ANU_News.Distribution].

After you have looked over NewsBuild.Com and NewsSite.H, execute NewsBuild.Com from the [.News_Src] subdirectory, in order to build a copy of News.Exe and News.Hlb, incorporating code for the NNTP transport you will use. In general, NNTP over TCP/IP is faster than NNTP over DECnet, so unless you need to take advantage of the user authentication DECnet provides, you should probably choose TCP as your transport protocol. You may wish to specify CLIENT as p4 to NewsBuild.Com, since you will only need the news reader and help library, and to define the C preprocessor symbol NNTP_CLIENT_ONLY, either via p5 of NewsBuild.Com or in NewsSite.H, in order to eliminate from News.Exe code which is unnecessary when running as a client.

Once you have executed NewsBuild.Com, create a directory to hold the binaries and configuration files used by News, and copy News.Exe and News.Hlb to it. You will later define the logical name News_Manager to point to this directory, so it is often referred to as 'the News_Manager directory', or just 'News_Manager'.

In addition, you may wish to create a NewsRC.Template file, which is used to establish the context (groups registered and profile settings) for first-time users. Two procedures for doing this are described below.

File protections should be set to allow world read access to the files in News_Manager, and to any other file specified through the logical names discussed below. Alternatively, you may install News with SysPrv, to allow it to access these files; it will disable privileges at all other times.

4.2 Setting up the News logical name environment

In order to provide site-specific information to News, the following logical names need to be defined. You will probably want to put these definitions into a DCL procedure to be executed manually now, and as part of system startup in the future. Several of these logical names are considered 'secure' by News, that is, News will use only translations obtained in accordance with the rules specified in NewsSite.H for secure logical name translation. This is indicated in the explanation of each logical name.

4.2.1 News_Address

This logical name is used as the name of the local host when constructing mail addresses, and when checking whether the current user is allowed access to or moderator privileges for a newsgroup. The rules for translating secure logical names are followed when attempting to translate News_Address. If this is not successful, the translation obtained for News_Node as specified below is used.

4.2.2 News_GMT_Offset

This logical name specifies the offset of your local time from GMT (UTC), as a string with the format hh:mm:ss, and its translation is used in converting between GMT in headers and local time. The string may begin with '+' or '-', indicating whether the offset should be added to or subtracted from UTC to obtain the local time; if the sign is absent, '+' is assumed. This logical name is normally defined in the system logical name table, although translation is via the normal lookup process, so definitions in the process, job, or group logical name table are used in preference to the system-wide value.

4.2.3 News_HelpLib (optional)

If this logical name is defined, it must translate to the file specification of the help library for News commands. When looking for its help library, News first tries to open the library specified by News_HelpLib. If this fails, it then looks for a library named News.Hlb, first in News_Root (which is generally not defined on a system which operates only as a NNTP client), then in Sys$Help, and finally in the same directory as News.Exe. This logical name is normally defined in the system logical name table, although translation is via the normal lookup process, so definitions in the process, job, or group logical name table are used in preference to the system-wide value.

4.2.4 News_Locked_Command

When you run NewsShutdown.Exe, if it successfully obtains an exclusive lock on the News resource, it spawns a subprocess and passes the translation of this logical name to the subprocess to be executed as a DCL command. Translation of this logical name is via the normal lookup process, and it is usually defined in the process table just before executing NewsShutDown.Exe.

In general, NewsShutDown.Exe will not be used on a system running only as an NNTP client, since it does not have a local News database. However, it may be useful on occasions when you need to prevent users from running News for some other reason.

4.2.5 News_Locked_Wait_Minutes

After NewsShutDown.Exe enqueues the request for an exclusive lock on the News database, it will wait for the number of minutes specified by the translation of this logical name before giving up and exiting with the status SS$_CANCEL, without executing News_Locked_ Command. If this logical name is not defined, then NewsShutDown.Exe will wait one minute before giving up. Translation of this logical name is via the normal lookup process, and it is usually defined in the process table just before executing NewsShutDown.Exe.

4.2.6 News_Manager

This logical name should translate to the name of the directory in which News.Exe and its ancillary files reside. When attempting to translate this logical name, the rules for translation of secure logical names are used.

4.2.7 News_Mgr_ID

This logical name should translate to the name of the rights identifier granted to News managers to allow them management privileges. When attempting to translate this logical name, the rules for translation of secure logical names are used.

4.2.8 News_NetPost (optional)

If this logical name is defined, its translation is used as the name of a rights identifier which, when held by a user, indicates that the user is allowed to post items to network newsgroups, regardless of whether that user also holds the identifier specified by News_NoNetPost. When attempting to translate this logical name, the rules for translation of secure logical names are used.

4.2.9 News_NNTP_Protocol

This logical name should translate to the name of the transport protocol you specified when compiling News. The syntax for transport names is identical to that used with the /NetProtocol qualifier to the News command. When choosing the transport protocol to use for a given session, News uses the following order of precedence:

1. protocol supplied as a value to the /NetProtocol qualifier on the command line
2. protocol stored in the NewsRC file from a previous session

3. protocol obtained by translating the logical name News_NNTP_Protocol

This logical name is normally defined in the system logical name table, although translation is via the normal lookup process, so definitions in the process, job, or group logical name table are used in preference to the system-wide value.

4.2.10 News_NNTP_Server

This logical name should translate to the name of the NNTP server you plan to use as your default server. If you're using DECnet as the transport protocol, then this should be the DECnet nodename, or the node number in the format *area**1024+*node* (*area**1024 is optional if the server is in the same area as your system). If you're using TCP, then this should be the host name of the server, or its IP address in dotted decomal form. When choosing the server to which to connect for a given session, News uses the following order of precedence:

1. server name supplied as a value to the /NetServer qualifier on the command line
2. server name stored in the NewsRC file from a previous session
3. server name obtained by translating the logical name News_NNTP_Server

This logical name is normally defined in the system logical name table, although translation is via the normal lookup process, so definitions in the process, job, or group logical name table are used in preference to the system-wide value.

4.2.11 News_Node

This logical name specifies the Usenet name for the local node, and is used to identify it in Usenet headers (e.g. Relay-Version:, X-News:). In addition, it serves as a default value for the name of the local host in Path: headers if News_Pathname is not defined, and in mail addresses if News_Address is not defined. If this logical name is undefined, another logical name may be used, depending on the NNTP transport you specified when building News:

NNTP transport	Logical Names (in order checked)
Multinet	Multinet_SMTP_Host_Name
	Multinet_Host_Name
TCPware	TCPware_SMTP_From_Domain
	TCPware_DomainName
UCX	UCX$Inet_Host
Wollongong	ARPANet_Host_Name
All	Internet_Host_Name
	Sys$Cluster_Node
	Sys$Node

Attempts to obtain a translation for this logical name and any of its fallbacks follow the rules for secure logical names. Finally, if no logical name is found, News prints an error message and exits.

4.2.12 News_NoNetPost (optional)

If this logical name is defined, its translation is used as the name of a rights identifier which, when held by a user, indicates that the user is not allowed to post items to network newsgroups. When attempting to translate this logical name, the rules for translation of secure logical names are used.

4.2.13 News_Organisation (or News_Organization) (optional)

If this logical name is defined, its translation is used in constructing the Organization: header of outgoing items. This logical name is normally defined in the system logical name table, although translation is via the normal lookup process, so definitions in the process, job, or group logical name table are used in preference to the system-wide value.

4.2.14 News_Pathname (optional)

If this logical name is defined, its translation is used as the name for this site in Path: headers, instead of the translation of News_Node. When attempting to translate this logical name, the rules for translation of secure logical names are used.

4.2.15 News_Post_CC (optional)

It some cases, when an item is posted into a particular newsgroup, it is useful to mail a copy to certain email addresses (e.g. a mailing list). If you wish to do this, you should create a file in the format specified for News_Manager:News_Post.CC (described in the section on configuration files in the News server setup section), and define the logical News_Post_CC to point to this file. (The file News_Manager:News_Post.CC is not itself used when posting as a client.) Note that the file to which News_Post_CC points supersedes the user's private file Sys$Login:News_Post.CC, if it exists. Note also that News_Post_CC is translated using the normal sequence of logical name tables and access modes, so a process-private definition will supersede a system-wide one.

4.2.16 News_*remhost*_Task (optional)

When attempting to establish a DECnet connection to node *remhost*, News will try to translate the logical name News_*remhost*_Task (e.g. when connecting to node VIRGIL, News will try to translate News_VIRGIL_Task), and, if successful, will use this translation as the name of the DECnet task to which to connect. This logical name is normally defined in the system logical name table, although translation is via the normal lookup process, so definitions in the process, job, or group logical name table are used in preference to the system-wide value.

4.2.17 News_Stop (optional)

If this logical name is defined in the system logical name table in executive mode, then each currently active News and NNTP server session will exit when it finishes executing the current command (or after 30 minutes in any case), releasing any locks they may have held in the News index files. (Exception: Multithreaded NNTP servers will not respond to this logical name while hibernating between connections, so the server process may have to be stopped explicitly by the process which requires exclusive access to the News database.) This is an older method of obtaining exclusive access to the News database, and NewsShutDown.Exe is the preferred method of accomplishing this in the current release of News. For more information and an example of performing a task requiring exclusive access to the News database, see the section in this manual on optimizing News index files.

4.2.18 News_Timezone

This logical name specifies the time zone notation to be added to Date: headers of outgoing items. It should translate to a standard time zone abbreviation (e.g. EST, PDT) acceptable according to RFC822. If this logical name is undefined, depending on the NNTP transport you specified when you built News, the translation of the following logical name is used:

NNTP transport	Logical Names
NNTP Transport	Logical Names
All	PMDF_Timezone
Multinet	Multinet_Timezone
TCPware	TCPware_Timezone
Wollongong	WIN$Time_Zone

These logical names are searched in the order given, and the search stops at the first match. If no appropriate logical name is defined, "GMT" is used as the time zone. This logical name is normally defined in the system logical name table, although translation is via the normal lookup process, so definitions in the process, job, or group logical name table are used in preference to the system-wide value.

4.2.19 News_Usage_Anonymous (optional)

If this logical name is defined, then all entries written to the usage log (see News_Usage_Log logical name below) do not contain the username. Instead, the value of the Account field of the SysUAF record for that username is used. When attempting to translate this logical name, the rules for translation of secure logical names are used.

4.2.20 News_Usage_Log (optional)

If this logical name is defined, its translation is used to specify a file to which a summary of each news session will be written. (News.Exe must have write access to the file, of course.) The summary contains the username (if News_Usage_Anonymous (q.v.) is not defined), the newsgroups accessed, and the number of items read and posted in each newsgroup. When attempting to translate this logical name, the rules for translation of secure logical names are used.

4.3 Setting up optional configuration files

The files described here are used to configure optional functions of News; none of them are required for a basic NNTP client setup.

4.3.1 Editing the News.Aliases file

News uses this file when processing the newsgroup names in the headers of incoming items. If a newsgroup name with an entry in this file is found in the header, it is replaced with the newsgroup name(s) specified in the entry. Note that all such changes are passed on to downstream sites as well. Entries in this file have the format:

newsgroup_name replacement_list

newsgroup_name - the name of the newsgroup to which this entry applies. *Newsgroup_name* cannot contain whitespace. It is separated from *replacement_list* by any amount of whitespace. In order for *replacement_list* to be substituted into an item's Newsgroups: header, a newsgroup name in the header must exactly match *newsgroup_name*. Comparison is case-insensitive.

replacement_list - a list of newsgroup names, separated by commas, which replaces *newsgroup_name* in the Newsgroups: header of the item. *Replacement_list* cannot contain any whitespace. It may be continued across line breaks by specifying the character '\' as the last character on a line.

All characters following the character '#' up to the end of the line are ignored.

Each newsgroup in the item's original Newsgroups: header is matched against every entry in News.Aliases, but the matching is not recursive (i.e. newsgroup names substituted into the header are not then matched against News.Aliases). Mapping of newsgroup names to their aliases is performed before other configuration files which direct actions based on newsgroup names are consulted.

If you need to remap newsgroup names in this fashion, you may set up this file in one of two ways. If you choose, you may edit the template file provided in the News_Dist ZIP file, and place the edited copy in News_Manager. You may also add, remove, or display entries in this file using the commands Add Alias, Delete Alias, and Show Alias, respectively, from within News; the file will be created when you use Add Alias for the first time. Modifying the alias file from within News causes all comments within the file to be lost.

4.3.2 Creating the News.Classes file

In additional to the automatic classes $net and $local, News can create and populate newsgroup classes on a system-wide basis, based on the contents of the file News_Manager:News.Classes. These classes cannot be modified from within News. Entries in this file should be in the following format:

class_name newsgroup_list

class_name - the name of the class to be created. If *class_name* does not begin with the character '$', which designates an unmodifiable class, then a '$' will be prepended to *class_name* to form the name of the actual class created. The class names '$net' and '$local' are reserved for automatic classes created by News, and entries specifying either of these names will be ignored. *Class_name* cannot contain whitespace, and is separated from *newsgroup_list* by any amount of whitespace.

newsgroup_list - a list of newsgroup patterns which specify the newsgroups to be added to the class. Within each pattern, the character '*' is a wildcard which matches any number of characters, while the characters '%' and '?' are wildcards which match any single character. Patterns in *newsgroup_list* may be separated by a comma or whitespace.

An entry can be continued across multiple lines by ending each line but the last with the character '\'. All text between the character '#' and the end of the entry is ignored.

If you wish to have certain newsgroups entered into specific classes by default, you should create this file yourself. For instance, a simple News.Classes file might say:

Example 4–1: Sample News.Classes File

```
fun_groups  rec.*,talk.*
science     bionet.*,sci.*
bargains    *.forsale
vms_stuff   comp.org.decus,comp.os.vms,vmsnet.*
```

4.3.3 Editing the News.Distributions file

This file contains a list of distribution keywords accepted by News, along with brief descriptions, in the following format:

keyword description

keyword - the keyword which would appear in the Distribution: header of an article posted to the distribution described by this entry.

distribution - a one-line description of the scope of the distribution denoted by the presence of *keyword* in the Distribution: header of an article. Note that this is just for information; the functional interpretation of keywords in Distribution: headers is via the *distributions* and *subscriptions* fields of entries in News.Sys.

When a user running News.Exe posts an item, all keywords specified in the Distribution: header must have entries in this file. If this file does not exist, then any keyword is considered acceptable.

You should edit the template file provided in the News_Dist ZIP file to reflect the distributions you accept and place the edited file in News_Manager.

4.3.4 Editing the News_Address.Cnf file

News_Address.Cnf contains a series of rewrite rules used by News to rewrite mail addresses for outgoing items. This allows you to transparently route mail to different sites by different mechanisms, or to specify a foreign mail transport without requiring users to type the prefix with every address. Entries in this file have the format

address_pattern rewrite_rule

address_pattern - a text pattern which is matched against the mail address to be rewritten. The '*' character is a wildcard which matches any number of characters in the address, while the '?' is a wildcard which matches any one character in the address. (Note that unlike similar patterns in other News configuration files, the '%' character is a literal, not a wildcard, since it often appears in mail addresses.) The '^' character is ignored in *address_pattern*. *Address_pattern* may not contain any white space. It is separated from *rewrite_rule* by any amount of white space (excluding line breaks).

rewrite_rule - a text pattern specifying how the address is to be rewritten, if it matches the pattern specified in *address_pattern*. Characters are copied directly from the rewrite rule into the rewritten address, except that each time the string '\n', where n is an octal number <24 (octal), is encountered, the string from the original address which matched the nth '*' wildcard in *address_pattern* is substituted into the rewritten address. For example, for the following entry:

IN%*@*.decnet.myuniv.edu \002::\001

the address 'IN%aeneas@virgil.decnet.myuniv.edu' would match the *address_pattern*, with the string 'aeneas' matching the first '*', and the string 'virgil' matching the second '*'. According to *rewrite_rule*, then, this address would be rewritten as 'virgil::aeneas'.

Any text following a '#' character, up to the end of the line, is ignored.

Entries are searched in the order encountered in the file, and the search stops with the first entry which matches the address to be rewritten. An address is converted to lower case before being rewritten. In the rewritten address, case is preserved for characters copied from the *rewrite_rule* pattern, but characters substituted from the original address are in lower case. Finally, note that only addresses seen by News are rewritten; addresses indirectly passed to VMS Mail (e.g. by specifying a distribution file at the To: prompt) are not.

Read the comments in the version of this file distributed in the News_Dist ZIP file, and, if you decide it would be useful to you, edit it and place the edited copy in News_Manager. If this file does not exist in News_Manager, then News uses the following default rewrite rules:

1. If the logical name SMTP_MailShr is defined or the file Sys$Share:SMTP_MailShr.Exe exists, then the following rewrite rules are used:

```
smtp%*          smtp%\001
psi%*           psi%\001
@*              @\001
*::*            \001::\002
*@*             smtp%"\001@\002"
*               \001
```

2. If the conditions for (1) are not true, but the logical name ST_MailShr is defined or the file Sys$Share:ST_MailShr.Exe exists, then the following rewrite rules are used:

```
st%*            st%\001
psi%*           psi%\001
@*              @\001
*::*            \001::\002
*@*             st%"\001@\002"
*               \001
```

3. If the conditions for (1) and (2) are not true, then the following rewrite rules are used:

```
in%*            in%\001
psi%*           psi%\001
@*              @\001
*@*             in%"\001@\002"
*::*            in%"\002@\001"
*               \001
```

4.3.5 Editing the News_Post.CC file

News_Post.CC contains entries in which newsgroup names are paired with mail addresses, so that when an item is posted to the newsgroup, a copy is also mailed to the address specified. Note that the posting also goes out via the normal mechanism specified for the newsgroup, so using this file is not equivalent to using the /MailList qualifier when creating a newsgroup. Entries in this file have the following format:

newsgroup_pattern address_list

newsgroup_pattern - a pattern matched against the newsgroup names in the Newsgroups: header of the item. The character '*' is a wildcard matching any number of characters, and the characters '*%*' and '?' are wildcards matching any one character. The pattern may not contain any whitespace. Comparison is case-insensitive.

address_list - a list of mail addresses to which a copy of the item is forwarded, when a newsgroup name in the Newsgroups: header matches *newsgroup_pattern*. The list is converted to lower case. It is separated from *newsgroup_pattern* by any amount of whitespace, and may contain whitespace within the list (though multiple spaces will be compressed to a single space when the list is used), but it may not cross a line boundary.

All text between a '#' character and the end of the line is ignored. Each newsgroup name in the item's Newsgroups: header is matched against every entry in this file.

In order to use this capability, edit the template file provided in the News_Dist ZIP file, and place the edited copy in News_Manager.

4.3.6 Editing the News_Post.Defaults file

This file is used to supply default values for the Distribution: and Followup-To: headers for items posted to certain newsgroups. This allows you to direct followups and limit distribution of posts to certain newsgroups automatically. The format of entries in this file is:

newsgroup_pattern
    header_keyword default_words
    [*header_keyword default_words*]

newsgroup_pattern - a pattern matched against the newsgroup names in the Newsgroups: header of the item. The character '*' is a wildcard matching any number of characters, and the characters '*%*' and '?' are wildcards matching any one character. Comparison is case-insensitive. When the matching is performed, a '*' is appended to the beginning and the end of *newsgroup_pattern*. If *newsgroup_pattern* is the keyword 'default', it matches any newsgroup name. *Newsgroup_pattern* must begin in column 1.

header_keyword - the name of the header for which defaults follow. Two keywords are recognized: 'distribution:' and 'followup-to:'. Recognition is case-insensitive. The line(s) containing *header_keyword* must begin with whitespace.

default_words - a comma-separated list of default values to be added to the header specified by *header_keyword* of any item containing a newsgroup in its Newsgroups: header which matches *newsgroup_pattern*. The text of *default_words* is converted to lowercase and added to the words entered into the header in question by the user. If a word is already present in the item header, it is not duplicated.

All text between a '#' character and the end of the line is ignored. Each newsgroup name in the item's Newsgroups: header is matched against every entry in this file.

If you find this useful, edit the template file provided in the News_Dist ZIP file, and place the edited copy in News_Manager.

4.4 Installing News.Exe

In order to function properly, News.Exe must have read access to the News configuration files in News_Manager, as well as to News.Hlb. You can grant this access by allowing world read access to each of these files, or you can restrict access to these files and allow News.Exe to obtain access by installing it with SysPrv. It will disable this privilege as soon as it starts up, and will enable it only when necessary.

In addition, if you wish to use the News lock mechanism (i.e. NewsShutDown.Exe) to prevent users from running News while certain other tasks are being performed, News.Exe must be installed with SysLck. If you are using message-ID caching, you must also add SysGbl to this list.

Finally, if you expect News.Exe to be in use frequently, you may may improve performance by installing it with its header resident and for shared access.

All of these things are accomplished via the Install utility, using a command like

```
$ Install Add News_Manager:News.Exe/Open/Header/Share -
          /Priv=(SysPrv,SysLck,SysGbl)
```

This command should be executed when you first install News, and as part of normal system startup.

4.5 Setting up a mail contact address for News matters

Any site exchanging news in the Usenet hierarchies is required to route mail sent to the address usenet@site to the person responsible for managing news usage at that site. In effect, usenet@site is to news what postmaster@site is to mail. Even if you don't exchange news feeds with other sites, if users from your site post to network newsgroups, readers outside your organization may try to contact this address to report technical problems (e.g. the From: addresses in posts from your site are unreplyable) or breaches of netiquette (e.g. a user at your site is posting chain letters). You should create a mail forwarding address or define a system-wide logical name which will direct incoming mail with this address to the appropriate person. For instance, you can do this using VMS Mail's Set Forward command:

```
$ Mail
MAIL> Set Forward/User=USENET myaddr
MAIL> Exit
```

where *myaddr* is the valid mail address to which you want incoming messages to USENET directed.

4.6 Providing a foreign command to run News

News.Exe is written to be invoked from DCL by a foreign command, and will parse its own command line. You should add a line to your system-wide login procedure which sets up this command, e.g.

```
$ News == "$News_Manager:News.Exe"
```

You may specify the name of the NNTP server and the NNTP transport protocol to use when connecting to the server in several ways; see the sections describing the logical names News_NNTP_Server and News_NNTP_Protocol for more information.

Note that if you have for some reason defined the logical name News_Root on your system, you will need to place at least the qualifier /NetServer on the command line when invoking News as an NNTP client. Otherwise, it will try to find the local News database, and when it does not, will either attempt to create the index files for the local database (if News.Exe can obtain SysPrv, and it is run by a user holding News_Mgr_ID), or exit after printing a message about first-time installation.

4.7 Moving News.Hlb

If you place News.Hlb in Sys$Help or the directory in which you have placed News.Exe, News will be able to find it automatically. If you place it in another directory, you should define the logical name News_HelpLib as the file specification for the help library. News.Hlb should be world readable.

4.8 Optional steps to assist users

In order to simplify use of News, especially for new users, you may wish to provide any or all of the following aids.

4.8.1 Documentation

The News_Doc ZIP archive contains plain text, PostScript, RTF and TeX sources for user documentation, as well as several useful or informative items for news admins (some of which we hope you've already read :-)). The documentation explains News concepts and commands. Further information on the current state of Usenet and hints on netiquette can be found in the newsgroup news.announce.newusers. We encourage new users to read the regular postings to this group in order to acquaint themselves with Usenet culture.

4.8.2 System help library entry

The News_Dist ZIP archive includes a file named DCLNews.Hlp, which contains a sample entry describing News for inclusion in a system help library. You may wish to edit this file appropriately and add it to Sys$Help:Help.Hlb or a site-specific help library.

4.8.3 NewsRC.Template file

When a user invokes News for the first time, he or she will not normally have a NewsRC file in Sys$Login to provide information on profile settings and newsgroup context. While News can fall back to default settings in the absence of a NewsRC file, this often leads to intimidating lists of 'New newsgroup:' messages at startup, or may make it harder for users to take advantage of specific features of your setup. In order to help customize the environment for new users, News will use the file News_Manager:NewsRC.Template as the NewsRC file for their first session, if this file exists. The simplest way to create such a file is to rename your NewsRC file, run News yourself, set up the user profile and group registration appropriately, exit News, and copy the resulting NewsRC file to News_Manager:NewsRC.Template. (Be careful not to use your regular NewsRC for this, as you will be including in the template file a record of items you have previously read, which may lead to confusion later.) Alternatively, you may use a DCL procedure like this to create the template file:

Example 4–2: Sample Make_NewsRC_Tempate.Com Procedure

```
$!---------------Make_NewsRC_Template.Com-------------------
$ Run_News _= "$News_Manager:News/NetServer/NoScreen"
$ Open/Write templ Sys$Login:NewsRC_Setup.
$! This is the first line from a valid NewsRC. file on your system -
$! any one will do - to point News to the proper server.
$ Write templ "~netnews.upenn.edu 2~2B643CC8 news.announce.important_4#"
$ Write templ "PROFILE"
$ Write templ "UNSEENSTACK"
$ Write templ "NEWREGISTER_=none"
$ Write templ "DISPLAY_=(L1,P1)"
$ Write templ "FROMSIZE_=15"
$ Close templ
$ Define/User NewsRC "Sys$Login:NewsRC_Setup."
$ Run_News
Set Profile/NewRegister_="*/confirm"
Deregister/All
Register news.announce.newusers
Select news.announce.newusers
Exit
$ Copy Sys$Login:NewsRC_Setup. News_Manager:NewsRC.Template
$ Delete/Log/NoConfirm Sys$Login:NewsRC_Setup.;*
$ Purge/Log/NoConfirm News_Manager:NewsRC.Template
$ Exit
```

In either case, you will want to update your template file periodically, so that it accurately reflects the current set of newsgroups available from your server.

Chapter 5
Setting Up News as a Local Reader/Server

In this configuration, News maintains local copies of all news items available to your users. These files, referred to as the local News database, are kept current by exchanging new items with other sites on the net, and by removing old items from the local store as they reach their expiration date. You can also use News in this configuration as a conferencing tool to manage discussion groups among users at your site, either alone or in addition to carrying network newsgroups. These local newsgroups are often superior to mailing lists, since they allow you to keep only one copy of each message in a central location, and they provide an overview of recent items in a thread which a user might not have retained in his or her mail file.

5.0.1 Setting up disk space for News use

In order to maintain a local news database, you will need to set aside sufficient disk space for the News images and support files, for the database itself, and for the temporary files News creates during various maintenance tasks. If you plan to carry a large number of newsgroups, this can require quite a lot of space, as well as some attention to tuning in order to prevent News maintenance jobs from bogging down. As you decide how to allocate space, it may be helpful to consider the following five areas.

5.0.2 News images and support files

These files are usually grouped in one directory, referenced by the logical name News_ Manager. This area is also often used as the root of a directory tree, referenced by the logical name News_Manager_Dev, for spool files holding articles which have arrived from feed sites, but have not yet been added to the local News database, and items awaiting transport to downstream sites. Depending on the size of your newsfeeds, this area can take up as little as a few thousand disk blocks or as much as a few hundred thousand.

