第16章 C-News

Netnews所用的软件包中，最常用的是 C-News。你可从ftp://ftp:cs.toronto.edu/pub/c-news/c-news.tar.Z下载它。它是为通过UUCP链接传输新闻的站点设计的。本章将详细讨论关于C-News的几个重要概念及其基本安装和维护。

C-News将自己的配置文件保存在 /usr/lib/news内，它的多数二进制程序都保存在/usr/lib/news/bin目录内，文章则保存在/var/spool/news目录下。实际上，你还应该保证这些目录内的所有文件都归用户新闻和组新闻拥有。许多问题都是因为 C-News不能对这些文件进行访问而引起的。对你而言，在行动之前，一定要利用 su，使自己成为用户新闻，这是非常有用的。唯一例外的是setnewsids，它用于设置某些新闻程序真正的用户 ID。它的拥有者必须是root，而且必须设置setuid位。

接下来，为大家详细讲解所有的 C-News配置文件，并向大家展示怎样才能使自己的站点正常运行。

16.1 新闻投递

文章被发送到C-News的方式有许多种。本地用户投递一篇文章时，新闻阅读机通常会把它交给inews命令，该命令将文章头信息补充完整。从远程站点发来的新闻，不管是一篇单独的文章，还是整个的batch，都被交给rnews命令处理，该命令将它保存在/var/spool/newsin.coming目录中，稍后再由newsrun从这个目录中将其剔出来。但是，这两种方法中，不管采用哪一种，文章最终都会被交给relaynews命令处理。

对每篇文章来说， relaynews命令都会首先检查它是否在本地站点出现过，也就是说在历史文件内查找其消息ID。重复性的文章将被丢弃。然后，relaynews命令再查看Newsgroup:header字段，找出本地站点是否请求得到新闻组内的文章。如果是，新闻组就会被列入 active文件，relaynews试着将本地站点请求的文章保存在新闻假脱机区内的相应目录下。如果这个目录不存在，它就会创建一个。然后，该文章的消息 ID被记入历史文件内。另一方面，如果本地站点没有请求文章， relaynews就会把它丢弃。

就“进入文章”(incoming article)已经被投往一个新闻组而言，如果 relaynews不能保存进入文章是因为这个组没有包含在 active文件内，那么这篇文章就会被移入垃圾组（你站点上存在的新闻组和你站点希望接收的新闻组是有区别的。举个例子来说，订阅列表可能指定comp.all，意思是comp结构下的所有新闻组，但对你的站点来说，则是只列出 active文件内comp新闻组的数目。投向这些新闻组的文章会被移入垃圾组）。relaynews还可以检查出陈旧的或过时的文章并将它们丢弃。如果由于其他原因，进入的 batch文件失败，它就会被移到/var/spool/news/in.coming/bad，错误消息也会被记录下来。

随后，利用为每个特定站点指定的传输方式，这篇文章被中转到其他的所有站点（它们都请求了这些组的新闻）。为了保证不将文章发到已经看过它的站点，要对每个目标站点和该文章的Path:header字段进行核对，该字段中包含一份站点列表，表明这篇文章已经经过了哪

些站点，它采用的是路径格式。只有目标站点名不在这份列表内时，这篇文章才会被转发给它。

虽然C-News可用于NNTP环境中，但它常用于UUCP站点间的新闻中转。为了把新闻投递到一个远程UUCP站点 —— 无论是单独的文章，还是 batch文件 —— 需要在远程站点上利用 uux来执行rnews命令，并按标准输入，将文章或 batch文件发送给它。

在具体站点上采用“成批处理法”（batching）时，C-News不会立即发送任何进入的文章，而是将其路径名添加到一个文件内，该文件通常是 out.going/site/togo。通过crontab条目，周期性地执行batcher程序（注意，这个条目应该是新闻的 crontab，目的是不干扰文件的访问许可权），把文章放入一个或多个文件内，对它们进行选择性地压缩，最后在远程站点把它们发送给rnews。

16.2 安装

要安装C-News，须通过untar操作，把文件放入恰当的位置，并对随后黑点列出的配置文件进行编辑（这些配置文件都位于 /usr/lib/news内）。其格式将在随后进行说明。

注意 看了下面的描述，大家便可了解新闻是如何通过relaynews的：

1) 你站点上存在的新闻组和你的站点希望接收的新闻组之间，是有区别的。举个例子来说，订阅列表可能指定comp.all，意思是comp结构下的所有新闻组，但对你的站点来说，则是只列出active文件内comp新闻组的数目。投向这些新闻组的文章会被移入垃圾组。

2) 注意，它应该是新闻的crontab，目的是不干扰文件的访问许可权。

■ sys —— 虽然使用all/all始终是万无一失的，但你仍然必须修改描述系统的 ME行。还必须为向其发送新闻的每个站点增加一行。如果你是叶子站点，只需要一行，将所有本地生成的文章发送到你的发送站点。假设你的发送站点是 moria，那么你的sys文件就应该像这样：

```
ME:all/all::
moria/moria.orcnet.org:all/all,!local:f:
```

