

NewsWatcher 2.0d17 To Do List

November 1, 1993

John Norstad
Academic Computing and Network Services
Northwestern University
j-norstad@nwu.edu

This document is a list of errors and problems to fix, features to implement, and other ideas for future versions of NewsWatcher. Some of these items are things we definitely plan to do. Some are things about which we haven't yet made up our minds. Some are just vague "wishes".

Improve John Werner's filtering code.

Support styles: underline, bold, italic, etc. Makes filters usable on B&W monitors.

Support "marker characters". For example "highlight all postings by Steve Dorner by marking them with the "!" character". Again, makes filters nicer on B&W monitors.

People should be able to use the color picker to select their own colors and add them to the highlight menu.

Should save filtering info with user group lists, not in separate prefs file.

Global filters and filters attached to arbitrary nodes in the news hierarchy (e.g., filters for comp.sys.mac.*, etc.). To do this, it might make sense to make the full group list window hierarchical with triangle collapse/expand controls. This would give the user a place to click to create/edit filters at various levels in the hierarchy.

A hierarchical full group list is a nice idea anyway.

Study how the various UNIX newsreaders do kill files. We need to provide at least as many features and as much flexibility.

Notes from John Werner:

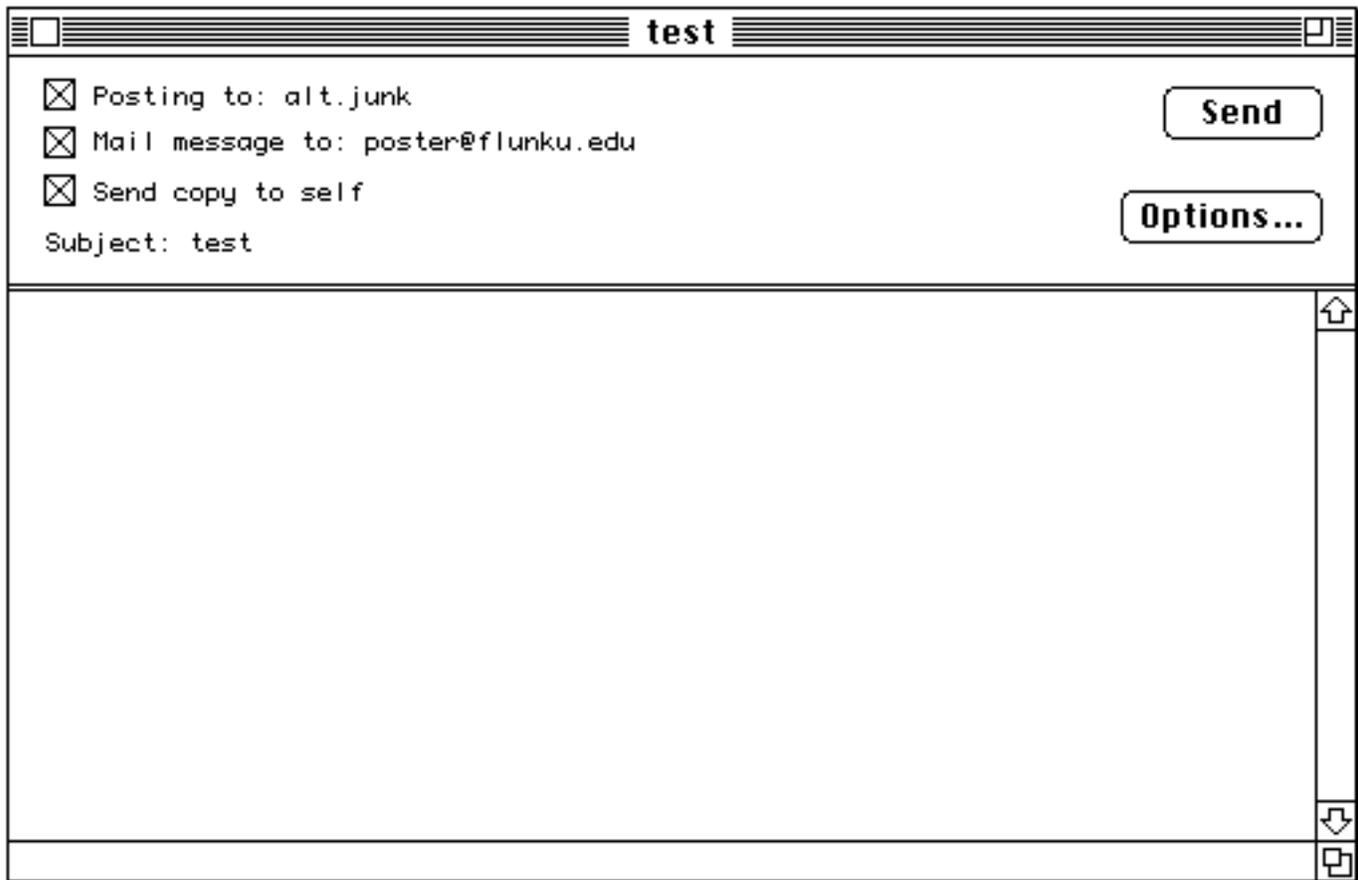
- 1) Right now it filters all of the articles every time you open a subject window. For filters that kill articles this isn't really necessary; it should only look at new articles that it hasn't filtered before. Filters whose action is "Keep" or "Highlight" still have to be done every time, since they affect how the article appears in the subject window.
- 2) If a group has only a few articles but many filters, sending lots of XPAT commands to do the filtering is probably slower than just grabbing the relevant headers and doing it locally. Especially if there's only one header (like From:) involved. The proper cutoff point probably varies according to the number of different headers involved.
- 3) An item from the old To Do list: allow killfile entries that expire after a given amount of time. I actually left a field in the data structure for this, but didn't do anything with it.