5.0.3 Index files for the local news database

The three ISAM files News.Groups, News.Items, and History.V60 are used by News to keep track of items in the local database. They are all located in the directory referenced by the logical name News_Root. In addition, if you are using NNTP message ID caching, the section file News.Cache is located here as well. The index files can grow quite large on a system carrying a full newsfeed (in particular, News.Items can be over 200,000 blocks in size), and performance is improved considerably if fragmentation of these files is kept to a minimum. For this reason, you may wish to place News_Root on a volume which has adequate space and which is not used to store large numbers of small files (e.g. the news item database), since this tends to increase fragmentation. In addition, you should periodically optimize the index files, as described later in the setup procedure. Finally, you can improve efficiency by using global buffers to cache the indexes of these files. (See the appendix *RMS Tuning of the News Index Files* for more information.)

5.0.4 Item files

Text files and directories containing the actual items in the News database are stored in a tree rooted by the logical name News_Device. These files, while small, may number into the hundreds of thousands, and so account for the majority of space taken up by News, requiring as much as 2 Gbytes of disk space for a full newsfeed. In addition, since most item files are small, and they are constantly being created and deleted as items are added and expired, the volume on which News_Device resides is prone to fragmentation. For these reasons, you should consider the following steps to prepare the volume which will contain News_Device:

- In order to keep fragmentation from adversely affecting other files, try to locate News_Device on a dedicated disk, or at least one on which there is relatively little creation or extension of other files going on. If the size of your newsfeed doesn't justify dedicating an entire spindle to News, consider using a package like VDDRIVER (available on some of the more popular VMS ftp sites) to create a "virtual disk" on which to place News_Device. This will isolate your item files within the container file of the virtual disk, and so will prevent them from fragmenting the rest of the spindle.

- Be sure to preallocate sufficient headers for the item files on the volume which will contain News_Device. You can estimate the number of item files which will fit on a volume by dividing its size (in blocks) by the figure:

  ((*clusters_per_file* * *cluster_size_in_blocks*) + 1)

  where *clusters_per_file* is the number of clusters taken up by an average item file (this will vary depending on which newsgroups you plan to carry; if you don't have an estimate from other sources, you can use an average size of 5 blocks as an initial rule of thumb, and divide that by the cluster size of News_Device to obtain the number of clusters used by the an average item file).

  You can preallocate headers on the volume by initializing it with the DCL command

```
$ Initialize/System/NoErase/NoHigh/Headers_=n -
            /Maximum_Files_=n dev: label
```

where *n* is the estimated maximum number of files obtained above. If the device is already in use, and you are running VMS 5.5-2 or later, you can perform an image backup of the device,

reinitialize it using the above command, and restore the image backup using the /NoInit qualifier to Backup. If you are running an earlier version of VMS you'll need to either obtain a fixed copy of Backup from DEC, or copy all files off the device, reinitialize it, and copy the files back, since versions of Backup prior to VMS 5.5-2 erroneously truncate IndexF.Sys even when the /NoInit qualifier is specified, and so remove any benefit you have gained by preallocating headers.

While not strictly required, these steps are strongly recommended if you intend to carry a large number of newsgroups. If you don't make these preparations, you may well run out of file headers on News_Device, causing News to crash, and forcing you to take these steps in a hurry at some inconvenient time. Verbum sapienti satis est.

5.0.5 Spool files

Incoming and outgoing items for exchange with other news sites are stored in spool files pending their addition to the local News database or transfer to the receiving site, respectively. The types of files and space requirements vary depending on the size of incoming and outgoing newsfeeds, and on the transport method used.

5.0.5.1 NNTP over TCP/IP or DECnet

Incoming files received by the NNTP server are placed in the News_Manager directory, unless you change the hard-coded value in NNTP_Server.C, so if you are expecting a large amount of incoming NNTP traffic, you should make sure you have enough space available on the volume which contains News_Manager. Since these NNTP batch files are larger than item files, and tend to be of a relatively constant size, they do not in themselves pose as much of a risk for fragmentation as the item files do, but it is still a good idea not to place them on a volume where many other files are being created. Outgoing NNTP traffic is handled by NNTP_Xmit using message ID list files, which are much smaller than batch files, and so do not require much space for even large newsfeeds. You specify their location when you set up the feed for a given site, and their location is more a matter of convenience than performance; a few approaches to organizing these files are discussed later.

5.0.5.2 Other methods of transport

Incoming and outgoing batches transferred by other means (e.g. file copy, UUCP) can be placed wherever you have space for them. Like incoming NNTP batches, these files can become rather large, depending on the number and size of your newsfeeds, but are fairly short lived, and remain at a fairly constant level over time. Once your feeds have been active for a few weeks, you should have a good idea how much space will be required for your spool areas, and can plan accordingly.

5.0.6 Scratch files

News uses temporary scratch files to hold items being processed (e.g. added to the local database, received by the NNTP server). These files are small and transient, so they don't impose any real burden on disk space. However, I/O to these files can become a limiting factor in processing large numbers of items, so you may wish to consider using a RAM disk for these files to speed up tasks like adding incoming batches to the local database. You can control the placement of these scratch files via the logical names News_Scratch (for files used by News.Exe) and NNTP_Scratch (for files used by the NNTP server).

5.1 Generating the required files

5.1.1 Building local copies of News images

Once you have unpacked the News distribution files as described above, read the comments in NewsBuild.Com describing how to use this procedure. In addition, you should read through the file NewsSite.H. In most cases, the options taken in the distributed version of NewsSite.H will work at your site. In some cases, however, you may need to make changes to accommodate unusual aspects of your situation. These settings fall into three categories:

- Parameters which may be changed freely to meet the needs of your particular site. This group includes CANCELMAIL, CONFIRM_NEW_GROUPS, DISPLAY_BY_NEWSGROUPS, EXP_TIME, FAST_LOAD, GRP_TIME, ITMLIFE_DEFAULTS, MAX_DAYS, MENU_SUPPORT, MIN_DAYS, NEWGROUP_REG, NEWS_BATCH_SIZE, NNTP_CACHE, NOALLOW_USER_FROM, SILENT_CLIENT_STARTUP, STARTLINE, STRIP_ERR_LINES, UNAPPROVED_NEWS, UNKCONTROLMAIL, USENET, and VMSV4/VMSV5.
- Parameters which should not be changed without a good understanding of the way in which they are interpreted, and the effects changes may have on News and related supporting code. This group includes ADDFILE_NEWSGROUPS_SIZE_LIMIT, ADDFILE_NEWSGROUPS_CROSSPOSTING_LIMIT, ADDFILE_STATISTICS_ENABLE, ARBITRON, DEBUG, DIRECTORY_ACCESS_CHECK, DIST_PATH_CHECK, DISTFILE, FULL_DIRECTORY_ACCESS_CHECK, JUNKNODOT, LOGGING, MAXUNK, MEM_CACHEMAX, NEWS_BATCH_SIZE, NEWS_BATCH_SIZE_HIGHLIM, NEWS_BATCH_SIZE_LOWLIM, NEWS_NNTP_BATCH_SIZE MEM_CHECK_QUOTA, MEM_DYNAMIC_QUOTA, MEM_RESERVE_SIZE, NEWSBATCH, NEWS_INIT, NNTP_USE_XGNOTICE, NNTP_USE_XGTITLE, NNTP_USE_XHDR, NNTP_USE_XOVER, NNTP_USE_XUPDGROUP, NOTE_RANGE_MAX, RECHECK_TIMER, SECLNM_MODE, SECLNM_TABLE, SECURE_LOGICALS, SELF_PATH_CHECK, and UWO.
- Parameters which identify the version of News you are running. These should not be changed under normal circumstances. This group includes NEWS_DDATE, NEWS_VERSION, NEWS_VDATE, and nntp_version.

If possible, you should build copies of all necessary binaries on your own system, in order to insure that the proper header files are used, and that you are linked to the proper RTLs. If you don't have access to a C compiler, consider obtaining and installing a free compiler such as GNU's gcc. If you cannot do this, or wish to get News running immediately, copies of binaries built with Multinet TCP support are provided in the News_Dist ZIP file, and copies of binaries built with Wollongong or UCX support are available for anonymous ftp at genetics.upenn.edu, in the directory Anon_Root:[ANU_News.Distribution].

After you have looked over NewsBuild.Com and NewsSite.H, execute NewsBuild.Com from the [.News_Src] subdirectory, specifying the NNTP transport you wish to use for exchanging news items with other systems. (If you do not intend to use NNTP over TCP or DECnet, you need not include any NNTP transport.) In general, NNTP over TCP/IP is faster than NNTP over DECnet, so unless you need to take advantage of the user authentication DECnet provides, you should probably choose TCP as your transport protocol. You may wish to run this procedure as a batch job, since it can take a while to compile and link all of the necessary files. When this process completes, you will have current copies of all the necessary images in the [.News_Dist] directory.

5.1.2 Creating the News_Manager directory

Create a directory in which to place News images and support files, and to serve as a temporary repository for incoming and outgoing spool files. If you have created a separate News manager account, you may wish to make this directory the login default directory for this account, but this is not necessary.

5.1.3 Installing News.Exe

Copy the News.Exe image you built to the directory in which you plan to keep it, and make it world executable. It is customary, though not strictly necessary, to put News.Exe in News_Manager. You must also install News with SysPrv, so it can create files in the News database area. (These directories should not be world accessible, or News will be unable to enforce restrictions on newsgroup access.)

In addition, if you wish to use the News lock mechanism (i.e. NewsShutDown.Exe) to prevent users from running News while certain other tasks are being performed, News.Exe must be installed with SysLck. Also, if you are using message-ID caching, you must add SysGbl to this list.

Finally, if you expect News.Exe to be in use frequently, you may may improve performance by installing it with its header resident and for shared access. All of these things are accomplished via the Install utility, using a command like:

```
$ Install Add News_Manager:News.Exe/Open/Header/Share -
          /Priv=(SysPrv,SysLck,SysGbl)
```

Remember to put a command to install News.Exe somewhere in your system startup procedure, in addition to doing it manually now.

5.1.4 Moving News.Hlb

If you place News.Hlb in News_Root, Sys$Help, or the directory in which you have placed News.Exe, News will be able to find it automatically. If you place it in another directory, you should define the logical name News_HelpLib as the file specification for the help library. News.Hlb should be world readable.

5.2 Setting up Configuration Files

A number of files are used to provide site-specific configuration information to various News images. These files are read at runtime, and can be freely changed as your configuration changes over time. Unless otherwise noted in the description of a particular file, these configuration files need not be world-readable; News will access them using SysPrv.

5.2.1 Editing the MailPaths. file

The file MailPaths. provides News with a mail address for the moderator of each moderated newsgroup. For network newsgroups, this is usually done by specifying a 'backbone' site to which all mail intended for moderators is routed, and which will sort out the precise address to which the post should go. Several of these sites are listed in the comments in the version of this file provided in the News_Dist ZIP file. The format for each entry is

newsgroup_pattern moderator_address

newsgroup_pattern - pattern specifying a set of moderated newsgroups. This pattern is matched against the name of the moderated newsgroup according to the rules used for the subscriptions field of a News.Sys entry (q.v.). In addition, if this field consists of only the keyword 'backbone', then the pattern is treated as if it were '*'. This pattern may not contain any whitespace, and is separated from the *moderator_address* field by any amount of whitespace (excluding line breaks).

moderator_address - the address of the moderator for the newsgroup specified by the newsgroup pattern. This address may contain a single '%s', into which is substituted the name of the moderated newsgroup, with each '.' converted to a '-'. This allows one to generate unique addresses using an entry whose *newsgroup_pattern* includes wildcards. For instance, the MailPaths. entry

```
bionet.* %s@net.bio.net
```

would match the newsgroup name 'bionet.announce', and would cause items posted to this group to be mailed to the address 'bionet-announce@net.bio.net'.

Any text following the '#' character, up to the end of the line, is ignored.

Entries are searched in the order encountered in the file, and the search stops with the first entry which matches the newsgroup in question.

You should edit the version of MailPaths. included in the News_Dist ZIP file, and include at least a 'backbone' entry specifying a backbone site near you, chosen from the list of sites included in the comments within the distributed version of MailPaths. Once you have modified this file appropriately for your site, you should place it in News_Manager. As you create moderated local newsgroups, News will add entries to this file automatically, so you should not have to edit it by hand after setting it up while installing News.

5.2.2 Editing the News.Aliases file (optional)

News uses this file when processing the newsgroup names in the headers of incoming items. If a newsgroup name with an entry in this file is found in the header, it is replaced with the newsgroup name(s) specified in the entry. Note that all such changes are passed on to downstream sites as well. Entries in this file have the format:

newsgroup_name replacement_list

newsgroup_name - the name of the newsgroup to which this entry applies. *Newsgroup_name* cannot contain whitespace. It is separated from *replacement_list* by any amount of whitespace. In order for *replacement_list* to be substituted into an item's Newsgroups: header, a newsgroup name in the header must exactly match *newsgroup_name*. Comparison is case-insensitive.

replacement_list - a list of newsgroup names, separated by commas, which replaces *newsgroup_name* in the Newsgroups: header of the item. *Replacement_list* cannot contain any whitespace. It may be continued across line breaks by specifying the character '\' as the last character on a line.

All characters following the character '#' up to the end of the line are ignored.

Each newsgroup in the item's original Newsgroups: header is matched against every entry in News.Aliases, but the matching is not recursive (i.e. newsgroup names substituted into the header are not then matched against News.Aliases).

Mapping of newsgroup names to their aliases is performed before other configuration files which direct actions based on newsgroup names are consulted.

If you need to remap newsgroup names in this fashion, you may set up this file in one of two ways. If you choose, you may edit the template file provided in the News_Dist ZIP file, and place the edited copy in News_Manager. You may also add, remove, or display entries in this file using the commands Add Alias, Delete Alias, and Show Alias, respectively, from within News; the file will be created when you use Add Alias for the first time. Modifying the alias file from within News causes all comments within the file to be lost.

5.2.3 Creating the News.Classes file (optional)

In additional to the automatic classes $net and $local, News can create and populate newsgroup classes on a system-wide basis, based on the contents of the file News_Manager:News.Classes. These classes cannot be modified from within News. Entries in this file should be in the following format:

class_name newsgroup_list

class_name - the name of the class to be created. If *class_name* does not begin with the character '$', which designates an unmodifiable class, then a '$' will be prepended to *class_name* to form the name of the actual class created. The class names $net and $local are reserved for automatic classes created by News, and entries specifying either of these names will be ignored. *Class_name* cannot contain whitespace, and is separated from *newsgroup_list* by any amount of whitespace.

newsgroup_list - a list of newsgroup patterns which specify the newsgroups to be added to the class. Within each pattern, the character '*' is a wildcard which matches any number of characters, while the characters '%' and '?' are wildcards which match any single character. Patterns in *newsgroup_list* may be separated by a comma or whitespace.

All text between the character '#' and the end of the entry is ignored. An entry can be continued across multiple lines by ending each line but the last with the character '\'.

If you wish to have certain newsgroups entered into specific classes by default, you should create this file yourself. For instance, a simple News.Classes file might say

```
fun_groups  rec.*,talk.*
science     bionet.*,sci.*
bargains    *.forsale
vms_stuff   comp.org.decus,comp.os.vms,vmsnet.*\
```

5.2.4 Ignoring the News.Distribution file

News.Distribution is an obsolete file which was used in the past to specify the distribution of items received from a particular site, in order to minimize redundant feeding of items to a site served both by you and the site from which you obtained the item. Since this can now be specified in News.Sys using entries of the form 'site1/site2,site3', the News.Distribution file is no longer used. It is still supported at present, however, for compatibility with old distribution schemes, and because it will, on rare occasions, allow a site to specify a particular distribution pattern which cannot be conveniently specified in News.Sys. The format for entries in this file is:

from_site to_sites

from_site - the name that the site to which this entry applies places in the Path: header of items forwarded to your site. *From_site* is compared to the first site name listed in the item's Path: header, and, if they match, then this entry is used to limit forwarding of the item. Comparison is case-insensitive. *From_site* cannot contain any whitespace, and is separated from *to_sites* by any amount of whitespace.

to_sites - a list of 'sites', separated by commas or whitespace, to which items which meet the criterion specified by from_site should be 'forwarded'. Only those entries in your News.Sys file whose *site* field name appears in this list will be applied to this item. Note that in order for the item to actually be added to the spool file specified by the News.Sys entry, it must also satisfy the criteria specified by the *subscriptions* and *distributions* field in that News.Sys entry. In other words, *to_sites* specifies a list of entries which are candidates for processing of this item; it does not mandate that the item actually be forwarded to any of these sites.

All text between the character '#' and the end of a line is ignored. An entry can be continued across multiple lines by ending each line but the last with the character '\'.

Unless you specifically need it, you should not use this file. Support for this file may be phased out in future releases of News.

5.2.5 Editing the News.Distributions file

This file contains a list of distribution keywords accepted by the local system, along with brief descriptions, in the following format:

keyword description

keyword - the keyword which would appear in the Distribution: header of an article posted to the distribution described by this entry.

description - a one-line description of the scope of the distribution denoted by the presence of *keyword* in the Distribution: header of an article. Note that this is just for information; the functional interpretation of keywords in Distribution: headers is via the *distributions* and *subscriptions* fields of entries in News.Sys.

When a user running News.Exe posts an item, all keywords specified in the Distribution: header must have entries in this file, in addition to matching the News.Sys entry for the local site. If this file does not exist, then any keyword is considered acceptable. Note that this requirement is not enforced by the NNTP server for items received by the NNTP server via the POST protocol or the IHAVE protocol.

In addition, the NNTP standard specifies that an NNTP server respond to the LIST DISTRIBUTIONS command with a list of the distribution keywords accepted at that site. News implements this by sending the contents of this file to the requesting system.

You should edit the template file provided in the News_Dist ZIP file to reflect the distributions you accept and place the edited file in News_Manager. While the News.Distributions file is, strictly speaking, optional, it is good netiquette to take the few minutes necessary to make this information available.

5.2.6 Editing the News.Sys file

News.Sys is the file which determines which newsgroups are accepted at your site, and which newsgroups are passed to sites which you feed. In addition, it specifies the type of spool files which are created for downstream sites. You should read the comments in this file, and create entries for yourself and for all sites which you will feed. Note that even if you are functioning as an 'end node' (i.e. you don't provide newsfeeds to any other sites), you will still need to send local postings back to your feed site for distribution to the net, so you should include an entry for your feed site which covers all newsgroups except those restricted to your site. Entries in News.Sys have the following format:

site[/*skip_sites*]:*subscriptions*:*flags*:\
spool_file[:*distributions*]

[The \ is used to continue the entry across a line break, and is removed by News when it reads the entry.]

site - The name of the site to which this entry applies. News will use this name to determine whether an item has already been transported through that site, by comparing the name in this field with the Path: header of the item. For this reason, it is essential that the *site* field you specify in a site's News.Sys entry be identical to the name it places in Path: headers. This may be different from the FQDN or UUCP name of that site. Note that this has no impact on the process of file transfer; you will specify the actual host name for transfer elsewhere. If News finds a site's name in an item's Path: header, the remainder of that site's News.Sys entry is ignored, and processing moves on to the next entry. When specifying the entry for your own site (used only to determine which newsgroups are accepted on the local system), you may use the keyword 'me' in place of your site's news pathname. Note that *site* need not actually specify a remote site, and dummy names are often used when constructing News.Sys entries whose only purpose is to generate spool files for some other purpose. For instance, if you wanted to archive all postings to the newsgroup vmsnet.sources, you might set up a News.Sys entry like

codesink:vmsnet.sources:M:Userdisk/[Userdir]VMSNet_Sources.Itm:*

and then, as part of your NewsAdd job, transfer these files to some place for long-term storage.

skip_sites - after the site name which identifies the entry, you may optionally specify a foreslash followed by a comma separated list of other site names. If News finds the name of any of these sites in the Path: header of an item, the remainder of this entry is ignored, and processing moves on to the next entry. As with the *site* field, the names used here must exactly match the names that the sites in question put into Path: headers. This option is designed to prevent redundant exchange of items among a set of interconnected feeds. For instance, if site *cicero* feeds both sites *virgil* and *pliny*, then site *virgil* can

assume that any item that has been through *cicero* will also have been sent to *pliny*, so it doesn't need to send those items on to *pliny* itself. In this case, the beginning of the News.Sys entry for *pliny* on *virgil* would begin

pliny/cicero:

subscriptions - this is a comma-separated list of patterns that determine which newsgroups are forwarded to the system whose News.Sys entry is being evaluated (or are accepted on the local system, if the entry being evaluated is that of the local system). The groups listed in the Newsgroups: header of the item are compared to the patterns in this field, using the following rules:

- comparison is case-insensitive
- if this field is empty, then a default value of '*' is used
- the '*' character is a wildcard, matching any number of characters in a newsgroup name, while the characters '%' and '?' are wildcards matching any one character in a newsgroup name.
- the keyword 'all', if it appears alone as a pattern or as the only text between '.' characters or the end of a pattern, is equivalent to '*' (i.e. comp.all is equivalent to comp.*, !alt.all.pictures is equivalent to !alt.*.pictures, but socall-things is not equivalent to soc.*-things).
- if a pattern consists only of the character '*' or the keyword 'all', then any newsgroup matches the pattern.
- a '!' at the beginning of a pattern indicates that newsgroups matching that pattern are not accepted by the site in question; otherwise newsgroups matching that pattern are accepted by the site in question.
- if a group in the Newsgroups: header exactly matches a pattern in the *subscriptions* field (without using wildcard matching), then the group is immediately considered to be accepted or rejected for transmission to that site (or at the local site, in the case of its News.Sys entry).
- if a group in the Newsgroups: header does not exactly match the current pattern, but does match the pattern using wildcard matching, or matches the pattern with a '.*' appended to it using wildcard matching, then the item is accepted or rejected based on the current pattern, AS LONG AS it does not exactly match any pattern appearing later in the *subscriptions* field.
- these rules are applied sequentially to each newsgroup in the item's Newsgroups: header, until a match is found which indicates that the site in question accepts a newsgroup to which the item is posted, or until the end of the Newsgroups: header is reached, in which case the item is not forwarded or accepted.

flags - this field specifies what type of spool file should be created to record items which should be forwarded to the site in question. Acceptable values for this field are:

M - the entire text of each item to be forwarded is placed in a separate file. This type of spool file is designed to accommodate forwarding of items by

mail, and is rarely used for newsfeeds. It is used now mainly to produce item files for input into news-mail gateways, UUCP map processing, and the like.

MN - same as 'M', but each line of each file is prefixed by the character 'N'. This is designed to minimize mangling of leading blanks by brain-damaged mailers.

B - the entire text of each item to be forwarded is appended to a news batch file, preceded by the one-line delimiter

#_!rnews <size_of_item_in_bytes> This continues until the maximum size specified for that site's batch files is reached (see below), in which case a new batch file is started. This format is generally used for forwarding to sites by file copy or UUCP, or for archiving items for local use.

BN - same as 'B', except that each line of of the batch file is prefixed by the character 'N'. This, too, is designed to minimize mangling of leading blanks by brain-damaged mailers, when the batch files must be transferred by mail.

N - only the message ID of the message is stored in the spool file. This type of spool file is used as input to NNTP_Xmit for NNTP transfers over TCP or DECnet links; NNTP_Xmit will retrieve the actual text of the item from the local News database at the time of transfer. Note that this means that the item must exist in the local News database at the time NNTP_Xmit is run. Since some downstream sites may accept items which you don't carry at your site, be careful not to use the /NoJunk qualifier with the Add File command when adding incoming items which might fall into this category, or you will not keep a local copy of the item in the 'junk' newsgroup long enough to forward the item. Similarly, make sure you run NNTP_Xmit before you skim these items out of your 'junk' newsgroup, which typically has a short retention time.

NX - same as 'N', but the spool file is a RMS indexed file, so multiple processes can read and write to the file simultaneously. If only the 'N' format is specified, then only one process can access the spool file at a given time (e.g you cannot be running Add File in one process to add new items to the spool file at the same time another process is running NNTP_ Xmit to forward the items to the downstream site.

If you use the 'B' format, the flag character(s) may be followed by a number, which is taken as the maximum size, in bytes, the spool file can reach before a new version is started. For instance, the flags field 'B250000' specifies the 'B' format, with a maximum size for a given batch file of 250,000 bytes. The default value for the maximum size of a batch file is specified by the compile-time constant NEWS_BATCH_SIZE in NewsSite.H.

Similarly, if you use the 'N' format, the flag character(s) may be followed by a number, which is taken as the approximate maximum number of message IDs the spool file can hold before a new version is started. For instance, the flags field 'N250' specifies the 'N' format, and that a new spool file should be started after approximately 250 message IDs have been written to the current spool file. The default value for the approximate

maximum number of message IDs per spool file is specified by the compile-time constant NEWS_NNTP_BATCH_SIZE in NewsSite.H.

If this field is empty, a default value of 'B' is used. This field is ignored in the entry for the local system.

spool_file - this is the file specification for the spool file for the system in question. It is a regular VMS file specification, except that the character ':', which is already used as a field delimiter in News.Sys, is replaced by a '/'. This field must be specified, as there is no default value. It is ignored in the entry for the local system.

distributions - This field contains a list of patterns which are matched against the contents of the item's Distribution: header, in order to determine whether the site in question accepts any of the distributions to which this item was posted. The pattern matching is according to the same rules by which the patterns in the *subscriptions* field are matched against the item's Newsgroups: header, with the additional case that trailing '.' characters in a keyword found in the item's Distribution: header are ignored. If the item is not accepted based on the patterns in this field of the News.Sys entry, or if this field is missing, then the Distribution: header is matched against the subscriptions field of the News.Sys entry, and the item is accepted or rejected based on this matching process.

Any text following the '#' character, up to the end of the line, is ignored. Entries may be continued onto the next line by ending the current line with the character '_\'. When an incoming item is processed, it is checked against every entry in News.Sys, and, if it matches patterns accepted by that site, it is added to the spool file specified (or to the local News database, if it matches the entry for the local site).

When specifying the criteria in each entry which are used to filter news items, remember that *both* the Newsgroups: header and the Distribution: header of an item must contain a match to the criteria specified in a News.Sys entry for the item to be passed to that site (or accepted locally, in the case of the entry for your site). Note that the keywords following the item's Distribution: header may match an element of the *subscriptions* or the *distributions* lists in the News.Sys entry, but that the keywords in the Newsgroups: header must match an element of the *subscriptions* list. The entry for the local site should include at least the newsgroups 'local', 'control', 'junk', and to.{site}, with the last being the name of the newsgroup upstream sites will use to send control messages specifically to you (usually {site} is the node name you use for News_Node or News_Pathname).