■ organization —— 公司或组织名。比如，Virtual Brewery公司。在你的家用电脑上，输入“private site”（私人站点）或其他自己喜欢的名字。如果没有自定义这个文件，你的站点就叫作配置不当。

■ mailname —— 站点的邮件名。比如vbrew.com。

■ whoami —— 供收发新闻之用的站点名。通常采用 UUCP站点名，比如vbrew。

■ explist —— 编辑这个文件时，应该让它能反映某些特殊新闻组的期满时限。磁盘空间此时显得非常重要。为了创建一个新闻组的初始结构，须从你的新闻发送站点获得 active和newsgroups文件，并把它们安装在 /usr/lib/news内，确保它们归news拥有，并且其许可模式是644。从active文件内删除所有的 to.*新闻组，增加 to.mysite和to.feedsite和垃圾组以及控制。to.*新闻组常用于交换 ihave/sendme消息，但你应该创建它们，不管你是否会用到ihave/sendme。随后，在active文件的第二和第三个字段内，替换文章编号，这是通过下面的命令来完成的：

```
# cp active active.old
# sed 's/ [0-9]* [0-9]* / 0000000000 00001 /' active.old active
# rm active.old
```

第二个命令是一个sed(1)调用，也是我本人比较喜欢的一个命令。这次调用将分别用零字串和000001字串来替换两个数位字串。

最后，创建新闻假脱机目录和子目录，这些目录供进入和外出的新闻使用：

```
# cd /var/spool
# mkdir news news/in.coming news/out.going
# chown -R news.news news
# chmod -R 755 news
```

如果你使用的是C-News的最新版本，还必须在新闻假脱机目录内创建 out.master目录。

如果你使用的新闻阅读机源于一个不同于自己正在运行的 C-News程序，你可能会发现新闻假脱机是在/usr/spool/news上，而不是在/var/spool/news内。如果你的新闻阅读机找不到任何文章，就应该在/usr/spool/news和/var/spool/news之间，创建一个象征性的目录。

现在，准备接收新闻。注意，除了前面列出的目录外，你不必创建任何目录，因为 C-News每次从一个还没有假脱机目录的新闻组收到文章时，都会创建它。

对交叉投递文章的所有新闻组来说，尤其如此。稍隔一会儿，你就会发现自己的新闻假脱机目录中充满了从未订阅的新闻组目录，比如 alt.lang.teco。怎样防止这类情况的发生呢？答案是：从 active删除所有不想要的新闻组，或定期运行一个外壳脚本，该脚本将删除/var/spool/news下面的所有空目录（当然， out.going和in.coming除外）。

C-News需要用户向其发送错误消息和状态报告。默认情况下，是 usenet。如果你采用的是默认设置，就必须为用户设置一个别名，该用户把自己的全部邮件都转发到一个或多个负责人处（第13和14章详细讨论了 smail和sendmail的具体做法）。另外，还要通过把环境变量NEWSMASTER设置为相应的主管名，改写这一行为。在新闻的 crontab文件内，以及每次手工调用一个管理工具时，都必须如此。只有这样，安装别名才可能比较简单。

在探测/etc/passwd时，一定要保证每个用户在密码文件的 pw_gecos字段（即第四个字段）内有其真名。对Usenet netiquette来说，发送端的真名出现在文章的 From:内。当然在使用邮件时，总希望如此。

16.3 sys文件

sys文件位于/usr/lib/news内，用于控制你收到的结构目录，并将其转发到其他站点。虽然addfeed和delfeed之类的维护工具可以使用，但我觉得最好手工对该文件进行维护。

sys文件内包含一些条目（针对你把新闻转发到的那些站点）和针对你将收到的新闻组的说明。典型条目如下所示：

```
site[/exclusions]:grouplist[/dist list][:flags[:cmds]]
```

利用反斜杠 \ 的话，条目可能会持续到新行。“ # ”号表示它是批注。下面是sys条目的定义：

■ site —— 该条目适用的站点名。通常选用站点的 UUCP名。sys文件内，也必须有用于你自己站点的条目，不然，你自己是不能接收任何新闻组文章的。

特殊站点名ME代表你的站点。ME条目定义了你希望本地保存的所有新闻组。与 ME行不匹配的文章将被置入垃圾新闻组。

由于C-News会将站点和Path:header字段内的站点名进行核对，所以，必须保证两者真正一致。有的站点采用的是这个字段内的完整形式的域名，有的则采用 news.site.domain之类的

别名。为避免文章被返回这些站点，必须把这些站点添加到例外列表（ exclusion list）中，中间用逗号隔开。

以moria站点为例，它的site字段中将包含moria/moria.orcnet.org这个名字。

■ grouplist —— 这是一个用于特定站点的新闻组和结构订阅列表，各组或分层结构间用逗号隔开。怎样指定分层结构呢？答案是：给出该分层结构的前缀（比如 comp.os，表示名字中有这个前缀的所有新闻组），再加上关键字all，后者仅供选择（比如comp.os.all）。

在分层结构或新闻组前加上感叹号后，这些结构或组便被排除在转发对象之外。如果对新闻组和列表进行核对，将采用最接近的匹配。比如，如果 grouplist中包含

```
!comp.comp.os.linux.comp.folklore.computers
```

