term

A terminal program for Amiga computers

Version 4.0
16 June 1994

by Olaf Barthel

Copyright © 1990-1994 Olaf Barthel

You may make and distribute verbatim copies of this documentation if the contents are unchanged or the author has agreed to any changes made.

No guarantee of any kind is given that the program described in this document are 100% reliable. You are using this material on your own risk.

The program 'term' and the data received/sent by it must not be used for the following purposes:

1. The construction, development, production or testing of weapons or weapon systems of any kind.
2. The construction, development, production or use of plants/installations which include the processing of radioactive/fissionable material.
3. The training of persons to deal with the abovesaid actions.

Listen to your conscience.

'Installer' copyright and license notice:

Installer 1.24 and Installer project icon © Copyright 1991-93 Commodore-Amiga, Inc. All Rights Reserved. Reproduced and distributed under license from Commodore.

INSTALLER SOFTWARE IS PROVIDED "AS-IS" AND SUBJECT TO CHANGE; NO WARRANTIES ARE MADE. ALL USE IS AT YOUR OWN RISK. NO LIABILITY OR RESPONSIBILITY IS ASSUMED.

1 Introduction

'term' is a telecommuncations program designed for use with any Commodore-Amiga computer running Kickstart 2.04 or higher. Its features include

- Fast built-in VT-220 terminal emulation
- Support for custom terminal emulation modules following the *XEM 2.0* standard
- Operates in any display environment, supports all screen display modes
- Support for file transfer modules following the *XPR* standard
- File- and printer-capturing functions
- Review-buffer support
- Powerful phone book and dialing functions
- *Amiga User Interface Style Guide* conformant user interface
- Online-help (requires AmigaGuide package)
- Built-in *ARexx* interface
- File upload list, which permits selecting the files to be transferred before the upload is started.
- Login script learn mode.
- Built-in keyword/response parser which makes it possible to have 'term' respond to BBS prompts and such with the user name, password, etc. without having to program the ARexx interface.
- Interface for external programs to rendezvous with 'term', taking over serial I/O processing (such as '`HydraCom`').

Although this program is freely-distributable, it is not entirely free. If you like it and use frequently, you are requested to send the author a donation which you suppose will do as a payment for the program. See the chapter registration (see Chapter 2 [Registration], page 3) for details.

Admittedly, 'term' is a fairly large program which consumes quite an amount of memory when running. You need to have at least 2 MBytes of system memory installed if you wish to run the program. Please note that this is the bare minimum configuration. It is recommended that you run 'term' on a faster machine, i.e. plain MC68000-driven Amigas will have trouble running the program.

2 Registration

'term' is made available under the concept of Gift-Ware, which is a variant of Share-Ware. Share-Ware software authors often release 'crippled' versions of their products, i.e. these programs do not support the same functionality as the registered versions you get when sending monetary contributions to the authors. It all comes down to 'pay for the software you are using' in Share-Ware terms. With Gift-Ware registration is different, you are not required to contribute money, but a gift will do. With 'term' you always get a fully functional program, there is no 'crippled' test release which you can try for a limited time and then have to pay for in order to receive the working registered version.

It may seem as if 'term' was free, but this is not the case. Although there is no need to pay the author in order to get a fully functional version of the program you should consider making a contribution. You don't need to feel guilty if you cannot or do not want to give me something in return for the work I have put into 'term'. Show me that it was worth spending so much time listening to users, updating, rewriting and enhancing this program. Your contributions will provide the motivation for me to keep developing the program.

The preferred method of 'payment' would be to order the 'Olsen Collection' CD-ROM. This CD contains almost every program I have written for the Amiga. The CD will be available in August 1994 and can be bought for US$ 35 directly from me or from your local software distributor.

If you don't have any CD-ROM drive or just don't want to buy the CD, here is a list of suggested contributions:

- Infocom games and Infocom hintbooks. Although I already own almost all games published in *The Lost Treasures of Infocom vol I+II* I still collect the originals (not those super-cheap Virgin Software re-releases). With the exception of *A Mind forever voyaging*, *Arthur*, *Hollywood Hijinx*, *Infidel*, *Leather Goddesses of Phobos*, *Nord and Bert couldn't make head or tail of it*, *Plundered Hearts*, *Shogun*, *Spellbreaker*, *Suspect*, *The Lurking Horror*, *Trinity*, *Wishbringer*, *Zork I*, *Zork Zero* (which I already own) I welcome any game in any format – it does not necessarily have to be an Amiga game, Apple II, Kaypro, Atari-XL, C64, etc. will be fine, too.
- The films *Jabberwocky*, *Brazil*, *The Fisher King* or *Time Bandits* on a PAL-VHS video cassette
- A CD by the Beatles (except for *Help*, *Rubber Soul*, *Revolver*, *Sgt. Pepper's Lonely Hearts Club Band*, *The white album*, *Magical Mystery Tour*, *Abbey Road* and *Past Masters, Volume II*), Little Feat (-1989), Weather Report, Paul Simon (1971-1985), Eric Clapton (-1985), Peter Gabriel (1977-1989) or Daniel Lanois
- Raymond Elias Feist's books *Magician: Apprentice* and *Magician: Master*
- A book by Michael Crichton, Bruce Chatwin, Raymond Chandler, Terry Pratchett or Steven Meretzky

- Old CinemaWare games – with the exception of *Defender of the Crown*, *The King of Chicago*, *The Three Stooges*, *Sinbad*, *It came from the desert*, *Lords of the Rising Sun*, *TV-Sports Basketball* and *TV-Sports Football*
- An old game by Activision (*Hacker II*, *Little computer people*, *Portal*, *Shanghai*, *Tass Times in Tonetown*), Electronic Arts (*Marble Madness*, *Return to Atlantis*, *Skyfox*, *The Bard's Tale*) or Epyx (*Temple of Apshai*, *Rogue*, *California Games*)
- An old Telarium/Trillium game for the C64 on 5.25" floppy disk (such as *The Amazon*, *Rendezvous with Rama*, *Fahrenheit 451*).
- An old Lucasfilm game for the C64 on 5.25" floppy disk (such as *Rescue on Fractalus*, *Ballblazer*, *Koronis Rift*, *The Eidolon*).
- Scenery disks for the SubLogic Flight-Simulator II – except for *Western European Tour*, *Japan*, *USA #7*, *USA #9* and *USA #14*.
- CD-ROMs and CD-ROM caddies
- Cash and checks (no credit cards – sorry) are always welcome.

Send your contribution to the following address:

Olaf Barthel
Brabeckstrasse 35
D-30559 Hannover
Federal Republic of Germany

If you wish to be notified when program updates become available or wish to order the next update as it becomes available you must include enough money to cover my expenses, see the chapter entitled orders (see Chapter 4 [Orders], page 7) for more information.

3 Commercial distribution

I don't mind if you make a copy of 'term' for a customer who is looking for a decent terminal program. But if you decide to distribute the program on a larger scale, such as by including a copy with each modem you sell, I want to know about it. The reason is this: by distributing 'term' you take a certain responsability; if your customer has difficulties in installing and configuring the program she/he will most likely ask you for assistance. I want to make sure that if a 'term' user is in bad need of help there is somebody who will be able to provide it. If you wish to become a 'term' distributor and want your name displayed in the 'About' window, contact me for details.

As a general rule, no profit must be made by distributing 'term', i.e. you may charge fees for copying, disks and for providing a printed manual but no money must be charged for the software itself. I consider taking legal actions against anyone who violates this rule. Payments for using the program should be made to the author, not to the provider.

4 Orders

You can order a copy via standard mail by sending a self-addressed envelope, including postage and disks (more on this below) to the following address:

Olaf Barthel
Brabeckstrasse 35
D-30559 Hannover
Federal Republic of Germany

Either include three 3.5" double-density disks or two 3.5" high-density disks. The postage must cover the mailing costs, this should be the equivalent to DM 8,- in your preferred currency for Europe, and DM 16,- for all other countries. All orders are shipped via air mail. If you cannot provide the disks and the envelope, add DM 1,50 per double-density disk or DM 3,- per high-density disk and DM 2,- for a padded envelope.

As of this writing I do not distribute printed copies of the documentation.

Orders must be made by mail only, not by phone or eMail.

Whenever a new release of 'term' becomes available I will try to make it known in the telecommunications networks.

The most current 'term' release will be available through ftp from `ftp.informatik.uni-oldenburg.de` (`134.106.1.9`), look into the '`/pub/amiga/term`' directory. All Aminet sites will also receive copies of the program.

'term' is available for download in a number of BBSes. *Careful please!* For reasons I have always failed to understand certain individuals take pride in patching program version numbers, copying archives to disk and compressing them into .DMS files, attach silly notes to archives or extract single files from archives, just to recompress and release them later. *Hands off these files!* 'term' is always distributed only in complete LhA-archives, never as a single program or as a .DMS-file. In addition to this file lists and signatures generated using the PGP program provide authenticity. If any of these signs are missing chances are that you have found a corrupted copy which most likely was not released by me. My public key can be found in the chapter PGP key (see Chapter 30 [PGP key], page 111).

I will not distribute 'term' via eMail, the program is too large to be mailed and since I am paying both for incoming and outgoing mail it would also be too expensive for me.

Unless requested, no update notifications will be sent via standard mail. I recommend that you include an international reply coupon for each update notification you wish to receive.

The author reserves the right to discontinue development of the ‘term’ program.

5 Letters to the author

It is always nice to receive feedback from users, to hear about critical comments and enhancement request. If you wish to communicate with me I suggest that you try electronic mail channels first. As of this writing there only exists a single Internet address (`olsen@sourcery.han.de`) you can send mail to. I have no Fidonet accounts or such, in fact I don't even visit any BBSes regularly. If you need to put a mail through to me and don't have direct Internet access it is recommended that you use the gateway facilities most nets offer, your local sysop or postmaster will be able to tell you more. Do not send mail larger than about 64K bytes, as it may get caught on the way to me and if it in facts arrives in my mail folder it will have cost me routing fees since I pay both for my incoming and my outgoing mail. Please don't expect me to make long-distance calls to your local BBS, German phone rates are rather steep and I am not that wealthy after all.

If you happen to be registered in a BBS which has no links to any net or if you don't have any email access at all the only alternative to dropping by and paying me a visit in person is to use the standard mail facilities: send a letter, mail a package. Although such mail will usually arrive safely and unharmed there still is a problem: I am slow at responding to 'real-world mail'. If a letter can be answered with a few lines of text chances are good for a snappy answer, but if the topic is a little more complex your letter may go into my mail service stack. To give you an impression how this stack looks like: it's a large pile of unanswered mail sitting on my desk which keeps falling over each time I open the window. Such mail will usually get answered at the end of the semester or when my bad conscience tells me to.

Be sure to mention it if you want me to send you an update to 'term' or the most recent version of the program and include some money to cover my expenses. If you want me to help you or to give advice please keep in mind that even though there is no denying the fact that I have written 'term' I am by no means an expert in telecommuncations matters. So if you need to know which modem brand to choose or which Bulgarian BBSes are worth trying I strongly suggest that you ask someone else.

In any case, feel free to make comments and to ask questions.

6 Known bugs & problems

During beta testing certain software did not work very well with 'term' (nothing serious I would say, but you may have a different opinion). There are also a few problems that would show up at the last minute (and at great expense) when it was too late to find a fix. Last but not least the following list also includes a few 'features'~for which I was unable to find a better solution.

- When running on the Workbench screen 'term' keeps eating processing time, even the mouse no longer moves smoothly.

  It's sad but true: if any other window partly obscures the status line display system performance will suffer greatly. Try to make 'term' the frontmost window or separate the status line display from the main window in order to get rid of this nuisance.
- As of Kickstart 2.04 slider and list displays look rather odd. The '`gadtools.library`' user interface support routines do not handle proportional-spaced fonts very well in v2.04. Try to use a different user interface font, preferably a fixed-width font.
- With '`MagicMenu`' running in the background the checkmark and Amiga symbols overwrite some menu entries.

  '`MagicMenu`' does not notice 'term' scales these symbols to fit the current screen display aspect ratio. Future versions of '`MagicMenu`' may address this problem.
- In the built-in screen mode requester some screen modes show up twice.

  No harm should be done. The built-in screen mode requester is only used under Workbench v2.04, you should upgrade to v2.1 or even to Kickstart 3.x.
- With small terminal window sizes the status line display no longer fits into the window.

  'term' always pays attention to the terminal settings, it does not take the status line width into account. No harm should be done.
- Not all the buttons in the file transfer window do what their labels say.

  The XPR library running the file transfer is responsible for listening to the commands 'term' sends to it when the '`Abort entire transfer`', '`Skip current file`' and '`Abort transfer batch`' buttons are pressed. Not all libraries will pay attention to these commands. So far only '`xprkermit.library`' responds to all commands. The '`xprzmodem.library`' included in the 'term' distribution will ignore the '`Abort transfer batch`' command. All other XPR libraries either treat all buttons the same way, i.e. they abort the transfer no matter which button is pressed, or just pay attention to the '`Abort entire transfer`' command.

7 Frequently asked questions

This section was written in order to answer the most frequently asked questions concerning ‘term’. Although I don’t suppose that the users who this section was intended for will really ready it, but it may nevertheless still be quite useful. Each entry in the following list explains why a particular feature works this way or the other and possibly why.

1. While scrolling ‘term’ quietly swallows characters and sometimes whole lines.

   Usually, ‘term’ cannot process incoming serial data while the terminal output processing is taking place. If output processing takes too much time you may end up losing incoming data. There are several way how to approach this problem. You could reduce the number of colours used for screen output, or enable the terminal emulation process via the ‘`Enable emulation process`’ switch in the section entitled terminal panel (see Section 19.5 [Terminal panel], page 53).
2. Right after a connection is established the modem hangs up the line.

   Following your configuration options ‘term’ makes sure that the modem is set up correctly prior to making a call. Does the modem drop the line right after the `CONNECT` message is received your modem or the remote modem may reponsible. Not unheard of are modems which due to firmware trouble fail to negotiate correctly with the remote modem over the transmission protocol to be used. With error correction enabled such modems would drop the line right after establishing the connection. Turning off the error correction mode would correct this problem. ‘term’ cannot do anything about the modem behaviour, it is up to you to find the correct setup.
3. Trying to run the modem at higher baud rates causes the modem to ignore commands, it does not even echo characters back.

   A number of modems respond only to a fixed set of baud rates. While for example 9,600 bps are fine there is no response at 14,400 bps. I suggest that you try all available baud rates ‘term’ supports until one is found to fit. Do not overdo it however, try to keep the baud rate below 57,600 bps.
4. During file transfers error #6 is reported over and over again.