5.2.7 Ignoring the News.SysD file

News.SysD is an obsolete file which duplicates the function of the *distributions* field in a News.Sys entry. It is a text file with entries of the format

site_name:*dist_list*

site_name - the name of a 'site' in the News.Sys file (i.e. the *site* field of an entry). *Site_name* is separated from *dist_list* by a colon.

dist_list - a list of distribution keywords, constructed according to the same rules governing the *distributions* field of a News.Sys entry. If non-empty, *dist_list* replaces the *distributions* field of the News.Sys entry whose *site* field matches *site_name*.

All text between the character '#' and the end of a line is ignored. An entry can be continued across multiple lines by ending each line but the last with the character '\'.

While this file is still supported for backwards compatibility, it serves no function not supported by the News.Sys file, and will probably not be supported in future releases of News.

5.2.8 Editing the News_Address.Cnf file

News_Address.Cnf contains a series of rewrite rules used by News to rewrite mail addresses for outgoing items. This allows you to transparently route mail to different sites by different mechanisms, or to specify a foreign mail transport without requiring users to type the prefix with every address. Entries in this file have the format

address_pattern rewrite_rule

address_pattern - a text pattern which is matched against the mail address to be rewritten. The '*' character is a wildcard which matches any number of characters in the address, while the '?' is a wildcard which matches any one character in the address. (Note that unlike similar patterns in other News configuration files, the '%' character is a literal, not a wildcard, since it often appears in mail addresses.) The '^' character is ignored in *address_pattern*. *Address_pattern* may not contain any white space. It is separated from *rewrite_rule* by any amount of white space (excluding line breaks).

rewrite_rule - a text pattern specifying how the address is to be rewritten, if it matches the pattern specified in *address_pattern*. Characters are copied directly from the rewrite rule into the rewritten address, except that each time the string '\n', where n is an octal number <24 (octal), is encountered, the string from the original address which matched the nth '*' wildcard in *address_pattern* is substituted into the rewritten address. For example, for the following entry

```
IN%*@*.decnet.myuniv.edu \002::\001
```

the address 'IN%aeneas@virgil.decnet.myuniv.edu' would match the *address_pattern*, with the string 'aeneas' matching the first '*', and the string 'virgil' matching the second '*'. According to *rewrite_rule*, then, this address would be rewritten as 'virgil::aeneas'.

Any text following a '#' character, up to the end of the line, is ignored.

Entries are searched in the order encountered in the file, and the search stops with the first entry which matches the address to be rewritten. An address is converted to lower case before being rewritten. In the rewritten address, case is preserved for characters copied from the *rewrite_rule* pattern, but characters substituted from the original address are in lower case. Finally, note that only addresses seen by News are rewritten; addresses indirectly passed to VMS Mail (e.g. by specifying a distribution file at the To: prompt) are not.

Read the comments in the version of this file distributed in the News_Dist ZIP file, and, if you decide it would be useful to you, edit it and place the edited copy in News_Manager. If this file does not exist in News_Manager, then News uses the following default rewrite rules:

1. If the logical name SMTP_MailShr is defined or the file Sys$Share:SMTP_MailShr.Exe exists, then the following rewrite rules are used:

```
smtp%*          smtp%\001
psi%*           psi%\001
@*              @\001
*::*            \001::\002
*@*             smtp%"\001@\002"
*               \001
```

2. If the conditions for (1) are not true, but the logical name ST_MailShr is defined or the file Sys$Share:ST_MailShr.Exe exists, then the following rewrite rules are used:

```
st%*            st%\001
psi%*           psi%\001
@*              @\001
*::*            \001::\002
*@*             st%"\001@\002"
*               \001
```

3. If the conditions for (1) and (2) are not true, then the following rewrite rules are used:

```
in%*            in%\001
psi%*           psi%\001
@*              @\001
*@*             in%"\001@\002"
*::*            in%"\002@\001"
*               \001
```

5.2.9 Editing the News_Post.CC file (optional)

News_Post.CC contains entries in which newsgroup names are paired with mail addresses, so that when an item is posted to the newsgroup, a copy is also mailed to the address specified. Note that the posting also goes out via the normal mechanism specified for the newsgroup, so using this file is not equivalent to using the /MailList qualifier when creating a newsgroup. Entries in this file have the following format:

newsgroup_pattern address_list

newsgroup_pattern - a pattern matched against the newsgroup names in the Newsgroups: header of the item. The character '*' is a wildcard matching any number of characters, and the characters '*%*' and '?' are wildcards matching any one character. The pattern may not contain any whitespace. Comparison is case-insensitive.

address_list - a list of mail addresses to which a copy of the item is forwarded, when a newsgroup name in the Newsgroups: header matches *newsgroup_pattern*. The list is converted to lower case. It is separated from *newsgroup_pattern* by any amount of whitespace, and may contain whitespace within the list (though multiple spaces will be compressed to a single space when the list is used), but it may not cross a line boundary.

All text between a '#' character and the end of the line is ignored. Each newsgroup name in the item's Newsgroups: header is matched against every entry in this file.

In order to use this capability, edit the template file provided in the News_Dist ZIP file, and place the edited copy in News_Manager.

5.2.10 Editing the News_Post.Defaults file (optional)

This file is used to supply default values for the Distribution: and Followup-To: headers for items posted to certain newsgroups. This allows you to direct followups and expand distribution of items posted to certain newsgroups automatically. The directives in this file are applied both to items posted locally and to items posted to the NNTP server by remote clients using the NNTP POST command, but not to items received by the NNTP server as part of an IHAVE transaction, or to items added to the local News database via the Add File command. The format of entries in this file is:

newsgroup_pattern
    header_keyword default_words
    [*header_keyword default_words*]

newsgroup_pattern - a pattern matched against the newsgroup names in the Newsgroups: header of the item. The character '*' is a wildcard matching any number of characters, and the characters '*%*' and '?' are wildcards matching any one character. Comparison is case-insensitive. When the matching is performed, a '*' is appended to the beginning and the end of

newsgroup_pattern. If *newsgroup_pattern* is the keyword 'default', it matches any newsgroup name. *Newsgroup_pattern* must begin in column 1.

header_keyword - the name of the header for which default values follow. Two keywords are recognized: 'distribution:' and 'followup-to:'. Recognition is case-insensitive. The line(s) containing *header_keyword* must begin with whitespace.

default_words - a comma-separated list of default values to be added to the header specified by *header_keyword* of any item containing a newsgroup in its Newsgroups: header which matches *newsgroup_pattern*. The text of *default_words* is converted to lowercase and added to the words entered into the header in question by the user. If a word is already present in the item header, it is not duplicated.

All text between a '#' character and the end of the line is ignored. Each newsgroup name in the item's Newsgroups: header is matched against every entry in this file.

If you find this useful, edit the template file provided in the News_Dist ZIP file, and place the edited copy in News_Manager.

5.2.11 Creating the NNTP message ID cache file (optional)

Message ID caching allows the NNTP server to track items coming in via NNTP feeds, and reject duplicate items when a remote site offers to transfer them. If you do not use caching, the server will accept all items in newsgroups carried on the local system, and duplicates will not be rejected until News tries to add them to the local database. Caching is often useful if you have incoming NNTP feeds from several sites with duplicate coverage. On the other hand, if you have only a single feed for any given newsgroup, caching is usually of limited value, since under normal circumstances most items will not be offered for transfer more than once. If you choose to use Message ID caching, you will need to create and initialize the cache file as follows:

```
$ Run Sys$System:SysGen
  Create News_Root:News.Cache/Size=1400
  Exit
$ Set Process/Privilege_=(SysLck,SysGbl)
$ Run [News_Dist]Cachem
Z ! to reset the cache
E ! Exit
```

5.2.12 Editing the NNTP_Access.News file (optional, but recommended)

If you plan to run an NNTP server, this file can be used to specify the type of access allowed to remote nodes when they connect to your server. Entries in this file are of the form:

site_pattern read/transfer_access [*post_access* [*newsgroup_list*]]

site_pattern - a pattern which is compared to the name of the remote host to determine whether this entry should be applied to the current connection. For DECnet connections, the remote nodename obtained from the NCB is used; for TCP connections, the remote host name is used if it is available, otherwise the IP address in dotted decimal form is used. The character '*' is a wildcard which matches any number of characters in the remote host name, while the characters '%' and '?' are wildcards which match any single character in the remote host name. If site pattern is the keyword 'default', then the entry will be applied to any host whose name does not match another entry in this file. Comparison is case-insensitive. *Site_pattern* cannot contain any whitespace, and is separated from *read/transfer_access* by any amount of whitespace.

read/transfer_access - a single keyword indicating the read and transfer access permitted to hosts matching this entry. Only the first character of the keyword is significant. It is interpreted as follows:

x - host is permitted to transfer items to the local server via the NNTP IHAVE protocol
v - host is permitted only to read items on the local server
r - host is permitted to both read and transfer items
any other character - host is permitted neither read nor transfer access

Read/transfer_access cannot contain any whitespace, and is separated from *post_access* by any amount of whitespace.

post_access - a single keyword indicating the posting access permitted to hosts matching this entry. Only the first character of the keyword is significant. It is interpreted as follows:

p - host is permitted to post items to this server using the NNTP POST protocol
any other character - host is not permitted to post items to this server

If this field is absent, post access is not allowed. *Post_access* cannot contain any whitespace, and is separated from *read/transfer_access* by any amount of whitespace.

newsgroup_list - a comma-separated list of newsgroup patterns which specify the newsgroups to which the remote host is allowed access. If the first character of a pattern is '_!', then the remote host is denied access to any newsgroup whose name matches the rest of the pattern. Within each pattern, the character '*' is a wildcard which matches any number of characters, while the characters '%' and '?' are wildcards which match any single character. For the purpose of comparison, each pattern has the characters '.*' appended to it. Comparison is case-insensitive. *Newsgroup_list* cannot contain any

whitespace. This list determines whether a given newsgroup name appears in the response to a NNTP LIST command, and whether the remote host can access a newsgroup via the NNTP GROUP command. In addition, if the remote host attempts to access an item directly via the message ID, it succeeds if that item is found in a newsgroup to which that host has access (regardless of whether the item also appears in other newsgroups to which that host does not have access). If this field is empty, then it is treated as if it were '*'.

If this file is not present, any host connecting to your server is given read, transfer, and post access to all newsgroups carried at your site.

If you would like to restrict access to specific hosts or functions, edit the template file provided in the News_Dist ZIP file, and place the edited copy in News_Root (not News_Manager; this is done for historical reasons).

5.3 Creating the News manager account and identifier

In a typical setup, News relies on periodic batch jobs to add new items fed to you by upstream sites to the local database, and to feed items to downstream sites. These jobs need to run under an account which meets the following criteria:

- It must have full (RWED) access to all News files, either by ownership or by possessing appropriate privileges (e.g. SysPrv).
- It must have sufficient disk quota available to handle incoming batch and mail files, outgoing spool files, and log files.
- If you plan to use message ID caching (described above), it will also need SysLck and SysGbl privileges.
- It must have access to any incoming mail which must be added to the News database (e.g. mailing list subscriptions which are gatewayed into local newsgroups). This can be accomplished by subscribing this account to the mailing lists, and sorting them into the proper newsgroups by searching for the list name in message headers, or you can give this account access to the mail folders of the accounts which are subscribed to the groups (this will probably require privileges; the exact requirements depend on your local configuration).
- If you plan to feed other sites by DECnet copy, it will need proxy access to the destination directories of downstream sites.

You may choose to perform these tasks under an existing account, or you may choose to create an account specifically for News. Since News uses this account only to run the periodic batch jobs, if you create a new account, you may disable all access except batch mode. Throughout this manual, this account is called the NewsMgr account.

You will also need to create a rights identifier which will be used by News to determine whether a user is allowed to execute News management commands. This identifier should be granted to all accounts who will perform management tasks, including the account described above. A separate rights identifier is used, rather than just checking the UIC identifier of the NewsMgr account, so that it can be granted to a number of users.

5.4 Setting up the News logical name environment

In order to provide site-specific information to News, the following logical names need to be defined. You will probably want to put these definitions into a DCL procedure to be executed manually now, and as part of system startup in the future. Several of these logical names are considered 'secure' by News, that is, News will use only translations obtained in accordance with the rules specified in NewsSite.H for secure logical name translation. (As distributed, News requires that these logical names be defined in the system logical name table in executive mode.) The section describing a particular logical name indicates whether News considers it 'secure'.

5.4.1 News_Address

The translation of this logical name is used as the name of the local host when constructing mail addresses, and when checking whether the current user is allowed access to or moderator privileges for a newsgroup. The rules for translating secure logical names are followed when attempting to translate News_Address. If this is not successful, the translation obtained for News_Node as specified above is used.

5.4.2 News_Cache (optional)

If this logical name is defined, News and the NNTP server will attempt to use message ID caching. If you choose to use message ID caching, you must first set up the cache file as described above. This logical name must be defined in executive mode in the system logical name table. If you wish to change this, you must edit the file Cache.C directly; the rules for secure logical names set in NewsSite.H are not respected here. (This is because the cache routines are linked into some support programs which do not contain the News logical name translation code.)

5.4.3 News_Device

This logical name must translate to a rooted directory specification, under which News will create directories in which it stores the actual text of items in the local database. It should point to an area with adequate space (a full Usenet/alt.*/vmsnet.* feed can take up ~2 Gbytes), where performance will not be adversely affected by frequent I/O and by the fragmentation which results from creation and deletion of many small, short-lived files as items are added and expired. Many sites choose to dedicate an entire disk to news items, or to use a virtual disk in a large container file in order to preserve the rest of the disk from fragmentation. In addition, you may need to extend IndexF.Sys on this disk to allow for the large number of small item files which will exist at any given time. You will probably need about 25000 file headers per day for a full feed including alt groups. This number will increase as the volume of news increases. It has been customary for News_Device to be the rooted form of the directory to which News_Root points, but this often adversely affects performance, since all News I/O is then concentrated on one disk, and the large index files often become fragmented amongst the item text files. For best performance, then, you should place News_Root on one

spindle and News_Device on another. You must create the root directory manually; News will create and delete subdirectories as necessary. When attempting to translate this logical name, the rules for translation of secure logical names are used.

5.4.4 News_GMT_Offset

This logical name specifies the offset of your local time from GMT (UTC), as a string with the format hh:mm:ss, and its translation is used in converting between GMT in headers and local time. The string may begin with '+' or '-', indicating whether the offset should be added to or subtracted from UTC to obtain the local time; if the sign is absent, '+' is assumed. This logical name is normally defined in the system logical name table, although translation is via the normal lookup process, so definitions in the process, job, or group logical name table are used in preference to the system-wide value.

5.4.5 News_GroupFile_BufferCount (optional)

If this logical name is defined, its translation should be a decimal number, which is used as the number of buffers RMS should use when the newsgroup index file is opened by News.Exe or the NNTP server (rab$b_mbf), up to a maximum of 127. This logical name is used principally to obtain better efficiency during Add File and Skim processing, and, since it operates on a per-process basis, can be defined in the process logical name table of processes performing these tasks.

5.4.6 News_GroupFile_DFW (optional)

If this logical name is defined, News.Exe and the NNTP server turn on the deferred write option when opening the newsgroup index file (fab$m_dfw bit in fab$l_fop). By default, RMS uses a very conservative buffering strategy, writing any changed buffer to disk immediately. Deferred write changes that behavior and defers the writing of a buffer until that buffer is needed for other blocks. However, if the process should terminate after the buffer has been changed, but before it is written to disk, the changes are lost, possibly leaving the disk file in an inconsistent state. Therefore, you should use this option ONLY if you are prepared to rebuild your newsgroup index file. If you don't know what this means, don't do it. The performance gains are not worth the pain of learning how to rebuild these files. This logical name is used principally to obtain better efficiency during Add File and Skim processing, and, since it operates on a per-process basis, can be defined in the process logical name table of processes performing these tasks.

5.4.7 News_GroupFile_FastDelete (optional)

If this logical name is defined, the RMS fast delete option is turned on when News.Exe or the NNTP server opens the newsgroup index file. This capability is provided for testing future revisions of newsgroup index file processing. Its use is discouraged at present, as it has not been tested on the current version of News. This logical name is used principally to obtain better efficiency during Add File and Skim processing, and, since it operates on a per-process basis, can be defined in the process logical name table of processes performing these tasks.

5.4.8 News_GroupFile_Global_BufferCount (optional)

If this logical name is defined, its translation should be a decimal number, which is used as the number of global buffers RMS should use when the newsgroup index file is opened by News.Exe or the NNTP server (fab$w_gbc). If the file has a global buffer count alrady associated with it (e.g. via Set File), News uses the greater of this value and the value specified by News_GroupFile_Global_BufferCount. This logical name is used principally to obtain better efficiency during Add File and Skim processing. However, when a file is shared by multiple processes, the global buffer count used for a file is determined by the first stream to connect to the file, so this logical name should be defined in a way that is available to all processes which might be running News.Exe or the NNTP server when you wish to turn this optimization on.

5.4.9 News_GroupFile_RTV (optional)

If this logical name is defined, when News.Exe or the NNTP server opens the newsgroup index file, all retrieval pointers are mapped to the window buffer (fab$b_rtv = 255). This technique, sometimes called using 'cathedral windows', eliminating the 'window turns' which may occur if the file is fragmented. This logical name is used principally to obtain better efficiency during Add File and Skim processing, and, since it operates on a per-process basis, can be defined in the process logical name table of processes performing these tasks.

5.4.10 News_HelpLib (optional)

If this logical name is defined, it must translate to the file specification of the help library for News commands. When looking for its help library, News first tries to open the library specified by News_HelpLib. If this fails, it then looks for a library named News.Hlb, first in News_Root, then in Sys$Help, and finally in the same directory as News.Exe. This logical name is normally defined in the system logical name table, although translation is via the normal lookup process, so definitions in the process, job, or group logical name table are used in preference to the system-wide value.

5.4.11 News_HistoryFile_BufferCount (optional)

This logical name is similar to News_GroupFile_BufferCount (q.v.), except that its translation is used when the history file is opened.

5.4.12 News_HistoryFile_DFW (optional)

This logical name is similar to News_GroupFile_DFW (q.v.), except that its translation is used when the history file is opened. Before using this logical name, see the cautionary notes under the description of News_GroupFile_DFW.

5.4.13 News_HistoryFile_FastDelete (optional)

This logical name is similar to News_GroupFile_FastDelete (q.v.), except that its translation is used when the history file is opened.

5.4.14 .hl 3 News_HistoryFile_RTV (optional)

This logical name is similar to News_GroupFile_RTV (q.v.), except that its translation is used when the history file is opened.

5.4.15 News_ItemFile_BufferCount (optional)

This logical name is similar to News_GroupFile_BufferCount (q.v.), except that its translation is used when the item index file is opened.

5.4.16 News_ItemFile_DFW (optional)

This logical name is similar to News_GroupFile_DFW (q.v.), except that its translation is used when the item index file is opened. Before using this logical name, see the cautionary notes under the description of News_GroupFile_DFW.

5.4.17 News_ItemFile_FastDelete (optional)

This logical name is similar to News_GroupFile_FastDelete (q.v.), except that its translation is used when the item index file is opened.

5.4.18 News_ItemFile_Global_BufferCount (optional)

This logical name is similar to News_GroupFile_Global_BufferCount (q.v.), except that its translation is used when the item index file is opened.

5.4.19 News_ItemFile_RTV (optional)

This logical name is similar to News_GroupFile_RTV (q.v.), except that its translation is used when the item index file is opened.

5.4.20 News_Locked_Command

When you run NewsShutdown.Exe, if it successfully obtains an exclusive lock on the News resource, it spawns a subprocess and passes the translation of this logical name to the subprocess to be executed as a DCL command. Translation of this logical name is via the normal lookup process, and it usually defined in the process table just before executing NewsShutDown.Exe.

5.4.21 News_Locked_Wait_Minutes

After NewsShutDown.Exe enqueues the request for an exclusive lock on the News database, it will wait for the number of minutes specified by the translation of this logical name before giving up and exiting with the status SS$_CANCEL, without executing News_Locked_Command. If this logical name is not defined, then NewsShutDown.Exe will wait one minute before giving up. Translation of this logical name is via the normal lookup process, and it usually defined in the process table just before executing NewsShutDown.Exe.

5.4.22 News_Manager

This logical name should translate to the name of the directory in which most News configuration and support files (and, by convention, News.Exe) reside. It is customary, though not absolutely necessary, to make this directory the login directory for the NewsMgr account. When attempting to translate this logical name, the rules for translation of secure logical names are used.

5.4.23 News_Manager_Dev

This logical name should translate to a rooted directory specification, under which the spool directories for outgoing news feeds are customarily created (as specified in the *spool-file* field of entries in the News.Sys file). Unless you follow this convention in your News.Sys entries, this logical is not used by any image distributed with News, except the multithreaded NNTP server which runs under UCX TCP. Sites often use the directory specified by News_Manager as the root of this tree, but this is not strictly necessary. You must create the root directory and any subdirectories manually; News will not automatically create directories in response

to News.Sys entries. This logical name is normally defined in the system logical name table, although translation is via the normal lookup process, so definitions in the process, job, or group logical name table are used in preference to the system-wide value.

5.4.24 News_Mgr_ID

This logical name should translate to the name of the rights identifier granted to News managers to allow them management privileges. When attempting to translate this logical name, the rules for translation of secure logical names are used.

5.4.25 News_NetPost (optional)

If this logical name is defined, its translation is used as the name of a rights identifier which must be held by a user in order for them to post to any network newsgroup (i.e. any neswsgroup not marked as local by using the /Local qualifier with the Create Newsgroup or Set Newsgroup command). Note that a user holding this identifier will be permitted to post, regardless of whether that user also holds the identifier specified by News_NoNetPost. Conversely, a user who does not hold this identifier will not be permitted to post, even if that user does not hold the identifier specified by News_NoNetPost. When attempting to translate this logical name, the rules for translation of secure logical names are used.

5.4.26 News_NNTP_Log (optional)

If this logical name is defined, its translation is used to specify a file to which a summary of each NNTP server session will be written. This summary contains the remote site's name, the newsgroups accessed, and the total number of items read, posted, accepted, and rejected in each newsgroup. When attempting to translate this logical name, the rules for translation of secure logical names are used.

5.4.27 News_NNTP_Protocol (optional)

This logical name should be defined only under the circumstances mentioned below for News_NNTP_Server. It should translate to the name of the transport protocol you specified when compiling News. The syntax for transport names is identical to that used with the /NetProtocol qualifier to the News command. When choosing the transport protocol to use for a given session, News uses the following order of precedence:

1. protocol supplied as a value to the /NetProtocol qualifier on the command line
2. protocol stored in the NewsRC file from a previous session
3. protocol obtained by translating the logical name News_NNTP_Protocol

This logical name is normally defined in the system logical name table, although translation is via the normal lookup process, so definitions in the process, job, or group logical name table are used in preference to the system-wide value.

5.4.28 News_NNTP_Server (optional)

This logical name should be defined only if you plan to run News with a local database in some sessions, and as an NNTP client *in separate sessions*. (This is distinct from the mechanism for creating served newsgroups in a local database, which is described elsewhere in this manual.) It should translate to the name of the NNTP server you plan to use as your default server.

If you're using DECnet as the transport protocol, then this should be the DECnet nodename, or the node number in the format *area**1024+*node* (*area**1024 is optional if the server is in the same area as your system).

If you're using TCP, then this should be the host name of the server, or its IP address in dotted decomal form. When choosing the server to which to connect for a given session, News uses the following order of precedence:

1. server name supplied as a value to the /NetServer qualifier on the command line
2. server name stored in the NewsRC file from a previous session
3. server name obtained by translating the logical name News_NNTP_Server

This logical name is normally defined in the system logical name table, although translation is via the normal lookup process, so definitions in the process, job, or group logical name table are used in preference to the system-wide value.

5.4.29 .hl 3 News_NNTP_Server_Always_Spool (optional)

By default, the NNTP server adds items received from remote clients via POST transactions directly to the local database, rather than spooling them in incoming batch files, unless the item contains a Control: header. If this logical name is defined, all items received via POST transactions are spooled to batch files, and local processing is deferred until the batch file is read using the Add File command. This logical name is used only by the NNTP server, so it can be defined in any way in which it is available to the process running the NNTP server.

5.4.30 News_NNTP_Server_Batch_Flush_Disable (optional)

By default, when the NNTP server flushes data added to incoming batch files to disk, it uses the fflush( ) and fsync( ) C RTL routines. If this logical name is defined, and its translation is an odd decimal number, or a string whose first character is one of TtYy, then the batch file is closed instead of being flushed, and is reopened when the next item arrives. If it translates to any other value, it has no effect. This logical name is used only by the NNTP server, so it can be defined in any way in which it is available to the process running the NNTP server.

5.4.31 News_NNTP_Server_Batch_Size (optional)

If this logical name is defined, its translation should be a decimal number specifying the maximum size (in bytes) of incoming news batch files. The minimum, maximum, and default values for this size are specified by the compile-time constants NEWS_BATCH_SIZE_LOWLIM, NEWS_BATCH_SIZE_HIGHLIM, and NEWS_BATCH_SIZE, respectively, in NewsSite.H. This logical name is used only by the NNTP server, so it can be defined in any way in which it is available to the process running the NNTP server. It affects the process of item transfer only; posting of items by remote clients uses NEWS_BATCH_SIZE_LOWLIM at all times.

5.4.32 News_NNTP_Server_Leave_Spool_Open (optional)

By default, the NNTP server leaves incoming batch files open until the maximum size is reached, or until 2 minutes have elapsed since the last item was received. If this logical name is defined, and its translation is an even decimal number, or a string whose first chracter is not one of TtYy, then the batch file is closed after each item is received. If its translation is an odd decimal number, or a string whose first character is one of TtYy, then the default behavior is unchanged. This logical name is used only by the NNTP server, so it can be defined in any way in which it is available to the process running the NNTP server.

5.4.33 News_Node

This logical name specifies the Usenet name for the local node, and is used to identify it in Usenet headers (e.g. Relay-Version:, X-News:). In addition, it serves as a default value for the name of the local host in Path: headers if News_Pathname is not defined, and in mail addresses if News_Address is not defined. If this logical name is undefined, another logical name may be used, depending on the NNTP transport you specified when building News:

NNTP transport	Logical Names (in order checked)
Multinet	Multinet_SMTP_Host_Name
	Multinet_Host_Name
TCPware	TCPware_SMTP_From_Domain
	TCPware_DomainName
UCX	UCX$Inet_Host
Wollongong	ARPANet_Host_Name
All	Internet_Host_Name
	Sys$Cluster_Node
	Sys$Node

Attempts to obtain a translation for this logical name and any of its fallbacks follow the rules for secure logical names. Finally, if no logical name is found, News prints an error message and exits.

5.4.34 News_NoNetPost (optional)

If this logical name is defined, its translation is used as the name of a rights identifier which, when held by a user, indicates that the user is not allowed to post items to network newsgroups. When attempting to translate this logical name, the rules for translation of secure logical names are used.

