GRn

Michael B. Smith and Markus Illenseer

COLLABORATORS			
	TITLE : GRn		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Michael B. Smith and Markus Illenseer	January 29, 2023	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

GRn

1.1 GRn

```
                    GRn - Gadtools Read News
          for the Amiga


   I.
                Copyright
                   II.
                Overview
                   III.
                Author and Support
                   IV.
                Parameters and Tooltypes
                   V.
                Installation
                   VI.
                Operation
                   VII.
                Pragmatics
                   VIII.
                Thanks
                   IX.
                Common Questions
```

1.2 Beta-Tester Information

```
                   For Copyright information, see
                Copyright
               .

   By agreeing to beta-test GRn, you agree to the following considerations:

     You will not distribute any beta version of GRn.

     You agree to provide me with descriptions of problems that you
     may have, and to work with me in resolving those problems.
```

I may be reached electronically at:

Usenet: mbs@adastra.cvl.va.us, mbs@hopper.itc.virginia.edu
BIX: mbs
MichaelS on IRC.

1.3 GRn Authors

Gadtools Read News
for the Amiga

by

Michael B. Smith
P.O. Box 6791
Charlottesville, VA 22906
mbs@adastra.cvl.va.us

Based on an original work
by
Michael H. Schwartz

1.4 GRn Copyright

GRn - Gadtools Read News

At this time, GRn is FreeWare. It is likely that this is the last freeware version of GRn. Donations are encouraged, and might help motivate me to continue efforts on GRn as freeware.

Copyright 1992 - 1994 by Michael B. Smith.
Copyright 1991 - 1992 by Michael H. Schwartz.

All Rights Reserved.

No warranties, express or implied, are available. Use at your own risk. The authors have taken pains to attempt to produce quality software, but by no means are we liable for anything it does.

1.5 GRn Overview

GRn is a fully-featured Intuition based Newsreader for the Amiga running 2.04 or above. GRn works with AmigaUUCP, C News, and NNTP.

Run time parameters may be set on the command-line for shell users, and in the included GRn.info icon for Workbench users.

GRn requires release 2.04 of AmigaDOS, and works with 2.1, 3.0 and 3.1.

GRn is designed to work with Dillon’s AmigaUUCP versions 1.16 and 1.17. It will work with local and NFS mounted C News partitions as well.

GRn’s NNTP support works with AS225r2, AmiTCP, DNet, AUW and a serial-port based NNTP using GRn-Term.

1.6 Installing GRn

There are four different ways to use GRn as a newsreader:

```
    (1) AmigaUUCP,
    (2) C News,
    (3) NNTP via Ethernet/SLIP (AS-225 or AmiTCP), and
    (4) NNTP via the serial port (includes using DNet and AUW).
```

If you are using AmigaUUCP, C News, or NNTP then you *must* already have these installed and operational. These instructions do *not* cover the process of installing those packages.

If you intend to use GRn to read news via NNTP with AS-225, you must also have socket.library installed. At this writing, socket.library is only available to registered developers of Commodore.

If you intend to use GRn to read news via NNTP with AmiTCP, you must use the special GRn.amitcp binary included in the archive UNLESS you are using AmiTCP 3.0 or above, and have the special socket.library compatibility library installed.

If you do not have AmigaUUCP installed, normal configuration information if supplied through environment variables .

If you already have installed GRn you just might explore the parameters and options.

Options and Parameters

Example Parameters

Description of the GUI

Environment Variables

The following instructions assume that the GRn archive file that you have is named "GRn-2.1.lha". GRn must be installed from the Shell (CLI).

```
0)  Use your existing newsreader, and read all your news. GRn maintains
    its own read/not-read list for each user.

1)  cd ram:
    lha -r -x x GRn-2.1.lha

2)  cd GRn-2.1
```

3) copy fonts fonts: all clone

4a) For non-AmiTCP users:

```
copy GRn UUCP:C clone
copy GRn.info UUCP:C clone
```

4b) For AmiTCP users:

```
copy GRn.amitcp UUCP:C/GRn clone
copy GRn.info UUCP:C clone
```

5) For NNTP via serial users only (includes DNet and AUW):

```
copy GRn-Term UUCP:C clone
```

6) copy GRn.guide UUMan:GRn.guide clone

7) For Shell users only:

```
stack 16000
```

(This command should be placed, using your normal editor, into the S:User-Startup file on your system. If a similar command, with a larger number, already exists, do not change it; and do not place this command in the file.)

8) path uucp:c add

(This command should be placed, using your normal editor, into the S:User-Startup file on your system. If this command already exists, do not change it, and do not place this command in the file.)

9) For Workbench users only:

Edit the TOOLTYPES contained in the project Icon UUCP:C/GRn.info to properly fit your environment.

You are now ready to use GRn.

You should refer to the description of the options and parameters that may be applied to GRn.

Options and Parameters

Example Parameters

Description of the GUI

Environment Variables

1.7 GRn Runtime Examples

By default, GRn assumes that it will be used with AmigaUUCP. If that is *not* the case, then you will be required to set some runtime parameters, either as command-line arguments (for Shell users) or in the GRn Icon (for Workbench users). Some quick examples:

(1) AmigaUUCP

```
GRn wrap 84 screen MyScreen
```

(2) C News

```
GRn cnews wrap 84 screen MyScreen
```

(3) NNTP with AS-225 or AmiTCP

```
GRn nntp nntpserver nntp.virginia.edu screen MyScreen
```

(4) NNTP with serial port

```
GRn nntp device serial.device unit 0 screen MyScreen
```

See GRn-Term.man for examples on using GRn with DNet and AUW.