   Data transfer to and from the serial hardware is a time requires quick responses by the handler routines, this is why interrupt processing is used for such time-critical jobs. Unfortunately, these interrupts cannot always be serviced as fast as possible. Read and write accesses to disk may temporarily slow interrupt processing down or even disable it in short intervals. If the send/receive buffers keep flowing over I suggest that you use a smaller file buffer size, so data gets written to disk in smaller chunks, making the intervals in which interrupt processing may be affected smaller.
5. I have saved the phonebook and the configuration files to disk and an older ‘term’ release reports that it cannot read them.

   ‘term’ stores version information with the configuration files it saves. Older ‘term’ releases will refuse to read files created by newer releases. Newer releases will almost always read configuration files by older ‘term’ releases.

6. I upgraded from an older ‘term’ release, but the program refuses to read my configuration and phonebook files.

   The phonebook and configuration file format was changed and greatly enhanced with the introduction of ‘term’ 3.1. Older files need to be converted to the new format, this is what the ‘`UpdateConfig`’ program is for that should be included in the ‘term’ distribution. The conversion is easy, just enter `UpdateConfig <old file name> <new file name>`, the program automatically determines whether it is reading a configuration or a phonebook file.

   Caution: the conversion program cannot read encrypted phonebook files, so they should be saved in unencrypted form first.

7. In some BBSes ANSI graphics and text output starts at the wrong screen position, especially after the screen contents are erased.

   The so-called BBS-ANSI terminal command set treats the ‘clear screen command’ different from the VT-100 specs, i.e. it expects the cursor to be reset to the home position. You can enable this feature using the `CLS' resets cursor position` switch which can be found in the emulation panel (see Section 19.6 [Emulation panel], page 55).

8. The text buffer window does not show any special characters, such as accented characters, but only dots (‘.’).

   The text buffer window cannot display characters which fall into the range between code #127 and code #159. In order to show any text at all these codes get replaced by the dot character.

9. When I upgraded from an older program release (1.6 - 2.3) to the new ‘term’ release the program would no longer find all its configuration files.

   Some of the configuration files were renamed in v2.4, and some were moved to different directories. While the files used to be present in ‘`ENVARC:term`’ ‘term’ now looks for them in ‘`TERM:config`’. If ‘term’ finds no ‘`TERM:`’ assignments, it will create one. If no ‘`TERM:config`’ directory can be found, it will also be created. The configuration files have been changed as follows:

   ‘`Preferences.term`’
   ‘`term_preferences.iff`’

   New name is now ‘`term.prefs`’

   ‘`Phonebook.term`’
   ‘`term_phonebook.iff`’

   New name is now ‘`phonebook.prefs`’

   ‘`Hotkeys.term`’
   ‘`term_hotkeys.iff`’

   New name is now ‘`hotkeys.prefs`’

   ‘`Speech.term`’
   ‘`term_speech.iff`’

   New name is now ‘`speech.prefs`’

'Macros.term'
'term_macros.iff'
'macros.prefs'
New name is now 'functionkeys.prefs'

'Fast!Macros.term'
'term_fastmacros.iff'
New name is now 'fastmacros.prefs'

In order to use 'term' and 'termcap' you need to rename the TERMPATH variable to TERMCONFIGPATH. Look into the 'ENVARC:' directory, rename the file and reboot.

Caution: it is not sufficient just to rename the file names, you will also have to take care of the phonebook entries.

10. I have added several phone numbers to the quick dial menu, but 'term' does not make them all available.

    'term' has room for only up to 50 quick dial entries, any further entries will be ignore.

11. When transferring files between computers connected via null-modem cables the transfer always aborts immediately complaining that the DTR or the carrier signal was lost.

    During a null-modem transfer there is no carrier or DTR signals present, only modems and such offer such features. Switch the handshaking mode to 'None' and turn off the 'Check carrier' feature in the serial panel (see Section 19.1 [Serial panel], page 45).

12. The dialer skips two entries at once when pressing the 'Skip' button.

    Pressing the 'Skip' button stops dialing, most modems will respond to this with an OK message. Some modems will however send NO CARRIER, which 'term' interpretes the same way as if BUSY had been sent. If this is what your modem does, change the ''NO CARRIER' = 'BUSY'' switch in the Serial panel (see Section 19.1 [Serial panel], page 45).

13. During ZModem file transfers using a fast modem the 'CPS' display first lists an incredibly high value, then drops sharply, followed by lots and lots of transfer errors occuring.

    This is effect is mostly seen if the 'Handshaking' mode is not set to 'RTS/CTS' when using a fast modem. The trouble is caused by data getting sent while the modem has already stopped accepting new data. It flags this states using the RTS/CTS line.

14. None of my ARexx scripts works any more.

    In v3.1 the 'term' ARexx interface was rewritten from scratch. If you wish to retain your old ARexx scripts they will have to be rewritten.

15. During file transfers errors show up while data is saved to and read from a hard disk drive.

    Some hard disk drive controllers temporarily disable interrupt processing while accesses take place. In such cases I recommend to download and upload from the ram disk or to upgrade the hard disk driver.

16. Even though the speech support feature is enabled, 'term' does not speak a single word.

    With the introduction of Workbench 2.1 Commodore ceased to support the speech synthesizer. If you don't have 'narrator.device' and 'translator.library' installed the speech support will not work.

17. When downloading files they don't end up in the right directory.

    The file transfer protocol usually has it's own opinion on where to place files it receives. This behaviour can be changed by editing the 'term' settings, open the miscellaneous panel (see Section 19.10 [Miscellaneous panel], page 62) and turn on the '`Override transfer path`' switch. Now you can select the names of the directories to store files received in using the transfer panel (see Section 19.12 [Transfer panel], page 65).

18. Sometimes 'term' stops processing input and output and just beeps when a key is hit.

    Look at the status line, if it displays '`Holding`' 'term' has received an xOFF character. Press `Control` + `S` to restart.

19. When I moved my configuration files into a different directory and updated my main configuration the phonebook entries still don't find their settings files.

    'term' uses the environment variable '`TERMPATH`' to locate its configuration files. However, the local phonebook entries may have different search paths set. Check the paths settings to see where they are pointing to.

20. Even though several files are selected for transfer only the very first file is sent.

    Not every file transfer protocol supports batch transfers. There is no way for 'term' to tell whether a protocol supports batch transfers, please consult your protocol documentation for more information.

21. Even though everything is set up correctly no ARexx scripts are executed.

    In order for ARexx to work the '`RexxMast`' program needs to be running. Usually, this program is located in the '`System`' drawer of your system partition. Drag it into the '`WBStartup`' folder in order to use it at system startup time. Also make sure ARexx knows where to find your ARexx scripts. Either give a complete path name or copy your file into the '`REXX:`' drawer. Do not rely upon an `Assign REXX: <drawer name> add` call in your '`S:User-Startup`' file to work, as for this writing ARexx does not support multi-volume assignments.

22. Even though the auto-download feature of the current file transfer protocol is enabled no auto transfer takes place while an ARexx script is running.

    Serial I/O processing only takes place if the main program takes care of terminal output.

23. Running 'term' twice from shell does not cause two 'term' processes to be started, instead only the first program is reactivated.

    By default starting 'term' more than once only brings an already running 'term' process to the front. Use the `NEW` keyword to suppress this feature.

(To be continued)

8 Reporting bugs

'term' is a rather complex program which is difficult to maintain – especially since there is only one person to take care of it: me. Although one tries to write correct, bug-free software, one cannot always achieve this goal. Tough – but that's life. It rains when you leave the umbrella at home. Toast falls buttered-side down. The phone rings while you are in the bath.

In case you come upon one of those nasty features which even the author was unable to track down and remove, follow these steps:

1. Keep calm. Shouting, cursing, crushing disks may help to cool your temper, but it will not help anybody (the least yourself!) if it results in a sudden cardiac infarction: you will have problems in reporting the problem.
2. Read the documentation! The bug you may want to report may be a deliberate feature.
3. Repeat previous step as often as possible. Yes, really, do so. Read the documentation. You will be glad you did.
4. Describe your problem elaborately. A comment like 'things fall down when dropped' may have inspired Sir Isaac Newton, but a similarly laconic comment 'downloads do not work' will most certainly fail to give any useful hints how to approach the problem. In case you encounter a problem with the built-in terminal emulation, try to make a verbatim file capture (i.e. turn off the `Capture filter`) of the session in which the offending codes were used and send it to me.
5. If you wish to report a bug in the ARexx interface include a sample ARexx script to produce the bug.
6. Do not forget to write it down! There is a difference between noticing a bug and reporting it (honestly!). Do not suppose that a bug will be fixed in a future program revision or rely on anybody else to report it: do it yourself. Send a letter to the author, preferably per electronic mailing services. The addresses are given at the end of this document.
7. State your system and program configuration. It helps a lot to know on which machine the program caused problems. Please include information such as memory expansion size, Amiga model (A500+, A600, A1200, A3000, A4000, etc.), graphics hardware (ECS, AGA, etc.), CPU type (MC68000, MC68020, MC68030, MC68040, etc.).

If you can please run the standard Commodore debugging tools ('`Enforcer`', '`Sushi`', '`tnt`', '`MungWall`', '`SegTracker`') in the background and capture the output. If you include '`Enforcer`' hit reports make sure that you have '`SegTracker`' running in the background or the '`Enforcer`' output will be worthless to me.

Please direct reports of problems with the file transfer and terminal emulation libraries to the respective authors, I am not responsible for maintaining the support libraries.

Error reports concerning the ARexx interface should include a sample script to illustrate what is going wrong.

9 Background

This program is a product of anger and despair; I was unable to find a telecommunications program to suit my personal needs – neither in the commercial area, nor in the public-domain.

Most programs had a lot of extras but lacked other more important, perhaps more sensible features (just to take an example: in revision 2.20c and after four years of constant development '`Handshake`' still fails to use the current keymap settings and also strips the high order bit when receiving text – sorry Eric, that's why I never registered!).

I have hesitated for a long time before starting my first attempt at writing my very own telecommunications program. When Kickstart 2.x was about to become widely available I took the opportunity to create 'term' always trying to use the new OS routines wherever possible. While this started to be quite a difficult task it also was a lot of fun (imagine Columbus wrecking his fleet four times on his journey to the West Indies due to unexpected leakages in all vessels and sudden changes in the ships' sailing manuals – that's how I felt!).

As far as computer-telecommunications are concerned, Germany appears to be a developing country. This is partly due to the Deutsche Bundespost, the federal mail/phone company whose telecommunications monopoly used to be protected by federal law. Until 1989 you would risk a heavy penalty if using a non-registered modem or telephone instead of the Bundespost-supplied hardware. So, if you have any complaints or miss a few extremely important features in 'term', don't boo and hiss, I am not as long in the telecomm business as you are (I have yet seen only a single '`DEC VT-101`' from afar!). Tell me what you need and I will try to add it in the next revision.

This project was started at December 24 1990 and completed by January 25 1991.

10 Future

I spent almost five years of my life programming and updating ‘term’. Support for new operating system features was added as soon as Commodore lifted the veil. ‘term’ grew both in functionality and size, it was difficult to make plans for the path development would follow. While I have no specific idea which turn development may take in the future there are a few things I definitely do not want to add:

- FAX support

  There are plenty of good commercial FAX solutions available for the Amiga. I lack both the time and the motivation to add FAX support to ‘term’. However, future ‘term’ releases may provide interfaces to FAX programs.
- More terminal emulations

  ‘term’ offers support for the XEM standard, making it possible to easily add external terminal emulation libraries. If you are looking for a RIP emulation or a certain Data General terminal emulation, look for external emulation library.
- Script language

  ‘term’ supports ARexx, I don’t see any reason why I should add another script language.
- More file transfer protocols

  Except for the ASCII transfer routines ‘term’ offers no built-in file transfer protocols. This is what the XPR interface is for, external file transfer libraries provide all the file transfer services. If you need a file transfer protocol, look for an external transfer protocol library.
- Reduced functionality

  It was ‘tough’ enough to implement all the features, bits & pieces that make up ‘term’. Honestly, I don’t have the heart to cut back features. It would not have been the first attempt to make ‘term’ smaller either; there have been numerous attempts to create programs to copy the functionality of ‘term’ or to build a smaller, scaled down terminal program based upon the ‘term’ source code. As far as I know none of these attempts was successful.

It is difficult to judge how the next ‘term’ release will look like. Also, v4.0 may be the last ‘term’ release, but then again maybe not. The future of ‘term’ depends on its users, and this includes you.

11 Acknowledgements

My thanks go to the following people for their invaluable help and assistance: Andreas Kirchwitz, Christoph Teuber, Christopher Wichura, Garry Glendown, Germar Morgenthaler, Henning Hucke, Holger Lubitz, Juergen Otte, Marc-Christian Schroer, Marko Kuechmann, Markus Stoll, Martin Berndt, Martin Taillefer, Matthias Zepf, Michael Vaeth, Michael Wolfgang Hohmann, Oliver Wagner, Peter Fischer, Ralf Thanner, Ralph Schmidt, Roby Leemann & AUGS, Stefan Becker, Thorsten Seidel, Till 'Dill-Prince' Prinzler, Udo Wolt, Ueli Kaufmann, Veith Schoergenhummer, Volker Ulle & the Aquila Sysop Team and to all those who supplied libraries & control sequence tables.

Special thanks go to John Burton of Papua New Guinea who revised and rewrote certain parts of the program, in particular the terminal emulation routines, Leo Schwab who discovered means to use interleaved screen bitmaps in a system-integrated manner and to Nicola Salmoria whose invaluable assistance helped to reduce the incredible number of bugs lurking in the source code.

Additional user interface wizardry and advice by Martin Taillefer. The file transfer section of this manual was rewritten to incorporate several suggestions made by Mike Safer.

The XPR libraries were created by Terence Finney ('`bplus`'), Kenneth Osterberg & Markus Pietz ('`jmodem`'), Marco Papa & Stephen Walton ('`kermit`'), Jack Rouse ('`quickb`'), Marc Boucher ('`xmodem`'), Ueli Kaufmann ('`ascii`', '`ymodem`' & '`vms`') and Rick Huebner & William M. Perkins ('`zmodem`').

The XPR standard was created by Willy Langeveld, the quicksort routine ('`QuickSort.asm`') was written by David Jones.

