

B News to C News Transition Guide

Geoff Collyer

Software Tool & Die
Brookline, MA

ABSTRACT

We describe the changes in installation and administration from B News to C News. The intended audience is B News administrators switching to C News.

Introduction

C News was written by two people (Geoff Collyer and Henry Spencer) who were experienced B News administrators (since 1982). C News was intended to be a much faster and more correct B News with some of the things that irritated us in B News changed. C News is thus administratively similar to B News, though not identical. We have no personal experience running B News later than 2.10.1, and thus may have missed subtleties in later releases, notably 2.11.

The single largest visible change from B News is probably that we wanted to allow the possibility of machines of different architectures sharing a single news data base (*/usr/spool/news* plus the data files in */usr/lib/news* in B News terms) via network file systems while keeping separate directories of executables elsewhere. We did this by pushing the executables into a new directory, often called */usr/lib/newsbin*. We also took the opportunity to push most of the binaries into subdirectories to reduce the clutter at the top level, as the old B News */usr/lib/news* had become awfully cluttered.

The next most visible change is probably that we only compile in default values for directory names (and a few other configuration parameters), but they can be overridden at run-time via environment variables (subject to restrictions to prevent spoofing). *notebook/config* (relative to the C News source subtree root) describes the variables. In a nutshell, *NEWSCTL* is the equivalent of */usr/lib/news* when referring to control files, *NEWSBIN* is the equivalent of */usr/lib/news* when referring to executables (and is often */usr/lib/newsbin*), and *NEWSARTS* is the equivalent of */usr/spool/news*. This capability can be used to run B News and C News in parallel for testing, feeding them identical inputs. It can probably be used instead of the **IHCC** ifdefs in B News. It is also used for regression testing, when we run known input data through the software in test directories and verify that we get known results out; **make f** in a given source directory will generally invoke the relevant regression test and **make rs** in *conf* will invoke all of them.

Changes in control file formats

In general, *news(5)* now describes the file formats.

Sys adds a few optional features and withdraws some old and little-used ones. Many obsolete flags draw fatal diagnostics, notably **N**, the old, unbatched *ihave/sendme* flag. The other unsupported flags are **A** (convert outgoing messages to A News format; *relaynews* is not a protocol converter), **H** (append history entry to named file?; sorry), **S** (duplicate the work of the shell to avoid one process; just delete this flag), **M**, and **O** (multicast flags; obsoleted by the batcher, as far as we can tell). New flags are **f** (append spool file name and size in bytes to the named file) and **n** (append spool file name and Message-ID to the named file; equivalent to **FI** in B News). There have been some changes to flag semantics: **I** in B News invoked all the *ihave/sendme* machinery *and* wrote message-ids to the batch file; C News just writes the message-ids. See below for how to set up *ihave/sendme* feeds. In C News, **FI** and **I** are equivalent; use **n** if you want file names and message-ids. All file names written are relative to *NEWSARTS*.

B News allowed comments to be continued; this is at odds with all other UNIX® software, including C News. To comment-out an entry under C News, you need to prepend '#' before *each* line of the entry, not just the first.

There is now an optional *distributions* list after the newgroups pattern, separated by a slash. If the fourth *sys* field is a relative command name, *NEWSCTL/bin* and *NEWSBIN/relay* will be searched before the standard search path. If the fourth field is a relative file name, *NEWSARTS/out.going* will be prepended. The ihave/sendme kludges of B news have been expunged; one must say what one means, using three to five entries, typically. See the section below on ihave/sendme.

Active has four fields (there is vestigial code in some, but not all, programs for two or three fields) and any number of digits in the article-number fields (five is too few for the long term). Two new values are understood for the fourth (flags) field: **x**, meaning "quietly discard articles for this group", and **=real-group** meaning "file articles for this group under *realgroup* instead". **=** is useful for coping with badly-run local newsgroups, often created from mailing lists. You must lock the news system with *locknews* before editing the active file.

History is in an extended B 2.10 news format: the second field consists of two subfields separated by a tilde: time received as an integer (a *time_t* in fact), and the value of the *Expires:* header, "-" if none. More subfields may appear in future. You must lock the news system with *locknews* before editing the history file, and run *mkdbm* afterward to rebuild the index files.

Log and *errlog* are only faintly similar to their B News counterparts. Error processing in general is due for an overhaul, which may further revise their formats. Currently, *errlog* is the unadorned standard error from *relaynews*(8) and any programs it runs. *Log* is a time-stamped log, down to the millisecond (where possible, offer void where prohibited by System V), consisting of one line per article, and always showing the name of the host that handed us this article and the Message-ID of this article. See *notebook/log* for details.

Locks are in the style of B 2.10: *NEWSCTL/LOCK* is the news system lock, but it never times out. One locks the news system by repeatedly attempting to link to the lock name until successful; one releases the lock by removing the lock name. There are other locks, all of which begin with the prefix *NEWSCTL/LOCK*. This scheme works well over network file systems, can be used from shell scripts and interactively, permits one to see trivially what locks are present, and is portable across UNIX variants. See *notebook/newslock* for details.