1.8 GRn Environment Variables

The following environment variables will affect GRn:

```
USERNAME     - the login name of the user running GRn
REALNAME     - the full name of the user
NODENAME     - the Usenet/UUCP/Internet name of your system
DOMAINNAME   - the name of the domain your system is a part of
HOSTNAME     - the fully qualified domain name of your system
ORGANIZATION - the name of the organization you are a part of
NEWSEDITOR   - the program to run to Edit outgoing News
MAILEDITOR   - the program to run to Edit outgoing Mail
SENDMAIL     - the program that sends outgoing Mail
POSTNEWS     - the program that sends outgoing News (non-NNTP)
UULIB      - the directory where the GRnrc is placed/found,
         and where the Config file may be found
UUNEWS       - the directory where news articles reside (non-NNTP)
UUSPOOL      - the directory where the GRn logfile is (by default)
NNTPUSERID   - the authorized NNTP user (NNTP only)
NNTPPASSWORD - the password for the above (NNTP only)
NNTPSERVER   - the hostname of the system providing NNTP access
GRNSAVEDIR   - the directory where "Save Article" requesters start
MUNPACK      - the program started by the 'x' command
REPLYPREFIX  - how to start text in replied news articles
```

USER is a synonym for USERNAME.
NODENAME and DOMAINNAME override HOSTNAME.

For example, here is what they look like on the author's system:

```
USERNAME  mbs
REALNAME  Michael B. Smith
NODENAME  adastra
DOMAINNAME  .cvl.va.us
ORGANIZATION  Private UUCP Node
NEWSEDITOR  dme
MAILEDITOR  dme
SENDMAIL  uucp:c/sendmail
POSTNEWS  uucp:c/postnews
UULIB   uulib:
UUNEWS    uunews:
UUSPOOL   uuspool:
```

The RealName, NodeName, DomainName and Organization are only used if you are running NNTP (for posting articles).

If any of these are not set, GRn attempts to open a file named S:UUConfig (or UULIB:Config if it isn’t present) and find them inside.

The only variable absolutely required for the user to specify is UserName. All others have defaults applied.

If you will be using NNTP , you should also specify RealName, NodeName, DomainName, and Organization. GRn will not normally use these except when you are using NNTP. PostNews is ignored when you are using NNTP, as is UUNews.

MUNPACK is initiated by the ’x’ command in Article Mode, and the current article is passed to it as standard input.

REPLYPREFIX is used to indicate the text which will preceed quoted text in a news article. It defaults to "> ".

1.9 GRn Parameters and Tooltypes

All of GRn’s parameters may be specifed as both Shell command-line arguments, and as Workbench Tooltypes in GRn’s .info file. They are:

```
7WIRE
     = <Switch>

BAUD
      = <Numeric Argument>

CNEWS
     = <Switch>

CUSTOMSCREEN
= <Switch>
```

```
DEVICE
     = <Device Name>

DISCARD
    = <Switch>

FONT
       = <Font Name>

FONTSIZE
   = <Font Size>

HEIGHT
     = <Numeric Argument>

HOST
       = <Local Host Name>,

LAST
       = <Numeric Argument>

LEFT
       = <Numeric Argument>

LOGFILE
    = <Log File>

MODEID
     = <Numeric Argument or "ask">

NNTP
       = <Switch>

NNTPSERVER
 = <Host Name>

NOECHO
     = <Switch>

NORESIZE
   = <Switch>

NOSCAN
     = <Switch>

OVERRIDE
   = <Switch>

PORTNAME
   = <Name of Message Port>

PRIORITY
   = <Numeric Argument>

PUBSCREEN
  = <Screen Name>
```

```
REALNAME
   = <Real Name of User>

SCANALL
    = <Switch>

SUBSCRIBE
  = <News Group List>

TOP
        = <Numeric Argument>

UNIT
       = <Device Name>

UPDATERC
   = <Switch>

USERNAME
   = <Login Name of User>

WIDTH
      = <Numeric Argument>

WRAP
       = <Numeric Argument>
```

1.10 GRn Argument: TOP

```
               Example:

  TOP=10

This tells GRn what the upper ('y') coordinate of its window
should be.

This defaults to "one character" in whatever font is used.

See also:
               LEFT
              ,
               WIDTH
              ,
               HEIGHT
              .
```

1.11 GRn Argument: LEFT

```
               Example:

  LEFT=10
```

This tells GRn what the left-most ('x') coordinate of its window should be.

This defaults to "one character" in whatever font is used.

See also:
TOP
,
WIDTH
,
HEIGHT
.

1.12 GRn Argument: WIDTH

Example:

```
WIDTH=640
```

This tells GRn how wide its window should be.

This defaults to 640.

See also:
TOP
,
LEFT
,
HEIGHT
.

1.13 GRn Argument: HEIGHT

Example:

```
HEIGHT=400
```

This tells GRn how high its window should be.

This defaults to 400.

See also:
TOP
,
LEFT
,
WIDTH
.

1.14 GRn Argument: CUSTOMSCREEN

```
                 Example:

   CS
   CUSTOMSCREEN

 This forces GRn to open a custom screen for it's use. Normally,
 GRn will only open a custom screen if the supplied
                HEIGHT
                and

                WIDTH
                parameters will not fit onto the current default public
 screen.

 When CUSTOMSCREEN is set, GRn determines the optimal display mode
 from the display database to be used in opening the screen,
 according to the
                WIDTH
                and
                HEIGHT
                parameters provided. If they
 are not provided, GRn uses 640x400.

 CUSTOMSCREEN may not be used with
                PUBSCREEN
               .

 See also:
                PUBSCREEN
               ,
                WIDTH
               ,
                HEIGHT
               .
```

1.15 GRn Argument: FONT

```
                 Example:
   FONT=grn.font
   MSGFONT=grn.font

 These two commands (which are synonymous) tell GRn which font is
 to be used for display newsgroups and articles.

 Fonts must be non-proportional.

 They *must* be specified in the format "FONT=grn.font". If not,
 GRn will not find the font.

 This defaults to "grn.font".

 See also:
```

FONTSIZE
.

1.16 GRn Argument: FONTSIZE

Example:

```
  FONTSIZE=8
  MSGFONTSIZE=8
```

These two commands (which are synonymous) tell GRn the size of the font to be used in displaying newsgroup names and articles.

This defaults to 8.

See also:
FONT
.

1.17 GRn Argument: LOGFILE

Example:

```
  LOG=UUSPOOL:LOGFILE
  LOGFILE=UUSPOOL:LOGFILE
```

These two commands (which are synonymous) tell GRn what file to use to log fatal error messages.