The current implementation of the external terminal emulation library interface was developed by Ueli Kaufmann, who also wrote the external terminal emulation libraries supplied with 'term'. Without the invaluable help of Martin Berndt the library interface would probably not be working at all.

Since time did not permit me to translate the full original German documentation into English, I had asked the Z-Net Amiga community for help. As a result this document was translated by three different authors (in order of translation): me, Marc Schroer and Henning Hucke. Garry Glendown took care of the original termRexx documentation – thanks to all of you!

The beta tester group, consisting of Abdelkader Benbachir, Alfredo Rojas, Andreas M. Kirchwitz, Bernd Ernesti, Bob Maple, Bodo Thevissen, Chris Hanson, Chris Mattingly, Christoph Guelicher, Christopher G. Newby, Christian Hechelmann, Dabe Murphy, Dean S. Pemberton, Eric W. Sommer, Florian Hinzmann,

Frank Duerring, Gary B. Standen, Gregory A. Chance, Holger Heinrich, Holger Lubitz, Jason C. Leach, Jay Grizzard, Joel E. Swan, Jonathan Tew, Juergen Zeschky, Julian Matthew, Kai Iske, Kay Gehrke, Keith A Stewart, Keith Christopher, Kenneth Fribert, Klaus Duerr, Leon D. Shaner, Mark Constable, Martin Berndt, Matthias Scheler, Matti Rintala, Michael Zielesny, Olaf Peters, Ottmar Roehrig, Peer Hasselmeyer, Peter L. Banville Jr., Piotr Kaminski, Robert L. Shady, Russell John LeBar, Sebastian Delmont, Stefan Becker, Stefan Gybas, Stefan Hudson, Steve Corder, Sven Reger, Tony Kirkland, William Michael Mushkin and Yves Perrenoud, took care of testing the program – thank you very, very much!

12 Source code

Since there are still only very few well-documented examples (or general programming examples) for Kickstart 2.x and Kickstart 3.x I have decided to include the full ‘C’ source code with the ‘term’ distribution.

The source code is *not* intended for commercial use. If you are about to include portions in commercial programs you will need to ask me for permission. Still you may use parts of the source code for non-commercial software development without my consent.

I sincerely hope that the release of the full ‘term’ source code will give Kickstart 2.x a better start (I’ve overcome quite a lot of obstacles) so that more programs to use the new OS features will be available soon.

13 Documentation and online help

'term' comes bundled with a number of documentation files, these are:

'term.doc'
: Human-readable english program documentation in standard ASCII format.

'term.guide'
: English program documentation in AmigaGuide format suitable to submit to 'AmigaGuide' or 'MultiView'.

'termRexx.doc'
: Human-readable english 'term' ARexx interface documentation in standard ASCII format. This file describes all the ARexx host commands 'term' supports and also gives a brief introduction how to use them.

'termRexx.guide'
: English ARexx interface documentation in AmigaGuide format suitable to submit to 'AmigaGuide' or 'MultiView'.

'xprascii.doc ... xprzmodem.doc'
: Human-readable documentation on the XPR transfer libraries supplied with 'term'.

In addition to the pure ASCII files documentation files are available in TeX-DVI format and Postscript. As space requirements do not permit to include them on the distribution disks they are only available directly from the author.

'term.dvi'
: English program documentation in a format suitable for printing using a utility to print TeX-DVI-output files, such as supplied with the packages AmigaTeX or PasTeX.

'term.ps'
: English program documentation in Postscript format. This file was generated from 'term.dvi' using the 'dvips' utility.

'termRexx.dvi'
: English ARexx interface documentation in a format suitable to printing using a utility to print TeX-DVI-output files, such as supplied with the packages AmigaTeX or PasTeX.

'termRexx.ps'
: English ARexx interface documentation in Postscript format. This file was generated from 'termRexx.dvi' using the 'dvips' utility.

In order to take advantage of the online-help feature, AmigaGuide and the file 'term.guide' are required. Copy the file to the drawer the 'term' main program is located in and configure the ''term' `help text file`' settings (see Section 19.11 [Path panel], page 64) to point to `PROGDIR:term.guide`. Once this has been done, pressing the `Help` key in any window to support online help will bring up a help window. Please note that context sensitive help requires AmigaGuide v39 or higher.

14 Foreign language support

'term' supports foreign language text catalog tables as introduced with Workbench 2.1 '`locale.library`'. As of this writing there is only a german translation of the program text available. If you wish to create your own national translation table you should consult the program source code file '`term-blank.ct`' which is a blank translation table. Before actually starting to fill in the translation table you should contact me first; there may already be someone preparing a translation. However, you should keep in mind that once you have created a foreign language translation of the program text I may ask you to update your translation for a future program release.

Once you are finished with the translation table, send it to me on disk or via eMail, *don't compile the catalog on your own!*.

So far, I have received a Dutch, French and Norwegian translations for v3.4. I hope to get them updated for v4.0 as soon as time permits.

15 Workbench and Shell

The program can be started both from Workbench and from Shell. Kickstart 2.04 (revision 37.175) and Workbench 2.04 (revision 37.67) are required to run 'term'.

The 'behaviour' of 'term' can be changed by adding tool type entries to the corresponding Workbench icon or by specifying additional command line parameters when running the program from the Shell. Supported keywords are:

'WINDOW'
: The console window specifier to be used when opening terminal output windows (this will override the default settings). The default is `CON:0/11//100/term Output Window/CLOSE/SCREEN %s`. The `%s` will be replaced by the name of the public screen 'term' uses.

'PUBSCREEN'
: The name of a public screen to open the 'term' window on. In case the public screen happens to be unavailable, 'term' will fall back to the Workbench screen. Note that 'term' assumes that the main window is to be opened on a public screen rather than on a custom screen if this option is in effect, regardless how the default settings may be configured.

'STARTUP'
: The name of an ARexx script file to be run on program startup.

'PORTNAME'
: The ARexx host port name 'term' is to use instead of the built-in default name. The port name will be translated to upper case characters as required by the ARexx host port naming convention. The resulting name must be unique or 'term' will fall back to its built-in default name.

'SETTINGS'
: This keyword determines where to read the default configuration file from. In order to read it from '`Work:term/config`', one would use `term Settings Work:term/config` from Shell or add the tooltype entry `SETTINGS=Work:term/config` from Workbench. This argument does not necessarily need give the name of a path to search, but can also specify the name of the configuration file to be used.

'UNIT'
: Similar to the '`DEVICE`' keyword the '`UNIT`' keywords affects the serial driver settings. It determines which serial driver unit is to be used instead of the one specified in the default configuration file. In order to use unit number 4 one would use `term Unit 4` from Shell or add the tooltype entry `UNIT=4` from Workbench.

'DEVICE'
: In order to use a different serial device driver than the one specified in the default configuration file, use this keyword. To use `duart.device` one would use `term Device`

`duart.device` from Shell or add the tooltype entry `DEVICE=duart.device` from Workbench.

‘`QUIET`’
: If this parameter is present, the program will not start opening a display but rather put an icon on the Workbench backdrop, waiting to be invoked. A double-click will bring it to life. This parameter will be ignored in case the `STARTUP`-Parameter is used along with it.

‘`SYNC (Shell only)`’
: If called from Shell ‘term’ will detach itself immediately allowing the Shell window to be closed afterwards. This effect can be avoided if `SYNC` is entered in the command line.

‘`NEW (Shell only)`’
: Usually, running ‘term’ twice will cause the screen of the other program to be popped to the front instead of creating a second ‘term’ process. To avoid this effect, enter `NEW` as a calling parameter. If called from Workbench, each program will run as a separate process.

‘`BEHIND`’
: This option will cause ‘term’ to open its screen behind all other screens and not to activate its window.

16 User interface notes

To operate a gadget, press the key corresponding to the letter highlighted in the gadget label. Suppose a slider is labeled `Baud rate` with the letter `r` of the word `rate` underlined; in order to increase the slider value one had to hit the key labeled `R`, to decrease the value one is to hold down either `Shift` key while pressing the key `R`.

With some requesters and windows text gadgets will be auto-activated. Pressing the `Return` key will cycle through all the available text gadgets, holding down either `Shift` key will break the cycle.

Windows in which only a single scrolling list is present cursor keys may be used to scroll the contents.

The `Return` and `Escape` keys are respected by most windows. A button surrounded by a recessed box represents the default choice in a requester, pressing the `Return` key will select it. The `Escape` key always selects the 'abort' or 'cancel' button usually to be found in the lower right corner of a window. Most windows to feature a close gadget in the upper left corner can be closed by pressing the `Escape` key.

Numeric entry fields accept input several notations, namely hexadecimal (`$..` and `0x..`), octal (`&..`) and binary (`%..`).

The user interface support library tries to make all control panel windows fit on the screen. If the first attempt fails a different font is used. If this did not help either a new screen will be opened for the window.

17 Screen

Unless configured to open only a simple window on a public screen the 'term' main screen will alway open to the full text-overscan size (which can be set using the '`Overscan`' tool in the '`Prefs`' drawer).

At the bottom of the screen a small window displays a few basic parameters. These are:

'`Status`' The current program operating status. This includes '`Ready`', '`Holding`' (`Control + S` was pressed), '`Dialing`' (the dialing function is at work), '`Upload`' (data is being sent), '`Download`' (data is being received), '`Breaking`' (a break signal is transmitted across the serial line), '`Hanging up`' (connection is being cancelled), '`Recording`' (a script is being recorded) and '`Rec.line`' (a line of text is being recorded).

'`Buffer`' Indicates whether the text buffer is currently recording incoming text or whether the current text buffer contents are frozen.

'`Protocol`'
The currently set data transfer protocol.

'`Emulation`'
The currently active terminal emulation mode.

'`Baud rate`'
The data transfer rate in bits per second.

'`Parameters`'
The current serial parameters (Data bits-Parity-Stop bits).

'`Time`' The current time of day.

'`Online`' The time elapsed after a connection was successfully established. This counter will be stopped as soon as the connection is cancelled (e.g. by hanging up) and is reset to 00:00:00 as soon as a new connection is made.

Every five seconds the time display will change to a calling fee display and the other way round.

The screen itself is opened as a public screen (called `TERM`) which is available to other programs for their purposes. If more than one 'term' process is running, the public screen name will change according to the number of the program (i.e. the first 'term' to be started will call the scren `TERM`, the second one will call it `TERM.1`, the third one `TERM.2`, etc.). The screen title bar will also display the name of the public screen.

18 Menus

For each requester and input window there exists a set of menu items to execute the commands associated with the buttons, dials and gauges in the requester/window. Press the right mouse button to have a look at the commands and their shortcuts.

The following text is to describe the menu items available in the 'term' main menu.

18.1 Project

'`Save screen as Picture/Text`'
: This menu serves to save the current terminal window contents either as plain ASCII text file or as a picture file.

'`Print Screen/Clipboard`'
: These menu entries are to output text on the printer. You can either print the contents of the main screen or the contents of the clipboard.

'`Capture to File/Printer`'
: Selecting one of these menu entries will toggle capturing incoming text to the printer and/or a file on disk.

'`Iconify`'
: Closes all screens and windows 'term' has currently open, resets and releases the serial driver and puts an icon into the Workbench window. Double-clicking this icon will cause 'term' to wake up and to return the state is was in before iconification took place.

While 'term' is iconified, most incoming ARexx-commands will be queued and the corresponding ARexx scripts will appear to 'hang'. In order to reactivate the program either double-click on the program icon or send the ARexx command `ACTIVATE`*. As soon as 'term' is 'awake' again pending commands will be processed again*

'`About`' Shows some information on the program.

'`Quit`' Terminates the program, hold down a `Shift` key to quit immediately, otherwise you will be prompted to confirm your decision.

After selecting the '`Quit`' menu item the program may take a few seconds to exit since the contents of the text buffer (see Chapter 23 [Text buffer], page 97) are freed first.

18.2 Edit

‘Copy’
: In order to transfer any currently marked screen text to the clipboard buffer, select this menu item. Text can be marked by double-clicking the select button while the mouse is over a word or by clicking the select button and dragging the mouse. Holding down either shift key will append the selected text to the current clipboard contents. In any other case the new text will replace the previous contents.

‘Paste’
: Pastes the contents of the clipboard at the current cursor position provided that the clipboard contains text data. Hold down either `Shift` key to have ‘term’ include the ‘`Paste prefix`’ and the ‘`Paste suffix`’ (see Section 19.7 [Clipboard panel], page 58) along with the clipboard contents.

‘Clear’
: Any currently marked text will be released as soon as any rendering operations are to be executed in the main window. To release marked text manually, select this menu item.

18.3 Commands

‘Execute AmigaDOS command’
: Enter the command you want to execute and its command line arguments here.

‘Execute ARexx command’
: This function calls the ARexx server to execute a script file. If the first input character is a ‘ or " the input will be considered as a small program in a single line. Note that this function will not be available if the ARexx server isn’t running.

  The ARexx command set supported by ‘term’ is described in the ‘term’ ARexx interface documentation.

‘Record script’
: Select this menu item to start/stop script recording. More on script recording can be found under script recording (see Chapter 28 [Script recording], page 107).

‘Record line’
: When in script recording mode this menu item will cause ‘term’ to temporarily switch into full line recording mode, rather than recording only single keystrokes. Pressing the `shift+return` keys has the same effect as calling the ‘`Record line`’ menu item. More on the topic of script recording can be found under script recording (see Chapter 28 [Script recording], page 107).

‘Edit traps...’
: This brings up the trap list editor. The list includes control sequences ‘term’ is to look for in the incoming data stream. When a sequence is found the corresponding command sequence (see Chapter 25 [Command sequences], page 101) is executed.

The window contains the usual list management tools (edit, add, remove, clear and move). Please note that the order of processing occurs top-down, i.e. of two entries with the same control sequence data only the top-most will be evaluated.

The buttons ‘`Load`’ and ‘`Save`’ will let you restore and save the current trap list. The ‘`Load`’ button appends the contents of a trap list file to the current trap list, so take care. On startup ‘term’ looks for a configuration file named ‘`trap.prefs`’ and tries to load it. Thus, there can be only one global trap list. It is up to ARexx to load and modify the contents of the trap list, although this editor window provides a graphical interface to the list. Please note that the more list entries ‘term’ has to filter through the input data stream the slower input processing may become, affecting mostly terminal output.

‘`Disable traps`’
: This menu item provides a shortcut to stop trap list processing. On startup ‘term’ will enable trap list processing if the trap list loaded from disk is non-empty.

