

TkNews II

A Multi-protocol News and Mail client

Robert Heller
Deepwoods Software
Wendell, MA, USA

December 7, 2007

This documentation was prepared with \LaTeX .
This document describes version 2 of the TkNews package.

Copyright ©2006 by Robert Heller D/B/A Deepwoods Software

All rights reserved. Permission is granted to copy this document in electronic form only, so long as it is with the software it documents.

The author, Robert Heller, may be contacted electronically (E-Mail) via the following:

FidoNet 1:321/153, Locks Hill BBS.

InterNet heller@deepsoft.com

Web site URL: <http://www.deepsoft.com/>.

Contents

1	Introduction	1
2	Tutorial	3
3	Reference	7
3.1	Main Window	7
3.2	Spool Window	8
3.3	TkNewsII Resource file	10

List of Figures

2.1	Invoking TkNewsII.	5
3.1	Main Window	8
3.2	Typical Spool Window	9

List of Tables

Chapter 1

Introduction

Chapter 2

Tutorial

1 Tutorial

In order to use TkNewsII, you will need to create a `~/ .TkNewsRc` file. This file contains the preferences and configuration information for the news servers and BBS's you read EchoMail on. See Section 3.3 for a detailed reference on the resources available.

To start with here is a minimal file for reading the local news server (eg talking to innd on the local machine), as shown in Listing 2.1.

The `spoolList` and `autoSpools` global resources list the spool name local and local is used in the resource for `useServer`. When TkNewsII starts up it will create not only its **Main Window** (see Section 3.1) but also a **Spool Window** (see Section 3.2) for the local news server.

The defaults will handle everything else, although you might want to include some additional global settings, as shown in Listing 2.2.

If you use a local BBS that uses QWK, you might include lines like those shown in Listing 2.3.

The script `~/bin/yoyobbs.sh` needs to do the following:

1. Connect to YoyoBBS (eg use `/bin/chat` if you dialing, or use `ssh`, `ftp`, `telnet`, or some other program if the BBS is network connected).
2. Look in `/tmp/spool/` and see if there is a file named `yoyobbs.rep`, and if there is, upload this file to to YoyoBBS and remove `yoyobbs.rep` from the local disk.
3. Ask YoyoBBS for a fresh QWK packet (and process the uploaded `yoyobbs.rep` file).

Listing 2.1: Simple NNTP local mail server.

```
*local*useServer: Yes
*spoolList: local
*autoSpools: local
```

Listing 2.2: Common global resource settings.

```
*organization: Your Organization
*from: Your name and <address>
```

Listing 2.3: Typical resource settings for a QWK BBS.

```
# Define our spools (only one for now).
*spoolList: yoyobbs
# Auto load our one spool.
*autoSpools: yoyobbs
# Directory where the group messages will reside.
*yoyobbs*spoolDirectory: ~/YoyoBBS/spool
# The file containing the counts of actual messages.
*yoyobbs*activeFile: ~/YoyoBBS/active
# File containing my subscribed groups and read messages.
*yoyobbs*newsRc: ~/YoyoBBS/newsrc
# Command to inject new public messages.
*yoyobbs*injectProgram: $QWKREPLY public %spool %active %group
# Command to inject new private messages.
*yoyobbs*emailProgram: $QWKREPLY private %spool %active NetMail
# The netmail group is really a private E-Mail group.
*yoyobbs*netmail*isEmail: yes
*yoyobbs*netmail*injectProgram: $QWKREPLY private %spool %active NetMail
# Place to keep saved messages.
*yoyobbs*savedNews: ~/YoyoBBS/Saved
# Place to put unsent drafts.
*yoyobbs*drafts: ~/YoyoBBS/Drafts
# Yes, enable the Clean function.
*yoyobbs*cleanFunction: Yes
# Command to connect with the BBS to transfer QWK packets.
*yoyobbs*qwkGetMailCommand: ~/bin/yoyobbs.sh
# Location to store QWK packets.
*qwkInSpool: /tmp/spool
*qwkOutSpool: /tmp/spool
```

```
/usr/local/bin/TkNewsII &
```

Figure 2.1: Invoking TkNewsII.