If GRn is started from a shell, these messages will also be displayed on the console.

A filename must be fully specified (such as: "LOG=UUSPOOL:LOGFILE").

This defaults to UUSpool:Logfile.

1.18 GRn Argument: WRAP

Example:

```
  WRAP=83
```

This tells GRn how many characters to display in an article before wrapping the line.

Using topaz/8 requires a wrap at about 64. Using grn/8 requires a wrap at 83. To use a different font, refer to the
FONT
and

FONTSIZE
parameters.

This defaults to 82.

See also:
FONT
,
FONTSIZE
.

1.19 GRn Argument: UPDATERC

Example

UPDATERC

This tells GRn to do a batch update of the news tree, open no screens, and display no output (other than errors).

Since it can take quite a while to scan the news tree, especially if you subscribe to any high-volume newsgroups, it is recommended that you place the command "GRn updaterc" in a CronTab, or in your UUPoll script to execute after the news transfer takes place.

If you do this, and use the
NOSCAN
keyword when you otherwise start
GRn, then GRn will start up very quickly.

This will NOT work for serial NNTP, unless you are VERY careful about how you establish communications with the NNTP server. Ethernet/SLIP NNTP connections work just fine.

This defaults to not being set.

See also:
SCANALL
,
NOSCAN
,
DISCARD
.

1.20 GRn Argument: CNEWS

Example:

CNEWS

This indicates to GRn that you are running C News.

This means that GRn will check every file for residence when pruning a news-group, and not assume that the presence of a low-numbered file indicates that all files above it are present. This will take longer, but results in a significant savings in memory. This is required because C News honors the "Expires:" article header, whereas AmigaUUCP does not.

GRn will use the file "UULib:News/Active" to get the valid Newsgroup list for subscriptions, instead of "UULib:NewsGroups".

This has no effect if NNTP is being used. There are interactions between this keyword, the NOSCAN keyword and the SCANALL keyword.

This defaults to not being set.

See also: SCANALL, NOSCAN.

1.21 GRn Argument: PUBSCREEN

Example:

```
  SCREEN=ScreenX
  PUBSCREEN=ScreenX
```

These two commands (which are synonymous) tell GRn which public screen to open it’s window on.

If GRn cannot open its window on the specified screen, GRn will display an error message (if started from the Shell) and/or write it to the logfile (if started from the Workbench). GRn will then abort.

GRn normally attempts to open its screen on the default public screen (which is usually the Workbench). If the default public screen is not large enough ("large enough" is 640x400 if the user does not set WIDTH and HEIGHT), GRn will open a custom screen (hi-res interlaced) that is large enough.

This option may not be used with CUSTOMSCREEN.

This defaults to the default public screen.

See also:
TOP
,
LEFT
, WIDTH ,
HEIGHT
,
CUSTOMSCREEN
.

1.22 GRn Argument: NNTP

Example:

```
NNTP
```

This tells GRn that articles are to be acquired from a remote computer that is has a server process for the Network News Transport Protocol (NNTP), as described in RFC-976.

This option must be set for the
DEVICE
,
UNIT
and
NNTPSERVER
commands to be utilized.

You may use NNTP across Ethernet or the serial port.

GRn will use the file "UULib:News/Active" to get the valid Newsgroup list for subscriptions, instead of "UULib:NewsGroups", if it is present. If it is not present, GRn will download the newsgroup list from the NNTP server. If GRn has to download the newsgroup list, the user will be asked as to whether or not to save the list. If the directory UULib:News does not exist, GRn will not be able to save the list.

This defaults to not being set.

See also:
NNTPSERVER
,
DEVICE
,
UNIT
.

1.23 GRn Argument: DEVICE

Example:

```
DEVICE=serial.device
```

This indicates to GRn that NNTP will occur across the serial port using the device named. Note that some devices, such as dnet.device, may not actually use the serial port, but they emulate it.

If NNTP is not specified, this option is ignored.

This defaults to "serial.device".

See also: NNTP , NNTPSERVER , UNIT

1.24 GRn Argument: UNIT

Example:

```
UNIT=0
```

This tells GRn that NNTP will occur across the serial port using the named (specified) unit of the previously named device. Note that some devices, such as dnet.device, require that UNIT be specified as a high number (such as 8193). Using the internal serial port will allow the user to default to zero.

If NNTP is not set, this option is ignored.

This defaults to 0.

See also: NNTP , NNTPSERVER , DEVICE .

1.25 GRn Argument: NNTPSERVER

Example

```
  NNTPSERVER=nntp.local.domain
```

This indicates to GRn that NNTP will occur across a TCP/IP link using AS225r2 or AmiTCP . If you do not have one of these packages installed, this will NOT work.

If NNTP is not specified, this option is ignored.

This has no default.

See also: NNTP , DEVICE , UNIT .

1.26 GRn Argument: PORTNAME

Example:

```
  PORT=GRN_PORT
  PORTNAME=GRN_PORT
```

These two commands (which are synonymous) tell GRn what the name of GRn’s message port should be. Currently, this is of no use. However, GRn checks to ensure that this port does *not* currently exist.

If it does, GRn informs you that another copy of GRn is already running.

You are strongly advised against running multiple copies of GRn.

This defaults to "GRN_PORT".

1.27 GRn Argument: PRIORITY

Example:

```
  TOOLPRI=2
  PRIORITY=2
```

These two commands (which are synonymous) tell GRn what priority to run at.

GRn normally inherits its caller’s priority, and does not set it.

1.28 GRn Argument: 7WIRE

Example:

```
7WIRE
```

This command tells GRn to set the 7-wire field for hardware flow control when using Serial based NNTP.

This defaults to not being set (inherited from Preferences/Serial).

See also:
NNTP
,
DEVICE
,
UNIT
.

1.29 GRn Argument: BAUD

Example:

```
BPS=9600
BAUD=9600
```

These two commands (which are synonymous) tell GRn at what baud rate the specified device and unit are to be set to, when using Serial based NNTP.