则comp分层结构中，除了 comp.folklore.computers和comp.os.linux下面的所有新闻组外，再已没有新闻组被投递到那个站点。

如果站点请求转发你收到的所有新闻，将 all作为grouplist输入即可。

■ distlist —— 向grouplist偏移一个斜杠，其中包含即将转发的程序列表。再次提醒大家注意，可在特定的程序前加一个感叹号，将其排除在外。要转发所有的程序，用 all来表示。省略distlist意味着采用all列表。

例如，你可以用一个程序列表 all,!local来防止只限于本地使用的新闻被投递到远程站点。

至少有两个程序是常用的，它们是 world和local。如果用户什么也没有指定，就会采用前者，它是默认设置。其他还有些程序应用于特定的区、州、国家等。最后，还有两个只适用于C-News的程序，那就是sendme和ihave，它们用于sendme/ihave协议。

这些程序的用法一直是业界争论的主题。比如，有的新闻阅读机只用顶级的分层结构，创建一个假的程序，例如向 comp.os.linux投递新闻的comp。应用于区的程序同样不可信，因为在通过因特网投递时，新闻可能根本不经过你所处的那个区。但是应用于某个公司的程序号却是相当有意义的，比如，它可防止公司的机密文件通过公司网络外泄。但要达到这样的目的，可采用更好的方式，那就是创建一个单独的新闻组或分层结构。

■ flags —— 描述发送站点的特定参数。该条目可以为空，也可以合用下面的标记：

- F —— 启用批处理法。
- f —— 完全等同于F标记，但允许C-News更精确地计算外出批处理文件的大小。
- I —— 令C-News建立一个适合于 ihave/sendme使用的文章列表。对 sys和batchparms文件所做的其他修改都需要启用 ihave/sendme。

■ n —— 为活动的NNTP传输客户机（比如 uutpxmit，参见第19章）创建批处理（ batch）文件。这个批处理文件中包含文章名及其消息 ID。

■ I —— 令C-News建立一个适合于 ihave/sendme使用的文章列表。对 sys和batchparms文件所做的其他修改都需要启用 ihave/sendme。

■ n —— 为活动的NNTP传输客户机（比如 uutpxmit，参见第19章）创建批处理（ batch）文件。这个批处理文件中包含文章名及其消息 ID。

■ L —— 要求C-News只传输投递你站点上的文章。该标记后面如果跟一个十进制数 n，则令C-News只传输 n次网关跳（从你的站点开始计数）范围内投递的文章。 C-News根据Path:field来判断跳次数。

■ u —— 要求C-News只对非中继新闻组的文章进行批处理。

■ m —— 要求C-News只对主持式新闻组的文章进行批处理。最多可用下列标记之一： F、f、I和n。

■ cmds —— 该字段中包含准备针对每篇文章执行的命令，启用批处理时除外。文章将作为标准输入发送给命令。它只适用于小型的文章发送；不然的话，发送和接收端系统的载入量都会大得惊人。

默认命令是

```
uux - -r -z system!rnews
```

将文章作为标准输入，把它发送到远程系统，便调用了 rnews。对该字段内给出的命令来说，它们所用的默认搜索路径是 /bin:/usr/bin:usr/lib/news/bin/batch。后一个目录中包含许多外壳脚本，这些脚本名均以via开头；具体情况参见本章稍后的说明。

如果利用F、f、I或n四个标记之一，启用了成批处理法， C-News就希望能够在该字段内找到一个文件名，而不是通过命令去找。如果文本名没有以斜杠 / 开头，就会被假定与/var/spool/news/out.going有关。如果该字段为空，它就会采用默认的 system/togo。

在设置C-News时，很可能你必须编写自己的sys文件。为了帮助大家进行编写，下面给出了一个示例文件，该文件是用于vbrew.com的，你可以从中复制自己需要的内容。

```
# We take whatever they give us.
ME:all/all::

# We send everything we receive to moria, except for local and
# brewery-related articles. We use batching.

moria/moria.orcnet.org:all,!to,to.moria/all,!local,!brewery:f:

# We mail comp.risks to jack@ponderosa.uucp
ponderosa:comp.risks/all::rmail jack@ponderosa.uucp

# swim gets a minor feed
swim/swim.twobirds.com:comp.os.linux,rec.humor.oracle/all,!local:f:

# Log mail map articles for later processing
usenet-maps:comp.mail.maps/all:F:/var/spool/uumaps/work/batch
```

16.4 active文件

active文件位于/usr/lib/news内，它列出了你的站点上已知的所有新闻组和目前在线的文章。一般很少碰这个文件，但为了完整起见，仍然为大家讲讲它。它其中的条目采用这样的形式：

```
newsgroup high low perm
```

newsgroup当然是指新闻组名。 low和high字段指目前能用的文章之最低和最高编号。如果目前无文章可用，低就等于高 +1。

这至少是low字段的本义。但是，出于提高效率的原因， C-News没有更新这个字段。如果没有依赖该字段的新闻阅读机，不更新这个字段根本就没什么大的损失。例如， trn会查看这个字段，进而判断是否可以将某些文章从自己的线程数据库内清除。为了更新 low字段，你