Integrate code contributions.

Bob Boonstra. 5/4. Miscellaneous changes.
Michael Reed. 3/11. Authorization.
Gerry Brown. 11/3, 2/24. Miscellaneous stuff.
Scott Lindsey. 12/1. Cancel articles. Save articles from subject window.
Jeroen Scheerder. 8/12. Cancel articles.
John Jenkins. 6/14. Chinese & Japanese.
Chris Fuhrman. 6/14 and 7/29. 8 bit conversions.
Michael Shappe. 9/2. Authorization.
Lasse Hillerøe Petersen <lassehp@imv.aau.dk>. 9/21. Latin1 version.

Completely redesign message windows.

The picture below shows the new window design.



This new "message window" is used for all kinds of messages: Postings, follow-ups, mail messages, and combinations.

Note the terminology I'm using here: An "article" is an existing news article. An article window is always read-only. A "message" may be a new article under

composition, a new mail message under composition, or both. Message windows are editable. A "follow-up" is a reply to an article which is posted as another article. An "email reply" is a reply sent via Internet mail to the poster. "Reply" is a generic term used to refer to either kind of reply.

The panel area at the top of the message window contains the following items:

1. "Posting to" checkbox and newsgroup list. The user can edit the newsgroup list. The checkbox turns posting on and off. Users can also drag and drop newsgroups to this list.
2. "Mail message to" checkbox and email address list. The user can edit the address list. The checkbox turns email on and off.
3. "Send copy to self" checkbox. If checked, a copy of the message is mailed to yourself.
4. Subject. The obvious.
5. Send button. The obvious.
6. Options... button. Brings up a dialog with extra options for less frequently used header lines: "Follow-up to poster via email" checkbox, "Followup-To:" header option, "Distribution:" header option (a typein popup menu - the user can edit the menu), "Reply-To:" header option, "Keywords:" header option. Also a textedit field which the user can use to add any other header lines he wishes.