5.4.35 .hl 3 News_Organisation (or News_Organization) (optional)

If this logical name is defined, its translation is used in constructing the Organization: header of outgoing items. This logical name is normally defined in the system logical name table, although translation is via the normal lookup process, so definitions in the process, job, or group logical name table are used in preference to the system-wide value.

5.4.36 News_Pathname (optional)

If this logical name is defined, its translation is used as the name for this site in Path: headers, and when looking for the local site's entry in News.Sys, instead of the translation of News_Node. When attempting to translate this logical name, the rules for translation of secure logical names are used.

5.4.37 News_Post_CC (optional)

News_Post_CC is an alternative method for specifying a file to perform the function of News_Manager:News_Post.CC (see description of this file above). It should translate to the file specification of a file in the format specified in the description of News_Post.CC. The file to which News_Post_CC points is used in addition to News_Manager:News_Post.CC, but supersedes the user's private file Sys$Login:News_Post.CC, if it exists. Note also that News_Post_CC is translated using the normal sequence of logical name tables and access modes, so a process-private definition will supersede a system-wide one.

5.4.38 News_Root

This logical name must translate to the directory in which the index files News uses to keep track of items in the local database will reside. These are large RMS indexed files which are frequently accessed, so it is a good idea to put them on a disk with adequate space and I/O bandwidth. You must create this directory manually. When attempting to translate this logical name, the rules for translation of secure logical names are used.

5.4.39 News_*remhost*_Task (optional)

When attempting to establish a DECnet connection to node *remhost*, News will try to translate the logical name News_*remhost*_Task (e.g. when connecting to node VIRGIL, News will try to translate News_VIRGIL_Task), and, if successful, will use this translation as the name of the DECnet task to which to connect. This logical name is normally defined in the system logical name table, although translation is via the normal lookup process, so definitions in the process, job, or group logical name table are used in preference to the system-wide value.

5.4.40 News_Stop (optional)

If this logical name is defined in the system logical name table in executive mode, then each currently active News and NNTP server session will exit when it finishes executing the current command (or after 30 minutes in any case), releasing any locks they may have held in the News index files. (**Exception:** Multithreaded NNTP servers will not respond to this logical name while hibernating between connections, so the server process may have to be stopped explicitly by the process which requires exclusive access to the News database.) This is an older method of obtaining exclusive access to the News database, and NewsShutDown.Exe is the preferred method of accomplishing this in the current release of News. For more information and an example of performing a task requiring exclusive access to the News database, see the section in this manual on optimizing News index files.

5.4.41 News_Timezone

This logical name specifies the time zone notation to be added to Date: headers of outgoing items. It should translate to a standard time zone abbreviation (e.g. EST, PDT) acceptable according to RFC822. If this logical name is undefined, depending on the NNTP transport you specified when you built News, the translation of the following logical name is used:

NNTP transport	**Logical Names**
NNTP Transport	Logical Names
All	PMDF_Timezone
Multinet	Multinet_Timezone
TCPware	TCPware_Timezone
Wollongong	WIN$Time_Zone

These logical names are searched in the order given, and the search stops at the first match. If no appropriate logical name is defined, "GMT" is used as the time zone. This logical name is normally defined in the system logical name table, although translation is via the normal lookup process, so definitions in the process, job, or group logical name table are used in preference to the system-wide value.

5.4.42 News_Usage_Anonymous (optional)

If this logical name is defined, then all entries written to the usage log (see News_Usage_Log logical name below) do not contain the username. Instead, the value of the Account field of the SysUAF record for that username is used. When attempting to translate this logical name, the rules for translation of secure logical names are used.

5.4.43 News_Usage_Log (optional)

If this logical name is defined, its translation specifies a file to which a summary of each News session will be written. (News.Exe must have write access to the file, of course.) The summary contains the username (if News_Usage_Anonymous (q.v.) is not defined), the newsgroups accessed, and the number of items read and posted in each newsgroup. When attempting to translate this logical name, the rules for translation of secure logical names are used.

5.4.44 NNTP_Disk_Danger and NNTP_Disk (optional)

If NNTP_Disk_Danger is defined, it should translate to the number of free blocks which must be present on the devices to whose names the logical name NNTP_Disk translates, in order for the NNTP server to accept incoming items. (NNTP_Disk may translate to a single device name or a comma-spearated list of device names.) If the available free space on any device listed in NNTP_Disk falls below this level, items are rejected and remote sites are asked to try again later. In attempting to translate NNTP_Disk_Danger, the rules for secure logical names are used. NNTP_Disk is translated according to the usual order of access modes and logical name tables.

5.4.45 NNTP_More_To_Offer

When NNTP_Xmit reaches the maximum number of message IDs you have allowed it to offer to a remote system, if there are more message IDs remaining in that system's spool file, NNTP_Xmit defines this logical name in the system logical name table in executive mode, with a translation of "YES". You can check for the presence of this logical name in the DCL procedure which invoked NNTP_Xmit, if you need to determine whether more items need to be offered to the remote system. You are responsible for deassigning this logical name once you have no more use for it.

5.4.46 NNTP_Scratch (optional)

If this logical name is defined, the NNTP server will use its translation as a scratch area for temporary files created while processing incoming items. The normal order of access modes and logical name tables is used for translating this logical name. If it is undefined, Sys$Scratch is used instead.

5.5 Setting up required mail addresses

5.5.1 Administrative contact address for News matters

Any site exchanging news in the Usenet hierarchies is required to route mail to the address usenet@*site* to the person responsible for managing news usage at that site. In effect, usenet@*site* is to news what postmaster@*site* is to mail. Even if you don't exchange news feeds with other sites, if users from your site post to network newsgroups, readers outside your organization may try to contact this address to report t echnical problems (e.g. the From: addresses in posts from your site are unreplyable) or breaches of netiquette (e.g. a user at your site is posting chain letters). You should create a mail forwarding address or define a system-wide logical name which will direct incoming mail with this address to the appropriate person. For instance, you can do this using VMS Mail's Set Forward command:

```
$ Mail
MAIL> Set Forward/User=USENET myaddr
MAIL> Exit
```

where *myaddr* is the valid mail address to which you want incoming messages to USENET directed.

5.5.2 Addresses for incoming news feeds

If you are accepting incoming news feeds by mail, when you make arrangements with the site that is feeding you, you will need to supply them with an address to which to mail the items they're sending you. Traditionally, this address has been either rnews@*site* or newsmail@*site*. If you are receiving feeds via mail, it will minimize confusion if you create mail forwarding addresses or define system-wide logical names to route mail to these addresses to a place where your periodic maintenance batch job will find the incoming items.

5.6 Providing a foreign command to run News

News.Exe is written to be invoked by a foreign command from DCL, and will parse its own command line. You should add a line to your system-wide login procedure which sets up this command, e.g.

```
$ News == "$News_Manager:News.Exe"
```

If you plan to use news as a local reader in some sessions, and as an NNTP client in separate sessions (e.g. for different hierarchies), then you should define two separate commands to invoke News, e.g. .indent 2

```
$ News == "$News_Manager:News.Exe/NoNetServer" ! for local sessions
$ RemoteNews ==
        "$News_Manager:News/NetServer_=site/NetProtocol_=protocol"
```

where *site* in the second command is the name of the NNTP server, and *protocol* is the name of the NNTP transport protocol you specified when you built News. Note that you may specify

the NNTP server name transport protocol in other ways (see the sections describing the logical names News_NNTP_Server and News_NNTP_Protocol for more information), but you must place at least the qualifier /NetServer on the command line if you want to invoke News as an NNTP client when the logical name News_Root is properly defined.

If you have multiple NNTP servers available to you, you should in general use one server consistently for a given set of newsgroups, since the item number assigned to a newsitem may vary from server to server, so the record of items read which is stored in you NewsRC file is valid only for that server. You can use different servers to read different newsgroups, and News will maintain different NewsRC files for each server (names Sys$Login:NewsRC.*server*, where *server* is the name of the NNTP server to which that file is tied, with all '.' characters in the server name replaced by the character '-'). In addition, News stores the name of the NNTP server and the transport protocol used to connect to that server in the NewsRC file, and will not use a NewsRC file which specifies a different server name than the one to which it is currently connected.

5.7 Running News for the first time

Once you have defined the News logical names, created the necessary directories, and set up the required files, you should login to an account to which the identifier specified by News_Mgr_ID has been granted, and invoke News. News will pause, and then print the message

NEWS - first time installation...

and will proceed to create its index files. You will then be left at an empty newsgroup directory screen. Using the Create Newsgroup command, you should create the newsgroups 'control', 'local', and 'junk', specifying the /Local qualifier for each to insure that these groups are not accidentally deleted later in response to network control messages.

Note

The /Local setting does not affect whether items in these groups are propagated to the net; you must be sure that the News.Sys entries for downstream sites do not include any groups whose contents you do not wish to forward.

News will use the 'junk' and 'control' newsgroups for maintenance tasks; the 'local' newsgroup is available for local postings. News is now ready for use.

5.8 Creating newsgroups on your system

Now that News is up and running, you'll need to create the newsgroups you plan to carry on your system. By default, News will not create new groups to accommodate incoming items which have unknown newsgroup names in their Newsgroups: header. This can be overridden by specifying the /CreGrp qualifier to the Add File command, but this is generally a bad idea, since it will result in the creation of many bogus newsgroups in response to typographic errors or pranks in the headers of incoming items. In fact, it's a good idea in general to manually review any group creation or deletion, so that you don't suddenly find your system in a mess because of bad network control messages.

About now, you may be thinking, with a touch of panic, "I'm planning to carry 2500 newsgroups, and I have to create every one of them **BY HAND???**" Relax - all is not lost. Since most of the newsgroups you're carrying probably fall into established hierarchies, you can take advantage of a mechanism known as a checkgroups message to streamline newsgroup creation.

A checkgroups message is a posting which contains a listing of newsgroup names and titles (one line descriptions of the groups' purposes), as well as information identifying moderated newsgroups. When this posting is fed to News as a control message, it will automatically create a DCL procedure containing the commands necessary to create the groups listed in the posting, and will mail this procedure to the local address USENET. A template for a checkgroups message which covers the Usenet groups is regularly posted to the newsgroup news.announce.newgroups (see if you can borrow temporary access to news on a system which is already up and running). If you are setting up your system to match the newsgroups carried on another system running News, you can ask someone on that system to execute the News command Extract/Directory=NewsGroupTitles to obtain a list of newsgroups you can use as the basis for a checkgroups message. In addition, the administrative newsgroups for various other hierarchies often carry checkgroups postings, or lists of newsgroups which can easily be edited into a checkgroups message, following the pattern of the one from news.announce.newgroups. Using this posting as an example, here is how to create the groups you want on your system:

1. Extract the template posting into a file.
2. Cut off all lines above and below the lines marked "cut to this line"
3. Change the Newsgroups: header so that the item will be added to the 'control' newsgroup you created earlier (and no other groups).
4. Be certain that the Distribution: header includes only distributions limited to your system.
5. Add an Approved: header to the list of headers, such as
   Approved: yourname@yoursite (the address you use is irrelevant).
6. Go through the list of newsgroups, and remove those you do not want to create on your system.
7. Save the edited message to a file.
8. Bypass your News profile settings by issuing the command
   $ Define/User NewsRC _NL: This is a useful practice in general when invoking News just to perform some management task, as it eliminates unnecessary processing at startup. It is required when processing checkgroups messages, because some profile settings (notably FASTLOAD) cause unpleasant side-effects during checkgroups processing.
9. Invoke News from an account holding the News_Mgr_ID identifier. Then execute the command
   Add File/GenID *filename* where *filename* is the name of the edited checkgroups file. News will generate a DCL procedure to create the necessary newsgroups and set the titles and moderation status properly. This procedure will be mailed to the local address USENET. Alternatively, if you specify the /Execute qualifier with the Add File command, News will create the newsgroups immediately.

10. When the mail message arrives, extract it to a file, check it over, edit off the mail headers, and execute it as a DCL procedure from an account with the News_Mgr_ID identifier. News will set up the newsgroups specified.

Once you have done this for each hierarchy you will carry, you should manually create any newsgroups not included in checkgroups messages (e.g. local newsgroups). This completes the setup of your local system. It is now ready to receive items from incoming feeds, and to allow local users to read and post items.

5.9 Setting up the NNTP server

If you are connected to a high speed DECnet or TCP/IP network, you may choose to set up the NNTP server supplied in the ANU News distribution. This will allow clients on other systems access to news from your local database, and can also serve as a means to receive incoming news feeds. Using the NNTP access file described above, you can control the types of access different remote hosts are allowed to your database.

The specifics of setting up the server differ depending on the transport protocol you will be using. The procedure for each transport supported by ANU News is described below. In addition, if you have more than one incoming feed for any substantial number of newsgroups, you may also wish to enable message ID caching by the NNTP server, as described above, in order to minimize redundant transfer and processing of items. (Remember that the NNTP server must have SysLck and SysGbl privileges if you are using message ID caching.)

5.9.1 NNTP server running over DECnet

There are two versions of the server provided in the News distribution, one a single-threaded server designed to be invoked in response to an incoming connection request to a known DECnet object, and the other a multithreaded server which will register itself as a DECnet object and process up to 32 simultaneous connections. In general, the multithreaded server is more efficient if you expect multiple simultaneous connections, but it is unable to obtain the remote host and username for each thread, so it will not allow you to control access to restricted newsgroups as the single-threaded server will.

Whichever server you use, you will need to determine under what local account the NNTP server will run. This need not be a privileged account, but the NetServer.Log files from incoming connections will be left in its Sys$Login, and it must have execute access to the server image. In most cases, the DECnet default account is fine for this purpose. If you choose another account, you must either provide remote sites with proxy access or with the appropriate access control information, or you must specify this information in a local DECnet object definition.

5.9.1.1 Setting up the single-threaded DECnet server

If the TASK DECnet object is not disabled on your system, you need only do the following:

- create a DCL procedure named NNTP.Com in Sys$Login of the account you are using to run the server, which contains the commands
  $ Run *dev*:[*dir*]NNTP_DEC.Exe
  $ Exit In addition, you may want to add commands to purge old copies of NetServer.Log, or set a reasonable version limit for this file.
- Copy the image NNTP_DEC.Exe from the [News_Dist] directory to the location you specified in NNTP.Com above, and make sure the account under which the server processes will run has execute access.
- Install NNTP_Dec.Exe with SysPrv. If you plan to use the News lock mechanism (e.g. NewsShutDown) add SysLck as well, and if you are using message-ID caching, add SysLck and SysGbl. You should use a command like:
  $ Install Add /Priv_=(SysPrv,SysLck,SysGbl) Be sure to add this command to your system startup procedure as well.

If the TASK object is disabled (as is frequently the case because of the security hole it creates), then you will need to create a DECnet object which can be used to direct incoming connection requests. It's impractical here to discuss all of the possible variations in configuration this method will support, so we'll stick to a few simple examples here. For more information, consult the VMS Networking Manual and the Network Control Program manual in the VMS doc set.

To define a DECnet object which runs under the default DECnet account, and to which users connect by specifying the task string "0=NNTP", perform the following, from an account with the requisite privileges:

```
$ Run Sys$System:NCP
NCP> Set Object NNTP Number 0
NCP> Define Object NNTP Number 0
```

If the file NNTP.Com is not in Sys$Login for the default DECnet account, add the FILE parameter to these commands. Similarly, if you want this process to run under an account other than the DECnet default, add the USER and PASS parameters to these commands.

If you would prefer that the task string used to specify the NNTP server is of the form "NNTP= ", then select an unused DECnet object number between 128 and 255, and set up the object as follows:

```
$ Run Sys$System:NCP
NCP> Set Object NNTP Number number
NCP> Define Object NNTP Number number
```

In this case, the file NNTP.Com must be in Sys$System; if it's not, use the FILE parameter to these NCP commands to specify it. Again, if you want this process to run under an account other than the DECnet default, add the USER and PASS parameters to these commands.

5.9.1.2 Setting up the multithreaded DECnet server

The procedure for setting up the multithreaded DECnet server is essentially identical to that for setting up the single-threaded server, with a few important exceptions:

- The image used is NNTP_DECM.Exe, not NNTP_DEC.Exe
- It must be installed with SysNam privilege, as well as those specified above, so it can declare itself as a permanent network object.
- The NNTP DECnet object must have the number 0. If the TASK DECnet object is not disabled on your system, you don't need to do anything as long as remote users will use the task string "TASK_=NNTP" to connect to the server. If the TASK object is disabled, or remote users will use the task string "0_=NNTP" to connect to the server, you must follow the directions given in the previous section for setting up a DECnet object corresponding to this task string. You may not use a task string of the form "NNTP_= " with the multithreaded server.

The first incoming connection request will cause the server to be started, and it will then accept all incoming connections to the NNTP task until the server process is deleted, the system is shut down, or the server process exits in response to a News resource lock generated via the News_Stop logical name or NewsShutDown.Exe.

5.9.2 Setting up the CMU-IP NNTP Server

There are two versions of the NNTP server available for use with the CMU-IP TCP/IP package, one single-threaded and the other multithreaded. The single-threaded server is designed to be started by the CMU-IP master TCP server. The multithreaded server can also be invoked by the master server in response to the first incoming connection request, and then handle successive requests itself, or it can be directly run as a detached process.

5.9.2.1 Setting up the single-threaded CMU-IP server

In order to set up the single-threaded server, you need to do the following:

- Copy the image NNTP_TCPCMU.Exe from the [.News_Dist] directory to a location accessible to the SYSTEM account (e.g. TCP$System).
- Add the following line to Sys$Manager:Internet.Config (line wrapped here to fit into the margins):

```
WKS:119:NEWSRV:TCP$NEWSRV:NETWRK:NETMBX,TMPMBX,SYSPRV,
        SYSLCK,SYSGBL,PHY_IO:4:5
```

  You need SysLck only if you are using the News lock mechanism (e.g. NewsShutDown) or message-ID caching; you need SysGbl only if you are using message-ID caching.

- Define the logical name TCP$Newsrv in the system logical name table as the filespec of NNTP_TCPCMU.Exe. Add a command to do this to Sys$Manager:IP_Startup.Com as well.
- Stop and restart the TCP master server process.

Incoming connection requests to TCP port 119 will now be connected to the NNTP server.

5.9.2.2 Setting up the multithreaded CMU-IP NNTP server

If you want the multithreaded server to be invoked by the TCP master server in response to the first incoming connection request on TCP port 119, follow the instructions for setting up the single-threaded server, with the following changes:

- Use the image NNTP_TCPCMU_M.Exe instead of NNTP_TCPCMU.Exe.
- The entry added to Sys$Manager:Internet.Config should be:

```
WKS:119:NEWSRV:TCP$NEWSRV:NETWRK:\
        NETMBX,TMPMBX,PHY_IO,SYSPRV,SYSLCK,SYSGBL:\
        BYTLM=65535,BIOLM=32767,DIOLM=32767,ASTLM=200,ENQLM=100:\
        SYS$NULL:SYS$NULL:SYS$NULL:4:5
```

  You need SysLck only if you are using the News lock mechanism (e.g. NewsShutDown) or message-ID caching; you need SysGbl only if you are using message-ID caching.

- Define Sys$Null as NLA0: in the system logical name table, and add a command to do this to Sys$Manager:IP_Startup.Com.
- Stop and restart the TCP master server process.

The first incoming connection request to TCP port 119 will cause the NNTP server to be started, and it will then accept all incoming connections to port 119 until the server process is deleted, the system is shut down, or the server process exits in response to a News resource lock generated via the News_Stop logical name or NewsShutDown.Exe. If a News resource lock is in effect, all incoming connection requests to TCP port 119 will be accepted, a message will be transmitted saying that the server is temporarily unavailable, and the connection will be closed.

Alternatively, you may run the NNTP server directly as a detached process. In order to start the server, use a command like

```
$ Run/Detached/Process_="NEWSRV" -
     /Privilege_=(NetMbx,TmpMbx,Phy_IO,SysPrv,SysLck,SysGbl) -
     /Input_=_NLA0:/Output_=_NLA0:/Error_=_NLA0:/Priority_=4 -
     /Buffer_Limit_=65535/IO_Buffered_=32767/IO_Direct_=32767 -
     /AST_Limit_=200/Enqueue_Limit_=100
     dev:[dir]NNTP_TCPCMU_M.Exe\
```

You need SysLck only if you are using the News lock mechanism (e.g. NewsShutDown) or message-ID caching; you need SysGbl only if you are using message-ID caching.

You may also want to specify a file with the /Error qualifier, so you can examine any error messages generated by the server.

The server process will start immediately, and it will then accept all incoming connections to port 119 until the server process is deleted, the system is shut down, or the server process exits in response to a News resource lock generated via the News_Stop logical name or NewsShutDown.Exe. Once the server process exits, all incoming connections to TCP port 119 will be rejected until you manually restart the server (for instance, from the process

which took out the exclusive lock on the News resource, once it's done whatever processing required exclusive access to the database).

5.9.3 Setting up the Multinet TCP NNTP server

The NNTP server provided for use with the Multinet TCP package is single-threaded, and is usually invoked by Multinet's master TCP server. In order to set up this server, you should

- Copy NNTP_TCPWINMultinet.Exe from the [.News_Dist] directory to Multinet:NNTP_Server.Exe
- Set up the service using the Multinet services configuration program:

```
$ Multinet Config/Serv
MultiNet Server Configuration Utility 3.2(32)
[Reading in configuration from MULTINET:SERVICES.MASTER_SERVER]
SERVER-CONFIG> add NNTP
[Adding new configuration entry for service "NNTP"]
Protocol: [TCP]
TCP Port number: 119
Program to run: Multinet:NNTP_Server.Exe
[Added service NNTP to configuration]
[Selected service is now NNTP]
SERVER-CONFIG> Sho/Ful NNTP
Service "NNTP":
 TCP socket (AF_INET,SOCK_STREAM, Port 119)
 Socket Options _= SO_KEEPALIVE
 INIT() = TCP_Init
 LISTEN() = TCP_Listen
 CONNECTED() = TCP_Connected
 SERVICE() = Run_Program
 Program = "MULTINET:NNTP_SERVER.EXE"
SERVER-CONFIG> Quit
Configuration modified, do you want to save it ? [NO] YES
```

A server process will be started by the master server in response to each incoming connection request on TCP port 119. The server process will exist for the duration of that connection only.

5.9.4 Setting up the TCPware TCP NNTP server

The NNTP server provided for use with Process Software's TCPware TCP package is single-threaded, and is usually invoked by the master TCP server. In order to set up this server, you should

- Copy NNTP_TCPWINMultinet.Exe from the [.News_Dist] directory to TCPware_Common:[TCPware]NNTP_Server.Exe.
- Add the following line to the file TCPware:Services.

```
nntp 119/tcp readnews # USENET News Transfer Protocol
```

- Define the service as follows:

```
$ Run TCPware:NetCU
Add/Service NNTP Stream TCPware:NNTP_Server.EXE -
/Process_Name = NNTP_SERVER -
/NoAccounting -
/NoAuthorize -
/UIC = [SYSTEM] -
/AST_Limit = 50 -
/Buffer_Limit = 10240 -
/Enqueue_Limit = 100 -
/Extent = 1024 -
/File_Limit = 50 -
/IO_Buffered = 16 -
/IO_Direct = 16 -
/Maximum_Working_Set = 4096 -
/Page_File = 10000 -
/Priority = 4 -
/Privileges = (NoSame,SysPrv,Phy_IO,NetMbx,TmpMbx,World,SysLck,SysGbl) -
/Queue_Limit = 8 -
/Working_Set = 1024 -
/Subprocess_Limit = 0
Quit
```

  You need SysLck only if you are using the News lock mechanism (e.g. NewsShutDown) or message-ID caching; you need SysGbl only if you are using message-ID caching.

- Add the NetCU commands to TCPware:Servers.Com as well, so that the service will be defined at subsequent TCPware startups.

A server process will be started by the master server in response to each incoming connection request on TCP port 119. The server process will exist for the duration of that connection only.

5.9.5 Setting up the UCX TCP NNTP server

There are two versions of the UCX NNTP server provided in the News distribution, one a single-threaded server designed to be invoked by the UCX master TCP server in response to an incoming connection request, and the other a multithreaded server which runs as a detached process and handles up to 16 simultaneous connections.

5.9.5.1 Setting up the single-threaded UCX NNTP server

At the time of this release, the single-threaded server is not functional; only the multithreaded server is supported. The source code for the single-threaded server is still provided, however, and anyone is welcome to clean up the code.

5.9.5.2 Setting up the multithreaded UCX NNTP server

In order to set up the multithreaded server, you should

- Copy NNTP_TCPUCXM.Exe from the [.News_Dist] directory to a location accessible to the UIC under which the detached process will run.
- Create the directory News_Manager_Dev:[News_Log], in which the server will create its log files (named NNTP_Server_Drvr.Log). If you wish to change this, change the line

  #define drvrlogfile "NEWS_MANAGER_DEV:[NEWS_LOG]NNTP_SERVER_DRVR.LOG"

  in NNTP_TCPUCXM.C, and run News_Build.Com to rebuild NNTP_TCPUCXM.Exe.
- As part of your News system startup, create a detached process running this image, using a DCL procedure like:

```
$! Change the following four lines as appropriate to your site.
$ outfile = "News_Manager:NNTP_TCPUCXM.Log"
$ errfile = "News_Manager:NNTP_TCPUCXM.Log"
$ server_image = "Sys$System:NNTP_TCPUCXM.Exe"
$ uic = "[NEWSMGR]"
$!
$ oldpriv = F$SetPrv("Detach,SysPrv,SysLck,SysGbl,PrmMbx")
$ Create &outfile
$ Create &errfile
$ Purge/Keep=5 &outfile
$ Purge/Keep=5 &errfile
$ Run/Detach/UIC=&uic -
     /Privileges=(NoSame,SysPrv,PrmMbx,NetMbx,SysLck,SysGbl) -
     /Priority=4 -
     /Enqueue_Limit=50/AST_Limit=20/Queue_Limit=8 -
     /Buffer_Limit=12288/File_Limit=20/Page_File=10000 -
     /Working_Set=600/Maximum_Working_Set=1000/Extent=2000 -
     /Input=NLA0:/Output=&outfile/error=&errfile -
     /Process_Name="UCX/NNTP server" -
     &server_image
$ If oldpriv .nes. "" Then foo = F$SetPrv(oldpriv)
$ Exit
```

If you expect the server to operate under a heavy load (e.g. many connections will be active at most times), you may need to increase some of the quota values given here.

The server process will start immediately, and it will then accept all incoming connections to port 119 until the server process is deleted, the system is shut down, or the server process exits in response to a News resource lock generated via the News_Stop logical name or NewsShutDown.Exe. Once the server process exits, all incoming connections to TCP port 119 will be rejected until you manually restart the server (for instance, from the process which took out the exclusive lock on the News resource, once it's done whatever processing required exclusive access to the database).