4. Download the fresh yoyobbs.qwk to `/tmp/spool/`.

The script should return an error status if it was not able to complete its processing. It might want to make a backup copy of an existing yoyobbs.qwk in `/tmp/spool/` before it starts (just in case).

It is possible to use multiple BBSs as shown in Listing 2.4.

Once you have your `.TkNewsRc` file set up, you can start TkNewsII, as shown in Figure 2.1.

TkNewsII takes one option, `-iconic`, which will start TkNewsII with its toplevel windows in iconic mode – this is useful if you are starting TkNewsII as part of your desktop startup and want TkNewsII to be started, but not pasted on the screen until a later time.

If you are using a local spool or a NNTP server, it will automatically fetch the active group list from the server on startup. For QWK BBSs it won't. For these you will need to use the File menu on the main window. There are three possible options:

Open QWK File Open a listed QWK file that has already been downloaded to the QWK in spool.

Get QWK File Fetch a QWK file for a selected spool.

Get All QWK Files Fetch all available QWK files for all spools that have a QWK get mail command.

Once the messages have been fetched, you will have a group list showing the available messages in the top part of the spool window. Double clicking on a group opens that group, with displays a list of available articles (messages) in the lower part of the spool window. Double clicking on an article line opens and displays that article (message).

To open any spools not opened automatically (listed in the autoSpools resource) or opened by the **Open QWK File**, **Get QWK File**, or **Get All QWK Files**, the **Review Spool** menu item on the File menu can be used.

Listing 2.4: Resource settings for two QWK BBSes, continued.

```

# Global definitions.
# Define our spools.
*spoolList: yoyobbs buckarooobbs
# Auto load one of them.
*autoSpools: buckarooobbs
# Location to store QWK packets and replys.
*qwkInSpool: /tmp/qwkfiles
*qwkOutSpool: /tmp/qwkreplies
# Buckaroo Banzi's BBS
# Our Id on Buckaroo Banzi's BBS.
*buckarooobbs*from: Banzi Group Leader 777 <BGL777@BuckarooBBS.org>
*buckarooobbs*organization: Banzi Group of Grover's Mills, NJ
# Directory where the group messages will reside.
*buckarooobbs*spoolDirectory: ~/BuckarooBBS/spool
# The file containing the counts of actual messages.
*buckarooobbs*activeFile: ~/BuckarooBBS/active
# File containing my subscribed groups and read messages.
*buckarooobbs*newsRc: ~/BuckarooBBS/newsr
# Command to inject new public messages.
*buckarooobbs*injectProgram: $QWKREPLY public %spool %active %group
# Command to inject new private messages.
*buckarooobbs*emailProgram: $QWKREPLY private %spool %active NetMail
# The netmail group is really a private E-Mail group.
*buckarooobbs*netmail*isEmail: yes
*buckarooobbs*netmail*injectProgram: $QWKREPLY private %spool %active NetMail
# Place to keep saved messages.
*buckarooobbs*savedNews: ~/BuckarooBBS/Saved
# Place to put unsent drafts.
*buckarooobbs*drafts: ~/BuckarooBBS/Drafts
# Yes, enable the Clean function.
*buckarooobbs*cleanFunction: Yes
# Command to connect with the BBS to transfer QWK packets.
*buckarooobbs*qwkGetMailCommand: ~/bin/buckarooobbs.sh
# Yoyodyne's BBS
# Our Id on Yoyodyne's BBS.
*yoyobbs*from: John Warffin <John.Warffin@Yoyodyne.com>
*yoyobbs*organization: Yoyodyne Corporation
# Directory where the group messages will reside.
*yoyobbs*spoolDirectory: ~/YoyoBBS/spool
# The file containing the counts of actual messages.
*yoyobbs*activeFile: ~/YoyoBBS/active
# File containing my subscribed groups and read messages.
*yoyobbs*newsRc: ~/YoyoBBS/newsr
# Command to inject new public messages.
*yoyobbs*injectProgram: $QWKREPLY public %spool %active %group
# Command to inject new private messages.
*yoyobbs*emailProgram: $QWKREPLY private %spool %active NetMail
# The netmail group is really a private E-Mail group.
*yoyobbs*netmail*isEmail: yes
*yoyobbs*netmail*injectProgram: $QWKREPLY private %spool %active NetMail
# Place to keep saved messages.
*yoyobbs*savedNews: ~/YoyoBBS/Saved
# Place to put unsent drafts.
*yoyobbs*drafts: ~/YoyoBBS/Drafts
# Yes, enable the Clean function.
*yoyobbs*cleanFunction: Yes
# Command to connect with the BBS to transfer QWK packets.
*yoyobbs*qwkGetMailCommand: ~/bin/yoyobbs.sh