There are four menu commands in the News menu which involve this new window:

1. "New Message" (CMD-N). Brings up a new message window. If a group list window is on top, the currently selected group name or names are used to initialize the "posting to" newsgroup list. If a subject list or article window is on top, the group to which the subject list or article belongs is used to initialize the "posting to" newsgroup list. Otherwise, the "posting to" newsgroup list is empty. In any case, the "posting to" checkbox is on, the "mail message to" checkbox is off, and the "subject" is empty.
2. "Reply" (CMD-R). Replies to the current (top) article. This command is used for follow-ups, email replies, or both kinds of replies. The newsgroup list and email address list are initialized according to the usual RFC rules, using the "Newsgroups", "Followup-To", "From", and "Reply-To" header lines of the article. The "subject" is set to "Re: original subject" as usual. The full body of the original article is quoted with ">" markers as usual.
3. "Forward" (CMD-K). Forwards the current (top) article. The "posting to" checkbox is off. The "mail message to" checkbox is on, with an empty address list. The subject is copied from the current article. The header and body of the original article are copied to the new message body, without quoting.
4. "Send Message" (CMD-E). Sends the message, same as clicking on the "Send" button.

There are several new preferences which involve this new window. These preferences are all in a new "Message Options" topic in the preferences dialog:

1. Reply via posting checkbox. On by default. If on, the "Reply" command opens reply windows with the "Posting to" checkbox turned on.
2. Reply via email checkbox. Off by default. If on, the "Reply" command opens reply windows with the "Mail message to" checkbox turned on.
3. Send copies to self checkbox. Off by default. If on, all message windows open with the "Send copy to self" checkbox turned on.
4. Extra header lines text field. Specifies any additional header lines which should be included in all postings. The default is no extra header lines.

Examples (assuming default prefs):

1. Post a new article. Select group in any group list window. CMD-N. Type subject, Tab, and body. CMD-E.
2. Post a follow-up. CMD-R. Edit body. CMD-E.
3. Post a follow-up with email copy to author. CMD-R. Turn on "Mail message to" checkbox. Edit body. CMD-E.
4. Reply to author via email. CMD-R. Turn off "Posting to" checkbox. Turn on "Mail message to" checkbox. Edit body. CMD-E.
5. Forward an article. CMD-K. Type email address. If you wish, add comments to the body. CMD-E.
6. Mail copy of article to self. CMD-K. Turn off "Mail message to" checkbox. Turn on "Send copy to self" checkbox. If you wish, add comments to the body. CMD-E.
7. Send "quick and dirty" mail message to someone. CMD-N. Turn off "Posting to" checkbox. Turn on "Mail message to" checkbox. Type email address, Tab, subject, Tab, body. CMD-E.
8. (This example answers a complaint from Steve Dorner). Suppose an article contains a "Followup-To: poster" header line. Even so, you want to override this and follow-up via a news posting rather than email. CMD-R. Turn on "Posting to" checkbox. Turn off "Mail message to" checkbox (if desired). Edit body. CMD-E.

The current dialogs which are presented whenever a user posts or replies would be completely eliminated.

I would no longer permit users to edit the headers for outgoing messages by hand. The "Show/Hide Header" command would no longer work for outgoing message windows.

There are lots of obvious possibilities for Option and Shift key modifiers with the new commands. I want to get the basic design right first before making any decisions about this.

Thanks to Steve Dorner for reviewing this.

Option-Reply: As in Eudora, only quote selected text.

Fully conform to the RFCs.

Check RFC 1036: From header line full names may not contain parentheses. If this is true, should check full name and complain if it contains any parentheses.

Implement full checking of all header info and don't permit the user to send messages with illegal header lines, or with missing required header lines.

RFC 822 header parsing, especially header line unfolding. Anybody who has actually read RFC 822 knows that full correct parsing is not trivial. But we really have to follow the standard, no matter how absurd it is.

Improve datetime.c "Date" header line parsing to fully implement all formats in RFC 822, including all the timezone formats.