18.4 Phone

‘`Phonebook`’
: The phonebook is one of the most powerful and complex functions of ‘term’ and will be described later in this document (see Section 20.8 [Phonebook], page 82).

‘`Redial`’
: Dialing list entries which the dialing routine was unable to establish a connection to are once again passed to the dialer.

‘`Dial`’
: To dial a single phone number select this menu item. The phone number entered will be passed to the dialing routine.

‘`Send break`’
: Sends a ‘break’ signal across the serial line.

‘`Hang up`’
: Tells the modem to hang up the serial line.

‘`Wait...`’
: Will cause ‘term’ to emit the character sequence `<Blank space><Backspace>` every second in order to fool the remote into believing that terminal input is currently taking place.

‘`Flush receive buffer`’
: Tells the serial driver to drop its input buffer contents and resets the state of the internal serial buffers.

‘`Release serial device`’
: The serial driver is released for other programs to use it. A requester will appear which allows you to reopen the serial driver or to quit ‘term’.

If the serial driver has been released by the ARexx interface and has not been reopened yet, this menu item will do it.

If you are still online, the '`Redial`' and '`Dial`' menu entries will be disabled. In order to make another call, hang up the line first.

18.5 Transfer

'`Upload ASCII file(s)`'
: This is a pure ASCII-file upload. It was added to allow poor BBS programs to receive text files (especially Paragon doesn't seem to like ZModem text file upload).

'`Download ASCII file(s)`'
: This is a pure ASCII-file download. Refer to '`Upload ASCII file(s)`' for features/options of this mode.

'`Upload text file(s)`'
: Sends a file/files to the remote receiver using the current transfer protocol. If possible this command will 'ask' the transfer protocol to transmit the file(s) in text mode (whatever that means) which may include CR/LF substitution and other gimmicks. Consult the library documentation to find out if your favourite transfer library supports text mode.

'`Download text file(s)`'
: Request a file/files using the current transfer protocol. Refer to '`Upload text file(s)`' for features/options of this mode.

'`Edit & upload text file`'
: Invokes the currently selected (see Section 19.11 [Path panel], page 64) text editor on a file to be selected using a file requester. 'term' pays attention to the '`EDITOR`' environment variable and will use the program indicated by it.

: 'term' will block and wait until the editor has returned.

: After the editor has returned, the user will be asked whether the file edited is to be transferred as plain ASCII or via text upload.

'`Upload binary file(s)`'
: Send a file/files to the remote receiver using the current transfer protocol. True batch upload is supported both through wildcard expressions ('`#?.txt`' will send all files whose names end with `.txt`) or through multiple selection. Refer to the documentation of your favourite transfer library to find out if batch file transfer is supported.

'`Download binary file(s)`'
: Receive a file/files using the current transfer protocol. If the protocol does not support batch download you are required to enter the name of the file to be received. Files which remain empty after the transfer are deleted automatically.

The transfer routines open an information window in which a number of transfer parameters are displayed (see Section 20.5 [Transfer progress panel], page 76).

18.6 Buffer

'`Clear buffer`'
: Clears the contents of the text buffer (see Chapter 23 [Text buffer], page 97). Any text will be discarded and *cannot* be recovered. Capture files are not affected by this command.

'`Display buffer`'
: Opens the text buffer screen (see Chapter 23 [Text buffer], page 97).

'`Close buffer`'
: Closes the text buffer screen but does not free the contents.

'`Freeze buffer`'
: This menu entry will, if enabled, stop the text buffer from filling up with new text.

'`Open buffer...`'
: Loads the contents of the text buffer from a file. If there are still text lines in the text buffer a requester will appear giving you the choice to discard the old data, append the new data, or to cancel the action.

'`Save buffer as...`'
: Saves the contents of the text buffer to a file. You will be notified if the file to save to already exists (you may discard the old file, append the new data or cancel the action).

18.7 Terminal

'`Clear screen`'
: Clears the whole 'term' screen and moves the cursor to the top left home position.

'`Reset font`'
: Will change the screen font back to the default screen font.

'`Reset styles`'
: Resets all character style attributes (bold, blinking, inverse video, underlined, etc.) and sets the text colour to the default pen.

'`Reset terminal`'
: Use this menu item to reset the state of the entire terminal emulation.

18.8 Settings

This is where you configure the standard preferences settings. If you select '`New entry`' in the phonebook window (see Section 20.8 [Phonebook], page 82) these standard settings will be used. Put in your most commonly used settings here. Change individual entries in the phonebook as needed.

'`Serial...`'
: See Section 19.1 [Serial panel], page 45

'`Modem...`'
: See Section 19.2 [Modem panel], page 47.

'`Screen...`'
: See Section 19.3 [Screen panel], page 49.

'`Terminal...`'
: See Section 19.5 [Terminal panel], page 53.

'`Emulation...`'
: See Section 19.6 [Emulation panel], page 55.

'`Clipboard...`'
: See Section 19.7 [Clipboard panel], page 58.

'`Capture...`'
: See Section 19.8 [Capture panel], page 59.

'`Commands...`'
: See Section 19.9 [Command panel], page 61.

'`Miscellaneous...`'
: See Section 19.10 [Miscellaneous panel], page 62.

'`Paths...`'
: See Section 19.11 [Path panel], page 64.

'`Transfer protocol...`'
: See Section 19.12 [Transfer panel], page 65.

'`Transfer protocol options...`'
: If the transfer protocol options menu item is selected, a transfer settings panel is displayed. The '`Default transfer library`' in the transfer panel (see Section 19.12 [Transfer panel], page 65) determines the contents of this control panel. If the selected default transfer library does not provide these facilities, a simple text requester will prompt for input. Consult the documentation of the chosen transfer protocol for legal options and the values to which they can be set.

  As an example, see Section 19.13 [XPR options sample], page 67.

'`Translation tables...`'
: See Section 19.14 [Translation panel], page 69.

'`Function keys...`'
: See Section 19.15 [Function key panel], page 71.

'`Fast! macros...`'
: See Section 20.1 [Fast macro panel], page 73.

'`Hotkeys...`'
: See Section 20.2 [Hotkey panel], page 74.

'Speech...'
: See Section 20.3 [Speech panel], page 75.

'Sound...'
: See Section 20.4 [Sound panel], page 75.

'Console window...'
: Whenever an AmigaDOS/ARexx command is executed an output window is opened. This menu item will bring up a requester allowing you to edit the size and position of the window to be opened (consult your AmigaDOS manual for a description of the window position string). If you do not want the window to appear, simply enter 'NIL:'. If the %s formatting parameter is used in the output specification it will be replaced by the name of the public screen 'term' uses.

18.9 Windows

'Status'
: This function opens a window to display program status information. A click on the 'Update' button will cause the information to be brought up to date:

  'Session start'
  : When was the program started?

  'Bytes received'
  : The number of bytes received.

  'Bytes sent'
  : The number of bytes sent.

  'Connection message'
  : The text returned by the modem when a connection was made.

  'BBS name'
  : If available, the name of the BBS the modem is currently connected to.

  'BBS phone number'
  : If available, the phone number of the BBS the modem is currently connected to.

  'BBS comment'
  : If available, the phone book comment corresponding to the BBS the modem is currently connected to.

  'BBS user name'
  : If available, the user name, as available through the phone book, corresponding to the BBS the modem is currently connected to.

  'Screen size'
  : The size of the terminal output window in characters (columns and rows).

'`ARexx port name`'
: If available, the name of the ARexx host 'term' is currently using.

'`Buffer size (bytes)`'
: The size of the text buffer (see Chapter 23 [Text buffer], page 97).

'`Free memory (bytes)`'
: The amount of free system memory.

'`Review`' A review window is opened which basically displays the same text as the text buffer screen. You can scroll through the text displayed both by mouse (see the right hand side scroller) and by cursor keys.

'`Packet`' Opens an input window in which a single line of characters to be transferred across the serial line can be entered. More on this feature is explained below (see Chapter 27 [Packet window], page 105).

'`Fast! macros`'
: Opens or closes the so-called fast macro panel (see Section 20.1 [Fast macro panel], page 73). For more information on fast! macros see Chapter 26 [Fast! macros], page 103.

'`Upload queue`'
: This will open a window featuring a list of files to upload. You can drag icons of files to upload on this window, their names will appear in the list. Dragging icons on the icon labeled '`term Upload queue`' has the same effect. Double-clicking this icon will open the upload list window on the Workbench screen. In order to start an upload either press the '`Binary upload`' or '`Text upload`' button.

  You can replace the icon 'term' uses by copying the icon of your choice into the directory 'term' resides in and naming it '`term_DropIcon`'.

18.10 The quick dialing menu

Note: this menu is available only in case any phone book entries have the special `Quick menu` switch set.

Selecting any item of this menu will cause the corresponding phone number to be dialled. Extended selection and drag-selection are also available when picking services to dial. Please note that you can 'check in' single items by selecting them, but you cannot remove any items from the dial list.

19 Control panels

This where all the control panels employed by 'term' are explained:

19.1 Serial panel

This is where the serial parameters may be changed.

'`Baud rate`'
: Transfer speed in bits per second.

'`Bits/char`'
: Number of bits per transferred character (7 or 8).

'`Parity`'
: Serial parity (none, odd, even, mark or space).

'`Stop bits`'
: Number of stop bits (1 or 2).

'`Handshaking`'
: Serial handshaking mode (RTS/CTS, RTS/CTS with DRS signal check, none).

'`Duplex`'
: Determines whether characters are echoed back to the terminal screen or not (full, half = local echo).

'`Buffer size`'
: The number to be specified here allows to set the serial driver I/O buffer size. *Every number you enter here will result in 'term' and the serial driver allocating twice the buffer size (this is only a word of warning for those among us who prefer buffer sizes of 256K and up).*

'`Break length`'
: Length of the break signal given in microseconds.

'`Strip bit 8`'
: For those guys and gals who need it: if this switch is effect each character received or transmitted by 'term' will have its high-order bit cleared.

'`High-speed mode`'
: Activates a special mode of the serial driver which is to skip a couple of internal parity and stop bits checks resulting in higher data throughput rate. *If this switch is turned on, the serial parameters will be reset to 8 bits per char, no parity and 1 stop bit.* Do not expect dramatic speed increases.

'`Check carrier`'
: 'term' will recognize the `NO CARRIER` message a modem emits when the carrier line signal drops back to low. For maximum safety, 'term' will also check the carrier signal line after

receiving the `NO CARRIER` message if this switch is enabled. This is to make sure that no accidentally appearing text causes confusion.

If this switch is enabled, 'term' will check the carrier signal during file transfers and will abort the transfer if the signal is lost.

Note: Be sure to switch this feature off in case you wish to transfer files using a null-modem cable!

'`Shared access`'
: This switch allows you to run the serial driver in shared access mode. Not all driver types will allow this to happen.

'`xON/xOFF`'
: This switch enables the processing of the flow-control characters `xON` (= `Control` + `S`) and `xOFF` (= `Control` + `Q`).

'`Pass xON/xOFF through`'
: If this switch is enabled, 'term' will pass the `xON`/`xOFF` characters through to the modem rather than swallowing them.

'`Use OwnDevUnit`'
: This switch controls whether Christopher Wichura's '`OwnDevUnit.library`' will be used to schedule access of multiple programs to the currently selected serial device driver.

'`Serial device`'
: The name of the serial driver to be used by 'term'. This is usually '`serial.device`' ('`modem?.device`' for the internal Supra modem, '`sxbios.device`' for ASDG's serial IO card).

'`Device unit number`'
: The device unit number of the serial driver selected above. This is usually left '0' but can also be used to address multiple serial IO ports.

'`Quantum`'
: The number of bytes 'term' tries to read in one 'gulp' and runs through the terminal emulation. The more bytes read, the more time it takes to process them. While the text is processed, mouse and keyboard input are delayed. Choose this value with care.

'`Use`'
: Accept the current settings.

'`Default (phone book only)`'
: Drop the current settings, making a connection to the corresponding phone number will leave the corresponding main configuration entry unchanged.

'`Cancel`'
: Don't use the current settings.

Most modems will not recognize a sudden change in the baud rate. Type `AT<RETURN>` to make the change known.

If '`RTS/CTS (DSR)`' handshaking is selected, 'term' will check to see if the `Data set ready` line happens to be high active. If not, it will default back to software handshaking (`xON`/`xOFF`). The `RTS/CTS` handshaking mode will ignore the presence or absence of the `DSR` signal.

19.2 Modem panel

This is where modem control text and other related parameters are configured.

'`Modem init command`'
: The text to send to the modem after successful program initialization. This text is optional and thus does not need to be present.

  The dialing routine will use the initialization text entered here before dialing a phone number.

'`Modem exit command`'
: The text to be sent to the modem shortly before the program terminates. Just like the modem init text it is optional and does not need to be present.

  The dialing routine will use the initialization text entered here before dialing the next telephone number.

'`Hangup command`'
: The text to be sent to the modem when asked to hang up the line. This text does have to be present if the '`Drop DTR on hangup`' switch is enabled.

'`Dialer abort hangs up`'
: Unless this switch is enabled the dialing procedure will try to abort a modem dial command by sending a plain carriage return character. If the '`Dialer abort hangs up`' switch is enabled the usual modem hang up procedure will be used instead.

'`Dial prefix`'
: The text to be used to prefix each dialing command. This is usually a variant of `ATDP` or `ATDT`.

'`Dial suffix`'
: The text to be used to append to each dialing command. This is usually the carriage-return character `\r`.

'`'No carrier' message`'
: The message the modem emits if the data carrier is lost. The program uses this to determine the length of the connection and to calculate how much the user is to pay for it.

'`'Connect' message`'
: The message the modem emits after detecting a carrier signal. 'term' uses this input to determine successful telephone connection, to reconfigure itself and to start the online timer.

‘`‘No dialtone’ message`’
: The message to be returned by the modem in case it does not encounter any dialing tone on the phone line.

‘`‘Voice’ message`’
: The message ‘term’ is expected to receive if the modem detects a voice call. If in dialing mode, ‘term’ will abort the process. The user will in any case be notified of the event.

‘`‘Ring’ message`’
: The message the modem emits if it receives a call, same effects as with the ‘`Voice message`’.

‘`‘Busy’ message`’
: The message the modem returns if the number which has just been dialed is busy.

‘`‘Ok’ message`’
: The message the modem returns if a command was successfully executed.

‘`‘Error’ message`’
: The message the modem returns if a command was not to be executed successfully.

‘`Redial delay`’
: The time to wait after walking through the whole dialing list without making any successful connection before another attempt is started.