```

Chapter 3

Reference

The TkNewsII program is a multi-protocol news / EchoMail client. It supports a local USENET news-feed access, NNTP access, and QWK access.

TkNewsII is started from the command line and takes one (optional) command like option: `-iconic`. If this option is present on the command line, the application starts up iconified. TkNewsII is configured with a resource file named `.TkNewsRc`. TkNewsII won't start without this file.

The TkNewsII's GUI consists of a **Main Window**, described in Section 3.1, and zero or more **Spool Windows**, described in Section 3.2.

3.1 Main Window

The **Main Window** consists of the menu bar and a list of QWK files in the QWK spool directory, as shown in Figure 3.1.

The **Main Window's** file menu provides these 6 main functions:

Open QWK File Open the selected QWK file and import its messages into its spool.

Get QWK File Get a QWK file for a selected spool. This generally involves running a user-supplied script that connects to the BBS or server in question and downloading a QWK file. The QWK file is then imported into the spool. Any outgoing messages are collected into a QWK reply file and uploaded.

Get All QWK Files Get QWK files for all spools that define a QWK spool script.

Review Spool (Re-)open a spool window and load its current state.

Rescan Rescan the QWK spool directory and look for fresh QWK files.

Make QWK Reply Collect outgoing messages for a spool into a QWK reply file.

Double clicking on a QWK file in the QWK spool directory listing opens the selected QWK file.

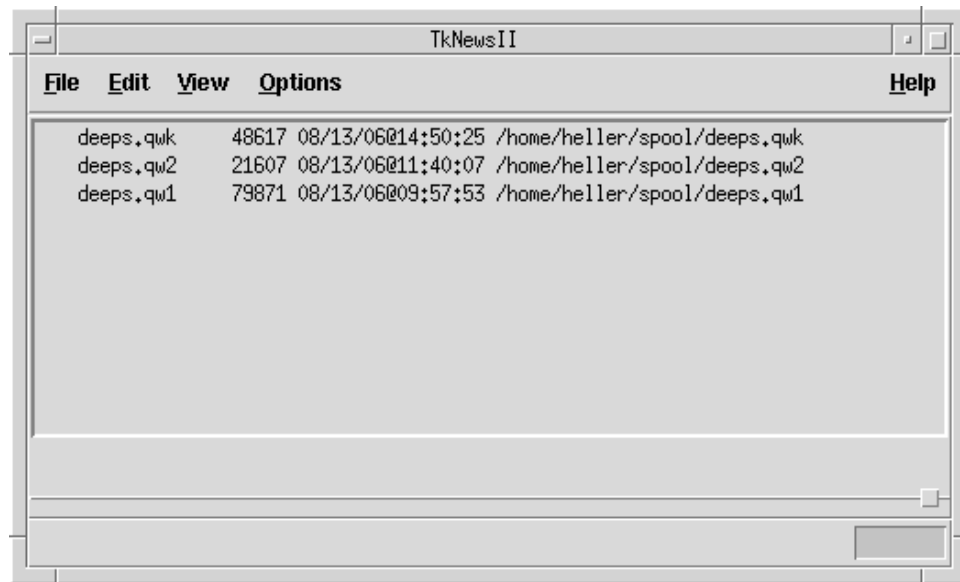


Figure 3.1: Main Window

3.2 Spool Window

Each **Spool Window** contains a menu bar, a group list, and a group level button menu, as shown in Figure 3.2. The file menu provides three main functions:

Read Group The selected group is opened and a list of available articles is shown in the article list.