Check for posting not allowed and handle gracefully. See RFC 977. On initial connection, server sends response code 200 if posting is allowed, 201 if not. Should remember this and permanently disable the "Posting To:" checkbox in all message windows for the duration of the session. Similarly, if the LIST command gives an "n" status for a group (local postings disallowed), notify the user if he tries to post (??? - check into this.)

No space in "Mail From:" and "Rcpt to:" commands to SMTP server? Someone complained about this. Check the RFC.

In personal info dialog, check email field for validity. The message id is generated using part of this field, and it must be valid.

When following up, filter out newsgroups which no longer exist. When sending any posting, make certain "To" and "Followup To" header lines contain only valid lists of extant newsgroups.

Implement authorization using the "AUTHINFO" command.

Support both the ref. imp. and inn, which have different requirements.

With the ref. imp., this feature can be used to require authorization for general reading and/or posting.

With inn, this feature can also be used to implement private "conference" newsgroups. In inn, the server administrator can restrict access to individual selected newsgroups to lists of authorized users.

With inn, the full group list as fetched by the "LIST" command is filtered to contain only the groups to which the user is authorized access. Thus, when the user changes his or her authorization information (username or password), NW must immediately send the server an AUTHINFO command with the new authorization information, and then use a "LIST" command to get a new full group list.

This feature requires more thought. Using inn to support private conference newsgroups on a large scale will also require some major enhancements to the inn server code to make it really useful.

Character sets.

Support non-English character sets. ISO stuff. Need to get a clue. Ask Dorner?

(Simon Tardell) "Filtering national character sets. This is an important one for non-anglosaxic users. In national 7-bit character standards some "less useful" characters in the US ASCII set have been replaced by "extra" characters of the national alphabet. Eg: to me } is minuscule a-ring. The problem is that in the Mac 8-bit ext. ASCII these characters have codes in the high 128 and (if no conversion is done) comes out as meta-ctrl-z and the like to other people (who don't use Mac, that is). The reverse is also a problem (a-ring coming out as a right brace to me). This should be easy to toggle (so you can read computer source code)."

Also support the Japanese/Chinese newsgroups.

Might find the following ISO 8859-1 (Latin-1) font useful: /info-mac/font/latin-1.hqx. Also write to Peter Lewis - he knows all about ISO 8859-1.

Russian? Just keeping the 8th bit would work if the right font was used. See Kjetil Ra Hauge's note of 7/22.

Some good ideas from Tuomas Eerola:

I haven't done much Mac programming but I anyways managed to put an taBL resource there (1.3d8), which seems to be quite good idea to character translation, it can be changed quite easily with just ResEdit. (That was from Eudora as well)

In GetArticle messages were copied one character at time and I did put it there and another was at SendMsg for sending out.

```
taBL=GetResource('taBL',1001)
(*txtHndl)[totalLength++] = *((*taBL)+(*current))
```

There should of course be some kind of error checking..

What I think would be best system (at least here) would be a general ISO-8859-1 translation for all messages and then there could be named resources (GetNamedResource) if some groups need some other translations or a table with no translation at all.

If there is resource called sfnet then all sfnet.something groups would use this translation. So when user opens a group then NW should check which translation to use (This would not slow much, because there is waiting for those subject anyways) use that translation with subjects and give handle of that translation table to GetArticle().

ISO-8859-1 translations also need that 8-bit transfer.

Read RFC 1342.

(Jean-Pierre Kuypers) Some software use taBL resources to have the translation tables. For the ISO 8859-1 <-> Macintosh translations, Andre Pirard (Pirard@vm1.ulg.ac.be) was proposing good tables. These tables are used now by Eudora, FTPd, Fetch, TurboGopher, NCSA/UCL Telnet 2.5 (and soon NCSA Telnet 2.6). MacMosaic will probably use it too. You can find a text about this subject on ftp.ulg.ac.be as /pub/docs/iso8859/iso8859.networking.

Other new commands.