‘`Number of dial retries`’
: The number of times the dialer walks through the dialing list trying to make a successful connection before giving up. Setting this value to ‘unlimited’ will cause the dialer to retry over and over again until it either makes a connection or the dialing procedure is aborted.

‘`Dial timeout`’
: The time to wait for a successful connection during dialing. After this time has elapsed, the dialer will skip to the next entry in the list.

‘`Redial after hanging up`’
: If this switch is in effect, ‘term’ will redial all the phone numbers still in the dialing list as soon as the line is hung up or the carrier signal is lost.

‘`Connect auto-baud`’
: Most modems echo the baud rate upon successful connection. If enabled the baud rate will be read and set for the serial driver.

Use this feature with care as it may have certain negative side-effects (such as the modem dropping the line just after connecting to a BBS)! If you happen to use a modern high-speed modem you will most certainly not need this feature. If in doubt whether you need to enable this feature you should rather disable it!

‘`Drop DTR on hangup`’
: Some modems will track the `data terminal ready` line in order to make sure that the terminal program is listening. Once the line goes back to low potential, these modems will drop the line and hang up. Use this button to make ‘term’ take advantage of this feature.

‘‘NO CARRIER’ = ‘BUSY’’
: If this switch is in effect, the dialing function will treat the modem response code `NO CARRIER` exactly as the `BUSY` response code.

‘Connect limit’
: This gauge is to set a certain period of time to be counted after a connection is made. When elapsed, a control sequence will be executed as to be set using the `Limit macro` settings. If set to `0:00` this function will be disabled.

‘Limit macro’
: A control sequence to be triggered when the time to be set using the `Connect limit` gauge has elapsed. If no text is entered this function will be disabled.

‘Time to connect’
: ‘term’ cannot measure the time to pass between the remote modem picking up the line and the local modem sending the ‘CONNECT’ message. This slider allows to set the length of this interval. Upon connection, it will be added to the total online time.

‘Use’ Use the current settings.

‘Default (phone book only)’
: Drop the current settings, making a connection to the corresponding phone number will leave the corresponding main configuration entry unchanged.

‘Cancel’ Keep original settings.

19.3 Screen panel

‘Display modes’
: All display modes available for the main screen are displayed here.

‘Colour’ This button determines the colour-mode the terminal emulation is going to use. Until now, three modes have been implemented:

‘4 Colours (Amiga)’
: Four colours, optionally blinking.

‘8 Colours (ANSI)’
: Eight colours, optionally blinking.

‘16 Colours (EGA)’
: Sixteen colours, as the EGA-palette, optionally blinking.

‘2 Colours (Monochrome)’
: Monochrome, two colours.

‘Palette’
: These buttons are used to select a colour of the screen palette that is to be changed.

'Red/Green/Blue'
: Use these sliders to modify the red, green and blue components of the currently active colour.

'Use default colours'
: Press this button to have the current colour palette set to the built-in default colours.

'Screen font'
: The name of the current user-interface font is displayed here.

'Select new screen font'
: Press this button to select a new font to be used for the user-interface design.

'Make screen public'
: If this switch is enabled, the 'term' main screen will be made available as a public area other programs may open their windows on.

''Shanghai' windows'
: This item, which is only displayed if 'Make screen public' has been selected, is closely related to the function before. If active, all windows that will normally be displayed on the Workbench screen will open on the 'term' main screen.

'Blinking'
: If selected the VT100-blink option is enabled. If running in eight colour mode, 'term' will only use three bitplanes instead of four for text display, so do not be surprised if display performance drops like a brick.

'Faster layout'
: Activating this switch will slow down display updates and window management. Oh well, not always, there may be a speed increase with some configurations, such as with external emulations activated or when using more than four colours on the screen.

  I suggest to experiment with the effects of this button before actually using it permanently.

'Screen title'
: If this button is enabled, the 'term' screen will contain a draggable title bar, if not, the title bar will be disabled, leaving more space for the terminal output window.

'Use public screen'
: 'term' does not necessarily open a custom screen, this switch will make the main and auxilary windows appear on a named public screen. 'term' will try to adapt colours and text rendering modes to its new environment. It will share the screen palette with other applications which makes it possible to run the built-in terminal emulation in eight or sixteen colours provided that enough shareable screen pens are available (note: Kickstart 3.0 required). 'term' will inherit the text font to be used for user-interface layout from the public screen it will open its window on.

  Note: only the built-in terminal emulation is guaranteed to take advantage of pen-sharing facilities, external emulation libraries will most likely fail to display text correctly!

  Take care when resizing the 'term' main window as a size change will reset the terminal emulation.

`Public screen name`
: The name of the public screen 'term' is to open windows on. 'term' will fall back to the Workbench screen if no proper name is given (i.e. no name is entered) or the desired screen is unavailable.

`Status line`
: This switch allows to disable the status line display or to change between two alternative status line displays:

  `Disabled`
  : No status line is displayed.

  `Standard`
  : The standard two status lines are displayed.

  `Compact`
  : A very condensed version of the status line is displayed, only the data is shown but no captions. The data is displayed in the following order:

    1. Status
    2. Terminal type
    3. Transfer protocol
    4. Baud rate
    5. Serial parameters
    6. Time of day
    7. Online time

`Online display`
: This switch determines what type of information is to be displayed in the bottom right corner of the status display:

  `Online time`
  : The time online

  `Online cost`
  : The amount of money to be paid for the connection

  `Time & cost`
  : Both time and money, the display will toggle between both of them every five seconds.

`Use standard pens`
: The user interface look is determined by the choice of on-screen rendering pens, i.e. which colour to use for highlighted text, active windows, inactive windows, etc. If this button is enabled the screen will be opened using a predefined standard set of drawing pens.

`Edit pens...`
: Pressing this button will open the pen panel (see Section 19.4 [Pen panel], page 52) which permits editing the drawing pens to use for this colour mode.

'Window border'
: The main window will be opened on a custom screen, featuring a drag bar and depth gadgets.

'Separate status window'
: A separate window will be opened for the status display window.

'Use' Use the current settings.

'Default (phone book only)'
: Drop the current settings; making a connection to the corresponding phone number will leave the corresponding main configuration entry unchanged.

'Cancel' Keep the old settings.

19.4 Pen panel

This control panel is available under Kickstart 3.0 and above only. Its functionality is not supported under previous operating system releases!

The user interface look is determined by the choice of on-screen rendering pens, i.e. which colour to use for highlighted text, active windows, inactive windows, etc. Changing the screen colour palette also affects the look of the user interface, this control panel permits to compensate for such changes: even if you reverse the terminal colours you can still retain the correct user interface look by adapting the pens. The controls available are similar to the Workbench Preferences editor program 'Palette':

'Background'
: The screen background colour to use, by default this is colour 0.

'Text' The default colour to render common text, such as in control panel labels.

'Important text'
: The colour to draw important text in, this should put a certain emphasis on the text rendered so the colour should be different from the 'Text' colour.

'Bright edges'
: The colour to use to render the bright edges of windows.

'Dark edges'
: The colour to use to render the dark edges of windows.

'Active window title bars'
: The colour to mark the currently active window and selected buttons and list entries.

'Active window titles'
: The colour to use when printing text over active window title bars, selected buttons and list entries.

'Menu background'
: The colour to render pull-down menus in.

'Menu text'
: The colour to use when printing the menu text.

'Use standard pens'
: Reset the current pen selection to default values.

'Use' Keep the current settings.

'Cancel' Return to previous settings.

19.5 Terminal panel

These settings control the basic behaviour of the terminal emulation 'term' uses. More specific emulation options are available in the emulation window (see Section 19.6 [Emulation panel], page 55).

'Bell' This is where you select the action(s) 'term' is to take whenever a 'bell' character turns up in the data stream:

- 'Visual' The screen will flash.
- 'Audible' An audible signal will be generated.
- 'Visual & audible' A combination of both effects.
- 'Ignore' Nothing will happen.
- 'System default' The system beep routines will be used.

'Alert' 'term' notifies the user of certain events, such as a connection being established or a file transfer action which has just been finished. This switch allows you to select the type of notification:

- 'Bell' A bell signal will be given.
- 'Screen' The 'term' screen will be brought to the front.
- 'Bell/Screen' A combination of the two actions above.
- 'None' Nothing will happen.

'Emulation'
: At this point the user may select the terminal emulation. First there is a solid VT102/VT220/ANSI emulation, then there is a so-called 'Atomic' emulation which does only support the very basic text output operations (CR, LF, Backspace, etc.), all special control sequences as supported

by the VT102 and ANSI specs are filtered out, a '`TTY`'-emulation which displays all data it cannot handle or interpret in raw control codes on the screen (great for debugging).

In addition to the built-in terminal emulation, 'term' also supports external terminal emulations. To select one, set the terminal emulation type to '`External`' and enter the name of the emulation in the text gadget below. In case anything goes wrong loading or initializing the terminal emulation library, 'term' will switch back to the built-in '`ANSI/VT`' emulation.

If in hex-mode the program will display all incoming characters in a special notation. The left hand screen side will show the character code in hexadecimal notation, the right hand side will - if possible - display the character as a glyph.

'`Emulation name`'
: The name of an external terminal emulation library to be used by 'term' instead of the built-in emulation code.

'`Font`'
: Here the type of the font to be used for text display in the terminal window can be selected:

  '`Standard`'
  : The standard text font selected under the '`Text font`' settings in this control panel.

  '`IBM-PC style`'
  : A font similar to the IBM-PC text font will be used. No matter how the translation tables (see Section 19.14 [Translation panel], page 69) are configured, outgoing Amiga characters are translated into PC character values.

  '`IBM-PC style (raw)`'
  : This selection has very much the same effect as '`IBM-PC style`' but no character translation is performed.

'`Text font`'
: The name of the standard or default terminal text display font. Please note that this font cannot be proportional-spaced.

'`IBM PC font`'
: The name of the font to use if the terminal is in IBM PC mode. Please note that this font cannot be proportional-spaced.

'`Send CR`'
: This button determines the sequence that is sent to the remote if a carriage-return character is to be transmitted.

  '`As CR`'
  : The carriage return character is sent without any change.

  '`As CR+LF`'
  : A line feed character is added.

  '`Ignore`'
  : The carriage return character is suppressed.

'`Send LF`'
: This button determines the sequence that is sent to the remote if a line feed charater is to be sent.

'As LF' The line feed character is sent without any change.

'As LF+CR'
A carriage return character is added.

'Ignore' The line feed character is suppressed.

'Receive CR'
'Receive LF'
These two buttons have largely the same effect as the `Send CR/LF` buttons, they are different in that they affect the incoming data rather than the data transmitted.

'Columns'
The number of columns to use for the terminal window. The minimum value is 20 columns, the maximum value is defined by the actual screen size.

'Lines' The number of lines to use for the terminal window. The minimum value is 20 lines, the maximum value is defined by the actual screen size.

'Keymap file'
If your installation requires that 'term' is to use a custom keymap layout, enter the keymap file name here.

At the time of this writing the program will not support custom keymap layouts with the packet window (see Chapter 27 [Packet window], page 105) due to operating system limitations.

'Enable emulation process'
If you are bold and daring you can have an external process handle the terminal text output, just turn on this switch. Please note that the external process will consume additional memory and text throughput speed is likely to suffer with fragmented memory. On the other hand the emulation process will relieve the main program of the tedious task of having to process the incoming data which. This helps the main program to keep up with the incoming data stream and makes it less likely that incoming text is lost.

'Use' Use the current settings.

'Default (phone book only)'
Drop the current settings, making a connection to the corresponding phone number will leave the corresponding main configuration entry unchanged.

'Cancel' Keep the old settings.

19.6 Emulation panel

The buttons found here allow you to change the internal parameters of the selected emulation.

The following text only describes the parameters to change when using the built-in terminal emulation; if an external terminal emulation is in effect, this menu will conjure up the corresponding parameters menu as supported by the external terminal emulation module.

'`Cursor keys`'
: If this mode is active the cursor keys transmit a control sequence instead of a cursor move event. If the remote computer needs these sequences, this mode - applications mode - can be activated with this button.

'`Numeric keypad`'
: If this mode is activated the keys of the numeric keypad transmit a control sequence instead of a number. If the remote computer needs these sequences, this mode - applications mode - can be activated with this button.

'`Lock key mode`'
: Certain applications may excessively change the cursor key/numeric keypad mode from standard to applications mode. If you don't want this to happen you can forbid it by turning on the `Lock key mode` button.

'`Wrap cursor moves`'
: According to the VT100-specifications the cursor movements have to stop at the edges of the screen. In spite of this the cursor may leave these borders, especially in ANSI-mode, and may appear at the other side of the screen. This button activates a more 'tolerant' mode.

'`Wrap characters`'
: This function activates the automatical carriage return function which is triggered as soon as the cursor crosses the right screen margin. To avoid unpleasant side-effects, this gadget should be activated all the time.

'`'CLS' resets cursor position`'
: As per the VT-100 specs, the control sequence to clear the screen is not to change the current cursor position. However, several applications expect it to be moved to the top left corner of the screen. This button will activate this behaviour.

'`Insert mode`'
: Normally, 'term' is in overwrite-mode (characters entered overwrite the contents of the screen). If this gadget is activated, typed characters are inserted by pushing all the characters right of the cursor towards the right margin.

  The insert-mode does only work for lines. If characters are pushed out of the screen they cannot be restored.

'`New-line mode`'
: This gadget activates a special mode in which some VT100 control sequences cause 'term' to perform a linefeed instead of clearing the screen or other serious changes of the contents of the screen.

'Font scale'
: VT-100 offers several different sizes of fonts. Some can be selected with this gadget:

  'Normal' The normal height of the font.

  'Half width'
  : Half width of font.

  The special size characters are produced in real-time, so text output may be slightly slowed.

'Lock font mode'
: If this switch is turned on, any terminal commands to change the font scale will be ignored.

'Scrolling'
: This button selects one of two different scrolling modes: smooth or jumping.

'Destructive backspace'
: Determines if the Backspace code, which will delete the character left from the cursor, only moves the cursor to the left or removes the character from the screen. 'Off' turns this feature off, 'Overstrike' clears the character below the cursor and 'Shift' shifts the line contents to the right of the cursor to the left.

'Swap 'Backspace' and 'Del' keys'
: For all VT100 groupies: if this switch is enabled the roles of the backspace key and the delete key are reversed. *This also applies to sequences such as* Control + H *which will produce a delete character instead of a backspace character.*

'Printer control enabled'
: 'term' supports the standard VT220 printer control commands. If you don't want the application on the other side of the line to play with the printer the corresponding support commands can be disabled with this switch. If disabled 'term' will act like a VT220 terminal with no printer attached.