必须定期运行updatemin命令（早期的 C-News版本中，则是upact）。

perm是一个参数，详细说明了被授权访问该新闻组的用户。它可采用下面的值：

■ y —— 允许用户向该新闻组投递文章

■ n —— 不允许用户向该新闻组投递文章。但是该新闻组仍然是可以读取的。

■ x —— 该新闻组已经被本地取消。通常发生在新闻管理员（或其上级）对投递到特定新闻组的文章采取攻击性行为时。

对该新闻组收到的文章来说，虽然它们被转发到了请求获得它们的站点，但不能实现本地保存。

■ m —— 表示主持式新闻组。用户试图向该新闻组投递文章时，一个聪明的新闻读取器将向她发出信号，并将文章发送给主持人。主持人的地址是从 /usr/lib/news内的（moderator）主持人文件内选出来的。

■ = real-group —— 这标明newsgroup作为另一组，即 real-group的本地别名。所有发送到newsgroup的文章将重定向到它。

C-News中，一般都不必直接访问这个 active文件。利用addgroup或delgroup，可在本地增添或删除新闻组。为整个 Usenet增添或删除新闻组，通常分别发送一条 newsgroup和rmgroup控制消息即可。绝不可以自行发送此类的消息！关于创建一个新闻组所需的步骤，可参考news.announce.newusers内定期发布的文章。

与active密切相关的一个文件是active.times。只要创建了一个新闻组，C-News就会将相关消息记入这个文件，该文件中有创建的新闻组名、创建日期，是由新闻组控制消息创建的，还是本地创建的，以及创建者是谁。对通知用户新近建立了哪些新闻组的新闻阅读机来说，这是非常方便的。该文件还可以供 NNTP的NEWSGROUP命令使用。

16.5 新闻组文章的批处理

新闻大批处理采用的特殊格式和 Bnew/s、C-News及INN一样。每篇文章都以这一行开头：

```
#! rnews count
```

count指的是文章内的字节数。采用成批处理法压缩之后，文件就被压缩为一个整体，以另外一行开头，表示该消息将用于解压。标准的压缩工具是 compress，用下面这行标记：

```
#! Cunbatch
```

有时，必须通过 mail软件（从所有数据中删除第 8位）发送批处理文件时，压缩过的批处理文件可利用一种名为c7编码的技术得以保护；这些批处理文件统统用 c7unbatch标记出来。

批处理文件被发送给远程站点上的 rnews时，它会检查它们的标记，并利用相应的解压方案对它们进行处理。有的站点还利用了其他的压缩工具，比如 gzip，所以它们采用的是gzipped标记，而不是zunbatch标记。C-News不能识别这些非标准的文章头，所以你必须修改源代码，以支持它们。

C-News中，文章的批处理是由 /usr/lib/news/bin/batch/sendbatches来执行的，它采用了一份取自site/togo文件的文章列表，并把这些文章放入几个新闻批处理文件内。根据通信量的大小，批处理文件应该每小时执行一次，或更为频繁。

批处理操作由/usr/lib/news内的batchparms文件控制。该文件描述了每个站点允许的批处理文件的最大字节数、准备采用的批处理和可选的压缩程序以及用于把它投递到远程站点的

传输程序等。你可以一个站点一个站点地指定批处理参数，对没有明显提及的站点，则采用默认参数集。

为了针对特定的站点执行批处理，可这样调用它：

```
# su news-c"/usr/lib/news/bin/batch/sendbatches site"
```

当不含参数情况下被引发时， sendbatch处理所有批处理队列。 all的意义依batch参数中缺省条目是否出现而定。如果出现，在 /var/spool/news/out.going下的所有目录都被检查，否则，将循环通过所有 batch参数条目。注意当浏览 out.going目录时， sendbatch只取站点名中不含。或@的那些目录。

当安装C-News时，你很可能在发布盘中找到batch参数文件，这个文件包含一些缺省条目，这样你就不会接触batch参数文件。我们现给出它的格式，每行有 6个字段，由空格分隔开：

```
site size max batcher muncher transport
```

■ site —— 条目所应用的站点名。该站点的 togo文件必须驻留在新闻假脱机目录下面的out.going/togo内。/default/站点名代表的是默认条目。

■ size —— 已创建的批处理文件的最大字节数（指压缩之前）。对大于这个数目的文章，C-News将执行一个违例，并把它们放入另一个单独的批处理文件中。

■ max —— 批处理文件的最大数目。这些批处理文件是在准备将批处理文件发送到特定站点之前，创建和安排传输的。这个字段非常有用，特别是远程站点长时间没有运行时，因为它可防止C-News把数额庞大的新闻批处理文件统统塞到你的 UUCP假脱机目录内。

C-News利用/usr/lib/news/bin内的排队脚本，判断排队等候批处理文件的编号。 Vince Skahan编写的newspak中包含了一个兼容BNU UUCP的脚本。如果你打算采用另类的假脱机目录，比如说泰勒式UUCP，就必须自行编写。如果不在乎假脱机文件的多少（因为你的计算机属你个人专用，而且没有编写兆字节的文章），就可以用一个简单的exit 0语句替换这个脚本的内容。