'Missing' control files

Aliases does not exist, since we believe that header munging is to be avoided if at all possible (we do update **Path:**, regenerate **Xref:**, and delete some large obsolete headers [**Relay-Version:**, **Posting-Version:**, **Date-Received:**, **Received:**, **Posted:**, **Illegal-Object:**]). It is possible to file articles locally under different newsgroups by use of the *active* file = flag.

Notify is replaced by *NEWSMASTER* in *NEWSCTL/bin/config*.

Moderators, *reemail* and *renews* are missing; we don't know what they do (yet).

Seq is gone; we don't need it.

New control files

Batchparms controls outbound batching. See *newsbatch*(8).

Explist controls expiry policies. The command-line options for *expire* are completely different from B News. See *expire*(8). The simplest possible *explist* is "**all x 7 -**" for 7-day expiry of all articles and history entries.

Mailname is the domain name of this machine for mail purposes; *whoami* is the news name of this machine (e.g. for **Path:** headers). If *replyusepath* exists, automated mail replies will use the **Path:** header for return addresses. If *server* exists, *inews* will *rsh* to the hostname found therein to run *relaynews*.

Setnewsids is an optional setuid-root program (in the wrong directory, it appears) to set user and group ids; it should be obsolete now, but instead POSIX and System V are making it necessary all too often.

ihave/sendme feeds

We didn't understand the implementation of ihave/sendme in B News for a long time and nearly didn't implement it, having a very dim view of ihave/sendme to start with, but we wanted to be RFC 1036 compliant. Upon discovering the tricks (including mandatory default values for various *sys* file fields) used internally by B News to implement ihave/sendme, there was immediate agreement that

- (a) we should document how ihave/sendme works, and
- (b) we should implement something more obvious and that would not mandate fixed values for fields that users might legitimately want to change from their default values.

We also see no use for the old unbatched ihave/sendme protocol: it is grossly wasteful of resources at today's traffic volumes and offers no clear advantage to sending lists of Message-IDs in each *ihave* or *sendme* message.

The result is that we have implemented batched ihave/sendme as multiple *sys* file entries, which is clearer but more verbose. One typically needs an entry to specify

- (1) which articles are to be offered to the other system via an *ihave* message,
- (2) that *ihaves* and *sendmes* are to be batched or sent immediately,
- (3) that the response to an *ihave* should be a *sendme* message posted to the appropriate **to.site** group, and
- (4) that the response to a *sendme* message should be batching the requested articles.

Note that any article requested by a *sendme* will be sent; there is no checking that the requesting site is permitted by its appropriate *sys* entry to receive this article. Such checking could be added, at some cost in performance. For now, disable ihave/sendme on systems with 'private' newsgroups (if that isn't an oxymoron).

See *notebook/ihave* for gory details including sample *sys* files.

Differences in behaviour

The *checkgroups* control message is non-destructive; it merely mails its output to **NEWSMASTER** (e.g. **usenet**). The distributed *checkgroups* script has problems, which will be fixed some day. In the mean time, some people have reported that the latest B News *checkgroups* script works fine for them; you could try it.

There is an *Also-Control:* header recognised, of which *Supersedes:* is a special case.

Interactions with NNTP

We had erroneously assumed when writing C News that the NNTP situation was not too disastrous. Since we didn't then run NNTP ourselves, we benignly ignored NNTP. Only once we started running NNTP did we realise what a performance disaster it was. We are still working on improving the situation.

The **CNEWS** *ifdef* of NNTP as distributed by Stan Barber makes the situation tolerable (I wrote the original version of the code inside the *ifdef*) if not wonderful. In particular, it will happily accept duplicate articles, only to have *relaynews* throw them away. We have an experimental *nntpd* based on NNTP 1.5.8 that we are reasonably happy with, though lock contention may still be an issue for a busy site. Alternately, Paul Vixie's *msgidd* or some code from Rich Salz may offer acceptable solutions.

More work is called for. Several people have simultaneously invented the idea of a multi-threaded *nntpd* handling multiple inbound transfer sessions and that may be a worthwhile direction to pursue. (It permits cheap duplicate rejection with little lock contention and few processes.) It has the disadvantage of not exploiting parallelism on multiprocessor machines, which seem to be what the biggest news relay sites are becoming.

dbm & dbz

There are now several redistributable *dbm(3)* clones available. Before they appeared, we fixed and enhanced Jon Zeef's *dbz* library; it is faster and has smaller files than any *dbm* or *dbm* clone, so we recommend using it for news. Whatever you elect to use, be sure to link NNTP and any other software that makes

history file lookups with the same library that C News uses (picking up *libcnews.a* is usually simplest).

Migration to C News

A good first step is to read the documentation in the C News *doc* source directory. You will want to refer to the contents of the *notebook* and *man* directories. If you have trouble formatting any of this documentation, get the *awf* formatter and use it. *Awf* can be had from your nearby *comp.sources.unix* archive site, or by *uucp* or *ftp* from *uunet* (*uunet.uu.net*) as *comp.sources.unix/volume23/awf.Z*. For side-by-side comparisons of B and C News where this document may have missed something, one could compare the corresponding sections for B and C News in *Managing uucp and Usenet*, by Tim O'Reilly and Grace Todino, O'Reilly & Associates, 1989, ISBN 0-937175-48-X, inquiries to 'nuts@ora.com' or 'uunet!ora!nuts', or 800-338-NUTS (in California, 800-533-NUTS), FAX 707-829-0104. See *doc/biblio* for a fuller bibliography.