Implement the "Find" and "Find Again" commands. Should work in list windows as well as in text windows. For segmented article windows, should work across segment boundaries. Include checkboxes for "ignore case" and "whole word". For list windows, a "find all at once" checkbox will also be useful. If an author matches the search, and the line is in a collapsed thread, expand the thread. If a subject matches, however, don't expand the thread - just select the first collapsed line. When searching a group list with the "find all at once" option, it would be nice to provide an option to open a new user group list window containing just the matching groups. When searching subject lists with the "find all at once" option, it would similarly be nice to provide an option to open a new subject list window containing just the matching articles. This should satisfy the people who miss the old "Reduction Filter" command. Rethink the existing Search command and see if we can come up with a combined approach which does everything cleanly.

Implement the "Paste as Quotation" command. When copying text from an article window to the clipboard, we should also copy the message id, author's name, and author's email address to the clipboard. The "Paste as Quotation" command would then paste this info into a posting or mail message window in the usual format, in addition to pasting the actual quoted text. This would make it easy to compose follow-ups which quote several different authors and articles. The message id of such a pasted quotation must also be added to the "References" header line.

Implement "Wrap Selected Text" and "Unwrap Selected Text" commands in the Edit menu.

Rethink and redo article and message saving.

Users shouldn't have to open articles just to save or print them. Permit saving and printing from the subject list. What to do if more than one article is selected in the subject list? Save all of them to one file as in Eudora? Marshall Clow would like the option of saving all to one file or saving to separate files in this case. If we do this, we should also permit reordering of items in subject list windows so that users could save multi-part postings in the proper order. Conrad Nobili also mentioned this problem. Matt Mora says: "Oooo I like this. When the user selects a range of messages and selects Save, ASK the user what she wants to do. I really like the reordering idea, but that sounds like a lot of work." Also should be able to save/print an entire thread easily from the subject list, again either as separate files or as one combined file.

(Ian Harries) How about an "append" option in addition to "save" in the File menu. For keeping those interesting threads in a single file, of course. Unless I am missing a simple way of doing this already ...

Add commands to easily download/upload multi-part uuencoded and BinHex'd files.

Probably the nicest way to do the downloading would be for NW to detect when an article being saved contains such a file (or the first part of such a file). If it does, present a dialog giving the user appropriate options. Do the best job possible finding all the parts. Look for the usual nn/mm in subject lines.

For uploading, we need something similar to Eudora's "Attach Document" command.

Perhaps saved article and message files should be NewsWatcher document files (type TEXT, creator NNTP). They could be reopened in NewsWatcher.

Perhaps something like Eudora's mailboxes. E.g., a collection of articles saved in a single file with a separate .toc table of contents file. When opened, presents a subject window listing the contents of the file. Add a Transfer menu as in Eudora?

Perhaps we should bring back article numbers as the first part of default saved article file names.

Need to completely rethink window titles for articles and names for saved files.

There are lots of possibilities here, and lots of design work to do.

Undo

The "Next Group" command is too powerful. People do it by mistake all the time. Make the "Undo" command work to undo at least this operation.

Should also be able to undo text editing, as in Eudora.

It would be nice to be able to undo/redo lots of operations, but it's hard work.

Human Interface Guidelines

On article open, present status dialog after 4 secs.

Menu items: check capitalization.

p67. Check all ellipses in menus.

p103. Check save changes alerts. Message should include app name and doc name.

p112. Undo Next Group / Redo Next Group

pp122,217. Size menu in prefs. Support all sizes.

p141. New group windows: untitled, untitled 2, etc.

p148. Default window placement rules.

Reference Threading

Implement reference threading in place of or as an alternative to subject threading. Perhaps a combination might work.

Reference threading presents problems. If we use the XHDR command to fetch the references header lines when opening subject windows, we add 50% to the time required to open the window. XOVER speeds this up, but many servers don't implement it.

Displaying reference trees and allowing users to manipulate them as in trn is another possibility.

Miscellaneous new features.