5.9.6 Setting up the Wollongong TCP NNTP server

The NNTP server provided for use with The Wollongong Group's WIN/TCP for VMS or PathWay for OpenVMS TCP package is single-threaded, and is usually invoked by the master TCP server. In order to set up this server, you should

- Copy NNTP_TCPWINMultinet.Exe from the [.News_Dist] directory to TWG$TCP:[Netdist.Serv]NNTP_Server.Exe.
- Add the following entry to the file TWG$TCP:[Netdist.Etc]Servers.Dat:

```
#ANU NNTP Server
service-name    NNTP
program         TWG$TCP:[NETDIST.SERV]NNTP_SERVER.EXE
socket-type     SOCK_STREAM
socket-options  SO_ACCEPTCONN _| SO_KEEPALIVE
socket-address  AF_INET , 119
working-set     200
maxservers      5
priority        4
INIT            TCP_Init
LISTEN          TCP_Listen
CONNECTED       TCP_Connected
SERVICE         Run_Program\|
```

- Add the following entry to the file TWG$TCP:[Netdist.Etc]Services.:
  nntp 119/tcp #ANU NNTP Server
- Stop the INET_SERVERS process using the DCL command Stop/ID.
- Restart the master server by executing TWG$TCP:[Netdist.Misc]Inetserv.Com

A server process will be started by the master server in response to each incoming connection request on TCP port 119. The server process will exist for the duration of that connection only.

5.10 Setting up the maintenance jobs

In order to keep your News database up to date, you will need to remove expired items, and add items received from upstream feed sites to the local database (postings by local users are automatically added at the time of posting). In addition, if you provide newsfeeds to other sites, you will need to actually transfer the items enqueued by News for delivery to each downstream site. These tasks are usually accomplished via periodic batch jobs running with a frequency determined by the speed with which you want to propagate items and the load these jobs place on your system. Each of these tasks is discussed in greater detail in the following sections.

5.10.1 Adding incoming items to the local News database

Items transferred to your site by upstream feed sites or received from mailing lists which you have gatewayed into News are added to the local database using the Add File command in News. You should consult the section in the manual which describes this command in detail to determine which qualifiers are appropriate for your site. For instance, the /Delete qualifier is a useful way to clean up after items are added, and is used routinely at most sites. Conversely, it is not a good idea to use certain qualifiers routinely:

/Accept - allows unapproved postings to be added to moderated newsgroups

/CreGrp - may result in creation of bogus newsgroups due to typos or pranks in item headers

/Execute - may result in unwanted creation or deletion of newsgroups in response to network control messages.

Also, if you plan to feed items to downstream sites which you don't carry locally, and you use a message ID queue file (News.Sys *flags* field value 'N'), you should not use the /NoJunk qualifier, since this will not allow any local copy of the item to be created, and NNTP_Xmit will not be able to retrieve the item text when it processes the message ID queue file.

Depending on the method by which the items were transferred, some preprocessing may be necessary to generate files suitable for input to News. If you are adding files from different sources, it is usually most efficient to do any required preprocessing first, and then add all of the resulting files in a single News session. This is usually done from a batch job, which is often called the 'NewsAdd job'. Examples of DCL procedures used to accomplish this are found in the News_Dist ZIP file (they all have NewsAdd in their names). Depending on the type of incoming news feeds you have and the load on your system, you may wish to run this job as often as once every thirty minutes or so, or as rarely as once a day. Remember that local users will not be able to read incoming items, nor will you be able to feed them to downstream sites, until you have added them to the local database.

5.10.1.1 Batches received by your NNTP server

All versions of the NNTP server provided with News store incoming batches in the News_Manager directory as files named NNTP_{long_unique_string}.Batch, which can be handed off directly to News, using a command like

```
NEWS> Add File/Delete News_Manager:NNTP_*.Batch
```

If the NNTP server was interrupted for some reason in the middle of receiving a batch, it may leave behind the items it received in a file named NNTP_{long_unique_string}.Incoming_Batch, so you may want to add a command like to your NewsAdd job

```
$ Set NoOn
$ Rename/Log/Before=Today News_Manager:NNTP_*.Incoming_Batch .Batch
$ Set On\|
```

before you invoke News to execute the Add File command.

5.10.1.2 Batches received by direct file transfer

These batches may or may not require preprocessing, depending on the arrangements you have made with the transferring site. In general, News expects the file passed as a parameter to the Add File command to contain a series of items either in 'rnews' format, or as a succession of mail messages. Most sites are able to provide batches in one of these formats, so you shouldn't have to convert the actual data. However, it is fairly common, especially among UUCP sites, to compress the batch files in order to minimize the bandwidth needed for transfer. The most common method of doing this is via the U*ix compress(1) tool, or its VMS equivalent, LZComp. Files received in this format must be uncompressed using the appropriate utility. For instance, files compressed using LZComp can be uncompressed using the command:

$ LZDcmp *compressedfile uncompressedfile*

while those compressed using U*ix compress(1) are uncompressed using:

$ LZDcmp -X 2 *compressedfile uncompressedfile*

In both cases, you must obtain the LZDcmp program and define LZDcmp as a foreign command to invoke the image ,i.e.

$ LZDcmp == "$*dev*:[*dir*]LZDcmp.Exe"

Note

If you don't already have LZComp and LZDcmp, you can get copies at most anonymous ftp sites carrying VMS software. For example, try ftp.spc.edu, in the directory [.macro32.savesets].

Once this is done, you can hand the uncompressed file off to News, using a command like

NEWS> Add File/Delete *uncompressedfile*

where *uncompressedfile* is the name of the uncompressed batch file.

5.10.1.3 Items received via mail

News is able to process items received via mail, and will attempt to remove header lines added during the mail transfer as it adds the item to the News database. This option is intended principally to allow you to gateway mailing lists into News. If at all possible, you should set up your news feeds based on file transfer or NNTP, since these are more efficient and less error-prone than mail transfer.

In order to gateway a mailing list into News, you should subscribe to the mailing list from an account to which the NewsAdd job has access. Then, from the NewsAdd job, you execute the following commands:

```
$ Mail
Set Folder Newmail
Extract/All News_Manager:News_Mail.Batch
Delete/All
Exit
```

in order to extract the items into a text file. Note that if you are gatewaying a number of mailing lists, and each list places a unique string in the From: header, you can subscribe to multiple lists from a single account, and sort them using the VMS Mail command Select, e.g.

```
$ Mail
Set Folder Newmail
Select/From_="LIST1-Tag"
Extract/All News_Manager:News_List1.Batch
Delete/All
Select/From_="LIST2-Tag"
Extract/All News_Manager:News_List2.Batch
Delete/All
. . .
Exit
```

If you're not able to distinguish among the lists using Select, you'll have to subscribe to them from different accounts, and use some other method to extract the items into files (e.g. DELIVER). If nothing else, you can do something like the following in your NewsAdd job:

```
$ list = "LIST1-L"  ! mailing list
$! Note: listuser account must have write access to News_Manager
$ listuser = "LIST1ACCT" ! account you have subscribed to the list
$ fnam = "Sys$Scratch:" + list + "_News.Com"
$ Open/Write cmdfil &fnam
$ Write cmdfil "$ Delete/NoLog/NoConfirm 'F$Environmane(""Procedure"")'"
$ Write cmdfil "$ Mail"
$ Write cmdfil "Set Folder Newmail"
$ Write cmdfil "Extract/All News_Manager:News_''list'.Batch"
$ Write cmdfil "Delete/All"
$ Write cmdfil "Exit"
$ Write cmdfil "$ Exit"
$ Close cmdfil
$ Submit &fnam /User=&listuser  ! Requires CmKrnl priv
$ Synchronize/Entry=&$Entry  ! Wait for the job to finish; >= VMS 5.4
```

or, alternatively:

```
$ list = "LIST1-L"  ! mailing list
$! mail file for account which you have subscribed to list
$! (account running NewsAdd job must have access to this file)
$ listuser_mailfile = "$Users:[List1Acct]Mail"
$ fnam = "Sys$Scratch:" + list + "_News.Com"
$ Open/Write cmdfil _&fnam
$ Write cmdfil "$ Delete/NoLog/NoConfirm 'F$Environment(""Procedure"")'"
$ Write cmdfil "$ Mail"
$ Write cmdfil "Set File ''listuser_mailfile'"
$ Write cmdfil "Set Folder Newmail"
$ Write cmdfil "Extract/All News_Manager:News_''list'.Batch"
$ Write cmdfil "Delete/All"
$ Write cmdfil "Exit"
$ Write cmdfil "$ Exit"
$ Close cmdfil
$ @&fnam
$ Synchronize/Entry=&$Entry  ! Wait for the job to finish; >= VMS 5.4
```

Once you have extracted the messages into a file, you can add them to News. Note that, unless the extracted messages are carrying news headers, you must tell News to generate a Newsgroups: header and a Message-ID: header when you add the items to News, by using appropriate qualifiers to the Add File command:

```
NEWS> Add File/Delete/GenID/Newsgroup="local.list1" -
        News_Manager:News_List1.Batch
NEWS> Add File/Delete/GenID/Newsgroup_="local.list2" -
```

News_Manager:News_List2.Batch

5.10.2 Transferring enqueued items to downstream sites

When News adds an item to the local database, whether as part of Add File processing or when a local user posts an item, it also enqueues it for transfer to other sites according to the entries in your News.Sys file. You will then need to actually transfer the enqueued items to the appropriate sites. Again, this is usually accomplished from a batch job; since transferring is usually a natural sequel to adding new items, this is often done from the NewsAdd job.

If the enqueued items are in single item (News.Sys *flags* field value 'M') or 'rnews' batch (News.Sys *flags* field value 'B') format, you just need to send them off to the other site via UUCP, DECnet copy, ftp, mail, or whatever means you've arranged with the other site. (Again, you should try to avoid feeding items by mail, and should group items into 'rnews' batches if possible, in order to transfer them efficiently and avoid damage done by unfriendly mailers.)

If you have enqueued message IDs rather than items (News.Sys *flags* field value 'N'), you can use the resulting queue file as input to the NNTP_Xmit utility, to conduct a transfer session with the downstream site using the NNTP IHAVE protocol. Read the section on NNTP_Xmit in this manual for details of the syntax used for invoking this program. Note that since NNTP_Xmit retrieves the actual item text from the News database, you should run it before any enqueued items have expired and been removed from the local database. This is often a concern if you feed newsgroups to downstream sites which you don't carry locally. These items will be placed in the newsgroup 'junk' on your system, and it is customary to set the retention time for this group to a short period (e.g. 1 day), in order to keep it from growing too large, so the items won't be around for long. The sample file NNTP.Com in the News_Dist ZIP file demonstrates one way of organizing the NNTP queue files for different downstream sites. Any scheme will work, as long as your batch job knows where to look for queue files for each site.

If you use different batch jobs to add items to the local database and to transfer items to downstream sites, you should run the transfer job after each NewsAdd job, if possible. Running it more frequently is of limited value, since it will only transfer items posted by local users since the last NewsAdd run. Running it less frequently will result in longer lag times before downstream sites receive, and can make use of, new items.

If you are functioning as an 'end node' (i.e. you don't feed any downstream sites), you will still need to run this job to transfer local postings back to your feed site. Since this typically involves a small number of items, you should run this job fairly frequently, so that postings from your site are propagated to the net with reasonable speed.

5.10.3 Removing expired items from the News database

Once newsitems reach their expiration date, they must be removed from the local database in order to keep it from growing indefinitely. This is accomplished via the Skim command in News, which can also be used to check the integrity of the database, and remove files that have become 'orphaned' due to errors in news processing. This task is usually performed in a batch job often called the NewsSkim job. You should read the section of this manual which describes the Skim command in detail in order to familiarize yourself with the options available when using this command.

Since item expiration is determined by date only, it is not useful to run the NewsSkim job more often than daily. If you have a large News database, you should run a basic NewsSkim job every day to expire items, using a command like

```
NEWS> Skim/Items/Served/NoDirectories/NoFileCheck/NoHistory
```

Since this can take a while, you may wish to run it during low-usage hours. Once every several days, you should run a complete Skim pass over the database to clear out any deleted newsgroup directories and 'orphaned' files, and to remove old entries from the history file, using a command like

```
NEWS> Skim/Items/Served/Directories/FileCheck/History
```

This will consume significant CPU and I/O resources, so it should be run during a quiet time on your system.

In addition to pruning the item database, it is usually worthwhile to periodically optimize the index files News uses to keep track of current newsgroups and items, as well as items deleted recently from the local database. This is done as follows:

- BEFORE the daily Skim job, run the Analyze/RMS/FDL utility on each of the index files:

  ```
  $ Analyze/RMS/FDL/Output=News_Root:Groups.FDL News_Root:News.Groups
  $ Analyze/RMS/FDL/Output=News_Root:Items.FDL News_Root:News.Items
  $ Analyze/RMS/FDL/Output=News_Root:Hist.FDL News_Root:History.V60
  ```

  This will provide you with a picture of file usage at its peak.

- Then optimize the FDL files using Edit/FDL:

  ```
  $ Edit/FDL/NoInteractive/Analysis=News_Root:Groups.FDL -
        News_Root:Groups.FDL
  $ Edit/FDL/NoInteractive/Analysis=News_Root:Items.FDL -
        News_Root:Items.FDL
  $ Edit/FDL/NoInteractive/Analysis=News_Root:Hist.FDL -
        News_Root:Hist.FDL
  ```

  If you want to specify other aspects of the index file structure (e.g. bucket fill factors, key compression), you can do this by keeping a "permanent" FDL file describing these changes for each index file, and supplying it as the parameter to Edit/FDL, while leaving the FDL file generated by Analyze/RMS as the argument to the /Analysis qualifier. This causes the current data gathered by Analyze/RMS to be merged into your FDL file, so that it can be used to optimize the index file while retaining your settings.

- Run the NewsSkim job, using at least the /Items/Served/History qualifiers.

- AFTER the skim job, optimize the index files using Convert/FDL:

```
$ Convert/NoSort/FDL_=News_Root:Groups.FDL -
        News_Root:News.Groups News_Root:News.Groups
$ Convert/NoSort/FDL_=News_Root:Items.FDL
        News_Root:News.Items News_Root:News.Items
$ Convert/NoSort/FDL_=News_Root:Hist.FDL
        News_Root:History.V60 News_Root:History.V60
```

Since the FDL files were generated and optimized based on the state of the index files before the skim, the resulting files will be preallocated to contain additional entries up to the size of the files before the skim.

Note that Analyze/RMS and Convert require exclusive access to the index files, so you will need to insure that no other processes have these files open when performing these tasks. This is described in detail in the next section. In addition, if your index files are large (as is typical if you're carrying many groups), optimizing the index files may take some time, and will require enough free space in News_Root to accommodate the new versions of each file created by Convert. Be sure you have enough space available, and remember to purge the old versions of the index files after Convert is finished.

5.10.4 Obtaining exclusive access to the News database

On some occasions, it is necessary to insure that one process has exclusive access to the News database. Most often, this is necessary when tuning the index files, as described above, since Convert requires exclusive access to the files it is optimizing. This may also be necessary when you are copying portions of the News database, or performing other operations during which the data could be lost if another process added or deleted items. News provides two mechanisms for doing this:

- If the logical name News_Stop is defined in executive mode in the system logical name table (N.B. this is *not* changed by the changing the 'rules for translating secure logical names' described elsewhere in this document), News will quit all active sessions within 30 minutes. Single- threaded NNTP servers will end the current session within 30 minutes as well, and then exit, but multithreaded servers will continue to hibernate with the news files open, so they must be stopped explicitly using the Stop/ID DCL command or its equivalent.
- If you have installed News.Exe and your NNTP server with SysLck, you can define the logical name News_Locked_Command to translate to the DCL command you would use to convert the index files, and then use NewsShutDown.Exe to invoke that command after obtaining an exclusive lock on the index files. For example, if you placed the Analyze/RMS commands described above for obtaining information about the News index files into the DCL procedure News_Manager:Analyze_Index.Com, and the Convert commands described above for tuning the News index files into the DCL procedure News_Manager:Convert_Index.Com, you could use NewsShutDown in a manner like this:

```
$ oldpriv = F$SetPrv("SysLck")
$! Stop all multithreaded NNTP server processes using Stop/ID
$ sts = 1
$ Define/User News_Locked_Command "@News_Manager:Analyze_Index.Com"
$ Define/User News_Locked_Wait_Minutes 30
$ Run News_Manager:NewsShutDown
$ If .not. $Status
$ Then
$   sts = $Status
$   Write Sys$Error -
    "%NEWSSHUTDOWN-F-ANALERR, unable to analyze index files, " + -
    "status _= ", sts
$   Goto all_done
$ EndIf
$ News/NoScreen
Skim/Items/Files/Directories/FileCheck/History/Served
Exit
$ Define/User News_Locked_Command "@News_Manager:Convert_Index.Com"
$ Define/User News_Locked_Wait_Minutes 30
$ Run News_Manager:NewsShutDown
$ If .not. $Status
$ Then
$   sts = $Status
$   Write Sys$Error -
    "%NEWSSHUTDOWN-F-CONVERR, unable to convert index files, " + -
    "status _= ", sts
$ EndIf
$ all_done:
$! Restart any multithreaded servers stopped above
$ If oldpriv .nes. "" Then oldpriv _= F$SetPrv(oldpriv)
$ Exit sts
```

In either of these cases, any other images which are forced to exit will return an exit status of SS$_DEADLOCK (decimal 3594). You can check for this return status in the DCL procedures which drive your add and skim jobs, and repeat the run if necessary after NewsShutDown releases the database lock or News_Stop is deassigned.

Note that in order to use NewsShutdown, *all* processes running News images (i.e. News.Exe, NNTP server, NNTP_Xfer, or NNTP_Xmit) must have SysLck enabled. Each News image checks this at startup, and if SysLck is not enabled, it will not participate in the News locking mechanism. It will not try to turn SysLck on before checking, nor will it turn SysLck off when it's finished using it.

5.11 Arranging to exchange news with other sites

Once you have decided which newsgroups you will carry on your system, you will need to make arrangements to exchange with other sites items in any groups not limited to your system. At first, you will probably want to function as an 'end node', that is, you will receive incoming news feeds, but will not establish any outgoing feeds, except as necessary to pass items posted by local users back to your feed sites. Once your setup is stable and you have a better idea of its impact on your system's performance, you may wish to provide outgoing feeds to other sites. We encourage as many sites as possible to do this, since distribution of news is based largely on cooperation among participating sites, and you'll benefit the net as

a whole, as well as your standing within the community, by providing a service as well as receiving it.

There are several places you might look for sites willing to provide you with a news feed. If you are bringing up a system on an existing network within your organization, it's possible that someone else on your network already has news running, and can feed you local and institutional hierarchies, as well as netwide groups. In the long run, this will benefit them as well, if you are capable of operating another NNTP server on your network to distribute the load generated by NNTP client newsreaders. If this isn't possible, you can try asking nearby sites (e.g. large university or corporate sites) to supply you with news - local colleagues may be able to point you in the right direction, or you can search out potential sites by consulting UUCP maps for your area. If you're looking for a feed for one of the smaller professional hierarchies, you'll often find information on obtaining feeds in the FAQ of the appropriate admin group, and can even post asking for a feed. Finally, some sites offer commercial newsfeeds, charging fees based on the volume of news transferred. You may end up using a combination of these approaches, with some sites providing local or regional groups, while others provide worldwide hierarchies.

If you're lucky enough to be able to choose among several feed sites, there are a number of factors you should take into account. First, consider the cost of obtaining a feed from a given site, not only in direct fees, but also in such indirect ways as line charges for UUCP calls. Second, make sure the sites you connect to are stable, as users will not appreciate it if your news feed is inconsistent or frequently disrupted. Thirdly, since you may be exchanging a lot of data, try to link up with sites with which you have high bandwidth connections, in order to minimize the impact of news traffic on your communications. Finally, if possible, you may want to spread the load out by taking partial feeds from several sites (e.g. Usenet from one site, alt.* from another, and regional hierarchies from a third); this is a bit more of a headache to set up, but may lead to the best performance and the most goodwill in the long run.

5.11.1 Obtaining the information necessary to set up an exchange

When making arrangements with a particular site, you should be sure to exchange the following information:

- the list of newsgroups you will exchange
- the name that each site places in Path: headers to indicate that an item has travelled through that site
- the format in which items are to be transferred. Options supported by News include:
  - individual items - each item is sent as a separate file. When adding items received by mail to the local database, News will attempt to strip off mail headers added during the transfer.
  - news batches (also called "rnews format") - multiple items are concatenated into a batch file, which is transferred as a unit.

    - "N blocking" - this option can be used with either individual items or news batches. The character 'N' is placed in the first column of each line in the file; the contents are otherwise identical to a file generated without this option. This was originally designed to circumvent problems with mailers which mangled files containing blank lines, and is rarely used anymore.
    - compressed batches (also called "compressed rnews format") - news batches are generated as usual, then compressed before transfer, using a tool such as LZComp. This is the preferred format for exchange among most UUCP sites.
    - NNTP NEWNEWS protocol - the receiving site queries the sending site for the message IDs of all items received since the last exchange, and then requests the items it does not already have. This is less efficient than a direct IHAVE setup, and is therefore rarely used.
    - NNTP IHAVE protocol - the sending site maintains a list of new items received since the last exchange with the receiving site, and exchange of these items only is negotiated during the next NNTP session. This is the preferred method for NNTP exchanges over direct connections.
- the method by which news is to be transferred. Commonly used options include:
    - UUCP - news batches are transferred during UUCP calls, with each site processing incoming batches and generating outgoing batches between calls.
    - DECnet copy or ftp - news batches generated independently at each site are copied to known locations at the other site, where the local News maintenance jobs look for new batches to process.
    - mail - individual items or batches are mailed between sites and processed locally. This method of transfer is rarely used anymore, as the volume of news has grown beyond the capacity of most mail systems to handle efficiently, and there are usually other methods better adapted to the high volume available.
    - direct NNTP connection - the two sites establish a connection over DECnet or TCP/IP links, and negotiate transfer using the NEWNEWS or IHAVE (preferred) protocols. This method is practically limited to high speed links, as the volume of data will overwhelm slower connections.
- the destination locations and login or proxy login information needed if news batches will be copied directly from one site to another.
- the addresses and TCP port numbers (default is 119) or DECnet object definitions for servers to be used for direct NNTP connections.
- the contact person at each site for news administration.

Remember that even if you plan to function as an 'end node' for now, you will still need to transfer items posted by users at your site back to your feed site for further distribution, so arrangements for bidirectional transfer must be made. Once this information is exchanged, each site can update its configuration accordingly. You should try a few small test transfers to insure that your configuration is correct before starting high volume transfers, and then check the logs of your maintenance jobs (and NNTP server, if you're using direct NNTP connections) carefully for a few weeks, in case any hidden bugs show up as the feed is broken in.

5.11.2 Checklist for integrating a new upstream site into News processing

Once you have made arrangements for an upstream site to feed you news items, you should be sure you have completed the following tasks:

5.11.2.1 Set up appropriate network connections

If the transfer is to be by NNTP over TCP or DECnet, you should make sure that the relevant NNTP server is running, and that you have edited the file News_Root:NNTP_Access.News to allow the upstream site transfer access to your site. In addition, if the transfer is over DECnet, you should make sure that the user identification information which will be used by the upstream site is valid.

If the transfer is by direct file copy, make sure that the user identification information which will be used by the upstream site is valid.

If the transfer is by UUCP, modify your UUCP schedule, if necessary, to contact the upstream site at appropriate intervals.

If the transfer is by mail, make sure that the address to which the items will be mailed is valid.

5.11.2.2 Set up the spool area for incoming files

Make sure that the incoming connection will leave the news items in the area you expect, that sufficient space is free to handle the incoming feed, and the your NewsAdd job will have access to the spool area.

5.11.2.3 Include the new items in your News add processing

Make sure that your NewsAdd job checks for incoming files, and performs any necessary preprocessing (e.g. uncompressing files, extracting mail). This may happen automatically, as when an additional site is integrated into a set of existing feeds using NNTP over TCP, or may require explicit action in your NewsAdd job, as when a mailing list is gatewayed into News. Finally, make sure that you perform any necessary cleanup (e.g. deleting intermediate files).

5.11.2.4 Test the incoming feed

Arrange for the upstream site to feed you a few items, and follow them through the entire processing cycle until they're incorporated into the local News database. If this succeeds, run the feed for a few days with a small number of items to insure that the regular processing cycle handles the feed appropriately, and that any bugs which do crop up don't cause a large volume of news items to back up. Once you're satisfied that this is stable, you can step up transfer to its working volume, and monitor it for a few days to make sure all is well (e.g. by examining the NNTP server and NewsAdd job log files).

This represents a relatively conservative approach, and you may be able to skip some of the intermediate steps if you're simply integrating a new site into a processing environment which is already working well.

5.11.3 Checklist for integrating a new downstream site into News processing

Once you have made arrangements to feed news items to a downstream site, you should be sure you have completed the following tasks:

5.11.3.1 Add the News.Sys entry for the new site

Make sure the *site* field of the entry is the name that site puts into Path: headers of news items. Look over the feed pattern for both your site and the downstream site, and determine whether you should include a *skip_sites* field. Make sure that the *subscriptions* and *distributions* fields specify all the newsgroups and distributions you want to transfer (including the to.*site* group for that site, if you are using one), and don't specify any you don't want to transfer. Set up the *flags* and *spool_file* fields appropriately, and insure that the spool file specified can be created (directory exists, ownership and privileges allow News.Exe access, and sufficient unused diskquota exists).

5.11.3.2 Set up appropriate network connections

If the transfer is by direct file copy, make sure that the user identification information which you will use to connect to the downstream site is valid.

If the transfer is by UUCP, modify your UUCP schedule, if necessary, to contact the downstream site at appropriate intervals.

If the transfer is by mail, make sure that the address to which the items will be mailed is valid.

If the transfer is to be by NNTP over TCP or DECnet, insure that you can connect to the receiving site's NNTP server, and that you are allowed to transfer items to that site.

5.11.3.3 Set up transfer of items to the downstream site

Items are spooled for transfer to a downstream site whenever local users post into a newsgroup which you are feeding to that site, or whenever Add File processing adds items to a newsgroup you are feeding to that site. However, you still need to actually transfer the spooled items to the receiving site. This is usually done as part of a regular News maintenance job (either a separate job or as part of the NewsAdd job). You should insure that this job runs frequently enough to insure timely transfer of new items to the downstream site.

If the transfer is via mail, UUCP, or file copy, this job should actually mail the items, or copy them to the appropriate location at the receiving site, and then delete the local spool files.

If the transfer is via NNTP over DECnet or TCP/IP, you should invoke the NNTP_Xmit utility to transfer the items using the NNTP IHAVE protocol. NNTP_Xmit will automatically delete the local spool file when all the items it contains have been successfully processed.

If the 'downstream site' does not actually exist (e.g. you are generating batch files containing the contents of groups you want to archive), you should perform whatever processing of the spool files you need.