This defaults to the user's Preferences/Serial default.

See also:
NNTP
,
DEVICE
,
UNIT
.

1.30 GRn Argument: SUBSCRIBE

Example:

```
SUBSCRIBE ng1 ng2 ngn
```

Use of this keyword causes GRn to subscribe to the named newsgroups as part of the GRn initialization.

This can significantly lengthen the time need for initialization, but simplifies the process of subscribing to multiple newsgroups.

This does work with normal NNTP, but not with serial NNTP.

This defaults to not being set.

See also:
SCANALL
,
NOSCAN
,
Subscribe Window
.

1.31 GRn Argument: DISCARD

Example:

```
DISCARD
```

When DISCARD is set, GRn will not write messages marked as READ back to the user's GRnrc. This can significantly reduce the amount of memory which GRn consumes at runtime, and decrease the size of the user's GRnrc file. However, the highest read message is never discarded (GRn requires it for bookkeeping).

On the downside, to re-read those message after they have been discarded requires Junking the newsgroup (unsubscribing) and then resubscribing, as GRn no longer knows about those messages.

To retain a message, you can
mark
it for retention.

The initial use of DISCARD can require a long pause during the time GRn is starting up (scanning the news tree).

This defaults to not being set.

See also:
Junk
,
mark
,
Subscribe Window
.

1.32 GRn Argument: NOECHO

Example:

```
NOECHO
```

Certain devices used for serial NNTP do not provide an "echo" like the serial.device does (e.g., dnet.device). When NOECHO is set, GRn will not expect that echo to be present.

This defaults to not being set.

See also:
NNTP
,
DEVICE
,
UNIT
.

1.33 GRn Argument: NORESIZE

Example:

NORESIZE

When NORESIZE is set, GRn does not allow it's window to be resized. This option is primarily for people running GRn on non-laced screens where every pixel is important (the resize gadget consumes 10 pixels of height -- or one line of text in the article mode display).

This defaults to not being set, and GRn will allow its window to be resized.

See also:
TOP
,
LEFT
,
HEIGHT
,
WIDTH
.

1.34 GRn Argument: SCANALL

Example:

SCANALL

SCANALL controls how new locally stored news articles are found, and how they are removed from the user's GRnrc. When SCANALL is set, all articles in a newsgroup are checked for existence in the news tree. New articles are added, and old articles are removed.

SCANALL does not have any effect for NNTP.

This option should be set for systems running C News, as C News does honor the Expires article header, and C News does not generate the

.next files which AmigaUUCP uses.

Users who are using the DCFS filesystem for their news partition should try SCANALL and determine whether it provides improved performance in their environment. It may, or may not, depending on the number of articles present.

Should a GRnrc become corrupted for any reason, using SCANALL can correct corruption in many instances.

Since SCANALL can consume a fair amount of time, user's should consider placing a "GRn UPDATERC" in Cron after each news transfer to their system, and execute GRn with the NOSCAN option to speed up GRn's startup in the normal case.

SCANALL defaults to not being set.

See also:
NOSCAN
,
UPDATERC
.

1.35 GRn Argument: HOST

Example:

```
HOST myhost.domain.com
```

Setting HOST as a GRn parameter is equivalent to setting the HOSTNAME environment variable. Specifying it as a parameter or tooltype will simply the process of using GRn in a multiple user environment (just have a different Icon for each user).

Setting NODENAME and DOMAINNAME as environment variables will override the use of the HOST argument (and the HOSTNAME environment variable, for that matter).

Using the parameter HOST overrides the environment variable HOSTNAME (it therefore also overrides the configuration file).

HOST defaults to not being set.

See also:
Environment Variables
.

1.36 GRn Argument: USERNAME

Example:

```
USERNAME johnny
```

Setting USERNAME as a GRn parameter is equivalent to setting the USERNAME environment variable. Specifying it as a parameter or tooltype will simplify the process of using GRn in a multiple user environment (just have a different Icon for each user).

Using the parameter USERNAME overrides the environment variable USERNAME (it therefore also overrides the configuration file).

USERNAME defaults to not being set.

See also:
Environment Variables
.

1.37 GRn Argument: REALNAME

Example:

```
REALNAME "Johnny B. Good"
```

Setting REALNAME as a GRn parameter is equivalent to setting the REALNAME environment variable. Specifying it as a parameter or tooltype will simplify the process of using GRn in a multiple user environment (just have a different Icon for each user).

Using the parameter REALNAME overrides the environment variable REALNAME (it therefore also overrides the configuration file).

REALNAME defaults to not being set.

See also:
Environment Variables
.

1.38 GRn Argument: NOSCAN

Example:

```
NOSCAN
```

When NOSCAN is set, GRn will not attempt to prune or add articles to the news tree. This will result in a significant improvement in GRn startup. However, it will not find any new articles which have become available since the last time a scan was executed, not will it discover which articles have expired.

NOSCAN only effects the initial startup scan. After startup, if the user clicks on the "Rescan News Tree" button in Group Mode, a full new article scan will occur.

This does affect NNTP operations.

NOSCAN defaults to not being set.

See also:
SCANALL
,
C News
.

1.39 GRn Argument: OVERRIDE

Example:

```
  OVERRIDE
```

By default, GRn will *not* allow you to cancel news articles which you did not author. However, there are times when this may be required. Setting OVERRIDE provides that opportunity.

USE THIS WITH CARE: Cancelling a news articles does not remove it from your disk, thus it does not save any disk space. Cancelling an article will not stop an article if its already propagated all the way around the world. At best, if used more than a couple of hours after you've submitted an article, its worthless.

1.40 GRn Argument: LAST

Example:

```
  LAST=50
  SUBSCRIBELAST=50
```

When using NNTP to subscribe to newsgroups, the transfer of the header information can consume a significant amount of time during the initial scan. The LAST (or SUBSCRIBELAST, which is synonymous) control how many headers will be transferred (and thus, how many of those messages will be read).

LAST only applies when NNTP is set.