(Steve Dorner). Every once in a while I wish I could close a group window without having NewsWatcher close the articles I have open from it.

(Rob Lentz). Select group in user group list, use delete key to unsubscribe.

Some way to easily retrieve an entire thread, restoring read articles in the thread.

Perhaps option-click on user group to open all articles, not just unread ones? Would perhaps make it a tad easier to go back to already read articles.

A suggestion from JQ Johnson:

I am finding that the "Next xxx" commands are too aggressive for my taste in marking unread articles as read. In rn I frequently use the "M" (mark) command while browsing through a large news group to mark articles that I don't have time to think about right now, then catch up on all of the rest of the articles in the group. The "M" command in rn, unlike the "m" command, implies that my marked articles stay unread but that the rest of the articles in the group are marked as read. Similarly in xrn, where marked articles behave as if marked with "M". My recommendation is that there be an option to determine whether the "Mark Unread" command is to behave like an rn "m" or "M" with respect to the "Next xxx" commands.

Support XGTITLE?

Some kind of "prev article" & "next article" navigation facility which ignores read/unread status. Use Command-up and down arrow as in Eudora?

(Quinn) Select multiple articles in subject window, Reply. Should build correct multiple references and quoted bodies. Email address should be unions of all From/Reply-To lines with duplicates removed. But what's the subject? - the first will do.

(P. Lewis) As in Eudora, any command which can be used with an open article window should also be usable with one or more selected subjects in a subject window. Same goes for commands used with an open subject window

(Dorner) Keypad 5 = page down if not at end, else next article.

Printing: Page headers and footers as in Eudora. Present status window while printing and let user cancel. Implement the new "Print Selection" command. Handle and report errors better: e.g., during initialization if no printer has ever been selected (virgin system: get unexpected error -43). Bigger left margin? In Print Monitor, name of document should be title of window being printed, minus any leading √. Printing doesn't work at all on my ImageWriter - fix it.

Option-click on check mark area: toggle read/unread?

Option to include/exclude full headers when saving/printing articles and messages?

Don't display redundant subjects in expanded threads?

(Marshall Clow). Indent expanded threads?

Gray separator lines in subject window (ala Eudora)? Group windows too?

Rethink default window placement and window position saving/restoring. At the very least:

1. Save and restore user group list window position.
2. Permit the user to have a preferred remembered location for child subject and grandchild article windows, at least for the first ones opened. NW should save this info on the parent user group list file and restore it when the user group list file is opened. This will solve Dorner's complaint about article windows obscuring their parent subject list windows.

During initialization, make sure full group list window pos is still on screen. Same for status window. There is code to do this now, but it is not 100% correct.

Tab key as in Eudora.

Documentation.

A complete document directed at beginners, similar to the NU Eudora doc.

System 7 balloon help. Also add balloon help for icons. See IM VI p. 9-34.

"NewsWatcher Shortcuts" command in Help menu. List keypad shortcuts, etc.

There's not much point in starting work on any of these things until the features have stabilized.

Mostly internal stuff and miscellaneous problems which need to be fixed.

Fine tune menu command enable/disable. E.g., enable "Cycle Windows" iff num windows > 1, enable Cut & Copy only if something selected, enable Paste only if clipboard not empty, etc.

When Status window is up, disable entire menu bar except for the Quit command, the Show/Hide Balloons command, and the application menu.

Add STR -16396 resource containing "NewsWatcher" to all docs (user group lists and prefs files), so Finder can display app name when app not found. See IM VI p. 9-21.

Check for required Apple event parameters. Also deal with interact with user option. See IM VI.

AETE resource.

Panel area in full group list and new groups list windows showing number of groups.

Open all references command should be active when a message window is active.

Permit ROT-13 in mail messages (should fall out of new message window design).

(Peter Lewis). When opening multiple articles, open in regular order, but use behind parameter to stack in desired order. User can be reading first one while the others are being opened. Nice especially on slow links.