5.11.3.4 Test the outgoing feed

First, try to transfer a small spool file by hand, to insure that the transfer pathway is properly configured, and that the downstream site handles incoming items appropriately. Then, try running a small feed for a few days (e.g. by altering the subscriptions field of the downstream site's News.Sys entry to include only a few newsgroups), and make sure that the periodic News maintenance jobs behave appropriately. When you are satisfied with this, widen the feed to its full volume, and follow processing for a few days to insure that all is well (e.g. by examining the NNTP_Xmit log files).

This represents a relatively conservative approach, and you may be able to skip some of the intermediate steps if you're simply integrating a new site into a processing environment which is already working well.

Chapter 6
Setting up News in Mixed Local/Served Configuration

In order to use News with a local database while relying on a remote server to supply you with specific newsgroups, you should follow the instructions for setting up News to use a local database, but do not create the newsgroups you want to be served by the remote site. Once you have completed this and run News for the first time, you create the served newsgroups using the News command:

NEWS> Create Newsgroup/Server_=*server_name*/Protocol_=*nntp_transport group*

where

group is the name of the newsgroup you wish to create,

server_name is the DECnet nodename or TCP host name of the server you wish to use for this newsgroup, depending on the NNTP transport you specified via the /Protocol qualifier, and

nntp_transport is the NNTP transport protocol you wish to use. You may specify DECnet or the TCP protocol specified when you built News.

If you wish to create a number of served newsgroups, you may use a wildcard pattern as *group*, and specify the /Inquire qualifier with the Create Newsgroup command, which will cause News to obtain a list of active newsgroups on the server, and, for each newsgroup whose name matches the pattern you specified, create an equivalent served newsgroup on the local system.

When a newsgroup is created /Served, the local system keeps a record of item numbers and selected information about each item (principally information used to construct the item index display). This information is updated by connecting to the server every time a Skim/Served is performed. When a user reads an article from a served newsgroup, News retrieves the text of the item from the server and displays it for the user. By default, News then retains the text locally for 2 days, and uses this local copy when other users read the item at a later time. (This can be customized by using the /Cache qualifier to Create Newsgroup and Set Newsgroup.) Because of the overhead incurred during the Skim/Served, it is not efficient to have more than a few served newsgroups on a system. If you cannot retain most newsgroups locally due to space constraints, consider shortening the item retention time for locally held groups, particularly large groups, or paring down your feed, and perhaps providing access to newsgroups you don't carry by using News in different sessions as a local reader or as

a diskless client to the remote server. While this is less convenient than having all groups available in one session, it is much more efficient than having a large number of individual served groups.

Finally, remember that at present the NNTP server provided with News cannot serve the text of items from a remote newsgroup to its clients. It will present the information which is available locally, but will not query the remote server for items. Of course, anyone with some spare time and the inclination to do so is welcome to add this capability to the ANU News server.

Chapter 7
News Support Utilities

The following sections describe various utilities which are part of the News distribution, and are used for various News maintenance tasks. Unless otherwise noted, these binaries are created by NewsBuild.Com, and can be found in the [News_Dist] directory. Since the NNTP server is usually set up along with News.Exe, and uses many of the same configuration files, it is described above in the section titled "Setting Up News as a Local Reader/Server".

7.1 Exchange of Items via NNTP Over DECnet or TCP

The utilities described in this section are used for exchanging items with other sites using NNTP protocols over high-speed DECnet or TCP links. These utilities are designed for transferring batches of items as part of incoming and outgoing newsfeeds. Retrieval or posting of individual items by a remote client is done via the local NNTP server, which is described elsewhere in this manual.

7.1.1 NNTP_Xfer

NNTP_Xfer is a utility used to retrieve news items from a remote NNTP server to the local system using the NNTP transfer protocols over a DECnet or TCP connection. Specifically, the local system sends NEWNEWS messages to the remote NNTP server, requesting that system to transmit the message identifier strings of items recently received at that site. The local system may then respond with a subsequent sequence of ARTICLE requests, to retrieve those items not held locally. This utility essentially makes the local system the active partner in the transfer. Because it is less efficient than NNTP_Xmit, and because the NEWNEWS command can impose a heavy load on the remote NNTP server, this method of news transfer is not recommended unless it is impossible for the remote site to run NNTP_Xmit to feed new items directly to you.

7.1.1.1 Installation

NNTP_Xfer.Exe is created in the News_Dist directory when NewsBuild is run. You should copy this image to a location, such as News_Manager, where it is accessible to the News maintenance job which will retrieve items from upstream sites, but is not accessible to other users. This job should invoke NNTP_Xfer using a foreign command definition, such as:

```
$ NNTP_Xfer == "$News_Manager:NNTP_Xfer"
```

7.1.1.2 Command Syntax

NNTP_Xfer can be invoked with a DCL-like syntax, using qualifiers to pass information to the image, with a U*ix-like syntax involving simple command line switches, or with an abbreviated syntax requiring three parameters. You may use either syntax, but you may not mix the two within the same command. In either case, NNTP_Xfer should be invoked via a foreign command definition; it will parse its own command line. The interpretation of all parameters is case-insensitive.

Sample DCL-like syntax:

```
$ NNTP_Xfer/Node=remhost/Protocol=transport/File=grpfile -
        [/Groups=grouplist][/DayOffset=days][/HourOffset=hours] -
        [/Task=decnet-task][/Stream][/LogFile=log-file][/Debug]
```

Sample U*ix-like syntax:

```
$ NNTP_Xfer -n remhost -p transport [-g grouplist] -
        [-d days] [-h hours] [-t decnet-task] [-s] -
        [-l log-file] [-o]
```

Sample abbreviated syntax:

```
$ NNTP_Xfer remhost transport log-file
```

If the first command line argument does not begin with '-' or '/', NNTP_Xfer assumes this abbreviated syntax is being used, and will interpret the command line arguments accordingly. As a consequence, the following is an illegal command:

```
$ NNTP_Xfer cicero decnet News_Manager:Cicero_Xfer.Log -g misc.legal.*
```

(The correct syntax would be:

```
$ NNTP_Xfer -n cicero -t decnet -
        -l News_Manager:Cicero_Xfer.Log -g misc.legal.*
```

in this case.)

/Node=*remhost*\BOLD - the name of the remote host. This is the actual address to be used in establishing the connection, not the name that site uses in news item headers. For DECnet connections this should be the DECnet nodename of the remote node, or its node number in the format *area**1024+*node* (*area**1024 is optional if the server is in the same area as your system). For TCP connections it should be the FQDN of the remote host, or its IP address in dotted decimal form.

[U*ix-like syntax: **-n *remhost*** Abbreviated syntax: first parameter]

/Protocol=*transport* - the transport protocol to be used when connecting to the remote node. This value should be 'TCP', or the name of the TCP transport you specified when building News, if you want to establish a TCP connection to the remote NNTP server; NNTP_Xfer uses the standard NNTP port (119). If any other value is specified, NNTP_Xfer will attempt to establish a DECnet connection.

[U*ix-like syntax: -p *transport* Abbreviated syntax: second parameter]

/File=*grpfile* - file specification of a file containing the list of newsgroup patterns to be passed to the remote server. *Grpfile* should be a text file containing newsgroup patterns separated by commas or record boundaries; a pattern cannot span a record boundary. All records beginning with the character '#' are considered to be comments, and are ignored. The list of newsgroup patterns read from the file is treated just as if it had been entered using the /Groups qualifier (except that files cannot be nested using the '@' for indirection).

If both /File and /Groups are specified, the /File qualifier is ignored. If neither /Groups nor /File is specified, then all newsgroups on the remote system are searched.

[U*ix-like syntax: -f *grpfile*. If both -g and -f are present on the command line, or if either is specified multiple times, only the last occurrence is used.]

/Groups=*grouplist* - a comma-separated list of newsgroup patterns specifying the newsgroups about which information is to be obtained. If the DCL qualifier /Groups was used, and first character of a newsgroup pattern is '@', the remainder of the pattern is treated as a file specification, as though it had been specified using the /File qualifier. The text of *grouplist* is converted to lower case, and all occurrences of '^' are replaced by '!'. The list is then passed to the remote NNTP server as the first parameter to a NEWNEWS command, and the server interprets the patterns. According to the NNTP standard, the following rules should be applied to each pattern:

- all newsgroups whose names match a pattern in *grouplist* should be searched for new items, as specified by the other parameters to the NEWNEWS command.
- the character '*' is a wildcard which matches any number of characters in a newsgroup name
- when being compared to newsgroup names, each pattern has the characters '.*' appended to it.
- if the first character in a pattern is '!', then any newsgroups whose names match that pattern should not be processed as part of the NEWNEWS search. Many servers will follow the same rule used to interpret patterns in News.Sys entries: a more specific match takes precedence over a less specific one. Thus the pattern comp,!comp.os,comp.os.vms would include all comp.* newsgroups except the comp.os.* groups, but would include comp.os.vms.

The total length of the NEWNEWS command cannot exceed 512 characters, so *grouplist* should not be longer than 487 characters.

If /Groups is not specified, or if no *grouplist* value is given, all newsgroups on the remote system are searched. If /Groups is specified, the /File qualifier is ignored.

[U*ix-like syntax: -g *grouplist*. If -g is specified, *grouplist* must be specified. The '@' character does not receive the same special interpretation as it does following the DCL-like /Groups qualifier; use the -f switch (described under the /File qualifier) to process files containing lists of newsgroup patterns. If both -g and -f are present on the command line, or if either is specified multiple times, only the last occurrence is used.]

/DayOffset=*days* - NNTP_Xfer maintains a record of the time at which it last contacted a system, and as part of the current NEWNEWS command, requests items which have been added on the remote system within *days* days before the last contact. Thus, if *days*=0, NNTP_ Xfer requests items added on the remote site after the time of the last contact, while if *days*=1, NNTP_Xfer requests all items appearing at the remote site since one day before the last contact. If the /DayOffset qualifier is not specified, *days* defaults to 0.

[U*ix-like syntax: -d *days*]

/HourOffset=*hours* - This qualifier functions similarly to the /Days qualifier, except that *hours* specifies the number of hours before the last contact to offset the time specified in the NEWNEWS command. This can be particularly useful since NNTP_Xfer records the time of the last c ontact in local time, while the NEWNEWS command requires that time be specified as GMT. Therefore, if your local timezone is two hours ahead of GMT, you should specify an *hours* value of 2 in order for the NEWNEWS command to properly correct its recorded time to GMT. If the /HourOffset qualifier is not specified, *hours* defaults to 0.

[U*ix-like syntax: -h *hours*]

/Task=*decnet-task* - the name of the DECnet object through which the NNTP server on the remote system is contacted. If the /Protocol value specifies a DECnet connection, NNTP_Xfer determines the remote task name as follows:

- if /Task is present, *decnet-task* is used. If *decnet-task* does not contain the character '=', the string "TASK=" is prepended to it. *Decnet-task* is converted to uppercase before being used.
- if /Task is not present, an attempt is made to translate the logical name News_*remhost*_ Task, and if this succeeds, the translation is used.
- if neither of the above cases is true, a default name of "TASK=NNTP" is used.

You should ask the remote site for the NNTP server's task name when making arrangements for a news feed, and pass the appropriate string to NNTP_Xfer either on the command line or via logical name. In particular, the names "TASK=NNTP" or "0=NNTP" are common on VMS systems, and "NNTP=" is common in Ultrix systems.

If /Protocol specifies a TCP connection, the /Task qualifier is ignored.

[U*ix-like syntax: -t *decnet-task* If the U*ix-like syntax is used, *decnet-task* is not checked for the '=' character, and "TASK=" is not added.]

/Stream - specifies that stream input calls should be used over a DECnet connection, rather than record-based input calls. This is normally necessary only when connecting to a NNTP server running on an Ultrix system. If /Protocol specifies a TCP connection, the /Stream qualifier is ignored.

[U*ix-like syntax: -s]

/LogFile=*log-file* - file specification of a file to which a one-line summary of the current session will be appended. This summary will list the time, *remhost*, *grouplist*, the number of message IDs returned by the remote system, and the number of items actually transferred.

[U*ix-like syntax: -l *log-file* Abbreviated syntax: third parameter]

/Debug - causes status messages tracking the progress of the session to be output to the terminal. In addition, all traffic over the connection to the remote server is echoed to the terminal.

[U*ix-like syntax: -o]

In addition, if NNTP_Xfer is invoked with the U*ix-like command line switch -he, a listing of the U*ix-like switches, with brief explanations, will be printed, and NNTP_Xfer will exit. There is no DCL-like equivalent for this option.

7.1.1.3 Transfer Session Summary

After parsing the command line, NNTP_Xfer checks to see whether there are any unprocessed message-ID files left over from previous sessions which terminated abnormally. NNTP_Xfer checks for old message-ID files of the same type as the current session, so, for example, if the current session is a selective search, then NNTP_Xfer will only check for message-ID files from prior selective searches of this remote site. If any old message-ID files are found, NNTP_Xfer will process them as it does any message-ID file (see below).

NNTP_Xfer then attempts to determine the time of the last connection to *remhost*, by looking for the file News_Manager:NNTP_*remhost*.LastCall. (When *remhost* appears in any file names, all occurrences of '.' in the host name are converted to '-'.). This file is normally written by NNTP_Xfer at the end of a NEWNEWS inquiry, and contains a hexadecimal number specifying the number of seconds elapsed between midnight January 1, 1970 and the last NEWNEWS inquiry. If NNTP_Xfer cannot read this file, it uses midnight January 1, 1970 as the time of the last inquiry. This time is then adjusted using the offsets specified by the /DayOffset and /HourOffset qualifiers. The adjusted time and *grouplist* are used to construct a NEWNEWS command, which is sent to the remote server. The list of message-IDs received from the server in response to the NEWNEWS request is written to a message-ID file in the News_Manager directory. The message-ID file is named as follows:

- if all newsgroups on the remote server are being searched, then the file name will be NNTP_*remhost*.IDs, where *remhost* is the name of the remote site, with all '.' characters converted to '-'.
- if a selective search is being performed (i.e. a *grouplist* other than '*' was specified), then the name will be NNTP_*remhost*_Selective.IDs, with *remhost* constructed as described above.

If the entire list is received without error, then the LastCall file is updated to reflect the time of the current connection.

Once the entire list of message-IDs has been received, NNTP_Xfer checks each message ID against the item index file and the history file on the local system. If the ID is not found on the local system, NNTP_Xfer requests the text of the item from the remote site, and stores it in a batch file in the News_Manager directory, named in a manner similar to the message-ID files, except that the file type is '.Batch' instead of '.IDs'. Once all of the message IDs are processed successfully, the message-ID file is deleted. The batch files should then be added to the local News database using the Add File command in News.Exe.

7.1.1.4 Compilation constants

The following C preprocessor constants, defined in NNTP_Xfer.C, may be useful in tailoring the behavior of NNTP_Xfer to your system:

- MAX_BATCH_SIZE - maximum size of a batch file (in bytes). Once a batch file reaches this size, NNTP_Xfer closes it and creates a new version. The distribution code sets this constant to 250000.
- MAX_RESTART_ATTEMPT - NNTP_Xfer will stay connected for considerable periods, particularly if there are low speed links between two systems, and the link may be broken by the underlying TCP software. NNTP_Xfer will attempt to restart a broken connection, which often is sufficient. To stop these restart attempts continuing indefinitely, NNTP_ Xfer will only restart a link a certain number of times, as specified by this constant. The distribution code sets its value to 20.
- CLIENT_TIMER - the maximum time which the local system will wait for remote data before terminating the link on a timeout. The distribution value is 250 (seconds).
- RESP_TIMER - The maximum time to wait when "pinging" the remote system to check that it is still alive. The distribution value is 30 (seconds).

7.1.1.5 Examples

The following command will cause NNTP_Xfer to contact the NNTP server on wombat.bush.au via TCP, and search all newsgroups on the remote system for message IDs appearing since the time of the last non-selective NNTP_Xfer connection to wombat.bush.au:

```
$ NNTP_Xfer/Node=wombat.bush.au/Protocol=TCP -
        /Logfile=News_Manager:NNTPXfer.Log
```

The message IDs will be stored in News_Manager:NNTP_wombat-bush-au.IDs, and items retrieved from the remote server will be stored in News_Manager:NNTP_wombat-bush-au.Batch. When the session is complete, a summary will be appended to News_ Manager:NNTPXfer.Log.

The following command will cause NNTP_Xfer to contact the NNTP server on the remote Ultrix system EMU, and search all newsgroups named animals.* for message IDs appearing on the remote system since one day before the last selective NNTP_Xfer connection to EMU. The connection will be over DECnet using stream I/O calls, with "NNTP=" specified as the task string in the connection request.

```
$ NNTP_Xfer -n emu -p decnet -s -t "NNTP=" -
```

```
-g animals -d 1 -l News_Manager:NNTPXfer.Log
```

The message IDs will be stored in News_Manager:NNTP_emu_Selective.IDs, and items retrieved from the remote server will be stored in News_Manager:NNTP_emu_Selective.Batch. When the session is complete, a summary will be appended to News_Manager:NNTPXfer.Log.

The following command will cause NNTP_Xfer to contact the NNTP server on the remote system KOALA, and search all newsgroups for message IDs appearing on the remote system since the last non-selective NNTP_Xfer connection to KOALA. The connection will be over DECnet using the default task string "TASK=NNTP".

```
$ NNTP_Xfer KOALA DECnet News_Manager:NNTPXfer.Log
```

The message IDs will be stored in News_Manager:NNTP_koala.IDs, and items retrieved from the remote server will be stored in News_Manager:NNTP_koala.Batch. When the session is complete, a summary will be appended to News_Manager:NNTPXfer.Log.

7.1.2 NNTP_Xmit

NNTP_Xmit is a utility used to send news items from the local system to a remote system using the NNTP transfer protocols over a DECnet or TCP link to the remote NNTP server. The actual protocol used is the local system sending IHAVE messages to the remote system notifying that system of locally held message identifier strings, to which the remote system will either respond with a "please send me" string to receive the actual text of the relevant item, or a rejection indicating that it already has the item. This utility essentially makes the local system the active partner in the transfer.

The list of message identifier strings is set up by News itself when processing incoming news batches via the Add File command or local postings via the Post command. In either case the News.Sys file is consulted, and if the item is to be forwarded to a remote site whose News.Sys *flags* field is N or NX, then the message ID is added to the spool file for that site.

The spool files are, by default, organized as sequential text files with one message ID string per line. In most cases this should be adequate, but where the news traffic level is high, or the remote system is connected with a low bandwidth line, there may be contention for this file between News.Exe (writing new message IDs to the file) and NNTP_Xmit.Exe (reading message IDs from the file during a transfer session). In such cases the NX flag should be used in the *flags* field of the News.Sys entry for that site, so that a shared access RMS indexed file is used to store the message ID strings.

7.1.2.1 Installation

NNTP_Xmit.Exe is created in the News_Dist directory when NewsBuild is run. You should copy this image to a location, such as News_Manager, where it is accessible to the News maintenance job which will forward items to downstream sites via NNTP over DECnet or TCP, but is not accessible to other users. This job should invoke NNTP_Xmit using a foreign command definition, such as

```
$ NNTP_Xmit == "$News_Manager:NNTP_Xmit"
```

7.1.2.2 Command Syntax

NNTP_Xmit can be invoked with either a DCL-like syntax, using qualifiers to pass information to the image, or a U*ix-like syntax, involving simple command line switches. You may use either syntax, but you may not mix the two within the same command. In either case, NNTP_Xmit should be invoked via a foreign command definition; it will parse its own command line. The interpretation of all parameters is case-insensitive.

Before invoking NNTP_Xmit, the process should insure that it has read access to all News item files in the local database. You can do this by running the job which invokes NNTP_Xmit under the same UIC as the owner of the News_Device directory tree (usually the News manager), or by granting the process SysPrv or ReadAll privilege, or by Installing the image with SysPrv or ReadAll. NNTP_Xmit does not turn off these privileges when not using them.

In addition, you should make sure NNTP_Xmit.Exe has SysNam privilege, if you want it do define the logical name NNTP_More_To_Offer when it exceeds the maximum number of message IDs you have specified for a single session, and there are more message IDs remaining in the remote system's spool file. You can do this by running NNTP_Xmit.Exe in a process which has SysNam, or by Installing the image with SysNam. NNTP_Xmit does not turn off SysNam when it is not using it.

Finally, if you plan to use the News resource lock mechanism (i.e. NewsShutDown.Exe) when you need exclusive access to the News database, NNTP_Xmit.Exe must have access to SysLck privilege. You can do this by running NNTP_Xmit.Exe in a process which has SysLck, or by Installing the image with SysLck.

Sample DCL-like syntax:

```
$ NNTP_Xmit/Node=remhost/Protocol=transport
        /IDFile=id-file[/Task=decnet-task][/Stream][/IndexID]
        [/LogFile=log-file][/DebugFile=dbg-file]
```

Sample U*ix-like syntax:

```
$ NNTP_Xmit remhost transport id-file
        [-t decnet-task] [-s] [-x] [log-file] [dbg-file] [-h]
```

/Node=*remhost* - the name of the remote host. This is the actual address to be used in establishing the connection, not the name that site uses in news item headers. For DECnet connections, this should be the DECnet nodename of the remote node, or its node number in the format *area**1024+*node* (*area**1024 is optional if the remote node is in the same area as your system). For TCP connections it should be the FQDN of the remote host, or its IP address in dotted decimal form.

[U*ix-like syntax: first parameter not beginning with - or immediately following -t.]

/Protocol=*transport* - the transport protocol to be used when connecting to the remote node. This value should be 'TCP', or the name of the TCP transport you specified when building News, if you want to establish a TCP connection to the remote NNTP server; NNTP_Xmit uses the standard NNTP port (119). If any other value is specified, NNTP_Xmit will attempt to establish a DECnet connection.

[U*ix-like syntax: second parameter not beginning with - or immediately following -t.]

/IDFile=*id-file* - file specification of the spool file to be used in offering items to the downstream site. This is the file specified in the *spool-file* field of the remote site's News.Sys entry.

[U*ix-like syntax: third parameter not beginning with - or immediately following -t.]

/Task=*decnet-task* - name of the NNTP server DECnet task on the remote node. This parameter is used only for DECnet connections, and is ignored for TCP connections. If the task name is not specified on the command line, NNTP_Xmit will attempt to translate the logical name News_*remhost*_Task, and, if successful, will use this as the task name. If this fails, a default task name of "TASK=NNTP" is used. The task name is converted to uppercase.

You should determine the DECnet task name of the remote NNTP server when you set up your newsfeed to the downstream site. In particular, the names "TASK=NNTP" or "0=NNTP" are common on VMS systems, and "NNTP=" is common in Ultrix systems.

[U*ix-like syntax: -t *decnet-task*]

/Stream - indicates that during a DECnet connection stream input routines should be used instead of record-based input routines. This is necessary when connecting via DECnet to a NNTP server running under Ultrix. This parameter is ignored for TCP connections.

[U*ix-like syntax: -s]

/IndexID - indicates that the spool file is an RMS indexed file, rather than a sequential text file (see discussion above).

[U*ix-like syntax: -x]

/LogFile=*log-file* - file specification of a file to which will be appended a one line summary of the transfer session on completion. This summary will list the number of identifiers offered to the remote system, the number of items accepted and rejected, and the number of local lookup failures.

[U*ix-like syntax: fourth parameter not beginning with - or immediately following -t .]

/DebugFile=*dbg-file* - the file specification of a file to which will be written a full traffic log of the transfer session. This file will expand quite rapidly when processing a large batch of identifiers, so the parameter should be used only in a controlled testing environment.

[U*ix-like syntax: fifth parameter not beginning with - or immediately following -t .]

The U*ix-like command line switch -h causes NNTP_Xmit to print a summary of the command syntax, as does invoking NNTP_Xmit with no parameters. There is no DCL qualifier which has this function.

7.1.2.3 Transfer Session Summary

NNTP_Xmit will establish a connection to the NNTP server on the remote system, and then commence a sequence of IHAVE messages to the remote server. The remote server will respond with either a request to send the full item or a rejection.

This process will continue until

- the input identifier file is fully read, or
- NNTP_Xmit is forced to exit when another process locks the News database, or
- the remote or local system terminates the transport link, or

- more than NNTP_MAX_OFFER identifiers are offered to the remote system. If this occurs, NNTP_Xmit will define the logical name NNTP_More_To_Offer in executive mode in the system logical name table, with the translation "YES", before exiting.

If the spool file is a sequential text file, it is opened in read mode, and a second work file named NNTP_*pid*.Tmp is opened in the same directory in write mode. If an item cannot be locally read, then the text will be written to this temp file for a subsequent retry. If the transfer is terminated before complete reading of the input file, all unoffered message identifier strings are then written into this work file. On exit the original input file is deleted, and the work file is renamed to the original input file name.

If the spool file is an indexed file, NNTP_Xmit makes a sequential pass through the file. All offers (i.e. remote acceptance or remote rejection) are followed by deletion of the corresponding record from the identifier file.

7.1.2.4 Compilation constants

The following C preprocessor constants, defined in NNTP_Xmit.C, may be useful in tailoring the behavior of NNTP_Xmit to your system:

- NNTP_MAX_OFFER - the maximum number of message IDs which can be offered to the remote system in a single session. Once NNTP_Xmit has offered this many identifiers, it closes the connection and quits. Before quitting, it attempts to define the logical name NNTP_More_To_Offer (qv). (This will succeed only if you have provided NNTP_Xmit.Exe with the necessary privilege.) The distributed copy of NNTP_Xmit.C uses the value 5000 for this constant.
- CLIENT_TIMER - the maximum time which the local system will wait for remote data before terminating the link on a timeout. The distribution value is 250 (seconds).
- RESP_TIMER - The maximum time to wait when "pinging" the remote system to check that it is still alive. The distribution value is 30 (seconds).