'Answerback message'
: The text to send across the serial line whenever an ENQ character is received. The text is - as usual - a command sequence.

'Max. prescroll'
: This is where you set the maximum number of lines the terminal emulation will scroll the screen contents up in one single 'jump'. This feature is often referred to as 'prescroll'. *Note: the data throughput rate plays an important part when counting the number of lines to scroll. The higher the rate, the more lines will be scrolled.*

'Max. jump scroll'
: Here is where you set the number of lines to move the screen contents up when the cursor moves beyond the last terminal screen line.

'Use' Use the current settings.

'Default (phone book only)'
: Drop the current settings, making a connection to the corresponding phone number will leave the corresponding main configuration entry unchanged.

'Cancel'
: Keep the old settings.

19.7 Clipboard panel

'Clipboard unit'
: The clipboard supports several units (0-255) which can be accessed independently. It can make sense to change this value but generally you will probably leave it as unit '0'.

'Character delay'
: When sending text this number determines how many seconds to wait before sending the next character.

'Line delay'
: When sending text this number determines how many seconds to wait before sending the line-termination character (carriage return).

'Line prompt'
: The character to wait for the receiver to issue after a line of text is send. These character may include control sequence tokens.

'Send timeout'
: If the 'Text pacing' mode is set to 'Wait for echo' or 'Wait for line prompt' the maximum time to wait for echo/prompt before the insertion is aborted.

'Text pacing'
: The mode to determine how text is sent to the remote:

  'Direct'
  : Each line will be sent without any delay.

  'Wait for echo'
  : The program will wait for each single character sent to be echoed by the remote.

  'Wait for any echo'
  : The program will wait for the remote to return any character in response to any character sent. Typically, this is the case with password prompts issued by BBSes.

  'Wait for line prompt'
  : The program will wait until the remote sends a certain line prompt text.

  'Character/line delay'
  : The program will respect the character/line delay values to be set using this control panel.

`Keyboard delay`
: The program will send character separated by a delay to be determined by the current system keyboard repeat delay.

Note: the 'echo' text pacing mode is to be used with greate care. Certain online services do not echo characters back to the sender as they run only in half-duplex mode. On the other hand most mailbox programs will not echo certain characters, such as escape codes, etc.

`Paste prefix`
: If enabled, the text to send before the clipboard contents are fed into the input stream, see Chapter 24 [Clipboard], page 99.

`Paste suffix`
: If enabled, the text to send after the clipboard contents are fed into the input stream, see Chapter 24 [Clipboard], page 99.

`Use`
: Use the current settings.

`Default (phone book only)`
: Drop the current settings, making a connection to the corresponding phone number will leave the corresponding main configuration entry unchanged.

`Cancel`
: Keep the old settings.

19.8 Capture panel

`Log actions`
: If enabled will write a protocol of each program action (uploads, downloads, dial attempts, etc.) to a file. Each action is listed along with time and date. Carrier-lost events will also note the approximate cost of the call.

`Log file`
: The name of the file in which the information on the actions executed by 'term' will be logged if `Log actions` is enabled.

`Log calls`
: If this switch is enabled, 'term' will create call-log files in a format compatible with the `NComm` terminal program.

`Call log file`
: The name of the file in which the information on calls made by 'term' will be logged if `Log calls` is enabled.

`Maximum text buffer size`
: To save memory, a high-water mark concerning the maximum amount of memory the text buffer (see Chapter 23 [Text buffer], page 97) will allocate for text may be specified. The

minimum value to be entered here is 2,000 bytes which are roughly equivalent to two text buffer pages of text. A value of 0 will cause the text buffer to always allocate as much memory as required to buffer all the incoming text.

'`Text buffer enabled`'
: This switch works in conjunction with the '`Freeze buffer`' menu entry (freezing the text buffer contents). In fact, the menu entry is adjusted according to the configuration settings whenever a new configuration is invoked (that is, whenever a new connection is made through the dialing panel or at program startup time). *'term' will only freeze the buffer if this switch is set, it will leave the buffer state (frozen or not) untouched if this switch is not enabled.*

'`Text buffer file path`'
: The path the file requester will bring up when saving the contents of the text buffer.

'`Connect-auto-capture`'
: If enabled will automatically open a capture file after successfully making a connection. Any other already open capture file will be closed before proceeding. The files created will bear the names of the corresponding phonebook entry.

'`Creation date`'
: By default 'term' will append the date of the call made to the name of the auto-capture file created. Alternatively, 'term' will use only the BBS name for auto-capture files, including creation date lines in the file.

'`Capture filter enabled`'
: If selected, command sequences are filtered out before the incoming characters are captured to disk or printer. This makes good sense with noisy lines generating random characters which might scare your printer. It also produces a text file that is much more readable than with all the control sequence codes cluttering up the text.

'`Capture file path`'
: This text gadget contains the path in which the the capture files will be created if '`Connect-auto-capture`' is enabled.

'`Open buffer window`'
: This switch controls which part of the text buffer contents the review buffer window will display when opened:

  '`Top`' Displays the top of the buffer contents.

  '`End`' Displays the end of the buffer contents.

'`Remember window position`'
: If this switch is in effect, the program will remember the text display position between calls rather than jumping to the top or the end of the text buffer.

'`Open buffer screen`'
: This switch controls which part of the text buffer contents the review buffer screen will display when opened:

'Top' Displays the top of the buffer contents.

'Remember'
Keeps the previous buffer position.

'End' Displays the end of the buffer contents.

'Remember screen position'
If this switch is in effect, the program will remember the text display position between calls rather than jumping to the top or the end of the text buffer.

'Buffer screen position'
The buffer screen will usually not be quite as wide as the system overscan settings permit. This switch determines the horizontal placement of the screen:

'Left' The screen will be left-edge aligned.

'Centre' The screen will be centred.

'Right' The screen will be right-edge aligned. This will bring the screen depth arrangement gadget in line with the other screens.

'Buffer line width'
The text buffer stores lines at a fixed size, this slider determines the maximum line width.

'Display mode'
This is where you choose the screen display mode the buffer screen should use.

'Search history size'
The text buffer search function maintains a backlog of all the search text entered. The number of texts to remember, before the oldest is discarded, can be set using the 'Search history size' control.

'Use' Use the current settings.

'Default (phone book only)'
Drop the current settings, making a connection to the corresponding phone number will leave the corresponding main configuration entry unchanged.

'Cancel' Keep the old settings.

19.9 Command panel

Here you will find entries for four command sequences which serve four functions:

'Startup command'
At the beginning of every session with 'term' and after a connection has been established by the dialing routine a command-sequence is executed. Do not use this command for auto-login

scripts and such, this is what the '`Login command`' is for. Note that the dialing procedure executes the '`Startup command`' after the '`Login command`'.

'`Login command`'
: This command will be executed immediately after the dialing procedure has established a connection. You should use this command for login scripts and such. Note that the dialing procedure first executes the '`Login command`' and then the '`Startup command`'.

'`Logoff command`'
: The command to execute when the line is hung up or the carrier signal is lost.

'`Upload command`'
: 'term' will execute this command after a successful upload has been made.

'`Download command`'
: 'term' will execute this command after a successful download has been made.

'`Use`'
: Use the current settings.

'`Default (phone book only)`'
: Drop the current settings, making a connection to the corresponding phone number will leave the corresponding main configuration entry unchanged.

'`Cancel`'
: Keep the old settings.

19.10 Miscellaneous panel

This is the place where options can be set which would not fit into other control panels

'`Program priority`'
: Use this slider to determine the priority under which the 'term' main process is to operate. Adjusting this value can make 'term' perform more reliably in a system which experiences heavy task loading. It is recommended to experiment with this value until a satisfactory state is found. Setting the program priority too high or too low may affect the performance of coprocess services such as the double-buffered file I/O routines.

'`Backup configuration`'
: By default the local program configuration saved along with a phonebook entry will replace the global configuration as soon as a successful connection is made. If this switch is enabled, 'term' will remember the global configuration in effect before the configuration data of a phonebook entry is adopted. As soon as the serial driver loses track of the carrier signal or the user chooses to hang up the line, 'term' will switch back to the previous global configuration.

'`Show fast! macros`'
: If this switch is enabled, the fast macro window (see Section 20.1 [Fast macro panel], page 73) will be opened whenever the current configuration becomes active (e.g. at startup time).

'Release serial device when iconified'
: By default the serial device driver is released when 'term' is iconified. Unfortunately, some modems drop the line when the device is closed so this switch allows you to keep the link.

'Override transfer path'
: Each batch file transfer protocol allows you to specify the name of the directory to place the files it receives in. By default 'term' will redirect the files to a directory to be specified in the path panel (see Section 19.11 [Path panel], page 64). If this switch is disabled, the internal settings of the current transfer protocol will be used.

'Use auto upload panel'
: Some users found the so-called '`ZModem auto-upload panel`' to be invoked whenever 'term' encounters the ZModem inquiry sequence rather disturbing, this switch will allow them to disable this feature.

'Transfer performance meter'
: When a file transfer is running, 'term' may optionally display the file transfer performance in a small resizable window. The lines drawn represent the following information:

  - Black line: This line displayes the current transfer performance (usually heavily oscillating).
  - Blue line: This line displays the average transfer performance.
  - White line: This line displays the smallest transfer performance (should be constant during the transfer).

'Set 'archived' bit'
: If enabled, this switch will cause 'term' to set the `Archived` bit on files transferred using the XPR-interface.

'Transfer file icons'
: This switch works in conjunction with the drag & upload feature (see Section 20.5 [Transfer progress panel], page 76). By default, 'term' will upload only the files whose icons are dragged on the main window. If this switch is in effect the icon files will be transferred as well.

'Create icons'
: If this switch is in effect 'term' will try to provide icons for all files it receives. The following file types (and the corresponding icon files) are supported:

  - Text file: icon '`ENV:sys/def_text.info`'
  - Sound file: icon '`ENV:sys/def_sound.info`'
  - Picture file: icon '`ENV:sys/def_picture.info`'
  - Tool: icon '`ENV:sys/def_tool.info`'
  - Archive file: icon '`ENV:sys/def_archive.info`'

- Preferences file

  icon 'ENV:sys/def_pref.info'
- Other file types

  icon 'ENV:sys/def_project.info'

Text and pictures saved by the program will also get icons attached.

'File comment'
: This is where the action to perform on downloaded files can be set:

  'Ignore' The file comment will not be touched.

  'File type'
  : The file will be examined and a guess will be made which type of file it is. The file comment will be set to the name of the file type.

  'Source and time'
  : The current BBS name and the time the file was received are placed in the file comment.

'Simple file I/O'
: This switch controls whether 'term' is to use double-buffered file management routines or not.

'I/O buffer size'
: This slider controls how much memory the double-buffered file management routines will allocate for each buffer. This means a value of 4096 bytes will result in an allocation of 8192 bytes in total.

'Use' Use the current settings.

'Default (phone book only)'
: Drop the current settings, making a connection to the corresponding phone number will leave the corresponding main configuration entry unchanged.

'Cancel' Keep the old settings.

19.11 Path panel

In this part of the settings all paths, which 'term' uses to save or load any data, can be determined.

'...Uploadpath'

'...Downloadpath'
: The directories in which the functions contained in the 'Transfer' menu will search and create files.

'Configuration storage directory'
: The directory that will contain all configuration files (phonebook, macro keys, etc.). The default configuration file is called 'term.prefs' and will be searched in the path defined

by the enviroment variable '`TERMCONFIGPATH`' (see Chapter 29 [Environment variables], page 109).

'`Text editor`'
: Contains the name and search path of the editor used by '`Edit & upload text file`' in the transfer menu (see Section 18.5 [Transfer], page 40).

'`"term" help text file`'
: This is where the name of the online-help text file is stored. By default this is '`PROGDIR:term.guide`'.

'`Use`' Use the current settings.

'`Default (phone book only)`'
: Drop the current settings, making a connection to the corresponding phone number will leave the corresponding main configuration entry unchanged.

'`Cancel`' Keep the old settings.

19.12 Transfer panel

This is where the libraries to be employed for file transfers are to be selected.

'`Default transfer library`'
: This library will be active most of the time, unless a special library is temporarily enabled. Auto-downloads and auto-uploads will be handled by this library, even if you select a text transfer in the auto-upload panel.

'`Mangle filenames for upload`'
: Certain transfer protocols running under MS-DOS get into serious trouble if told to receive files with names which do not match the local naming scheme (8 characters for the name + `"."` + 3 characters for the extension). For example, in such situations ZModem will restart the file transfer over and over again in a row without getting anywhere. To steer clear of trouble you can turn on the '`Mangle filenames for upload`' switch which will cause the file transfer protocol to report 'condensed' file names to the remote receiver. A special algorithm will shrink the file names to the MS-DOS file name template, clearing potentially dangerous character combinations on the fly. *Note: the algorithm may map two different Amiga file names to the same MS-DOS file name, so watch out!*

'`ASCII upload library`'
: The XPR library to be used when sending plain ASCII text as triggered by seleting the `Upload ASCII file(s)` menu item. This is usually '`xprascii.library`'. If no name is given, the `Default transfer library` will be used.

'`Use built-in upload routine`'
: Instead of the ASCII upload library 'term' will use the built-in routines.

'ASCII download library'
: The XPR library to be used when receiving plain ASCII text as triggered by selecting the `Download ASCII file(s)` menu item. This is usually 'xprascii.library'. If no name is given, the `Default transfer library` will be used.

'Use built-in download routine'
: Instead of the ASCII upload library 'term' will use the built-in routines. Incoming text will be received in a fashion similar to the capture and review buffer functions, but no filtering, etc. will be performed. As the download routine cannot always tell when all the data to be received has arrived, you may need to stop the download by hand.

'Text upload library'
: The name of the XPR library to be used when selecting the menu item `Upload text file(s)`. If no name is given, the `Default transfer library` will be used.

'Text download library'
: The name of the XPR library to be used when selecting the menu item `Download text file(s)`. If no name is given, the `Default transfer library` will be used.

'Binary upload library'
: The name of the XPR library to be used when selecting the menu item `Upload binary file(s)`. If no name is given, the `Default transfer library` will be used.

'Binary download library'
: The name of the XPR library to be used when selecting the menu item `Download binary file(s)`. If no name is given, the `Default transfer library` will be used.

'Use' Use the current settings.