Post The system opens an article edit window or starts an external editor process.

Clean This function “cleans” messages from a group directory. This function only makes sense for downloaded spools, that is QWK-based spools.

The group list displays the groups subscribed you are subscribed to and also contains a saved message areas as well.

The group menu buttons include:

Unread Group Mark all messages in the group unread.

Read Group The selected group is opened and a list of available articles is shown in the article list.

Close Group Close the group.

Catch Up Group Mark all messages in the group as read.

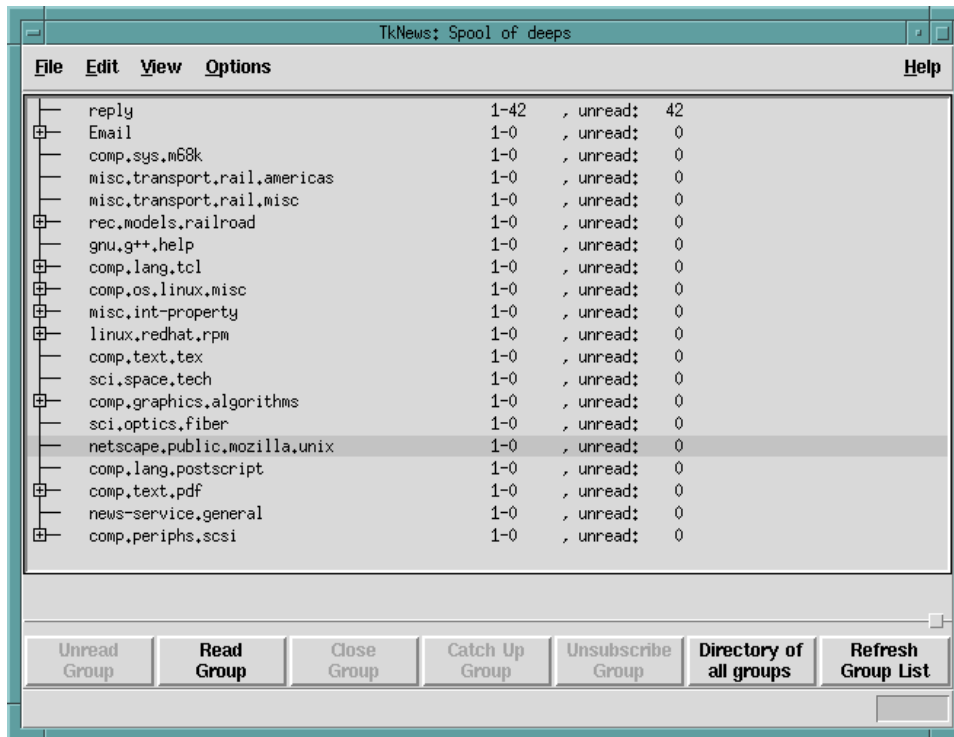


Figure 3.2: Typical Spool Window

Unsubscribe Group Unsubscribe from the group. For QWK spools this only effects TkNewsII and you will need to log onto the BBS and update your profile there.

Directory of all Groups Produce a directory listing of all known groups that match a specified pattern.

Refresh Group List Rescan the group list, updating the message counts.

Double clicking on a group listed in the group list does a **Read Group** on the selected group.

When a group is open an article list button menu, an article list, and an article button menu is added.

The article list button menu contains buttons that:

Compose Same as post.