对batcher字段来说，其中包含的命令用于生成一个批处理文件，其依据是 togo文件内的文章列表。对定期发送的文章来说，这个字段通常是 batcher。出于其他的目的，还可提供别的batcher。例如， ihave/sendme协议要求文章列表被转换为 ihave或sendme控制消息，这些消息被投递到新闻组to.site。文章列表的转换是由batchih和batchsm来执行的。

muncher字段指定压缩所用的命令。通常，这个命令是 compcun，它是一个生成压缩批处理文件的脚本（以C-News为例， compcun采用12位选项来压缩文件，这是多数站点都不采用的。如果你可以对其进行复制，比如 compcun16，采用的就是16位压缩选项。但和前面的压缩选项相比，改进并不明显）。另外，也可自行提供一个采用 gzip的muncher，比如说gzipcun（强调：你必须自行编写它）。除此以外，还必须保证远程站点上有解压命令，并且能够识别采用gzip压缩的文件。

如果远程站点上没有解压命令，可能需要指定 nocomp，表示不做任何压缩。

最后一个字段是transport，它描述了准备采用的传输命令。针对不同类型的传输，可采用不同的标准命令，这些命令以 via.开头。sendbatches把命令行上的目标站点名投递给它们。如果batchparms条目不是/default/，它就从site字段中截取站点名，具体做法是剔除该字段内的第一个句点或斜杠及其以后的内容。如果batchparms条目是/default/，就采用out.going中的目录名。

有两个命令利用 uux，在远程系统上执行 rnews。它们是：viauux和viauuxz。后者为旧版

本的uux设置了-z标记，阻止它为已投递的文章返回成功消息。另一个命令， viamail，通过邮件，向远程系统上的用户 rnews发送文章批处理文件。当然，这要求远程系统为其本地新闻系统发送rnews邮件。要想得到一份完整的传输命令列表，可参考 newsbatch(8)手册。

后三个字段的所有命令都必须位于 out.going/site或/usr/lib/news/bin/batch内。这些命令之中，大部分都是脚本，所以你可以根据自己的需求，轻松地定制新工具。它们被当作管道调用。文章列表被当作标准输入，发送给 batcher，后者再生成作为标准输出的 batch文件。然后，batch通过管道输送到muncher等等。

下面给出一个示例文件。

```
# batchparms file for the brewery
# site        | size   |max    |batcher  |muncher    |transport
#-------+-----+---+-----+------+------
/default/       100000  22      batcher   compcun     viauux
swim             10000  10      batcher   nocomp      viauux
```

16.6 对新闻进行过期处理

Bnews中，期满一直由一个名为 expire的程序执行，该程序采用新闻组列表作为参数，还有一个时间说明，表示文章的到期日是多少。为了让不同的分层结构在不同的时间期满，你不得不编写一个脚本，令其针对每个结构，单独调用 expire。C-News提供了一个更为方便的解决之道：在一个名为 explist的文件内，指定特定新闻组及其期满时间间隔。这个名为 doexpire的命令利用cron，通常每天运行一次，并根据 explist文件内的新闻组列表，对所有的新闻组进行处理。

有时，你可能想在特定新闻组已经到期之后，仍然持有这些新闻组内的文章；比方说，打算保存已投递到 comp.sources.unix的程序。这叫作“ 归档 ”。explist允许你将新闻组标记为“ 归档 ”。

explist文件内的条目如下所示：

```
grouplist perm times archive
```

grouplist（新闻组列表）是该条目所应用的一个新闻组列表，各组间用句点隔开。分层结构的指定是这样的：给出新闻组名的前缀，后面可以选择性地加上 all。比如，以应用于 comp.os下面所有新闻组的一个条目为例，在 grouplist这个地方，既可以输入 comp.os，又可以输入comp.os.all。

在判断一个新闻组的新闻到期时，应该按照指定的顺序，将该新闻组名和 explist内的所有条目进行核查。然后，采用第一条与之匹配的条目。例如，要在 4天之后，把大部分 comp内的新闻丢弃，你自己想保留一星期的 comp.os.linux.announce不计入内，你只须选用后者的条目（指定7天的到期时间），然后才是用于comp的条目（指定4天的到期时间）。

perm字段详细说明了该条目是应用于主持式新闻组、非主持式新闻组还是任何类型的新闻组。它可采用的值有：m、u和x，分别代表主持式组、非主持式组和任何类型的新闻组。

第三个字段times，通常只包含一个单独的数字。如果这些文章头的 Expires:字段内，没有为其人为地分配到期日期，这个数指出哪些文章将在多少天后到期。注意，这里的天数是从文章抵达你的站点那一天开始计算，而不是投递之日。

然而，times字段还可以更为复杂。其中可包含的数字多达 3个，中间用破折号“ - ”隔开。

第一个数指出，多少天后，对文章进行过期处理。这个数除了零值以外，较少采用别的值。第二个数是前面提过的默认到期值。第三个数指出，多少天后，无条件地对文章进行过期处理，不管它是否有Expires:字段。如果只指定中间那个数，其他两个数都将采用默认值。这些值是利用特殊的/bounds/条目来指定的，详情随后介绍。