1.41 GRn Argument: MODEID

Examples:

```
  MODEID ask
  MODEID 0x290014
  MODEID 136723

  SCREENMODE ask
  SCREENMODE 0x290014
```

SCREENMODE 135723

MODEID and SCREENMODE are synonymous.

When "MODEID ask" is specified, on Workbench Release 2.1 (V38) or above, GRn will present the ASL ScreenMode requester to allow the user to choose what type of custom screen that GRn should operate on. If on Release 2.0 (V37) "modeid ask" is not allowed.

When "MODEID <number>" is specified, GRn will attempt to open a custom screen of the specified type. Various utilities provide the numbers for available screenmodes. This option for MODEID is available on Release 2.0.

Specifying MODEID implies CUSTOMSCREEN and overrides PUBSCREEN.

1.42 GRn Basic Information

GRn's icon is named "GRn.info".

The GRn.info file that comes with this release has examples of most possible tooltypes. All are commented out, but you can see the exact format that is required.

If the
TOP
,
LEFT
,
WIDTH
, and
HEIGHT
arguments are specified
then GRn will NOT attempt to open its normal custom screen. Also, if you specify the screen dimensions, they will override GRn's concept of what they should be. This can be dangerous, as a screen that is too small could cause GRn to have major problems placing gadgets. GRn will not use a screen smaller than 600x200.

Using serial-based NNTP *requires* an error-free connection. If you don't have an MNP modem, or a V.42 modem, don't even try it. If GRn doesn't receive an expected response or an appropriate error, it will appear to hang. The serial routines will return an error if they receive a CTRL-C. If starting from a Shell, you *must* give the CTRL-C from the shell where GRn was started, not on the GRn window. When using GRn-Term, the CTRL-C must be given using the C:Break command, as GRn is started as an independent process.

If you are not using the internal serial port, you may need to specify "NOECHO", if GRn appears to hang at startup (and there is no modem activity).

Using Ethernet-based NNTP requires the Commodore AS225 package, along with (currently available to developers only) socket.library. Alternately, you can use the special GRn.amitcp with AmiTCP 2.2 or above.

PostNews is ignored for NNTP, but SendMail is still required. GRn does not do SMTP. Get INetUtils for that.

If NNTP is set, GRn tries to use UULib:News/Active for a valid newsgroup list for subscription purposes. If it doesn't exist, GRn requests a list of valid newsgroups from the NNTP server and saves it throughout a session.

GRn builds temporary article files in T: when using NNTP. Large articles can take a long time to transfer, as well as consume a great deal of memory.

1.43 GRn GUI

GRn operates in a single window, which can have two different looks to it.
The two modes of operation are Group Mode and Article Mode. In Group Mode, the user performs operations on newsgroups. In Article Mode, the user performs operations on the articles in the selected newsgroup.

There is a close gadget on GRn's window. Use this gadget ONLY if you want to quit without updating your GRnrc file! The "Quit" button on the Group Mode window is the normal method of quitting which saves updated news tree information to your GRnrc file.

Both Group Mode and Article Mode have a Text gadget near the bottom of the window which is used to display status messages. For Group Mode, the total number of newsgroups, articles, and unread articles is displayed. For Article Mode, the article number, remaining article count, and the name of the next newsgroup is displayed.

Group Mode

Article Mode

1.44 GRn GUI - Group Mode

In Group Mode, you are presented with a list of all the newsgroups, along
with the number of articles in each newsgroup and the number of unread articles in each newsgroup. Below this list of groups are a set of buttons, most of which operate on the selected newsgroup.

You must select a newsgroup as the first step for any of the newsgroup related functions (that is, click on the newsgroup name). Click once on a newsgroup to select it and the various buttons become active as appropriate. For example, it doesn't make sense to be able to read a newsgroup which has no articles, so the "Read" button will be disabled. Double-clicking on the newsgroup is the same as clicking once and then clicking "Read".

The buttons are:

```
                 Quit

                 Read

                 Post

                 Catchup

                 Publish

                 Junk

                 Subscribe

                 Bug Report

                 Rescan News

                 UUCP Bug Report

                 Save/Update GRnrc
```

1.45 GRn GUI - Group Mode - Quit

```
Keyboard Equivalent: Q

Quit GRn, saving the updated news tree if needed.
```

1.46 GRn GUI - Group Mode - Read

```
                Keyboard Equivalent: R  (requires newsgroup selection)

Read articles in the indicated newsgroup. GRn will transfer to

             Article Mode
            .

If there are not any articles in the indicated newsgroup, you will
not be allowed to select "Read".
```

1.47 GRn GUI - Group Mode - Post

```
                Keyboard Equivalent: P  (requires newsgroup selection)

Post an article to the indicated newsgroup.

Your specified editor will be executed for you to enter the article.
After you exit your editor, GRn requests you to confirm that you
still wish to post the article. If you confirm, GRn does so. If not,
```

the article is discarded. GRn also detects whether or not you actually changed the article. If you did not, GRn asks you whether to abort the posting or not.

See also:
Environment Variables
.

1.48 GRn GUI - Group Mode - Catchup

Keyboard Equivalent: C (requires newsgroup selection)

All articles for the indicated newsgroup will be marked as "read".

1.49 GRn GUI - Group Mode - Publish

Keyboard Equivalent: none (requires newsgroup selection)

Post a specified file, uuencoded and split, to the indicated newsgroup.

A file requester will be presented for you to choose the desired file to publish. GRn will automatically uuencode and make as many 500 line posts as needed to completely post the file. The subject line of each message will be formatted as:

```
    Subject: <filename> (part x/n)
```

where <filename> is the file being published, ’x’ is the current post, and ’n’ is the total number of posts required.

The user is given the option to prepare an introductory article for the published file. If the user chooses to prepare that articles the subject line will be:

```
    Subject: <filename> (part 0/n)
```

Each post will have two additional lines in the following form:

```
    Archive-Name: <filename>/partx
    Submitted-By: <e-mail address>
```

This is to simplify the process some archival software must use.

See also:
Environment Variables
.

1.50 GRn GUI - Group Mode - Junk

Keyboard Equivalent: J (requires newsgroup selection)

Unsubscribe to the indicated newsgroup.