Printing/saving rot-13 text doesn't work. User user's rot-13 command to unscramble text, but when articles is printed or saved, text is still scrambled.

Fix keypad keys on non-English keypads. Don't check ASCII codes?

Complete the campaign to carefully check and revise or rewrite all the code, with special attention to error detection, reporting, and recovery. I'm about half done with this so far. There's still lots of this work to be done.

Implement memory error handling. Keep reserved block for exclusive use by the toolbox and OS. Preflight large toolbox and OS memory needs. Make the program as robust as possible under low memory conditions. See new IM "Memory" volume.

Give English error message for common errors like "disk full" instead of "unexpected errors" with error numbers.

Check for exceeding the 32K limit while editing (e.g., on key events and paste operations).

Use Steve Falkenburg's XferIt trick: Patch ExitToShell to close any open TCP connections. If program crashes or user exits to shell, user doesn't have to restart the Mac. For now I'm using Steve's ZapTCP extension, which does this system-wide. But users probably won't have ZapTCP, so we should do this for them.

Make sure creator and file types are registered with DTS.

All lists: Check for >16384 list elements.

Stress test with HeapQC, low memory tests. Adjust default and minimum memory partition sizes to more reasonable values. Check for memory leaks and memory fragmentation.

Add entry/exit conditions internal documentation to all functions.

Put all strings in STR# resources.

mark.c/MarkXRefs: Needs rewrite. Should use FindHeader.

MakeNewWindow must call CalcZoom before MyOffset.

Improve keyboard navigation. Rewrite for speed. The current binary search algorithm for navigating the full group and new group lists is goofy - it doesn't work exactly right. Try to fix it. In subject windows, must also find authors.

Only set info->changed if something actually changes. E.g., in subject.c/OpenGroupCell.

Implement arrow modifier keys. Command: Go to edge, then page up/down. Shift: extend selection.

(James Nauer). Down arrow key in lists, when gets to bottom of list, causes "jumpy" scrolling. LAutoScroll is brain dead. Try to fix this.

SysBeep on all alerts and save changes dialogs.

International and A/UX canceling (command-period).

(Eiji Hirai) However, NewsWatcher seems to consider groups with a flag in the 4th field in the active file entry being "x", "n" or "=" to be readable/postable groups. This is not the case and they shouldn't appear in the Full Group List. Ideally, you should only show groups with the "m" or "y" flag.

(Robin Walker) Expand tabs in articles. (Not just when typing in message windows).

(Robin Walker) When get new groups, add to unsubscribed list so when newsrc is written back, UNIX newsreaders don't ask the user to subscribe again.

Wishes, dreams, and other ideas.

This stuff probably won't make it into 2.0 final. It's food for thought for future versions.

Optimize building subject window. Coagulate ranges of unread articles into bigger ranges, throw away unneeded ones. Larry Rosenstein: "I wonder if it's better to make one NNTP request to get all the subjects and just throw away the ones that have already been read." Need to test this and tune it. Optimal algorithm will depend on whether the server is using an xover database for xhdr commands, and on the connection latency. An adaptive algorithm would probably be best.

Async prefetch: next article in current thread, first article in next thread, subject/author info for next group. Observe user's behavior and adapt prefetch algorithm to optimize response time (by group). Larry Rosenstein: "Another idea would be to fetch information in the background, while you are doing something else. If you've ever tried NewsGrazer on the Next, this is what it does. (It makes it more difficult to sort things by subject, however.)" Conrad Nobili also mentioned this idea. Does anybody remember Harry Chelsey's HyperCard newsreader? - it did this kind of prefetching of the next article in

the background. My experience with our server on our network is that fetching the next article doesn't take all that much time, so this is not a super high priority item for me. Matt Mora: "This would be nice while in couch potato mode (when I'm going to read all the articles in a row and just space bar and n keying merrily along) that it prefetch the next article to be read. Also I would like to be reading the article as its arriving (i.e., not wait for the article and then display it). So if its an article I don't care to read, I can go to the next one. NW would then stop downloading the rest of this article and start getting the next one."