7.1.2.5 Examples

The following command will contact the NNTP server on the remote system wombat.bush.au using TCP, and will offer the items whose message IDs appear in the file News_Manager_Dev:[IHave_Wombat]Wombat.Ids. A summary of the session will be appended to the file News_Manager:NNTPXmit.Log.

```
$ NNTP_Xmit/Node=wombat.bush.au/Protocol=TCP -
_$ /IDFile=News_Manager_Dev:[IHave_Wombat]Wombat.Ids -
_$ /LogFile=News_Manager:NNTPXmit.Log\ |
```

The News.Sys entry for wombat.bush.au would be something like:

```
wombat:comp,misc,news,rec,soc,talk,vmsnet,au,to.wombat:N:\
        News_Manager_Dev/[IHave_Wombat]Wombat.Ids:world,net,au
```

The following command will contact the NNTP server on the remote Ultrix system EMU using DECnet, and will offer the items whose message IDs appear in the file News_Manager_Dev:[IHave_Emu]Emu.Ids. A summary of the session will be appended to the file News_Manager:NNTPXmit.Log.

```
$ NNTP_Xmit EMU DECnet -s -t "NNTP=" -
_$ News_Manager_Dev:[IHave_Emu]Emu.Ids -
_$ News_Manager:NNTPXmit.Log
```

The News.Sys entry for EMU would be something like:

```
emu:*,!alt,!control,!junk,!to,to.emu:N:\
        News_Manager_Dev/[IHave_Emu]Emu.Ids:*
```

The following command will contact the NNTP server on the remote VMS system KOALA using DECnet, and will offer the items whose message IDs appear in the RMS indexed file News_Manager_Dev:[IHave_Koala]Koala.Ids. A summary of the session will be appended to the file News_Manager:NNTPXmit.Log, and a detailed record of the session will be written to the file News_Manager:Koala_NNTPXmit.Test.

```
$ NNTP_Xmit/Node=KOALA/Protocol=DECnet -
_$ /IDFile=News_Manager_Dev:[IHave_Koala]Koala.Ids/IndexID -
_$ /LogFile=News_Manager:NNTPXmit.Log -
_$ /Debugfile=News_Manager:Koala_NNTPXmit.Test\ |
```

The News.Sys entry for KOALA would be something like:

```
koala:aarnet,vmsnet,to.koala:NX:\
        News_Manager_Dev/[IHave_Koala]Koala.Ids:world,net,au,aarnet
```

7.2 Testing NNTP connections over DECnet or TCP/IP networks

The programs described below are designed for use in testing DECnet or TCP/IP connections to an NNTP server (yours or another site's), or allowing you to converse directly with the server using NNTP commands. These are useful principally for tracking down problems with an NNTP server, and do not need to be used during normal operations.

7.2.1 NNTP_Client

NNTP_Client is a utility used to test a connection to a remote NNTP server. The program attempts to establish an interactive session with the remote server, allowing the user to send NNTP commands to the server. In the case of a connection made over TCP, this is equivalent to a telnet session to the remote NNTP port (i.e. Telnet/Port=119). There is no direct equivalent with DECnet.

7.2.1.1 Installation

NNTP_Client.Exe is created in the [.News_Dist] directory by the NewsBuild procedure. The image should be placed in a location, such as News_Manager, which is not accessible to general users. This is necessary because NNTP_Client can be used to easily make fraudulent postings to an NNTP server, since it contains no authentication checks like those in News.Exe.

7.2.1.2 Command syntax

NNTP_Client is invoked via the DCL Run command, and takes no command line parameters or qualifiers. It will prompt you for all necessary information before connecting to the remote NNTP server.

At the Node: prompt, enter the name of the remote host. This can be a DECnet node name, a DECnet node number in the format *area**1024+ *node* (*area**1024 is optional if the server is in the same area as your system), a TCP host name, or an IP address in dotted decimal form.

At the Channel: prompt, enter the transport over which you wish to establish the connection (DECnet or TCP). If you just press <Return>, the transport defaults to DECnet.

If you are establishing a DECnet connection, you will also be asked for the DECnet object name for the remote server. Enter the object name, or press <Return> to accept the default string TASK=NNTP. You will also be asked if you want to use DECnet stream input calls, instead of record input calls. This is usually necessary only if the remote server is running on an Ultrix system.

7.2.1.3 Session Summary

Once it has obtained the necessary information, NNTP_Client will connect to the remote NNTP server, and allow you to conduct an interactive session, using NNTP commands to communicate with the server. Lines you type are sent to the server when you press <Return>, and the server's responses are displayed on the terminal. For more information on the NNTP command set, see the NNTP standard, RFC977, which is provided in the News_Doc ZIP file.

When you have finished the session, use the NNTP QUIT command to close the connection, and then exit from NNTP_Client by pressing <Ctrl-Z >.

7.2.1.4 Examples

The following example connects to the NNTP server running on wombat.bush.au using TCP as the transport:

```
$ Run [.News_Dist]NNTP_Client
Node: wombat.bush.au
Channel (DECNET, WINtcp)? [DECNET]: wintcp
200 ANU NEWS/NNTP server wombat (V6.1.1). Ready at Fri Feb 25 02:19:55 1994 (posting ok).
quit
205 closing connection - goodbye!
<Ctrl-Z>
$
```

The following example connects to the NNTP server running on the Ultrix system KOALA, using DECnet as the transport:

```
$ Run [.News_Dist]NNTP_Client
Node: KOALA
Channel (DECNET, WINtcp)? [DECNET]: decnet
Decnet NNTP Objects are commonly of the form:
 TASK=NNTP used by most VMS NNTP server implementations
 NNTP=  used by Ultrix NNTP servers
 NEWS=  used by some VMS NNTP server implementations
Decnet Object string? [TASK=NNTP] : NNTP=
Use DECnet stream input calls (normally only with Ultrix NNTP server)? [N]: y
200 ANU NEWS/NNTP server KOALA (V6.1.1). Ready at Fri Feb 25 02:19:55 1994 (posting ok).
quit
205 closing connection - goodbye!
<Ctrl-Z>
$
```

7.2.2 NNTP_TTY

NNTP_TTY is a direct interface to the ANU NNTP server code in NNTP_Server.C, which allows you to simulate a connection to the NNTP server for debugging or informational purposes. Note that this includes only a portion of the actual NNTP server: the command interpreter, authorization checks, and routines which handle items once they have reached the local system. In effect, running NNTP_TTY is identical to establishing a DECnet connection to your local NNTP server, but without involving any of the transport level code. If you want to establish an actual test connection to your local NNTP server (i.e. via a DECnet or TCP connection to your own system), use NNTP_Client instead.

In order to run NNTP_TTY, you will need at least read access to the News database. In addition, if you plan to use the POST command, you will need write access to the News database, and if you intend to 'send' an item via the IHAVE command, you will need write access to News_Manager.

7.2.2.1 Installation

NNTP_TTY.Exe is created in the News_Dist directory by NewsBuild.Com. You should leave it in a location, such as News_Manager, where it is available to you when you want to test the NNTP_Server.C code. It does not pose a security hazard, as it can only be run successfully by users who already have direct access to the local News database.

7.2.2.2 Command syntax

NNTP_TTY is invoked using the DCL Run command; it takes no parameters or qualifiers.

7.2.2.3 Session summary

When invoked, NNTP_TTY simulates a connection to the ANU NNTP server on your system. It performs all authorization checks as if you had established a DECnet connection (i.e. using your username and DECnet nodename). You interact with NNTP_TTY using NNTP commands; the server's responses are displayed at your terminal. The NNTP command set is described in detail in the NNTP standard, RFC977, which is distributed in the News_Doc ZIP file.

Note that, while the connection to the local system is simulated, any actions you take do affect the local News database. Thus, any POSTed items appear in the appropriate newsgroups (and will be forwarded to downstream sites), and any items 'forwarded' using the IHAVE command will appear in batch files in the News_Manager directory.

7.2.2.4 Example

The following session results when NNTP_TTY is invoked from the account NEWSMGR on node WOMBAT:

```
$ Run [.News_Dist]NNTP_TTY
201 ANU NEWS/NNTP server wombat (V6.1.1). Ready at Sat Feb 26 16:03:55 1994 (posting ok).
xconn
281 NNTP connection information: newsmgr@wombat.
quit
205 closing connection - goodbye!
$
```

7.3 Copying News index files

These utilities allow you to recover as much as possible of damaged News index files, or to copy intact index files using locking to prevent other processes from modifying the files you're copying. In general, you won't need to use these programs unless your index files have been scrambled in some way.

It's pretty rare that News index files become corrupted in ways that aren't fixed routinely on the next full Skim pass. As this is written, I haven't heard of a group index file dying (except with the disk it resides on) yet. The item index file is a bit less bulletproof; in particular, some sites report that duplicate key values occasionally creep into their file, causing an RMS error when records with that key value are manipulated. A lot of people would love to know how this happens (and woe betide the bug then), but it's rare enough that it's been difficult to track it down.

However, the world being what it is, bad things do happen occasionally to good files, so News is distributed with these utilities, to allow you to recover as much information as possible. You'll generally find out about a damaged index file pretty quickly, as News will begin to report errors, particularly during Add File and Skim passes, since these actually modify the index files. If you do develop problems with an index file, it's a good idea to shut down all News activity as soon as possible, so as not to compound the problem. The sections of this manual describing NewsShutDown and the logical name News_Stop describe ways to do this. In addition, it's critical that you not perform a Skim pass before you've done whatever

rebuilding you plan on, since Skim is designed to clear up discrepancies in the News database, mostly by deleting "orphaned" items, and you can lose valid newsitems if you skim when the index files are scrambled.

If your item index file has become corrupted, you should first use NewItmFile to recover as much of the damaged item index file as possible. NewItmFile will tell you how far it got before it ran into an error. Then, use Rebuild_Item_File to reconstruct item index records for the rest of the items in your database. This process can take quite some time, if you have a large database. If there are particular groups you don't consider worth the effort of rebuilding, you can directly delete the item files in those groups before running Rebuild_Item_File. News stores the item files on News_Device, in a directory with the same path name as the group name, so the item files for the group news.software.anu-news are located in News_Device:[News.Software.ANU-News]*.Itm;*.

If your group index file is damaged, you should first use NewGrpFile to recover as much of the old file as possible. Then, from within News, re-create all newsgroups which are still missing, either by hand or using a checkgroups message. After this is done, perform a Skim/Newsgroups on any groups you had to re-create, so that the item counts in that group's index record will be accurate. Finally, you'll have to manually reset restricted newsgroups, using the Set Newsgroup/Restrict command. The old membership information is still there, and will be used; you just have to tell News to look for it.

In either case, if you decide that the amount of work involved in recovering parts of the existing database is too great, you're always free to quickly recover what information you can with NewGrpFile or NewItmFile, and then let the next full Skim sort out any remaining inconsistencies, or even to write off your entire current database and start fresh.

7.3.1 NewGrpFile

NewGrpFile generates a new version of the newsgroup index file News_Root:News.Groups by copying the existing version record by record, until an error is encountered. It can be used to partially recover a damaged News.Groups file, and so minimize the number of groups which must be recreated manually, in the event that this file becomes corrupted. How much of the existing file is recovered depends on where in the file the first bad record is encountered. Any records not recovered must be created manually using the News commands Create Newsgroup and Set Newsgroup. Be careful not to execute the News command Skim before the newsgroup index file is fully rebuilt, or else valid item files and directories may be deleted.

In order to use NewGrpFile, you must have SysLck privilege, and read and write access to the News_Root directory.

7.3.1.1 Installation

NewGrpFile.Exe is created in the News_Dist directory by NewsBuild.Com. You should leave it in a location, such as News_Manager, where it is available to you if you need to recover the newsgroup index file.

7.3.1.2 Command syntax

NewGrpFile is invoked using the DCL Run command; it takes no parameters or qualifiers.

7.3.1.3 Session summary

When NewGrpFile is invoked, it attempts to acquire an exclusive lock on the News resource, in order to prevent other processes from modifying the newsgroup index file as it is being copied. It will wait up to 20 minutes for the lock to be granted, and if it is unsuccessful, it will print an error message and proceed to copy the file while allowing shared access.

Once the lock is acquired or the request has timed out, NewGrpFile copies the existing News_Root:News.Groups record by record to News_Root:News.Groups_Wrk. The work file is created with an initial size that is 87% of the size of the existing News.Groups, and is contiguous; if this is not possible, a message is printed, and the file is created contiguous-best-try. Its extension size is 6% of the size of the existing News.Groups. If no errors are encountered, the work file is renamed to News.Groups, and the old copy of News.Groups is deleted. Be careful of this - if NewGrpFile was unable to obtain an exclusive lock on the News resource, another process may still have the old copy of News.Groups open, in which case any changes this process makes will affect only in the now-deleted version of this file, and will not be reflected in the new version.

If any errors are encountered, copying stops immediately, and a message is printed which notes the RMS status code for the error. The work file is not renamed, and the original News.Groups is left unchanged.

7.3.1.4 Example

The following results represent a successful run of NewGrpFile against a small newsgroup index file. Note that the News resource lock was not obtained, so the news manager must insure that no other processes have the old (now deleted) copy of the newsgroup index file open.

```
$ Run [News_Dist]NewGrpFile
Unable to aquire Exclusive Access to the News System:
operation canceled
Continuing . . .

 NEWSGROUP INDEX (#0)
 bionet.molbio.genbank.updates (#6)
 control (#7)
 junk (#3)
 univ.announce (#28)
 wombat.announce (#10)
 wombat.genbank.updates (#11)
 wombat.misc (#9)
 wombat.old_sys_welcome (#8)
End of file reached, 9 records read, 9 records written
Renaming newly created NEWS_ROOT:NEWS.GROUPS_WRK to NEWS.GROUPS
```

7.3.2 NewItmFile

NewItmFile functions like NewGrpFile, except that it operates on the newsitem index file News_Root:News.Items. Similarly to NewGrpFile, NewItmFile can be used to try to recover part of a damaged item index file. Any part of the file which is not recovered must be generated manually by retrieving the individual item files from the appropriate directories under News_Device, and re-adding them using the News command Add File/Retry. Be careful not to execute the News command Skim/File before the newsitem index file is fully rebuilt, or else valid item files may be deleted.

If you are unable to recover a portion of the index file using NewItmFile, you may wish to use Rebuild_Item_File, which creates new item index file records from scratch, instead. Beware that running either of these programs may require quite some time (up to many hours), if there are many items in your News database.

In order to use NewItmFile, you must have SysLck privilege, and read and write access to the News_Root directory.

7.3.2.1 Installation

NewItmFile.Exe is created in the News_Dist directory by NewsBuild.Com. You should leave it in a location, such as News_Manager, where it is available to you if you need to recover the newsitem index file.

7.3.2.2 Command syntax

NewItmFile is invoked using the DCL Run command; it takes no parameters or qualifiers.

7.3.2.3 Session summary

When NewItmFile is invoked, it attempts to acquire an exclusive lock on the News resource, in order to prevent other processes from modifying the newsgroup index file as it is being copied. It will wait up to 20 minutes for the lock to be granted, and if it is unsuccessful, it will print an error message and proceed to copy the file while allowing shared access.

Once the lock is acquired or the request has timed out, NewItmFile copies the existing News_Root:News.Items record by record to News_Root:News.Items_Wrk. The work file is created with an initial size that is 75% of the size of the existing News.Items, and is contiguous; if this is not possible, a message is printed, and the file is created contiguous-best-try. Its extension size is 12% of the size of the existing News.Items. A status message is issued for each group copied completely, and when processing groups with large numbers of items, a running count is displayed. If no errors are encountered, the work file is renamed to News.Items, and the old copy of News.Items is deleted. Be careful of this - if NewItmFile was unable to obtain an exclusive lock on the News resource, another process may still have the old copy of News.Items open, in which case any changes this process makes will affect only the now-deleted version of this file, and will not be reflected in the new version.

If any errors are encountered, copying stops immediately, and a message is printed which notes the RMS status code for the error. The work file is not renamed, and the original News.Items is left unchanged.

7.3.2.4 Example

The following shows a successful run of NewItmFile on a system with a small News database:

```
$ Run [.News_Dist]NewItmFile
Unable to aquire Exclusive Access to the News System:
operation canceled
Continuing . . .
      Newsgroup: junk, Count: 1
      Newsgroup: control, Count: 1
      Newsgroup: wombat.old_sys_welcome, Count: 1
      Newsgroup: wombat.misc, Count: 4
      Newsgroup: wombat.announce, Count: 1
      Newsgroup: wombat.genbank.updates, Count: 1344
End of file reached, 1375 records read, 1375 records written
Renaming newly created NEWS_ROOT:NEWS.ITEMS_WRK to NEWS.ITEMS
```

7.3.3 Rebuild_Item_File

Rebuild_Item_File allows you to reconstruct part or all of the News item index file, in case your current copy has been damaged. It differs from NewItmFile in that it does not try to recover data from the corrupted item index file, but uses the item text files to accumulate as much data as possible, and builds a new index file from scratch. In general, it's best to try recovering as much of a damaged item index file as possible with NewItmFile first, since it's much faster than Rebuild_Item_File. If you don't get back what you need, you can then fall back to Rebuild_Item_File.

If your item index file has been damaged, and you're considering rebuilding it, keep these facts in mind:

- Rebuild_Item_File reads the item text files to obtain information necessary to reconstruct the item index file records. This means the item text files need to be around when Rebuild_Item_File runs. However, the Skim/File command in News deletes any item text files whose item index file record is missing. Therefore, it's crucial that you do not run Skim/Files before rebuilding the index file.
- Since it's reading lots of item text files, Rebuild_Item_File can take a **long** time to run (many hours to a few days, depending on how many newsgroups you carry, system load, I/O rate on News_Device, etc.), during which time News cannot be used, you cannot add new items, and so forth. In light of this, you may decide it's easiest to just write off the lost item files and start fresh. I'm not saying that Rebuild_Item_File is a bad idea - it isn't, but know what you're getting into.

In order to run Rebuild_Item_File, you must have read access to the item text files (usually on News_Device), and write access to the item index file into which new records will be placed (usually in News_Root).

Since you wouldn't be running Rebuild_Item_File if there weren't a problem with the index files, you'll want to keep the other News images from running until the item index file is rebuilt. This can be done via the News_Stop logical name, or by running Rebuild_Item_File under NewsShutDown. Both of these approaches are described in more detail in previous sections of this manual.

7.3.3.1 Installation

You can build Rebuild_Item_File.Exe with the following commands:

```
$ Set Default [.News_Build]
$ CC [-.News_Src]Rebuild_Item_File.C
$ Link/Exec=[-.News_Dist] Rebuild_Item_File.Obj, -
   NewsMsg.Obj, [-.News_Src]Options_C_Link.Opt/Option
$ Delete Rebuild_Item_File.Obj;
```

You may keep this image in any convenient location (e.g. News_Manager).

7.3.3.2 Command Syntax

Rebuild_Item_File should be invoked via a foreign command definition; it will obtain the necessary parameters from its command line. The syntax for invocation is

$ Rebuild_Item_File [*indexfile*] [*itmfil_pattern*]

indexfile - The file specification of the item index file. If this parameter is omitted, it defaults to News_Root:News.Items.

itmfil_pattern - A wildcard file specification which matches all item text files you want Rebuild_Item_File to process. If this parameter is omitted, it defaults to News_Device:[000000...]*.Itm;* (i.e. all the item text files in the local News database).

7.3.3.3 Session Summary

When Rebuild_Item_File starts up, it tries to open the group index file News_Root:News.Groups, and exits if unsuccessful. It then opens the item index file specified by *indexfile*, creating a new file if one does not already exist.

Once the index files are open, Rebuild_Item_File begins processing files which match *itmfil_pattern*. For each file, it first checks that the directory in which the file is found matches the name of a newsgroup in the group index file. If so, it then looks for that item's entry in *indexfile*, and, if it finds the entry, does nothing. If there is no index entry, it build a new record as follows:

- The itm_from field is filled in using the From: header of the item. If this header is missing, the itm_from field is left empty.
- The itm_title field is filled in using the Subject: header of the item. If this header is missing, the itm_title field is left empty.

- The itm_postdate field is filled in using the Date: header of the item. If this header is missing, the current date and time are used.
- The itm_lines field is filled in using the Lines: header of the item. If this header is missing, the itm_lines field is left empty, and the record is marked to show that this field is invalid.
- The itm_id field is filled in using the Message-ID: header of the item. If this header is missing, a message ID of <*grpnum*: *itmnum*> is used instead, where *grpnum* is the group index number of the newsgroup name constructed from item file's directory, and *itmnum* is the item number read from the item file's name.
- The itm_recvdate field is set to the creation date of the item file, or, if that cannot be obtained, to the current date and time.
- The remaining fields in the item index record are zeroed, except for the LINESVALID flag, which is set or reset depending on whether an accurate line count was obtained.

Rebuild_Item_File then attempts to write the new record into the item index file. If successful, it continues on to the next item text file. If it fails because a duplicate key is detected (indicating that another item has the same message ID as this item), it prints an error message, discards the new record, and continues. If any other error occurs, it is signalled immediately, and Rebuild_Item_File continues or exits based on the severity of the error.

Rebuild_Item_File displays a running count of item files processed, and when it is done, displays a message showing the total number of items processed.

7.3.3.4 Example

The following example uses Rebuild_Item_File to create records for all items in the local.* hierarchy in the alternate index file News_Manager:Local.Items. (This has little practical use, but I don't intend to run Rebuild_Item_File against my whole database if I don't need to!).

```
$ Rebuild_Item_File News_Manager:Local.Items -
                    News_Device:[Local.Test]*.Itm;*
Scanning Item files: news_device:[local.test]*.itm;*:

Added 50 news articles.
```

7.4 Miscellaneous maintenance utilities

This section describes those programs supplied in the News distribution which are designed to ease some common tasks which crop up in news management. Feel free to use them or not, as your situation warrants.

7.4.1 BatchNews

BatchNews is a utility designed to convert groups of text files into rnews batch format, so they can be easily fed into news traffic, archived, etc. Each file in the input list is added to the batch as a single newsitem.

7.4.1.1 Installation

BatchNews.Exe is created in the News_Dist directory by NewsBuild.Com. You may place it in any convenient location (e.g. News_Manager).

7.4.1.2 Command syntax

BatchNews should be invoked via a foreign command; it will obtain and parse its own command line. It behaves according to DCL syntax, and accepts the following parameters and qualifiers:

$ BatchNews/Before=*beftime*/Since=*aftime*/NPrefix/Log -
/Size=*batchsize*/Max_Size=*totalsize*/BatchName=*outname* -
/FileList=*listfile input_list*

input_list - A comma-separated list of input file specifications to be processed. Wildcards are allowed in any of the file specifications.

/Before=*beftime* - Specifies that files created before *beftime* are to be processed. *Beftime* should be a valid VMS absolute time. This qualifier cannot be negated.

/Since=*aftime* - Specifies that files created after *aftime* are to be processed. *Aftime* should be a valid VMS absolute time. This qualifier cannot be negated.

/NPrefix - Causes the character 'N' to be prepended to each line of the output batch file. /NoNPrefix cancels the effect of a previous /NPrefix.

/Log - Causes the names and sizes of input files processed to be displayed. In addition, if multiple output batch files are created, the name and size of each batch file is displayed when it is closed. /NoLog cancels the effect of a previous /Log.

/Size=*batchsize* - Specifies that a new batch file should be created when size of the current batch file exceeds *batchsize* bytes. BatchNews finishes writing the current input file before creating a new batch file, so newsitems do not span batch files. Successive batch files have the name *outnameNNN*, where *NNN* is a three digit decimal number beginning at 001 and increasing by one for each batch file. If files with names *outname*001 to *outnameXXX* already exist, then BatchNews starts the current set of batch files at *outnameNNN*, where *NNN* = *XXX* + 1. *Batchsize* may be a decimal number, an octal number beginning with '%O', or a hexadecimal number beginning with '%X'. /NoSize cancels the effect of a previous /Size. This qualifier is ignored if the batch is being written to Sys$Output.

/Max_Size=*totalsize* - Specifies that all processing of input files should stop after *totalsize* bytes have been written to batch files. BatchNews finishes writing the current input file before stopping. *Totalsize* may be a decimal number, an octal number beginning with '%O', or a hexadecimal number beginning with '%X'. /NoMax_Size cancels the effect of a previous /Max_Size.

/BatchName=*outname* - Specifies the name template of the output batch file. If /BatchName is not specified, the output batch is written to Sys$Output. When writing a batch to Sys$Output, only a single output file (not limited in size) will be produced. This qualifier cannot be negated.

/FileList=*listfile* - Specifies that the names of input files are contained in *listfile*. *Listfile* should be a text file containing one input file specification per line. Wildcards are not allowed. If files are specified both via *input_list* and *listfile*, the files in *input_list* are processed first, followed by those in *listfile*. If batching is limited by *totalsize* while files still remain in *listfile*, a new version of *listfile* is created, which contains the remaining (unprocessed) names from the original *listfile*.

/Help - Displays a summary of BatchNews command syntax on Sys$Error and then exits.

7.4.1.3 Session Summary

When invoked, BatchNews simply reads each file matching the input file specification, and copies it into the output file, adding an appropriate rnews line before the file. The contents of the input file are not checked or modified in any way. This process continues until the list of input files is exhausted, or until the number of bytes specified with the /Max_Size qualifier has been exceeded. Output is directed to Sys$Output, unless the /BatchName qualifier is specified, in which case its value is the output file name. If the /Size qualifier is specified, several output files may be created, with names derived as described above.

Note that if the size specified by /Size or /Max_Size is exceeded, BatchNews does not take action until it has finished processing the current input file. This means that output files may be larger than the value specified by /Size, and more data may be copied than specified by /Max_Size. Therefore, you should consider these qualifiers guidelines to be used by BatchNews, rather than hard upper limits.

7.4.1.4 Examples

The following example combines all files with type .Rno in the current directory into batch files with the base name Runoff.Batch. When a batch file exceeds 51000 bytes (~100 disk blocks), subsequent input files are placed in a new batch file. If more than one batch file is generated, their names will be Runoff.Batch001, Runoff.Batch002, Runoff.Batch003, and so on.

```
$ BatchNews/BatchName=Runoff.Batch/Size=51000 *.Rno
```

The following example copies the files named in BatchList.Txt into an rnews batch, which is written to Sys$Output (e.g. the log file of a batch job).

```
$ Create BatchList.Txt
Project_Common:Stats.Today
Project_Common:Transaction_List.Rpt
Project_Admin:BugFix.InProgress
<Ctrl-Z>
$ BatchNews/FileList=BatchList.Txt/Max_Size=20000
```

If, say, Stats.Today is 2 kB and Transaction_List.Rpt is 25 kB, BatchNews would not process BugFix.InProgress, but would instead create a new version of BatchList.Txt containing only BugFix.InProgress.

7.4.2 Cachem

If you are using NNTP message ID caching, the Cachem utility allows you to examine and modify the message ID cache file. It is used principally for debugging the cache code. Under normal circumstances, the cache takes care of itself, so you do not need to run this program unless you develop a problem with the cache, or you're curious about its innards.

In order to run Cachem, you must have SysLck and SysGbl privileges set.

7.4.2.1 Installation

Cachem.Exe is created in the News_Dist directory by NewsBuild.Com. You may place it in any convenient location (e.g. News_Manager).

7.4.2.2 Command syntax

Cachem is invoked using the DCL Run command; it takes no parameters or qualifiers. At the prompt

Option (h for Help)?

it accepts the following single character commands:

A - Add a message ID to the cache. Cachem will prompt for the message ID to add.

C - List the number of message IDs inserted into the cache or rejected because they were already in the cache.

D - Dump the contents of the cache for a range of entry numbers. Cachem will prompt for the first and last index (entry number) in the range.

E - Exit Cachem.

F - Find a message ID in the cache. Cachem will prompt for the message ID to find, and will display a message indicating whether the message ID was found in the cache.