Selecting Junk will unsubscribe the user, and remove all reference to the newsgroup from the user's GRnrc. It does not affect whether AmigaUUCP, C News or NNTP continues to process the newsgroup.

If the gadget is clicked, GRn will ask for confirmation. If the keyboard is used, GRn does *not* ask for confirmation.

The user's GRnrc is not actually updated until either GRn is exited normally, or until Save GRnrc is selected.

See also: Save GRnrc, Quit, Subscribe.

1.51 GRn - GUI - Group Mode - Subscribe

Keyboard Equivalent: S

Subscribe to newsgroups.

Selecting subscribe will cause GRn to obtain the list of valid newsgroups for this system (for AmigaUUCP, UULib:Newsgroups; for C News, UULib:News/Active; for NNTP, UULib:News/Active if it exists, from the server if not) and present the user with a list of newsgroups which may be subscribed to.

GRn will prevent the user from subscribing to a newsgroup which is already subscribed.

The user is asked to confirm the subscription prior to GRn actioning it.

To exit Subscribe Mode, click on the close button for the Subscribe Window, or enter ESCAPE.

For GRn to acquire the articles, if any, which currently exist for the newly subscribed newsgroups, click on RESCAN.

Subscribe Mode supports a variety of keyboard shortcuts to simplify the process of moving through the newsgroup list. They are:

Enter:	down one newsgroup
down-cursor:	down one newsgroup

```
up-cursor:     up one newsgroup
shift-up-cursor:   top of list
shift-down-cursor: bottom of list
any-alpha-char:    move to the first newsgroup starting
       with that character
space:       forward one page of newsgroup names
backspace:     back one page of newsgroup names
```

To remove a newsgroup from the Subscription list, use the Junk command.

The user's GRnrc is not actually updated until either GRn is exited normally, or until Save GRnrc is selected.

See also: Junk, Save GRnrc.

1.52 GRn - GUI - Group Mode - Bug Report

Keyboard Equivalent: B

Send e-mail to "grnbugs@adastra.cvl.va.us" (GRn maintainer), or to the address specified by the environment variable BugAddress.

This allows the user easy access to GRn's maintainer for the processing of bug reports and/or enhancements.

The user specified mail editor will be executed to enter the message. After the editor has completed, GRn requests the user to confirm that the message should be sent. If the user confirms, GRn then mails the message. If not, the message is discarded.

GRn also detects whether or not the user actually changed the message. If no change is detected, GRn will ask whether to abort the mail, or send it anyway.

See also: Environment Variables, Mail Editor.

1.53 GRn - GUI - Group Mode - UUCP BugReport

Keyboard Equivalent: U

Post a message to the newsgroup alt.sys.amiga.uucp.

This simplifies the process of posting a bug report on UUCP to the alt.sys.amiga.uucp newsgroup.

The user specified news editor will be executed to enter the article. After the editor has completed, GRn requests the user to confirm that the article should be posted. If the user confirms, GRn then posts the article. If not, the article is discarded.

GRn also detects whether or not the user actually changed the article. If not, GRn asks to abort the posting.

The user does not need to be subscribed to alt.sys.amiga.uucp.

See also:
News Editor
.

1.54 GRn - GUI - Group Mode - Save

Keyboard Equivalent: none

Update the user’s GRnrc file.

This will take a current snapshot of the user’s article and newsgroup tree and save it to disk. This is the same operation that occurs when the user selects
QUIT
from Group Mode.

Updated information from the
Junk
and
Subscribe
commands is not
stored to disk until either this button is selected, or GRn exists normally.

See also:
Junk
,
Subscribe
.

1.55 GRn GUI - Group Mode - Rescan News Tree

Keyboard Equivalent: none

This causes GRn to rescan the news tree (for local C News and AmigaUUCP) or to reinterrogate the server (for NNTP) to see if any new articles have arrived.

Note that some versions of NNTP (notably the NNTP 1.5.11 reference implementation) will not work properly with rescan. INN does.

If so, GRn's newstree is updated.

See also:
Save
.

1.56 GRn - GUI - Article Mode

In Article Mode, the window takes a different display format ↩
than that
of Group Mode. The list at the top becomes a list of the articles in the selected group, including the status (READ/UNREAD), the article number, the sender (parsed from the "From:" field of the article), and the subject.

There are two checkboxes below the list, named "Hide Headers" and "Hide Read Messages". When "Hide Headers" is checked, the browser window at the bottom of the window will not display article headers. When "Hide Read Messages" is checked, the articles list will only show UNREAD articles. Following the two checkboxes on the same line, is a
Sort by
cycle gadget
for sorting the order in which articles are presented.

These three options are maintained on a group-by-group basis. Each group can have these options set independently.

The articles may be sorted by Number, From, or Subject fields. This allows the user a great deal of flexibility as to how the "next message" and "previous message" functions operate.

Below the checkboxes is a row of buttons. The buttons are:

Groups

prev

next

unread

thread

Catchup

Next Group

Prev Group
Below the buttons are text gadgets used to display the

From:
Date:

Organization:
Subject:

headers from the currently selected article. When the Hide Headers option is selected, this is typically the only header information that is significant. If the user wishes to see other information in the headers, this can be enabled with a click of the mouse.

Below the text gadgets is the article browser window. This window will contain the text (with or without headers) of the selected article. The user has a choice of using the mouse to scroll this window or of using the keyboard.

Intermixing use of the mouse and the keyboard is not recommended on releases of Workbench lower than 3.0.

Keyboard commands for controlling the article display are:

Enter:	down one line
cursor-up:	up one line
cursor-down:	down one line
shift-cursor-up:	top of article
shift-cursor-down:	bottom of article
space:	forward one page
backspace:	backward one page

cursor-right:	next article
cursor-left:	previous article
shift-cursor-right:	next newsgroup
shift-cursor-left:	previous newsgroup
0:	catchup current newsgroup (mark all articles as READ)

Keyboard commands for controlling the article list are:

ALT-cursor-up:	move list up by one item
ALT-cursor-down:	move list down by one item
ALT-shift-cursor-up:	move list to top
ALT-shift-cursor-down:	move list to bottom
ALT-space:	move list forward by one "page"
ALT-backspace:	move list backward by one "page"

Below the browser window is another row of buttons that operate on the currently selected article.