(Larry Rosenstein) "I've never tried using the threads code (from develop magazine) so I don't know how well it might work in this case. Another possibility is to divide NewsWatcher into a set of cooperating apps under System 7 (one to download subjects, one to post messages, etc.)" Matt Mora: "We could create an app that just posts messages. It would look in the post messages queue folder and post them when it has time."

Asynchronous sending of messages (postings and mail). Posting takes too long on some servers. When user sends, put message at end of new internal "messages to be sent" queue, close message window, and return control to user immediately. Process queue via idle time task called by main event loop. Use separate nntp connection to send postings. Add new "Message Queue" window. Add new command to windows menu to show/hide this window. Give user some way to cancel pending messages in queue. If error occurs, notify user via alert and reopen window containing message that failed. When user quits, if queue not empty, present alert giving user choice of cancelling the quit or saving the queue on disk (at the end of the prefs file perhaps?). The next time the program runs, it will restore the queue and begin processing it again. Alternatives to this scheme for asynchronous stuff include using MacTCP completion routines, using the Threads package from Develop magazine, and using cooperating faceless background processes under System 7. I think the idle time task is simplest and easiest to implement, even if it's not as sexy (high-tech) as the alternatives.

Add support for the "XOVER" command (if it's available). This would make it feasible to do things like display article sizes in subject windows, or do "true" threading using the "References:" header information.

Replying to info-mac postings and other digest subarticles easily.

For digests, present a popup menu or other style of table of contents to permit easy jumping to individual parts of the digest. That is, some kind of "dedigestify" feature.

Add a preference to check for new news continuously in the background for all open user group lists and subject lists, with the frequency of refresh controlled via a preference. This would be kind of like Eudora's periodic checks for new mail.

Object model Apple events.

Multiple signatures.

MIME.

(Larry Rosenstein) "I often mark articles unread so that I can read them at a later time. (I don't know if this is a common way of using NewsWatcher or

not.) It would be nice if the program could (optionally) cache the subject information locally, rather than having to get it from the server again."

(Larry Rosenstein) "I would like to see the UI rethought, with the idea of making the program more efficient to use. (Figure out how people like to use the program and make those things easier.) Perhaps making use of on-screen buttons vs. menu commands." Perhaps we could hire Tog away from SUN? :-). One possibility is to add buttons to the window panels. We could even let people configure exactly which buttons they wanted, if any, in the panels.

Offline news reading/posting. Scheduled online news fetching/sending. Make NW work with UUCP news feeds.

(Jim McCoy) Here is an idea you may want to consider for Newswatcher, it is something that I am currently playing around with on tin, seeing if it is possible under the current code:

An "archive" command. This works sort of like saving an article, it places the article in a flat dir structure (~/Archive/nwu.general for example) and maintains an active file in ~/Archive. Someone can transfer to the archive at the group selection level (or tin's news spool selection level) and browse a mini news structure of their saved messages (with a real "active" and any threading index files I end up including) that is handled the same way the regular tree is handled (e.g. I will probably just try to hook back in the code to read off the regular news spool)

Basically, I think that the current method of saving articles in every newsreader I have tried is lacking. For starters you lose all the structure that news article definitions imposes and there is no way to do any real work with headers and info structure because the headers all all mushed up with the messages. Hopefully this method will be one easy for a newsreading program to incorporate, familiar to the user, and one that maintains the same message format as the original information.

Cooperate with other network apps via Apple events. E.g., user selects an FTP site and file pathname in a posting. NewsWatcher invokes Fetch or whatever to fetch the file (NeXT newsreaders can already do this!). Lot of other possibilities here, but we all need to implement full object model Apple events before any of it can happen.

Reconfigurable field widths, especially author name field in subject windows (drag dividing line and resize like in a spreadsheet?).

Spell checking via Apple events, as in Eudora (Spellswell 7).