第四个字段archive，指出是否将新闻组归档以及归入哪个目录。如果不打算采用归档，就应该采用破折号“ —— ”。不然，就采用一个完整路径名（指向一个目录）或“ @ ”符号。“ @ ”符号代表默认的归档目录，这个目录是必须为 doexpire指定的，通过在命令行上采用 -a 标记，就可以完成指定了。归档目录应该属 news拥有。例如， doexpire对comp.source.unix的文章进行归档处理时，它会把它保存在归档目录下的 comp/sources/unix目录内，如果没有这个目录的话，它就创建一个。但是，归档目录本身是不能创建的。

作为doexpire命令基础的explist文件，有两个特殊的条目。采用的不是新闻组列表，而是两个关键字：/bounds/和/expired/。/bounds/条目中包含的默认值是前面提过的times字段的三个值。

/expired/字段表示C-News将把文章对应的行保存在历史文件内的时间。这个字段是必要的，因为文章一旦到期， C-News不会从历史文件内删除与之对应的行，仍然将其保存在历史文件内，这样一来，就可能导致文章重复的情况。如果你只有一个发送站点，可把这个时间值设小一些。UUCP网络中，则根据你从这些站点获取文章的时限来定，一般建议设为两周。

下面是一个explist示范文件：

```
# keep history lines for two weeks. Nobody gets more than three mont
/expired/                       x       14      -
/bounds/                        x       0-1-90  -

# groups we want to keep longer than the rest
comp.os.linux.announce          m       10      -
comp.os.linux                   x       5       -
alt.folklore.computers          u       10      -
rec.humor.oracle                m       10      -
soc.feminism                    m       10      -

# Archive *.sources groups
comp.sources,alt.sources        x       5       @

# defaults for tech groups
comp,sci                        x       7       -

# enough for a long weekend
misc,talk                       x       4       -

# throw away junk quickly
junk                            x       1       -

# Archive *.sources groups
comp.sources,alt.sources        x       5       @
# defaults for tech groups
comp,sci
                                x       7       -

# enough for a long weekend
misc,talk
                                x       4       -
```

```
# throw away junk quickly
junk                            x       1       -

# control messages are of scant interest, too
control                         x       1       -

# catch-all entry for the rest of it
all                             x       2       -
```

C-News中，采用过期处理时，可能还存在许多问题。其一，你的新闻阅读机可能会依赖于active文件内的第三个字段，该字段内包含在线文章的最低限量。在对文章进行过期处理时，C-News没有对这个字段进行更新。如果你需要（或希望）让这个字段代表实际情形，就需要在每次运行doexpire之后，运行一个名为updatemiin的程序（在旧版本的C-News中，是由一个名为upact的脚本来完成的）。

其二，C-News在进行过期处理时，不会查看新闻组的目录，只查看历史文件，从而得知文章是否已经到期（自1970年1月以来，文章的期满日期一直保留在历史列中间的字段内，以秒计）。如果你的历史文件莫名其妙地没有同步，新闻组文章就可能一直待在你的磁盘上，因为C-News已经当它们不存在了（为什么会这样，我也不得而知，但对我本人来说，这样的事的确常有发生）。补救办法是利用/usr/lib/news/bin/maint内的addmissing脚本，这个脚本将把丢失的文章添加到历史文件或mkhistory内，从头重新建立整个历史文件。别忘了在调用它之前，先成为news用户，不然你就以C-News不能阅读历史文件而告终。

16.7 其他文件

可控制C-News行为的文件有许多，但都不能从根本上控制它。所有文件都驻留在/usr/lib/news内。下面简要谈谈它们。

■ newsgroups（新闻组）—— 它是active文件的附加文件，其中包含新闻组名列表和一行主题说明。C-News收到新闻检查消息时，便自动更新这个文件（参见第18章）

■ localgroups（本地新闻组）—— 如果你有许多本地新闻组，不希望每次收到新闻检查时C-News的抱怨，就可把这些新闻组的名字及其说明放入这个文件内，它们就会在新闻组文件内出现。

■ Mailpaths —— 该文件内包含各主持式新闻组主持人的地址。每一行内包含新闻组名和主持人的邮件地址（偏移一个制表符）。

有两个特殊条目是默认提供的。它们是backbone（骨干网）和Internet（因特网）。两者均以bang路径表达式提供了到最近的骨干站点和可识别RFC 822式地址（user@host）的站点的路径。默认的条目如下所示：

```
internet      backbone
```

如果你已安装了smail和sendmail，没必要更改Internet条目，因为它们能够识别RFC 822式的地址。

只要有用户向其主持人没有显示列出的主持式新闻组投递文章，就可采用backbone条目。如果该新闻组名是alt.sewer，而backbone条目中包含path!%，C-News就会将文章邮寄到path!alt-sewer，希望骨干主机能够对该文章进行转发。采用什么路径，可向你的新闻组发送站点询问。最后才考虑uunet.uu.net!%s这条路径。

■ distributions —— 这个文件不是一个真正的 C-News文件，但它可供有些新闻阅读机和nntpd使用。它包含了你的站点能够识别的程序列表及其结果（或目的）说明。例如， Virtual Brewery就有下面这些文件：