L - List a range of message IDs to a file. Cachem will prompt for a file name, and the first and last indices (entry numbers) in the range you wish to list. The resulting file is a normal (Stream-LF) text file with 1 message ID per line.

R - Reset the insertion/rejection counters used to generate the counts displayed by the C command.

S - Search for a hash list greater than a certain depth. Cachem prompts for the hash list index at which to start and the list depth, and prints the first list found after the index given whose depth is greater than or equal to the depth value given.

V - Verify integrity of the cache and print statistics. If an error is found in any of the cache entries, a message is displayed. Once all entries have been searched, statistics describing the cache are displayed (see example below). As it examines cache entries, it displays a running count of its progress.

Z - Zero the cache. This deletes all entries, and leaves the cache empty.

7.4.2.3 Session summary

When Cachem starts up, it writes an entry indicating this to the cache error log (News_ Manager:News_Cache.Log). After that, it's pretty straightforward - Cachem prompts for a command, and then does what you ask. When you are finished, it writes an entry to the cache error log indicating that it is exiting. This allows you to keep track of when Cachem was run, and potentially used to reset the cache.

In order to understand the statistics Cachem displays, it helps to know a little about how the message ID cache works. When a message ID string is passed to the cache code, it generates a 32 bit hash value from the string, and uses the low order 14 bits of this hash value as an index into an array of linked lists containing the message IDs in the cache with that hash list index value. It then walks the list in order to determine whether the message ID string it is checking is already in the cache. If it was asked to find the message ID string, it will just return 1 if the ID string is present, or 0 if it is absent. If it was asked to add the message ID string to the cache, and the ID is not already present, it will add a new entry to the appropriate linked list and increment the insertion counter; if the ID was already present, it does not reinsert it, but increments the rejection counter. As new IDs are added to the cache, old ones are removed to make space for them.

When generating statistics in response to the V command, it checks each list, and displays the total number of lists, as well as information on the distribution of list depth (the number of entries in a given hash list). In addition it displays the number and percentage of multiple links (i.e. the number of items which are not the first element in their hash list), and the number and percentage of collisions (i.e. the number of times the code has had to compare a string to an existing entry and move on, instead of finding it immediately when it looks up the hash index).

7.4.2.4 Examples

The following example uses Cachem to obtain information about the usage of the cache:

Example 7–1: Using Cachem to Obtain Cache Usage Information

```
$ Run [.News_Dist]Cachem
Option (h for Help)? C
Number of insertions     = 15695
Number of rejections     = 16155
Percentage of rejects    =  50.7
Current index in cache   = 7505
Date counts last reset   =  6-JUL-1993 16:06:40.36
Option (h for Help)? V
Invalid back pointer in hash index 13250 (ptr=318) :
- bptr = 0, should be 5414
16000...

Summary :
---------
Number of terminated hash linked lists = 16383
Number of non-terminated hash linked lists = 0
Number of zero-length hash linked lists = 9872
Longest hash linked list has depth = 5
Average hash linked list depth = 1.258025
Total number of links = 8191
Number of used hash links = 8191

Histogram of hash linked list depths :
Depth = 0 : 9872
Depth = 1 : 5081
Depth = 2 : 1213
Depth = 3 : 187
Depth = 4 : 27
Depth = 5 : 3
Number of multiple links = 1430
Percentage of multiple links = 17.458186
Number of collisions = 1680
Percentage of collisions = 20.510316
Option (h for Help)? E
$
```

This example shows how to check whether a particular message ID is present in the cache:

Example 7–2: Using Cachem to find a Particular Message ID

```
$ Run [.News_Dist]Cachem
Option (h for Help)? F
Id? <2lfvt6$3n6@net.bio.net>
Found in cache list
Option (h for Help)? F
Id? <carthago$delenda$est@senatus.roma.org>
Not found in cache list
Option (h for Help)? e
$
```

7.4.3 FeedCheck

FeedCheck examines the logs of your NewsAdd job and produces summaries of traffic to and from adjacent nodes. In addition, it can generate statistics describing items junked during Add File processing.

As of the current release, a few minor bugs remain in FeedCheck, so you may encounter some spurious error messages, but it's still a very useful tool for quickly getting an overview of news traffic.

7.4.3.1 Installation

FeedCheck.Exe is created in the News_Dist directory by NewsBuild.Com. You may place it in any convenient location (e.g. News_Manager).

7.4.3.2 Command Syntax

FeedCheck should be invoked via a foreign command; it will obtain and parse its own command line. It behaves according to DCL syntax, and accepts the following parameters and qualifiers:

```
$ FeedCheck/HostName=remhost/Junk/Verbose=msglvl -
        /Before=beftime/Since=aftime/Output=outfile input_list\ |
```

input_list - A comma-separated list of input file specifications to be processed. Wildcards are allowed in any of the file specifications. This parameter is required.

/Hostname=*remhost* - Default host to use as the source of incoming items from batches which do not include a host name. At present this qualifier is not used by FeedCheck, and *remhost* is ignored. This qualifier cannot be negated.

/Junk - Directs FeedCheck to print a summary of items which have been junked during Add File processing, including newsgroups in the headers of junked items, and the sites from which you received them.

/Verbose=*msglvl* - Indicates that FeedCheck should print status messages as it processes input files. The value of *msglvl* is interpreted as follows:

Msglvl	Effect
1	Print a summary of the disposition of items to and from each site in each file processed, as well as printing an overall summary for each site.
2	Print a status message each time a new input file is processed ("Processing Log File:") and each time the log entries for a new batch of news items is processed ("Analyzing Log for Batch:"). In addition, the summary report for each log file shows the name of the file, the number of lines in the file, and the number of lines which could not be parsed.
3	Print a status message each time a new input file specification matching *input_list* is checked ("Checking File:"). This differs from the "Processing Log File:" message in that the latter is printed only when a file is actually examined and its contents included in the statistics generated by FeedCheck(i.e. is not excluded by *beftime* or *aftime*, and can be read successfully).

The effects of each level are cumulative, that is, specifying a *msglvl* of 2 produces both the effects for level 1 and level 2, etc. If *msglvl* is not specified with /Verbose, it defaults to 1.

/Before=*beftime* - Specifies that files created before *beftime* are to be processed. *Beftime* should be a valid VMS absolute time. This qualifier cannot be negated.

/Since=*aftime* - Specifies that files created after *aftime* are to be processed. *Aftime* should be a valid VMS absolute time. This qualifier cannot be negated.

/Output=*outfile* - Directs FeedCheck to write results to *outfile*, instead of Sys$Output.

/Help - Displays a summary of FeedCheck command syntax on Sys$Error and then exits. You must provide *input_list* on the command line when using this qualifier, but it is ignored. This qualifier cannot be negated.

7.4.3.3 Session Summary

FeedCheck scans each file matching the criteria set forth in *input_list*, *beftime* and *aftime*, looking for the output produced by News' Add File command (which explains why it's usually run on log files from your NewsAdd job). When it finds these messages, it uses them to compile statistics about the number and size of items received from upstream sites, what happened to those items during Add File processing, and what was forwarded to downstream sites. Once all input files have been read, it displays a table on Sys$Output, which lists the following statistics, by site:

- For each site from which more than one batch was received, the number and total size of incoming batches ("Files"), the number of items successfully added ("Accept"), junked ("Junked"), rejected as duplicates ("Dups"), or rejected for other reasons ("Reject"). In addition, it lists the percentage of all accepted and rejected items which originated at that site.

- For each site to which items were forwarded, it lists the number of message IDs added to a type 'N' spool file ("I've") and the number and size of items added to type 'M' or 'B' spool files ("Xmit" and "Bytes"). (The "I've" and "Xmit" fields aren't related - there's no way to tell from an Add File log what fraction of message IDs offered to a downstream site using IHAVE were actually transmitted to that site.)
- Totals for each category for all sites.

You can get additional information, or direct the output to a file, using the command line qualifiers described above.

There are a few minor bugs remaining in FeedCheck, so don't be surprised if you see a number of messages about unexpected host changes. Also, since it only prints statistics for upstream sites sending you more than one batch, the totals for received items shown at the bottom of the report may not equal the sum of the individual entries displayed. These idiosyncrasies aside, FeedCheck is a very nice way to keep up with traffic at a glance. For instance, you may want to do something like:

```
$ FeedCheck/Before=Today/Since=Yesterday/Junk -
           /Output=Sys$Scratch:FeedCheck.Tmp -
           News_Manager_Dev:[Log]NewsAdd.Log;*
$ subj = "FeedCheck daily report for " + -
         F$CvTime("Yesterday","Absolute","Date")
$ Mail/Subject="''subj'" Sys$Scratch:FeedCheck.Tmp Me
$ Delete/NoLog/NoConfirm Sys$Scratch:FeedCheck.Tmp;*
```

every day, so you'll develop a feel for traffic patterns, and spot any problems quickly.

7.4.3.4 Examples

The following example generates a basic traffic report from the previous day's NewsAdd logs for node SENATUS (ca. 190 B.C):

```
$ FeedCheck/Before=Today/Since=Yesterday/Junk -
_$News_Manager_Dev:[Log]NewsAdd.Log;*
```

The output from this command might be something like this (the table has been wrapped to fit within the margins of this document):

```
Site:     Files:Accept: Dups:Junked:Reject:Total:
=================================================
carthago  2(2.5K)    2     .      .     .      2
causae    3(2.5M)  717     .      .     .    717
censores       .     .     .      .     .      .
scipio         .     .     .      .     .      .

            Files:Accept: Dups:Junked:Reject:Total:
TOTAL:   72(2.5M)  720     .      .     .    720

 %Bad:%TAcpt: I've: Xmit:Bytes:
==============================
   .   0.3%     .     .     .
   .  99.6%     .     .     .
   .      .     .   717  2.5M
   .      .     2     .     .

 %Bad:%TAcpt: I've: Xmit:Bytes:
   .      .    2.   717  2.5M
```

The following example shows the result of scanning the same logs with this slightly modified command (again, the table has been wrapped to fit within the margins):

```
$ FeedCheck/Before=Today/Since=Yesterday/Junk/Verbose=3 -
_$News_Manager_Dev:[Log]NewsAdd.Log;*
```

The output in this case might be something like:

```
Checking File: NEWS_MANAGER_DEV:[LOG]NEWSADD.LOG;143
Checking File: NEWS_MANAGER_DEV:[LOG]NEWSADD.LOG;142
Checking File: NEWS_MANAGER_DEV:[LOG]NEWSADD.LOG;141
Processing Log File: NEWS_MANAGER_DEV:[LOG]NEWSADD.LOG;141
Checking File: NEWS_MANAGER_DEV:[LOG]NEWSADD.LOG;140
Processing Log File: NEWS_MANAGER_DEV:[LOG]NEWSADD.LOG;140
Checking File: NEWS_MANAGER_DEV:[LOG]NEWSADD.LOG;139
Processing Log File: NEWS_MANAGER_DEV:[LOG]NEWSADD.LOG;139
Analyzing Log for Batch: USER1:[NEWS]NNTP_900308171410_8C27.BATCH;1
Checking File: NEWS_MANAGER_DEV:[LOG]NEWSADD.LOG;138
Processing Log File: NEWS_MANAGER_DEV:[LOG]NEWSADD.LOG;138
Checking File: NEWS_MANAGER_DEV:[LOG]NEWSADD.LOG;137
Processing Log File: NEWS_MANAGER_DEV:[LOG]NEWSADD.LOG;137
Checking File: NEWS_MANAGER_DEV:[LOG]NEWSADD.LOG;136
Processing Log File: NEWS_MANAGER_DEV:[LOG]NEWSADD.LOG;136
Analyzing Log for Batch: USER1:[NEWS]NEWS_LOCAL.BATCH;1
Checking File: NEWS_MANAGER_DEV:[LOG]NEWSADD.LOG;135
Processing Log File: NEWS_MANAGER_DEV:[LOG]NEWSADD.LOG;135
Checking File: NEWS_MANAGER_DEV:[LOG]NEWSADD.LOG;134
Processing Log File: NEWS_MANAGER_DEV:[LOG]NEWSADD.LOG;134
Checking File: NEWS_MANAGER_DEV:[LOG]NEWSADD.LOG;133
Processing Log File: NEWS_MANAGER_DEV:[LOG]NEWSADD.LOG;133
Checking File: NEWS_MANAGER_DEV:[LOG]NEWSADD.LOG;132
Processing Log File: NEWS_MANAGER_DEV:[LOG]NEWSADD.LOG;132
Analyzing Log for Batch: USER1:[NEWS]NNTP_900308101817_863D.BATCH;1
Checking File: NEWS_MANAGER_DEV:[LOG]NEWSADD.LOG;131
Processing Log File: NEWS_MANAGER_DEV:[LOG]NEWSADD.LOG;131
Checking File: NEWS_MANAGER_DEV:[LOG]NEWSADD.LOG;130
Processing Log File: NEWS_MANAGER_DEV:[LOG]NEWSADD.LOG;130
Checking File: NEWS_MANAGER_DEV:[LOG]NEWSADD.LOG;129
Processing Log File: NEWS_MANAGER_DEV:[LOG]NEWSADD.LOG;129
Analyzing Log for Batch: USER1:[NEWS]NNTP_900308034825_823A.BATCH;1
Checking File: NEWS_MANAGER_DEV:[LOG]NEWSADD.LOG;128
Processing Log File: NEWS_MANAGER_DEV:[LOG]NEWSADD.LOG;128
Analyzing Log for Batch: USER1:[NEWS]NEWS_LOCAL.BATCH;1
Checking File: NEWS_MANAGER_DEV:[LOG]NEWSADD.LOG;127
Processing Log File: NEWS_MANAGER_DEV:[LOG]NEWSADD.LOG;127
Checking File: NEWS_MANAGER_DEV:[LOG]NEWSADD.LOG;126
Processing Log File: NEWS_MANAGER_DEV:[LOG]NEWSADD.LOG;126
Checking File: NEWS_MANAGER_DEV:[LOG]NEWSADD.LOG;125
Processing Log File: NEWS_MANAGER_DEV:[LOG]NEWSADD.LOG;125
Checking File: NEWS_MANAGER_DEV:[LOG]NEWSADD.LOG;124
Checking File: NEWS_MANAGER_DEV:[LOG]NEWSADD.LOG;123
Checking File: NEWS_MANAGER_DEV:[LOG]NEWSADD.LOG;122
Checking File: NEWS_MANAGER_DEV:[LOG]NEWSADD.LOG;121
Checking File: NEWS_MANAGER_DEV:[LOG]NEWSADD.LOG;120
Checking File: NEWS_MANAGER_DEV:[LOG]NEWSADD.LOG;119
    .      .       .
Checking File: NEWS_MANAGER_DEV:[LOG]NEWSADD.LOG;3
Checking File: NEWS_MANAGER_DEV:[LOG]NEWSADD.LOG;2
Checking File: NEWS_MANAGER_DEV:[LOG]NEWSADD.LOG;1

Site:     Files:Accept: Dups:Junked:Reject:Total:
==================================================
Log File: NEWS_MANAGER_DEV:[LOG]NEWSADD.LOG;132,
gallia                    1       0       0
censores       .     .     .      .     .      .
Log File: NEWS_MANAGER_DEV:[LOG]NEWSADD.LOG;136,
carthago       1       0       0        0
Log File: NEWS_MANAGER_DEV:[LOG]NEWSADD.LOG;128,
carthago       1       0       0        0
carthago   2(2.5M)    2     .      .     .      2
scipio         .     .     .      .     .      .
Log File: NEWS_MANAGER_DEV:[LOG]NEWSADD.LOG;129,
causae                   10       0       0
Log File: NEWS_MANAGER_DEV:[LOG]NEWSADD.LOG;132,
causae                  166       0       0
Log File: NEWS_MANAGER_DEV:[LOG]NEWSADD.LOG;139,
causae                  541       0       0
causae    3(2.5M)  717     .      .     .    717

          Files:Accept: Dups:Junked:Reject:Total:
TOTAL:   72(2.5M)  720     .      .     .    720
```

Note in this report that one item was received from site gallia, but since that site sent only one batch, it didn't appear in the brief report shown in the previous example, so the "TOTAL" statistics appeared to be off by one item.

7.4.4 NewsShutDown

NewsShutDown is designed to let you perform arbitrary tasks which require exclusive access to resources (particularly the local database) used by News. In most cases, it will produce faster results than using the older News_Stop logical name to force other News images to exit.

In order for NewsShutDown to function properly, all programs in the News distribution must participate in a common locking scheme, so that everyone is aware of everyone else's presence. From your perspective, this means that any time you run an image from the News distribution, it must have SysLck privilege available to it. You can do this by running the image in a process with SysLck set, or you can install the image with SysLck. If you don't do this, the image will assume you didn't want it to participate in the locking scheme, and will sail happily along. This won't hurt it, but there will be no record of that image's activity, so a utility like NewsShutDown may think it has gained exclusive access to the News resource, when in fact it hasn't. The News images are a little picky here - they check whether SysLck is already set, but won't attempt to set or reset it themselves.

7.4.4.1 Installation

NewsShutDown.Exe is created in the News_Dist directory by NewsBuild.Com. You may place it in any convenient location (e.g. News_Manager).

7.4.4.2 Command Syntax

NewsShutDown should be invoked via the DCL Run command. It takes no parameters or qualifiers on the command line, but gets the information it needs to function from two logical names:

- News_Locked_Command - should translate to the command you wish executed once an exclusive lock on the News resource has been obtained.
- News_Locked_Wait_Minutes - should translate to the number of minutes (decimal) NewsShutDown should wait for an exclusive lock to be granted before giving up. If this logical name is not defined, NewsShutDown will wait for 1 minute before giving up.

Both of these logical names are translated using the usual order of logical name tables and access modes, so they can (and usually are) defined in the process table in supervisor or user mode just before invoking NewsShutDown.

7.4.4.3 Session Summary

When NewsShutDown is run, it tries to translate the logical name News_Locked_Command, and, if unsuccessful, prints an error message and exits. If it succeeds in obtaining the command, it tries to enqueue an exclusive lock on the News resource. This will trigger blocking ASTs put in place by other News images which are currently running. These ASTs set a flag which the image checks the next time it looks for user input, and, if the flag is set, the image exits with status SS$_DEADLOCK (decimal 3594). News.Exe, NNTP_Xmit, and NNTP_Xfer will exit within 20 minutes in any case. The single threaded NNTP servers will close the connection and exit as soon as they notice that the flag is set, but not before, so they may wait some time if the connection is idle or slow. The multithreaded NNTP servers will close all open connections when they notice that the flag is set, but will remain active and keep the News index files open. These servers must be stopped manually in order to release the index files.

If an error occurs in enqueueing the lock, NewsShutDown exits immediately with the error status as its exit status; otherwise, it will wait News_Locked_Wait_Minutes for the lock to be granted. If the lock request still hasn't been granted, NewsShutDown exits with status SS$_CANCEL (decimal 2096).

Once it has obtained the exclusive lock, NewsShutDown spawns a subprocess, passing it the translation of News_Locked_Command as the command to be executed. When this command completes, the subprocess is deleted, and NewsShutDown exits with the final status of the subprocess as its exit status. If you want more than one command to be executed in the subprocess, place them in a DCL procedure, and invoke that procedure via News_Locked_Command. Remember that NewsShutDown is holding an exclusive lock on the News resource throughout the life of the subprocess, so you cannot execute News images via News_Locked_Command.

7.4.4.4 Examples

The following example uses NewsShutDown to force other News images to exit as soon as possible, and then keeps them from restarting using the News_Stop command. This is one way to perform an orderly shutdown of News activity on a node for an extended time.

```
$ oldprv = F$SetPrv("SysLck,SysNam")
$ Define/User News_Locked_Command "Define/System/Executive News_Stop 1"
$ Define/User News_Locked_Wait_Minutes 30
$ Run News_Manager:NewsShutDown.Exe
$ sts = $Status
$ If .not.sts Then -
    Write Sys$Error "%NEWSSTOP-F-ERROR, unable to shut down ANU News"
$ oldprv = F$SetPrv(oldprv)
$ Exit sts
```

For an example which uses NewsShutDown to obtain exclusive access to the News index files in order to optimize them, see the section above titled "Obtaining exclusive access to the News database".

Appendix A
Reporting Problems with ANU News

If you run into any difficulties setting up or using ANU News, the best source for advice is the newsgroup news.software.anu-news, or the associated BITNET mailing list ANU_News@UBVM.BitNet. (There is a bidirectional gateway between the list and the newsgroup, so you need only report problems by one of these routes.) When reporting a problem to the newsgroup, please try to include the following information:

- The version of News (this appears at the bottom right hand corner of the display when News.Exe is invoked, or in the startup banner for the NNTP server). Please include the version number and date (e.g. News 6.1b10 23-Jan-1995).
- The names of any patches you have applied to the base release at your site. If you use NewsPatch.Com to apply patches, you can simply include PatchList.H in your posting.
- A brief description of the problem, including what you were trying to accomplish when the problem occurred.
- The hardware model and VMS version under which you are running News.
- If you built the executable images at your site, the C compiler you used, and any options you specified when invoking NewsBuild.Com.
- The complete specifications of any relevant News logical names. You can simply include the output of:

  ```
  $ Show Logical/Full News*
  $ Show Logical/Full NNTP*
  ```

  if you're not sure which logical names are relevant.
- If the problem occurs when connected to a remote site, the NNTP transport protocol you are using to communicate with that site.
- The text of any commands you entered when the problem occurred.
- The complete text of any error messages generated by News. News.Exe displays error messages on the screen or in the batch logfile. If you are running News in screen mode, you can review these messages using the Show Message command. If an updated NewsRC file is written at the end of the session, News records the error messages at the end of

this file. Error messages generated by the NNTP server are written to the NNTP server log file.

- If the problem occurs when feeding items to a downstream site, the News.Sys entry for that site.
- Relevant portions of other News configuration files.
- Any other information about News or system configuration you htink is relevant. When in doubt, err on the side of including additinal information.

This may seem like a lot of information to gather, but in most cases, only some of these items apply, and accurate information will greatly assist others in figuring out what you're seeing.

When you post your problem report, please remember to include a valid return address (if at all possible, place it in a From: or Reply-To: header). This is, of course, particularly important if the problem has disrupted your access to news.

Appendix B
RMS Tuning of the News Index Files

This appendix is based on notes compiled by Rand Hall <rand@merrimack.edu> and posted to the newsgroup news.software.anu-news in 1990. It explains how to use RMS global buffers on your News index files to improve efficiency of lookups by caching the index records in memory.

To do this you need to have exclusive access to the index files News.Groups and News.Items. (See above for explanation of locking the News database.) First, you'll need to analyze the structure of the index files. From the News_Root directory, say

```
$ Analyze/RMS/FDL/Output=Groups.FDL News.Groups
$ Edit/FDL Groups.FDL
  . . .
        Main Editor Function            (Keyword)[Help] : Invoke
  . . .
        Editing Script Title            (Keyword)[-]    : Optimize

        An Input Analysis File is necessary for Optimizing Keys.

        Analysis File file-spec (1-126 chars)[null]
        : Groups.FDL
  . . .
        Graph type to display           (Keyword)[Line] : Line
  (Press <Return> to accept defaults.)
        Which File Parameter    (Mnemonic)[refresh]     : FD
        Text for FDL Title Section      (1-126 chars)[null]
        : <Return>
```

In the display, note the maximum values for Number of Buckets in Index, Suggested Bucket Size, and Pages Required to Cache Index. Press < Return> (or select FD at the graph) through all of the KEY 1 stuff (technical, eh?) until you get back to the main menu. Select Exit.

Buckets in Index should be Pages Required to Cache Index divided by Bucket Size. Edit Groups.FDL and add the line GLOBAL_BUFFER_COUNT *n* in the FILE section of the FDL file, where *n* is the value of Pages Required to Cache Index.

If you want to be more productive, look at the compression stats in the original (before Edit/FDL was invoked) Groups.FDL, in the Analysis of Key sections. If a key or record is compressed (look in KEY description at top of file) and the stats are lousy (<50%) turn compression off by editing the new Groups.FDL.

The parameters in the FDL file are now optimized. To apply it to News.Groups, obtain exclusive access to teh News database, and say:

```
$ Convert/NoFast/FDL=Groups.FDL News.Groups News.Groups
```

Now (huff, pant) do the same for News.Items (this takes a looooong time).

Each file you slap global buffers on needs one global section, so you may need to increase the SYSGEN parameter GBLSECTIONS. You may also need to boost the SYSGEN paramters GBLPAGES, GBLPAGFIL, and RMS_GBLBUFQUO. The following DCL procedure will show you what you need for each file:

```
$ !  RMSGLOB_SYSGEN.COM
$ !+
$ !  Author: Rand P. Hall  <rand@merrimack.edu>
$ !
$ !  Command procedure to estimate RMS global buffer SYSGEN
$ !  units needed for one indexed file.  This command procedure
$ !  can be adapted to accumulate these units across a number
$ !  of indexed files.
$ !
$ !  CAUTION:  This sample command procedure has been tested
$ !  using VMS 5.4-3.  However, we cannot guarantee its
$ !  effectiveness because of the possibility of error in
$ !  transmitting or implementing it.  It is meant to be used
$ !  as a template for writing your own command procedure, and
$ !  it may require modification for use on your system.
$ !
$ !-
$ ON WARNING THEN GOTO DONE
$ IF P1 .EQS. "" THEN $INQUIRE P1 "FILE SPEC "
$ P1 = F$SEARCH (P1, 1)                  ! Returns full filespec
$ gbc = F$FILE_ATTRIBUTES (P1,"GBC")     ! GBC available as of
$ IF gbc .EQ. 0 THEN GOTO NOGBC          ! VMS V5.2
$ bks = F$FILE_ATTRIBUTES (P1,"BKS")
$ RMS_GBLBUFQUO = gbc
$ GBLSECTIONS   = 1
$ k             = 64 + (gbc * 48)        ! GBH and GBD descriptor bytes
$ GBLPAGFIL     = ((gbc * bks * 512) + k + 1023)/512
$ GBLPAGES      = GBLPAGFIL + 2          ! plus 2 stopper pages
$ WRITE SYS$OUTPUT ""
$ WRITE SYS$OUTPUT "SYSGEN values needed by ",P1
$ WRITE SYS$OUTPUT ""
$ WRITE SYS$OUTPUT "  RMS_GBLBUFQUO = ",RMS_GBLBUFQUO
$ WRITE SYS$OUTPUT "  GBLSECTIONS   = ",GBLSECTIONS
$ WRITE SYS$OUTPUT "  GBLPAGES      = ",GBLPAGES
$ WRITE SYS$OUTPUT "  GBLPAGFIL     = ",GBLPAGFIL
$ GOTO DONE
$ NOGBC:
$   WRITE SYS$OUTPUT ""
$   WRITE SYS$OUTPUT "Following file did not have global buffers set: "
```

```
$   WRITE SYS$OUTPUT "     ",P1
$ DONE:
$   EXIT
```