Save

Reply

Forward

Followup

Post

Print

Cancel

mark

The following buttons are not listed, but are valid commands:

Rot13

Extract

1.57 GRn - GUI - Article Mode - Sort by

Articles may be sorted by Number, From, or Subject fields. This allows the user a lot of flexibility as to how the next and prev functions work. If sorted by Number, articles are presented in the order the messages were received by the system. If sorted by From, articles are presented in alpha order by author, then by article number. If sorted by Subject, articles are presented in alpha order by Subject, then by article number. Sorting by Subject provides a cheap threading mechanism.

1.58 GRn - GUI - Article Mode - Save

Keyboard Equivalent: S

Save the current article to a file.

GRn will present a standard file requester for the user to select a filename to save the article in.

The initial directory of the file requester is the directory the user was in when GRn was initiated, unless GRnSaveDir was specified as an environment variable by the user.

If the file already exists, GRn will append the article to the end of the existing file.

See also:

Environment Variables

.

1.59 GRn - GUI - Article Mode - Reply

Keyboard Equivalent: R

Reply to the current article via e-mail.

The user is asked whether to include the text of the current article in the reply.

The user specified mail editor will be executed to enter the message. After the editor has completed, GRn requests the user to confirm that the message should be sent. If the user confirms, GRn will execute the user specified mail program to send the message. If not, the message is discarded.

GRn will also detect whether or not the user actually created a message. If not, GRn asks to abort the mail.

See also:
Environment Variables
.

1.60 GRn - GUI - Article Mode - Forward

Keyboard Equivalent: none

Forward this article via e-mail.

GRn prepends an "ORIGINALLY FROM" line to the article prior to calling the user's mail editor. Headers from the message are not included.

The user specified mail editor will be executed to enter the message. After the editor has completed, GRn requests the user to confirm that the message should be sent. If the user confirms, GRn will execute the user specified mail program to send the message. If not, the message is discarded.

GRn also detects whether or not the user actually created a message. If not, GRn asks to abort the mail.

See also:
Environment Variables
.

1.61 GRn - GUI - Article Mode - Followup

Keyboard Equivalent: F

Post a follow-up article to the current article.

The user is asked whether to include the text of the current article in the followup article.

The user specified news editor will be executed to enter the article. After the editor has completed, GRn requests the user to confirm that the article should be posted. If the user confirms, GRn then posts the article. If not, the article is discarded.

GRn also detects whether or not the user actually created an article. If not, GRn asks to abort the posting.

See also:

Environment Variables

.

1.62 GRn - GUI - Article Mode - Post

Keyboard Equivalent: none

Post a new message in the current newsgroup.

The user specified news editor will be executed to enter the article. After the editor has completed, GRn requests the user to confirm that the article should be posted. If the user confirms, GRn then posts the article. If not, the article is discarded.

GRn also detects whether or not the user actually created an article. If not, GRn asks to abort the posting.

See also:

Environment Variables

.

1.63 GRn - GUI - Article Mode - Print

Keyboard Equivalent: none

Print the current article on PRT:.

GRn will attempt to open PRT: and write the current article to that device. GRn sends a formfeed after all of the article has been printed.

1.64 GRn - GUI - Article Mode - Cancel

Keyboard Equivalent: D (for Delete)

Submit a Cancel control message for the current article.

Use with care! GRn will build a control message, based on the header information in the current article, that will cause C News, NNTP, and B-News sites to cancel the article.

The user is asked to confirm (twice) this action, prior to GRn performing the operation.

You are only allowed to Cancel articles which you authored.

AmigaUUCP ignores control messages.

1.65 GRn - GUI - Article Mode - ROT13

```
Keyboard Equivalent: ctrl-X     (no button exists for this command)

Causes the current message to be redisplayed, with a rotational 13
translation to be applied to it.

Rot-13 is often used in news articles which may contain offensive
material.
```

1.66 GRn - GUI - Article Mode - Extract

```
                Keyboard Equivalent: X      (no button exists for this command ↩
                   )
Keyboard Equivalent: x

Causes MUNPACK to be initiated with the current article as its
standard input. MUNPACK is a program that can deal with certain
types of MIME messages, as well as UUENCODED messages.

The output of MUNPACK is placed in the directory that GRn was
started from.

See also:
             Environment Variables
            .
```

1.67 GRn - GUI - Article Mode - Groups

```
Keyboard equivalent: G

GRn will return to Group Mode when this is selected.
```

1.68 GRn - GUI - Article Mode - prev

```
Keyboard Equivalent: p
Keyboard Equivalent: b
Keyboard Equivalent: cursor-left

GRn will display the prior article, if one exists.
```

1.69 GRn - GUI - Article Mode - Next

```
Keyboard Equivalent: n
Keyboard Equivalent: cursor-right

GRn will display the next article, if one exists.
```

1.70 GRn - GUI - Article Mode - Unread

Keyboard Equivalent: u

GRn will display the next unread article, according to the newsgroup's specified sort sequence.

If there are no unread articles, the article displayed will not change.

See also: {" Sort By " link SORTBY},
Mark
.

1.71 GRn - GUI - Article Mode - Thread

Keyboard Equivalent: t

GRn will display the first unread article in the next thread which has unread articles (threading is controlled by "References:" headers in the article header).

Proper use of this function requires that "Hide Read" NOT be set, and that DISCARD not be set. As this may be too excessive for many people, the user may use the Sort-by Subject cycle gadget to simulate threading.

If no thread has unread articles, the article displayed will not change.

See also:
Article Mode options
,
DISCARD
,
Sort by
.

1.72 GRn GUI - Article Mode - Catchup

Keyboard Equivalent: C
Keyboard Equivalent: 0

All articles for the indicated newsgroup will be marked as "read".