- world
- everywhere in the world
- local
- Only local to this site
- nl
- Netherlands only
- mugnet
- MUGNET only
- fr
- France only
- de
- Germany only
- brewery
- Virtual Brewery only

■ log —— 该文件中包含对所有C-News活动的记录。通过运行newsdaily，便可定期获得这个文件；旧的日志文件副本保存在log.o、log.oo等文件内。

■ errlog —— 这是一个记录所有错误消息的文件，这些错误是 C-News引起的。该文件内没有包含投错新闻组之类的垃圾文章。如果该文件是非空的，就会定期被 newsdaily自动邮寄给newsmaster（新闻主管，是usenet的默认设置）。errlog的清除也由newsdaily负责。旧的错误消息日志文件副本被保存在errlog.o、errlog.oo等文件内。

■ batchlog —— 记录sendmatches的所有运行情况。这个文件较少使用。它也是由 newsdaily来负责的。

■ watchtime —— 每次运行newswatch时创建的空文件。

16.8 控制消息

Usenet新闻协议可识别另类的特殊文章，这些文章引起新闻系统的特定反应或动作。它们叫作控制消息。根据文章头的 Control:字段（其中包含即将执行的控制操作），就能够识别出它们。控制消息有许多类，统统由外壳脚本来操作，这些脚本位于 /usr/lib/news/ctl下。

许多控制消息在C-News处理文章时，没有通知新闻主管，自动执行自己的动作。默认情况下，只有checkgroups（新闻组检查）消息才被交给新闻主管（ RFC-1036上，有一个有趣的原型：实施者和管理员既可允许控制消息得以自动实现，也可安排它们等候一年一次的处理），但你可以通过编辑脚本的方式，修改这个默认设置。

16.8.1 cancel消息

最广为人知的控制消息就是 cancel（取消），用户可用这类消息来取消以前发送的文章。如果文章还在的话，这类消息便有效地将它从假脱机目录中删除。对被该消息影响的新闻组

内的文章来说，只要站点收到过它们， cancel消息就会被转发到这些站点，不管它们是否见过这些文章。这是因为考虑到这种可能：原始文章已经被作废消息延迟。有的新闻系统允许用户取消其他人的消息；不过，这当然是禁忌之事。

16.8.2 newgroup和rmgroup消息

涉及到新闻组创建和删除的消息有两条，它们是 newgroup和rmgroup。对usual结构下的新闻组来说，只有在 Usenet读者举行讨论和选举之后，才能创建。适用于中等结构的创建原则允许一些看似混乱的新闻组。关于这方面的详情，可参考 news.announce.newusers和news.announce.newgroups定期发布的贴子。千万不要自行发送 newgroup和rmgroup消息，除非你能确定你已经得到了许可。

16.8.3 checkgroups消息

checkgroups消息是由新闻管理员发送的，用于令网络内的所有站点将自己的活动文件和Usenet同步。例如，商业性的因特网服务供应商可能会向其客户的站点发送此类的消息，每月一次，针对主要结构的“ 正式 ” checkgroups消息通过comp.announce.newgroups的主持人，被投递到这个站点。但是，该消息是作为一篇普通文章来投递的，而不是控制消息。为了执行checkgroups操作，要把这篇文章保存到一个文件比如 /tmp/check内，然后删除所有内容，只留下控制消息本身，再利用下面的命令，把控制消息投递给 checkgroups脚本；

```
# su news -c "/usr/lib/news/bin/ctl/checkgroups" < /tmp/check
```

这样就将localgroups（本地新闻组）内列出的新闻组添加到你的 newsgroups文件，对其进行了更新。旧的newsgroups文件就会被移到newsgroups.bac中。注意，在本地投递该控制消息是不会有效的，因为inews拒绝接受这样大的文章。

如果C-News发现checkgroups列表和active文件不匹配，它就会出示一个命令列表，这些命令将刷新你的站点并把它投递给新闻管理员。其典型输出如下所示：

```
From news Sun Jan 30 16:18:11 1994
Date: Sun, 30 Jan 94 16:18 MET
From: news (News Subsystem)
To: usenet
Subject: Problems with your active file
The following newsgroups are not valid and should be removed.
alt.ascii-art
bionet.molbio.gene-org
comp.windows.x.intrisics
de.answers

You can do this by executing the commands:
/usr/lib/news/bin/maint/delgroup alt.ascii-art
/usr/lib/news/bin/maint/delgroup bionet.molbio.gene-org
/usr/lib/news/bin/maint/delgroup comp.windows.x.intrisics
/usr/lib/news/bin/maint/delgroup de.answers

The following newsgroups were missing.
comp.binaries.cbm
comp.databases.rdb
comp.os.geos
comp.os.qnx
```

```
comp.unix.user-friendly
misc.legal.moderated
news.newsites
soc.culture.scientists
talk.politics.crypto
talk.politics.tibet
```