It is probably safest to create parallel C News trees, at least for the equivalent of */usr/lib/news*, then populate them and rename the B and C trees when you are satisfied that C News is installed and set up correctly. The C News installation procedure (*build*) will not overwrite existing control files in *NEWSCTL*, so copying your old *active*, *sys*, and possibly *history* files to *NEWSCTL* adapting them as necessary (see below), and then installing C News there normally should get you started.

One advantage of setting up a parallel tree for C News is that it is possible, though *build* doesn't know how, to run all incoming news into both news systems for a day or two to convince yourself that everything is working right. If you decide *not* to switch, you can just remove the C News tree. In any case, running both systems simultaneously avoids the desperate panic of having to cut everything over at once and get it right first time to avoid losing news; this just should not be necessary any more. The change is just to ensure that all the commands that used to invoke *rnews* (*inews*, *rnews*, *cunbatch* and probably a few others) now hand the incoming batch to both news systems. Something like this should suffice (and has worked in the past):

```
# parallel rnews
f=/tmp/rn$$
cat >$f
brnews <$f
crnews <$f
rm $f
```

Sys should only need to be scanned for unsupported flags (I think **S** can just be deleted without ill effect), and entries with **I** flags rewritten as multiple *ihave/sendme* entries. *Active* can optionally be edited to refile groups locally, to compensate for lack of *aliases*. *C expire* can read a correct B *history* file, but B News sometimes produced incorrectly-formatted history files. To be safe, and to pick up information that B News didn't store in the history file, run *mkhistory* to regenerate *history* and its index files. *Mkhistory* will take quite a while; it has to open every article in the news system, which can take hours on some machines.

The normal installation procedure is described in *doc/install* and really just involves running *conf/build*, answering its (many) questions, and following its instructions. (We intend that eventually *build* will be run once per site for all time (or until the hardware or OS change drastically) and that thereafter components may be built using *make*.) Do be careful to use the correct userid for each step of installation; doing the whole thing as *root* will result in incorrect file ownerships. Two major by-products of *build* are *libcnews.a* in the root of the source subtree and the directory *include*.

There are some differences in configurable options. All the length and size limits are gone. Essentially all the B News *makefile* options are now queried for by *build*. *DFLTEXP* and *HISTEXP* are now specified in *explist*. Only a trivial newsreader, a *readnews* replacement, is included with C News, so little of the reader configuration is relevant. *NOTIFY* is now *NEWSMASTER* and *build* queries for it. *DFTXMIT* and *UXMIT* are now *CMDPFX* and *CMDSFX* in *relay/sys.c* with no simple override (though an override is in the works); the default command, in case anyone still uses it, is **uux - -r -z system!rnews**. *MANUALLY*, *NONEWGROUPS* and *UUPROG* are replaced by policy in shell scripts; edit *relay/ctl/** to taste. *BATCH* is gone; the unbatcher is built into *relaynews*. *OLD* is gone (*relaynews* is not a protocol converter). *DOXREFS* is gone; **Xref:** is always generated for cross-posted articles. *SENDMAIL* and *MMDF* are gone;

we just use *mail*. DEADTIME is gone; locks do not time out.

This table summarises the disposition of the remaining B News options. File names in the 'build file' column are relative to *NEWSCTL* and refer to files generated by *build*. A 'flag' in the 'note' column indicates that the presence or absence of the file *is* the flag. A 'gone' in the 'note' column indicates that the option has vanished. If the 'build file' column is empty, the option has just been absorbed into *build*.

box, center; c c c l l c .	B option	build file	note	_SPOOLNEWS	rnews.newsrun	flag	INTER-	
NETreplyusepath	flag	MYDOMAIN	mailname	MYORG	organization	UUNAME	whoami	_
CHEAP	gone	NICENESS	gone	FASCIST	gone	ORGDISTRIB	gone	
MULTICAST	gone	_UNAME	GHNAME	BSD4_2	BSD4_1C	LOCKF	HIDDENNET	
SMALL_ADDRESS_SPACE								

If you compile and install the software as two steps, it is simple, though time-consuming, to run regression tests for the major subsystems: **cd** into a source directory (e.g. *expire*, *batch*, *relay*) and type **make r**. If you get no complaints, the software is believed (after testing) to work correctly on your machine.

Sites with a lot of outgoing news feeds (over 60, say), should be aware that batch file writing will probably dominate *relaynews*'s elapsed time. We hope to fix this.

After installation, errors will generally be reported by mail (to *NEWSMASTER* [typically *usenet*]). To watch the progress of incoming news, tail *NEWSCTL/log* and *NEWSCTL/errlog*. *errlog* should be empty on a smoothly-running system.