'Default (phone book only)'
: Drop the current settings, making a connection to the corresponding phone number will leave the corresponding main configuration entry unchanged.

'Cancel' Keep the old settings.

Settings for each transfer library are saved in text files in the 'ENVARC:' and 'ENV:' drawers.

Leave the default transfer library set to the one you intend to use most.

19.13 XPR options sample

You will find an excerpt of the 'xprzmodem.doc' documentation file for the ZModem file transfer protocol below which is the default transfer protocol 'term' is shipped with. Please note that other file transfer protocols will sport different options and controls, you should consult the corresponding documentation for more information.

'`Text translation mode:`'

'`Y = Text Yes`'

If receiving, translate CR/LF pairs or solo CR chars to normal Amiga LF chars. Ignore data past ^Z. If sending, suggests to receiver that they should receive this file in text mode.

'`N = Text No`'

Receive file verbatim, without changes. If sending, suggest to receiver that they receive this file verbatim, without translations.

'`? = Text status unknown`'

If receiving, use sender's suggestion as to whether to do end of line translations or not. If sending, tell receiver to use default mode, since we don't know either.

'`C = Text mode set by Comm program`'

The library asks the communications program whether or not to use Text mode for each file. If the communications program does not support the necessary '`xpr_finfo()`' call, or if the call fails, this option acts like T?. From the user's point of view, what this option normally does is set the Text mode to match the communications program's built-in text/binary/end-of-line/translation mode, if any.

'`Overwrite mode:`'

'`Y = Overwrite Yes`'

If about to receive file with same name as one which already exists, delete the old file and receive the new file in its place.

'`N = Overwrite No`'

If about to receive file with same name as one which already exists, append ".dup" onto the name of the new file to keep them separate.

'`R = Overwrite Resume`'

If about to receive file with same name as one which already exists, resume receiving file data from the current end of the existing file.

'`S = Overwrite Skip`'

If about to receive file with same name as one which already exists, skip this file, we don't want it. Batch transfers will move on to the next file in the set, if any.

'`Buffer size:`'

'`xprzmodem.library`' adds a layer of file I/O buffering in addition to whatever the comm program may or may not provide. This option sets the size of XPRZModem's file I/O buffer in kilobytes. The minimum value is 1 KB, for those using RAM drives or fast hard drives, or those whose comm programs already provide sufficient buffering. The maximum value is as much contiguous RAM as you have available in your Amiga. If you specify more than is actually available, XPRZModem will keep decrementing the buffer size requested by 1 KB until the memory allocation works. That way, if your RAM is too fragmented to use the amount you request, XPRZModem simply uses the largest block available. Buffering is especially helpful for floppy drive users; it keeps your drive from continuously gronking and slowing things down all through the transfer.

Additional note for 'term' users: this option is practically replaced by the '`I/O buffer size`' *settings in the miscellaneous panel (see Section 19.10 [Miscellaneous panel], page 62).*

'`Frame size:`'
: Although normally avoided, ZModem has the ability to require an ACK to be sent from the receiver to the sender every X-many data bytes. Normally you don't want to use this feature, because not waiting for ACKs is part of how ZModem works so fast. However, this feature can be very useful in conjunction with file I/O buffering on slow devices (namely those floppy drives). If you set up a large I/O buffer to avoid gronking your floppy so often, you'll find that when the buffer finally *does* get around to being flushed that it can take a looonng time; so long, in fact, that the delay can cause timeouts and errors. But if you set your ZModem to require the sender to wait for an ACK every buffer's-worth of data, the sender will politely wait for you to flush your buffer to the slow floppy and send it an ACK saying it's OK to continue now. This value should be set to 0 to disable ACKs (normal mode), or set it to the actual number of data bytes allowed between ACKs. For example, if you set the Buffer size to 64KB because of your floppy, you should also set the Frame size to 65536 bytes.

'`Error limit:`'
: This allows you to set the number of sequential errors which will be required to convince ZModem to abort the transfer. The normal value is 10, meaning that 10 errors must happen in a row with no valid data being transferred in order to cause an abort. This setting is provided for those using XPRZModem with a BBS, who may wish to use a relaxed setting, or those with really lousy phone lines who are desperate and patient enough to want the transfer to continue in spite of horrible performance.

'`Auto-activate mode:`'
: '`Y = Auto-activate Yes`'
  : If the comm program supports the ability, the library will automatically go into receive mode when the start of a ZModem download is detected.

  '`N = Auto-activate No`'
  : Don't try to automatically start downloading, make the user activate it.

'`Delete after sending:`'
: '`Y = Delete Yes`'
  : Delete each file after it has been Successfully sent.

  '`N = Delete No`'
  : Don't delete files after sending them.

'`Keep partial files:`'

'Y = Keep Yes'
: Keep the fragment of a file received so far if file reception is aborted. This allows you to use the Overwrite Resume option above to pick up where you left off on your next attempt.

'N = Keep No'
: Delete any partially-received file after an aborted transfer.

'Send full directory path:'

'Send path Yes'
: Send full filenames including directory path to receiver.

'Send path No'
: Send only simple filenames, not including directory path.

'Default received path:'
: Store all received files in this directory, if option "Use received path" is not checked. Ignored entry if option 'Use received path' is checked. The path can be any valid existing directory, with or without trailing / (e.g. 'df0:', 'Comm:hold', etc.).

Additional note for 'term' users: the default received path option is ignored if the 'Override transfer path' *switch in the miscellaneous panel* *(see Section 19.10 [Miscellaneous panel], page 62)* *is enabled.*

You will also find the familiar 'Use' and 'Cancel' buttons here which will either keep or discard the changes you made to the settings.

19.14 Translation panel

'term' is capable of replacing any incoming and outgoing character with custom text. This may come in handy with the numerous incarnations of the dreaded IBM PC font. The control panel to be opened features a large list of buttons, each single one representing a single character. Some characters are shown with their corresponding glyphs, some with their symbolic names and some as plain numbers. Clicking on one of the buttons will bring up a control panel which allows setting the text to be received by the terminal emulation when a certain character is received and the text to be sent when a certain characters is transmitted.

Both texts can consist of standard command sequences (see Chapter 25 [Command sequences], page 101), except for the following commands which are not supported: '\a', '\c', '\d', '\g', '\i', '\p', '\u' and '\x'.

Please note that the translation does not come for free, terminal input and output speed may suffer.

As of this writing only a few translation table files (see below) are included in the distribution. If you wish to create translation tables for IBM doorway mode, national IBM-PC style font variants, etc. feel free to send them to me. I will try to include them in the next 'term' release.

Currently included in the 'term' distribution are the following translation table files:

'`ISO-4-(GB).prefs`'
: British 7 bit (ISO code 4) character set.

'`ISO-10-(S).prefs`'
: Swedish 7 bit (ISO code 10) character set.

'`ISO-11-(S).prefs`'
: Swedish 7 bit (ISO code 11) character set.

'`ISO-15-(I).prefs`'
: Italian 7 bit (ISO code 15) character set.

'`ISO-16-(P).prefs`'
: Portuguese 7 bit (ISO code 16) character set.

'`ISO-17-(E).prefs`'
: Spanish 7 bit (ISO code 17) character set.

'`ISO-21-(D).prefs`'
: German 7 bit (ISO code 21) character set.

'`ISO-60-(N).prefs`'
: Norwegian 7 bit (ISO code 60) character set.

'`ISO-61-(N).prefs`'
: Norwegian 7 bit (ISO code 61) character set.

'`ISO-69-(F).prefs`'
: French 7 bit (ISO code 69) character set.

'`PC-8.prefs`'
: Character translation for standard IBM-PC style font. If you wish to use these translation tables, make sure to set the '`Font`' type in the terminal panel (see Section 19.5 [Terminal panel], page 53) to '`IBM-PC style (raw)`'.

Unfortunately, there is no translation available for the Norvegian and Danish variants of the PC-8 character set as I do not yet have a fitting Amiga font available. Similar reasons have yet prevented to implement PC-850 character set support.

19.15 Function key panel

This control panel allows setting user definable texts for all ten function keys. All texts are considered command sequences (see Chapter 25 [Command sequences], page 101), a topic which will be covered later in this document.

`Modifier`
: All in all 40 keys may be covered with user defined command sequences (Chapter 25 [Command sequences], page 101). As the Amiga keyboard only has ten function-keys this button switches between the modifier keys (`Shift`, `Control`, `Alt`) which, if pressed in addition which a function key, will execute one of the 40 command sequences.

`Load`
: Load the function key settings from a file.

`Save`
: Save the function key settings to a file.

`Use`
: Use the current settings.

`Cancel`
: Keep the old settings.

As the definition of the function keys with command sequences contradicts the standard definition of the four functions keys of a VT-100-terminal, the keys `F1-F4`, which may be executed by pressing the `Shift` key and the approriate function-key simultaneously, are mapped to the standard sequences for function-keys. The user may - of course - change these settings.

If an external terminal emulation happens to be active, those function keys the emulation has allocated for itself will be disabled and cannot be edited.

20 Cursor key panel

This control panel both works and looks similar to the function key panel (see Section 19.15 [Function key panel], page 71), the only difference is that it is to assign command sequences to the cursor keys rather than to the function keys. Displayed are the assignments for all four cursor keys and the following buttons:

'Modifier'
: Any cursor key can be pressed along with one of the modifier keys (`Shift`, `Control`, `Alt`). This button will switch between the different assignments.

'Load' Load the cursor key settings from a file.

'Save' Save the cursor key settings to a file.

'Use' Use the current settings.

'Cancel' Keep the old settings.

20.1 Fast macro panel

The design and implementation of the settings to be configured in this menu are closely related to the menu entry function key panel (see Section 19.15 [Function key panel], page 71) discussed before. The only difference to be seen in the fact that the fast! macros are mapped to buttons rather than function keys (more on this topic later in this document, see Chapter 26 [Fast! macros], page 103).

'Macro list'
: The list of macros entered yet, to edit one of these, select it by clicking the mouse button with the mouse pointer on it.

'Macro' The name of a macro by which it is listed in the fast! macro list.

'Macro code'
: The command sequence (see Chapter 25 [Command sequences], page 101) associated with a fast! macro. Command sequences are discussed later in this document.

'New' Appends a new macro to the list. The user may then select and customize it.

'Delete' Removes the currently selected macro from the list.

'Load' Loads the macro list from a file.

'Save' Saves the macro list to a file.

'Top' Places the currently selected macro at the top of the list.

'`Up`' Moves the currently selected macro one entry up.

'`Down`' Moves the currently selected macro one entry down.

'`End`' Places the currently selected macro at the end of the list.

20.2 Hotkey panel

This is where the key sequences used to arrange screens and to execute special functions are to be configured.

'`term screen to front`'
: The keys to press to bring the 'term' screen to the front.

'`Buffer screen to front`'
: The keys to press to bring the screen of the text buffer to the front.

'`Skip dial entry`'
: As an alternative to the '`Skip`' button, pressing these keys will skip a dialing entry if the dialing function is currently active.

'`Abort ARexx command`'
: An ARexx script started from within 'term' can be aborted by pressing these keys. Use this function only if pressing `Control + C` does not stop the program execution.

'`Commodity priority`'
: The commodity priority to assign this task to. You may want to change this value if you have more than one program running which uses the same key sequences as 'term'. The program with the higher commodity priority will receive the keystrokes first.

'`Hotkeys enabled`'
: Whether the hotkeys are enabled or not can be toggled by clicking on this button, or by using the '`Exchange`' program to be found in the '`Tools/Commodities`' drawer.

'`Load`' Loads the hotkey settings from a file.

'`Save`' Saves the hotkey settings to a file.

'`Use`' Use the current settings.

'`Cancel`' Keep the original settings.

'term' will refuse to accept invalid keyword combinations. You will be notified by a brief screen flash/bell signal and the cursor will reappear in the text entry field whose contents are rejected.

20.3 Speech panel

If enabled, the Amiga speech synthesizer will be used to alert the user of certain actions, such as carrier lost, connection made, etc. This feature makes sense if 'term' is running in the background where the user cannot see what is actually happening on the main screen. By default this feature is disabled.

Note: speech synthesis is no longer available since Workbench v2.1 was introdured!

`'Rate'` Speaking speed in words per minute.

`'Pitch'` The greater this value, the higher the voice appears to be speaking.

`'Frequency'`
Voice frequency in Hertz.

`'Volume'` The volume of the voice.

`'Sex'` Enabled female or male voice.

`'Speech enabled'`
Toggles the activity of the speech synthesizer.

`'Speak!'` Speaks a small sample text, note that speech must be enabled for this function to work.

`'Load'` Loads the speech settings from a file.

`'Save'` Saves the speech settings to a file.

`'Use'` Use the current settings.

`'Cancel'` Keep the original settings.

20.4 Sound panel

As an option 'term' will associate sounds with special program functions and events. This is where the sounds are configured:

`'Terminal bell sound'`
The sound to be played whenever a `BEL` character is output on the terminal screen.

`''Connect' sound'`
The sound to be played when a connection is established.

`''Disconnect' sound'`
The sound to be played when a connection is lost.

‘‘File transfer finished’ sound’
: The sound to be played when a file transfer is finished successfully.

‘‘File transfer failed’ sound’
: The sound to be played when a file transfer is finished unsuccessfully.

‘Modem ‘ring’ sound’
: The sound to be played when the modem detects a call by a different modem.

‘Modem ‘voice’ sound’
: The sound to be played when the modem detects a phone call.

‘Volume’
: This slider affects the volume of all sounds produced by ‘term’. Setting it to zero suppresses sound output.

‘Preload sound files’
: If this switch is enabled ‘term’ will load all sound files immediately rather than accessing and loading them on demand. This may save access time when a sound is to be played but may eat up precious memory.

‘Load’
: Load the sound settings from a file.

‘Save’
: Save the sound settings to a file.

‘Use’
: Use the current settings.

‘Cancel’
: Keep the old settings.

There is no fixed size limit to sound files, the amount of available system memory matters. The sound files may be compressed, mono or stereo files.

20.5 Transfer progress panel

The transfer routines open an information window in which a number of transfer parameters are displayed. Additionally, the file transfer can be aborted by clicking either of the three buttons (‘Abort entire transfer’, ‘Skip current file’ or ‘Abort transfer batch’). *For most transfer protocols all buttons have the same effect.* Consult the documentation to see if different levels of abort are supported by your favourite transfer protocol.

The following information is displayed in the transfer window:

‘Protocol’
: The name of the transfer protocol currently running.