List/Search Articles Produce a listing off all articles that match a specified pattern.

Read Article Read a selected article. An **Article Read Window** is displayed showing the selected article

Refresh Article List Refresh the article listing.

Manage Saved Articles (Only for saved articles). A **Manage Saved Articles Menu** is displayed,

Double clicking on an article in the article list does a **Read Article** on the selected article.

The article button menu contains buttons that:

Reply To All For E-Mail groups, initiate a reply, using all addresses in From:, Reply-To:, To: and Cc: header lines.

Followup For groups that are not email based, reply to the newsgroup.

Mail Reply To Sender Send a private E-Mail to the sender of the current article.

Save Save (append) the current message to a flat file of messages.

File File the current message in a saved messages directory.

3.3 TkNewsII Resource file

The file `~/TkNewsRC` is the TkNewsII resource file. It is in standard X11 option database format and consists of key pattern value lines. Lines starting with a `#` are comments. Otherwise lines consist of option name patterns containing either names or class names of components or window pathname elements the wild cards (`*`) or periods separating these elements. There is a colon at the end of the pattern and there is a value after that.

The TkNewsII top level window has a class name of `Tknewsii`, so this should be used to introduce resources in the global X11 option database, that is in `.Xdefaults` or `.Xresources` files. In the `.TkNewsRc`, this is not needed. For spool and group based resources, including the name of the spool or group (with periods replaced with underscores) name (all lower case) or class (first character capitalized, the rest lower case) will specialize the resource to the specific spool or group within the spool. Spool windows are always children of the main top level and group windows are always children of the spool window they are part of. Article read windows are also children of the spool window as well. More specialized resources take precedence over less specialized resources.

Resources:

- Global, matched against the **Main Window**:

- **Name:** `spoolList`
Class: `SpoolList`
Value: Space separated list of spool names.
Default: Empty list.

This is the list of defined spools. Each spool is for a single news server or QWK BBS.

- **Name:** `autoSpools`
Class: `AutoSpools`
Value: Space separated list of spool names.
Default: Empty list.

This is the list of spools to load on startup.

- **Name:** `iconic`
Class: `Iconic`
Value: Boolean flag.
Default: 0 (false)

This flag corresponds to the `-iconic` option. If true, the main window and the auto spool windows come up iconified.

- **Name:** `tocGeometry`
Class: `TocGeometry`
Value: X11 Geometry specification.
Default: Empty string.

This is the initial geometry setting for the Main Window.

- **Name:** `qwkInSpool`
Class: `QwkInSpool`
Value: Directory pathname.
Default: `~/spool`

This is the directory where inbound QWK files land.

- **Name:** qwkOutSpool
Class: QwkOutSpool
Value: Directory pathname.
Default: ~/spool

This is the directory where outbound QWK rep files are placed.

- **Name:** qwkWorkDir
Class: QwkWorkDir
Value: Directory pathname.
Default: ~/TkNewsWork

This is the directory where QWK files are unpacked and processed. Also where QWK Reply packages are built.

- **Name:** qwkArchiver
Class: QwkArchiver
Value: Command line.
Default: zip -j

This is the command line used to create a reply file. The name of the reply file and the names of the files going into it are appended to this command string.

- **Name:** qwkUnarchiver
Class: QwkUnarchiver
Value: Command line.
Default: unzip -jo

This is the command line used to unpack a QWK file. The name of the QWK packet is appended to the command like.

- **Name:** printCommand
Class: PrintCommand
Value: Command line.
Default: lpr -

This is the command used to send articles to the printer. The articles are passed as a piped data stream to stdin of this command.

- Per Spool, matched against each **SpoolWindow**:

- **Name:** qwkGetMailCommand
Class: QwkGetMailCommand
Value: Command line.
Default: Empty string.

This is the command (script) used to fetch a QWK packet. If empty for a given spool, that spool does not use QWK to get articles.