从你的新闻系统收到此类控制消息时，不要盲目地相信它。取决于投递 checkgroups消息的人，该消息可能需要少数新闻组或整个结构内的新闻组，所以在删除新闻组的时候，你一定要小心行事。如果发现列出的新闻组是你站点上没有的，就必须利用 addgroup脚本，将它们添加到自己的站点上。把所缺新闻组的列表保存在一个文件内，再将该文件投递给下面的小型脚本：

```
#!/bin/sh
cd /usr/lib/news

while read group; do
    if grep -si "^$group[[:space:]].*moderated" newsgroup; then
        mod=m
    else
        mod=y
    fi
    /usr/lib/news/bin/maint/addgroup $group $mod
done
```

16.8.4 sendsys、version和senduuname

最后是用于查找网络结构的三条消息：sendsys、version和senduuname。它们令C-News分别向发送端返回sys文件、一个软件版本字串和 uuname(1)的输出。C-News的版本消息是相当简明扼要的；它返回一个简单的、没有任何说明的“ C ”。

再次提醒大家注意，你绝对不要执行此类消息，除非你充分肯定它使你（地区性）的网络宕掉。对sendsys消息的响应能够使你的UUCP迅速瘫痪（我还没有在因特网上试过）。

16.9 NFS环境中的C-News

在一个局域网内分发新闻，最简单的方法是把所有新闻保存在一个中央主机内，并通过NFS导出所有的相关目录，以便新闻读者可以直接浏览文章。在 NNTP上采用这种方法的好处是：获取和传输新闻耗费的开支明显降低。另一方面，在一个由不同成分组成的网络内使用NNTP，显然好处多多。这类网络由不同种类的主机组成，或者说其用户在服务器主机上没有对应账号。

在使用NFS时，对于本地主机上投递的文章来说，它们必须被转发到中央主机，因为如果不这样的话，访问管理文件可能会令系统处于不安全的状态，使得文件不连贯。此外，你可能还想通过只读方式导出新闻假脱机，对它进行保护，它也需要被转发到中央主机上。 C-News对此类情况进行的处理是透明的。在你投递一篇文章时，你的新闻阅读机通常会调用inews，把该文章注入新闻系统。这个命令对该文章运行了大量的检查，完成文章头的处理，并在/usr/lib/news内核实文件服务器。如果发现该文件服务器存在，而且其中包含一个不同于本地主机名的主机名，就通过 rsh，在那台服务器主机上调用 inews。由于inews脚本要采用C-News的大量二进制命令和支持文件，所以你必须在本地安装C-News，或通过服务器装入新闻软件。

为了让rsh调用正常运行，每个用户在服务器系统上，都必须有对应的账号，也就是说，他／她能够在不要求密码的情况下登录系统。

另外，要保证服务器内指定的主机名语义上与服务器主机上 hostname(1)命令的输出相匹配，如若不然，C-News在试着投递文章时，就会无休止地循环下去。

16.10 维护工具及任务

尽管C-News非常之复杂，但新闻管理员的生活仍然可能非常之轻松，因为 C-News提供了相当多的维护工具。其中一些通过 cron定期运行，比如newsdaily。使用这些脚本，可大大减轻日常维护和新闻发送工作。

除非特别说明，这些命令通常都位于 /usr/lib/news/bin/maint内。注意，在调用这些命令之前，务必保证自己成为news user。以超级用户的身份运行它们，以便能够对 C-News不能访问的文件进行访问。

■ newsdaily —— 正如其名，这个命令是一天运行一次。它是一个非常重要的脚本，有助于你保证日志文件不至于太大，保持前三次运行记录的副本。另外，它还试着探测异常情况，比如进入和外出目录中的失效批处理文件，试着探测被投递到未知或主持式新闻组的目录等。运行产生的错误消息都将被发送给新闻主管。

■ newswatch —— 这是一个应该定期运行的脚本，用于查找新闻系统中的异常情况，一般一小时运行一次。其目的是探测故障，并把故障报告发送给新闻主管。这些故障将立即对新闻系统的操作性产生影响。它检查的内容包括：没有及时删除的失效锁文件、空的输入批处理文件和磁盘空间是否缺乏。

■ addgroup —— 在你的站点上，本地添加一个新闻组。正确的调用是：

addgroup groupname y|n|m|=realgroup

第二个参数的含义和 active文件内的标记一样，意思是任何人都可向指定组（ y），无人主持式新闻组（n）和主持式新闻组（m），或采用别名的另一个新闻组（realgroup）投递文章。

此外，在新近建立的新闻组内第一篇文章先于 newgroup控制消息（打算建立改新闻组）抵达时，还可打算采用addgroup。

■ delgroup —— 允许你本地删除新闻组。其调用形式如下：

delgroup groupname

仍然必须删除新闻组假脱机目录中保留的文章。另一种办法是把它留给自然事件处理(比如过期处理)。

■ addmissing —— 把遗漏的文章添加到历史文件内。有些文章“ 阴魂不散 ”时，可运行这个脚本。

■ newsboot —— 该脚本应该在系统启动时运行。它把关机时被杀死的新进程遗留下来的锁文件删除，并关闭和执行 NNTP连接遗留下来的所有批处理文件。这里的 NNTP连接已在系统关机时被中断。

■ newsrunning —— 这个脚本驻留在/usr/lib/news/bin/input内，可用于允许对进入的新闻进行批处理。也可通过下面的调用关掉“ 取消批处理 ”：

/usr/lib/news/bin/input/newsrunning off