1.73 GRn - GUI - Article Mode - Next Group

Keyboard Equivalent: N
Keyboard Equivalent: shift-cursor-right

GRn will change the current group to the next group which has unread articles. If there are no newsgroups with unread articles, GRn changes the current group to the next group in the user’s newsgroup list. If there is only one group in the user’s newsgroup list, GRn doesn’t do anything.

If there are any unread articles in the new newsgroup, GRn will display the first unread article according to the newsgroup’s sort sequence.

If there are no unread articles in the new newsgroup, GRn will display the last article in the newsgroup.

1.74 GRn - GUI - Article Mode - Prev Group

Keyboard Equivalent: P
Keyboard Equivalent: shift-cursor-left

GRn will change the current group to the previous group in the newsgroup list.

If there are any unread articles in the new newsgroup, GRn will display the first unread article according to the newsgroup’s sort sequence.

If there are no unread articles in the new newsgroup, GRn will display the last article in the newsgroup.

1.75 GRn - GUI - Artical Mode - mark

Keyboard Equivalent: m

GRn will mark the current article as UNREAD, and advance to the next unread article. If the current article is the last article in the group, GRn will mark the article as UNREAD, and not advance.

1.76 GRn Thanks

Each person that has submitted a has submitted a bug report efforts are appreciated. Some folks effort deserves extra recognition. Those are:

Christopher A. Wichura - for assistance in getting GRn "clean" of Enforcer hits and Mungwall violations; and for the basic code that reads tooltypes.

Scott Ellis - for help debugging SerialNNTP (internal serial.device).

Bill Coldwell - for help debugging SerialNNTP with DNet.

Bob Lindabury - for help making GRn work well with C News.

Markus Illenseer - for the initial AmigaGuide format documentation.

To the many people (Markus I., Petra Z., Chris W., Paul K., Willy L. and others) who worked with me making GRn work properly via Ethernet/SLIP NNTP.

1.77 GRn - Problem Resolution

```
                If you think additional problems should be listed in this area, ↩
                   please
contact the GRn maintainer.

1)  "I can’t get GRn to execute SendMail or PostNews".

  Hmm. This shouldn’t happen anymore on 1.16f+, UNLESS you don’t have
  the directory where SendMail and PostNews are in the Workbench path
  when you started GRn.

  Try quiting GRn, "loadwb newpath" and then restarting GRn.

  If that doesn’t work, then:

1) Put GRn in the same directory with the other programs (normally
   uucp:c).

2) Explicitly name the full pathnames to the programs in your
   uulib:config file.

  or:

3) Set the environment
               variables
               SENDMAIL and POSTNEWS to point
   to the full pathnames of the programs in your uulib:config
   file.

2)  "GRn doesn’t open on my custom/public/whatever screen"

  As a Workbench user, you simply need to edit the
               tooltypes
               in GRn’s icon
  (grn.info) to set LEFT, TOP, HEIGHT, WIDTH, and PUBSCREEN. Then, GRn will
  use your existing screen UNLESS GRN WILL NOT FIT. GRn requires a minimum
  screen size of 600x200.

  As a Shell user, just pass these as arguments when you start GRn.

  If you ALWAYS want a CUSTOM screen, then use the CUSTOMSCREEN argument
  to GRn.

3)  "GRn doesn’t use my custom font"

  As a Workbench user, you simple need to edit the
               tooltypes
               in GRn’s icon
  (GRn.info) to set MSGFONT and MSGFONTSIZE. The MSGFONT must be in the
  "anyfont.font" style, or GRn will fail.
```

As a Shell user, just pass these are arguments when you start GRn.

4) "GRn doesn't scroll properly with my custom font"

Well, I thought I fixed this problem. Maybe not. Send me your font, and I'll try to fix the problem. Make sure you tell me what size screen you are using GRn on, and whether you are PAL or NTSC.

5) "GRn doesn't use my custom font in the gadgets"

True. It's a pretty big bother to have variable sized gadgets. At this time, I've decided against them. If enough people ask for them, I can probably be convinced to change my mind.

7) "How do I use GRn with C News?"

Use environment
variables
:

USERNAME - the login name of the user running GRn
REALNAME - the full name of the user running GRn
NEWSEDITOR - the program to run to Edit outgoing News
MAILEDITOR - the program to run to Edit outgoing Mail
SENDMAIL - the program that submits Mail
POSTNEWS - the program that submits News

You STILL must have a directory named UUNEWS:, which is the directory where the top-level news directories exists. You must also have a UULIB: directory that contains signature files, etc. It may contain a file named CONFIG that contains standard AmigaUUCP configuration information. GRn will use those redirections, if they exist, for UULIB and UUNEWS. GRn will first check environment variables before opening the CONFIG file.

See also the CNEWS, SCANALL and NOSCAN arguments.

8) "How do I use GRn with
NNTP
?"

It's pretty easy.

Use environment
variables
:

USERNAME - the login name of the user running GRn
REALNAME - the full name of the user running GRn
NEWSEDITOR - the program to run to Edit outgoing News
MAILEDITOR - the program to run to Edit outgoing Mail
SENDMAIL - the program that submits Mail to UUCP

GRn still uses the UULib: directory to look for the username.grnrc and username.signature, etc. It may contain a file named UULib:Config that contains standard AmigaUUCP configuration information. If you haven't set the UULIB environment variables, but HAVE redirected in that file, GRn will use those redirections, if they exist.

When you start GRn, specify the NNTP and NNTPSERVER arguments, to tell GRn that you are using NNTP.

9) "I’m using GRn with C News, and GRn doesn’t seem to notice new articles in some newsgroups. What can I do?"

The C News EXPIRE program can leave "holes" in article numbers. This is not a bug, just a different way that the C News EXPIRE works.

To minimize this problem, ensure that "GRn CNEWS UPDATERC" is executed BEFORE each EXPIRE run. This will make sure that GRn is as up-to-date as possible, and that there are as few holes as possible.

You can also start GRn using the SCANALL option after each RELAYNEWS run in your Crontab, and then use the NOSCAN when you start GRn. Read up on those options.