- **Name:** spoolNumGroups
Class: SpoolNumGroups
Value: Integer, positive non-zero.
Default: 20

Number of group lines to display in the group list.

- **Name:** activeFile
Class: ActiveFile
Value: Pathname to active file.
Default: /var/lib/news/active

This is the name of the active file. It is a local file in the spool area when the spool is fed via QWK packets. It is not used when a NNTP server is being used.

- **Name:** cleanFunction
Class: CleanFunction
Value: Boolean value.
Default: no

If set to yes, the [clean] menu option is enabled, which allows flushing all of the local copies of the articles in a group. Should only be used with QWK-based spools.

- **Name:** newsRc
Class: NewsRc
Value: File pathname.
Default: ~/.newsrc

This is the name of the file to keep track of read and unread articles and subscribed groups.

- **Name:** savedNews
Class: SavedNews
Value: Directory pathname.
Default: ~/News

This is the directory where news articles are saved via the File article button.

- **Name:** drafts
Class: Drafts
Value: Directory pathname.
Default: ~/Drafts

This is the directory where unsent drafts are stored.

- **Name:** serverName
Class: ServerName
Value: Hostname.
Default: localhost

Name of the NNTP server to use. Only used if useServer is true.

- **Name:** spellChecker
Class: SpellChecker
Value: Command line.
Default: /usr/bin/X11/xterm -e ispell

This is the command line to start up a spell checker. A file name is appended to this command.

- **Name:** externalEditor
Class: ExternalEditor
Value: Command line.
Default: /usr/local/bin/emacs

- **Name:** spoolDirectory
Class: SpoolDirectory
Value: Directory pathname.
Default: /var/spool/news

This is the directory where news articles live. Not used if useServer is true.

- **Name:** useServer
Class: UseServer
Value: Boolean flag.
Default: no

If this flag is true, news articles are fetched via a NNTP server.

- **Name:** spoolGeometry
Class: SpoolGeometry
Value: X11 Geometry
Default: empty string.

This is the geometry of **Spool Windows**.

- **Name:** spoolNumArticles
Class: SpoolNumArticles
Value: Positive, non-zero integer.
Default: 10

This is the number of articles to show in the article list box.

- Per Group, matched against a (fake) group window:

- **Name:** isEmail
Class: IsEmail
Value: Boolean value.
Default: no

If set true, the group is really an E-Mail group or area.

- **Name:** signatureFile
Class: SignatureFile
Value: File pathname.
Default: Empty string.

This is the name of a file containing a signature or tag message.

- **Name:** followupWithXCommentTo
Class: FollowupWithXCommentTo
Value: Boolean value.
Default: no

If set to true, followups include a “X-Comment-To:” header line (needed / used by ifgate / fidonet echo mail).

- **Name:** followupEmailTo
Class: FollowupEmailTo
Value: E-Mail address to send followups to (moderated newsgroups).
Default: empty string.

If not empty, followups are E-Mailed to this address instead of posted.

- **Name:** organization
Class: Organization
Value: String.
Default: Yoyodyne

This is the name of the organization to use with the Organization: header line.

- **Name:** from
Class: From
Value: String.
Default: John Bigbooty

This is the text to include in the From: header line.

- **Name:** ccSelf
Class: CcSelf
Value: Boolean value.

Default: yes

If set true, the Cc: header include your address.

- **Name:** injectProgram
Class: InjectProgram
Value: Command line.
Default: /usr/bin/inews -h

This program is used to inject a news article. The article text is piped to this program's stdin.

- **Name:** emailProgram
Class: EmailProgram
Value: Command line.
Default: /usr/lib/sendmail -em -i -t

This program is used to send an E-Mail message. The message is piped to this program's stdin.

- Per Article, match against each **Article Read Window**:

- **Name:** articleGeometry
Class: ArticleGeometry
Value: X11 Geometry.
Default: Empty string.

This is the geometry used for **Article Read Windows**.

Additional general resources can be specified, such as foreground and background colors as well as fonts and other appearance related resources.