'Information'
: A list to contain error message, the names files transferred and miscellaneous other messages addressed to the user.

'File'
: The name of the file being transferred.

'Next file'
: The name of the next file to be transferred.

'Space left'
: The space left on the destination device. 'term' will try to calculate the number of blocks the file being received will take on the destination device and display a warning the file in question is probably not going to fit.

  Caution: 'term' only makes a very likely guess which may or may not come true. The guess may be wrong if the destination device happens to be a kind of ram-disk which shrinks and expands as memory requirements come and go. Such devices are usually 100% full. In most other cases you will probably be able to make room for the file being received before any space problem turns up.

'Completion time'
: If the corresponding information is available, the point of time when the current file will be transferred completely.

'File size'
: If available, the size of the file.

'Bytes transferred'
: Number of bytes transferred yet.

'Total size'
: The total size of all files to be transferred.

'Total bytes transferred'
: The total number of bytes transferred yet.

'Files transferred'
: The number of files transferred yet and the number of files to go.

'Blocks transferred'
: Number of data blocks transferred yet.

'Characters/second'
: The effective transfer speed in characters per second.

'Character delay'
: The delay between two character being sent.

'Packet delay'
: The delay between two packets being sent.

'Packet type'
: A short description of the data block type employed for data transfer.

'`Block check type`'
: The method employed to verify the integrity of the data blocks being transferred (this usually is a form of cyclic redundancy checking).

'`Block size`'
: Size of a data block in bytes.

'`Expected time`'
: The time the transfer protocol expects the transfer will take.

'`Elapsed time`'
: The time elapsed during transfer.

'`Message`'
: A message of the transfer protocol addressed to the user.

'`Number of errors`'
: The number of errors occured during file transfer.

'`Number of timeouts`'
: The number of timeouts occured during file transfer.

If the currently active transfer protocol provides the necessary information, two bars will be displayed at the bottom of the transfer window indicating the amount of transferred data and of time to go before the transfer is finished.

'term' knows about the Z-Modem data-inquiry sequence the remote receiver issues when expecting files. If recognized, this sequence will cause 'term' to display a requester asking for the type of data upload: text or binary. One could call this feature 'auto upload'. You also have the opportunity to select '`Abort`' which will transfer the ZModem abort sequence or to click on the '`Ignore`' gadget which will plainly ignore the fact that the ZModem inquiry sequence has been recognized. *The Z-Modem abort sequence will also be transferred if you select the '`Cancel`' button in the file requester to appear after selecting text- or binary-upload.* If the '`Upload from queue`' option is in effect the contents of the transfer queue will be uploaded.

Some transfer protocols will allow you to enter a default receive path the library is supposed to create files it receives in. On request (see Section 19.10 [Miscellaneous panel], page 62) 'term' will ignore these settings and use the settings to be changed in the '`Settings/Paths`' (see Section 19.11 [Path panel], page 64) menu instead.

Each file that is received and which does not remain empty is examined briefly to find out about the file type. If recognized successfully and the corresponding feature is enabled, a small comment indicating the file type will be attached to the file. 'term' currently knows about 83 different file types.

If the 'term' main window is opened on the Workbench screen, you can select and drag icons on it in order to upload the corresponding files. A requester will be opened to ask for the upload style (either binary or text).

In case a file transfer terminates with an unrecoverable error (*note: the transfer protocol is responsible for reporting error conditions to 'term'*) the file transfer window will stay open until explicitly closed by the user so the transfer error report list can be viewed.

20.6 ASCII-transfer panel

The built-in ASCII transfer routines as to be enabled in the transfer panel (see Section 19.12 [Transfer panel], page 65) display transfer progress information in a special window (note that sending and receiving will open different windows). Here is a description of the controls and displays:

'`Bytes transferred`'
: The number of bytes sent/received.

'`Lines transferred`'
: The number of text lines sent/received.

'`Information`'
: Transfer progress information and error display.

'`Character delay`'
: When sending text this number determines how many seconds to wait before sending the next character.

'`Line delay`'
: When sending text this number determines how many seconds to wait before sending the line-termination character (carriage return).

'`Text pacing`'
: The mode to determine how text is sent to the remote:

  '`Direct`' Each line will be sent without any delay.

  '`Wait for echo`'
  : The program will wait for each single character sent to be echoed by the remote.

  '`Wait for any echo`'
  : The program will wait for the remote to return any character in response to any character sent. Typically, this is the case with password prompts issued by BBSes.

  '`Wait for line prompt`'
  : The program will wait until the remote sends a certain line prompt text.

'`Character/line delay`'
: The program will respect the character/line delay values to be set using this control panel.

'`Keyboard delay`'
: The program will send character separated by a delay to be determined by the current system keyboard repeat delay.

Note: the 'echo' text pacing mode is to be used with greate care. Certain online services do not echo characters back to the sender as they run only in half-duplex mode. On the other hand most mailbox programs will not echo certain characters, such as escape codes, etc.

'`Quiet ASCII transfer`'
: This switch controls whether incoming text will be displayed in the terminal window. You may want to watch how the remote responds to the data sent/received.

'`Abort entire transfer`'
: Stops the ASCII data transfer.

'`Skip current file`'
: Stops sending the current file and proceeds to the next.

In case a file transfer terminates with an unrecoverable error the file transfer window will stay open until explicitly closed by the user so the transfer error report list can be viewed.

20.7 ASCII-transfer settings

'`Text pacing`'
: The mode to determine how text is sent to the remote:

'`Direct`' Each line will be sent without any delay.

'`Wait for echo`'
: The program will wait for each single character sent to be echoed by the remote.

'`Wait for any echo`'
: The program will wait for the remote to return any character in response to any character sent. Typically, this is the case with password prompts issued by BBSes.

'`Wait for line prompt`'
: The program will wait until the remote sends a certain line prompt text.

'`Character/line delay`'
: The program will respect the character/line delay values to be set using this control panel.

'Keyboard delay'
: The program will send character separated by a delay to be determined by the current system keyboard repeat delay.

Note: the 'echo' text pacing mode is to be used with greate care. Certain online services do not echo characters back to the sender as they run only in half-duplex mode. On the other hand most mailbox programs will not echo certain characters, such as escape codes, etc.

'Character delay'
: When sending text this number determines how many seconds to wait before sending the next character.

'Line delay'
: When sending text this number determines how many seconds to wait before sending the line-termination character (carriage return).

'Line prompt'
: The character to wait for the receiver to issue after a line of text is send. These character may include control sequence tokens.

'Send timeout'
: If the 'Text pacing' mode is set to 'Wait for echo' or 'Wait for line prompt' the maximum time to wait for echo/prompt before the insertion is aborted.

'Send CR' This button determines the sequence that is sent to the remote if a carriage-return is issued.

'As CR' Without any change

'As CR+LF'
: Carriage-return and linefeed

'Ignore' Suppress any carriage-return

'Send LF' This button determines the sequence that is sent to the remote if a linefeed is issued.

'As LF' Without any change

'As LF+CR'
: Linefeed and carriage-return

'Ignore' Suppress any linefeed

'Receive CR'
'Receive LF'
: These two buttons have largely the same effect as the Send CR/LF buttons, they are different in that they affect the incoming data rather than the data transmitted.

'Ignore data past terminator'
: With this option enabled the receiver will search for a termination character in the incoming data stream. If this character is found the transfer will be terminated.

'Terminator character'
: Enter the ASCII code of the terminator character to be used for the 'Ignore data past terminator' feature here.

'`Quiet ASCII transfer`'
: If this switch is not enabled, the built-in ASCII upload/download routines will display the outgoing/incoming data in the terminal window. This option is intended to let you watch the progress of the file transfer, so that, for example, if the remote does not respond to the data you send, you may want to stop and restart the upload.

'`Strip bit 8`'
: For those guys and gals who need it: if this switch is effect each character received or transmitted by 'term' will have its high-order bit cleared.

20.8 Phonebook

The functions described in the following can be found in the '`Modem`' menu and relate to the menu entries '`Phonebook`', '`Dial`' and '`Redial`'.

'term' is equipped with a telephone number management system, the phonebook, which is described in the following lines.

'`Name list`'
: The names of all phonebook entries are displayed here.

'`BBS name`'
: Name of the last selected telephone book entry.

'`Comment`'
: A comment to associate with a phonebook entry.

'`Phone number`'
: The telephone number of the last selected telephone book entry.

If a BBS supports multiple lines, the phone number of each line may be entered, each one separated by a vertical bar | character (example: '`123456|654321`' would cause the dialing routine to dial the numbers '`123456`' and '`654321`'). The dialing routine will process all these phone numbers before proceeding to the next phonebook entry.

The | character also works for the modem init, modem exit and dial prefix sequences. Whenever the dialing routine dials another phone number from a list separated by bars, it will try to find a matching init/exit/dial prefix sequence. If more phone numbers are specified than sequences are available, it will use the last sequence given (an example: a phone number may be given as '`123456|654321|12345`', the dial prefix text may be '`ATDP|ATDT`'; the dialing routine will call the first number using '`ATDP123456`', the second number using '`ATDT654321`' and the third number, since no special dial prefix is available, again using '`ATDT12345`').

'`Quick menu`'
: If this switch is in effect, the corresponding phone book entry will be put into the '`quick dialing menu`' (see at the right hand side of the main menu). Selecting the menu entry will dial the corresponding phone number. Note: only up to 50 phone numbers can be put into the list.

'`New entry`'
: Generates a new telephone book entry with standard settings and places it at the end of the telephone book.

'`Clone entry`'
: Will duplicate the currently selected phonebook entry and place it at the end of the list.

'`Remove entry`'
: Removes the last selected telephone book entry from the telephone book and frees the memory allocated for this entry.

'`Dial list`'
: Will pass the list of currently marked phonebook entries to the dialing routine.

'`Clear list`'
: Will clear the marks set for each telephone entry in the dialing list.

'`Use entry`'
: Takes over the local configuration settings saved with the currently selected phone book entry. Also installed are the associated password and user name entries.

'`Load list`'
: Loads the contents of a telephone book from a file.

'`Sort list`'
: If any phonebook entries have been selected to be dialled, the phonebook entries will be sorted in the order of dialing. The remaining phonebook entries will be sorted in ascending alphabetical order.

'`Save list`'
: Saves the contents of a telephone book to a file.

'`Copy config`'
: A lot of time can be saved by copying selected parts of the global configuration to a local configuration which is part of a phonebook entry. Selecting this button will invoke a control panel which allows to select which parts of the global configuration should be copied. The control panel also remembers which parts were copied when it was invoked the last time, see Section 20.10 [Copy panel], page 87 for more information.

'`Password`'
: Press this button if you wish to save a special access password with the currently active telephone book file. You will then be asked to enter the password. What you type will not appear on the screen.

To clear an existant password and to save the phonebook file without encryption, just press return when asked to enter the new password.

The next time you save the phonebook data, the password will be encrypted and saved with it, the phonebook data itself will be encrypted using the password.

Whenever an encrypted phonebook file is loaded, it will take longer to load than an ordinary phonebook file, the same applies to saving phonebook data.

'`Print`' This button will cause another control panel window to be opened, see Section 20.12 [Printing panel], page 89 for more information.

Another list is located at the right hand side of the window. Each entry refers to a control panel to be invoked on the currently selected phonebook entry.

'`Settings`'

'`Serial`'

'`Modem`'

'`Screen`'

'`Terminal`'
'`Emulation`'
'`Clipboard`'
'`Capture`'
'`Commands`'

'`Misc`'

'`Paths`'

'`Transfer`'
'`Translations`'
'`Function keys`'
'`Cursor keys`'
'`Fast! macros`'
These entries refer directly to the settings main menu entries of the same name.

'`User/Password`'
This entry will open a control panel in which the password and user name to be used for the current phonebook entry can be entered. Both password and name are made available from within the 'term' ARexx interface to allow auto login script files to set up a connection.

'`Rates`' 'term' will count the minutes you are online and connected to a BBS as soon as a connection is made through the dialing routine. This entry opens a control panel which allows setting the necessary data (see Section 20.9 [Rate panel], page 85).

If you are still online, the '`Dial list`' gadget will be disabled. In order to make another call hang up the line first.

To put a phonebook entry into the dialing list, Shift-click (i.e. hold down either shift key, then click once on the list entry) its name. The number appearing to the left of its name indicates the precedence of entries in the dialing list. To remove an entry from the list, Shift-click it again. Instead of Shift-clicking on an entry, the space bar may be pressed as well.

Double-clicking on a name will immediately dial the selected entry.

To dial the list of selected entries, press the '`Dial list`' button, control will be passed over to the dialing panel.

As I have been asked several times: For dialing a telephone number the dialing prefix specified for this telephone number is used. If the MNP-error correction for a certain mailbox has to be specifically switched on via the dial text, this has to be done in the modem settings for this mailbox and not in the global settings of 'term'. The '`Modem init command`' and '`Modem exit command`' entries of the telephone book can also be used for initialization.

20.9 Rate panel

'term' will count the minutes you are online and connected to a BBS as soon as a connection is made through the dialing routine. As soon as the connection is lost or you hang up, 'term' will use the information to be specified in this control panel to calculate the amount of money to be paid for the call.

'`Pay/unit`'
: The amount of money to be paid for each single time unit when online. This fee must be given in the smallest currency unit available (pence, cents, centimes, etc.).

'`Seconds/unit`'
: This is where you enter how many seconds each time unit lasts.

There are two different groups of the two entries listed above available: one for the first unit and one for all following units. So, if you only pay for the call you make but not for the time you spend making it, just enter the fee in the first group and set the second group to zero.

'`Days and dates`'
: This list contains the default rate settings and exceptions for certain dates and days of the week. Each line displays the type of the entry and a comment (separated by the '>>' character). The following types are available:

  '`Day(s)`' Settings for certain days of the week

'12. Jan (example)'
Settings for a specific date

If there is no special type available for an entry, it's probably the default settings you are dealing with. These settings are used whenever 'term' cannot find an entry for the current day.

For each entry in this list there is at least one associated starting time available which defines when the associated rate settings are to be used. You will find the time settings in the list titled 'Time'. To add a new time use the 'Add time' button. To edit an existing entry use the 'Edit time' button. To remove an entry, press the 'Remove time' button.

'Add date'
Will invoke a control panel to create a new rate entry to be used on a specific date. Use the sliders and button to select the day the settings will be valid for.

The control panel will allow you to enter nonexistant dates, such as '31. Feb' so you will have to make sure that the date in question really exists.

'Add day(s)'
Will create a new rate entry referring to one or more days of the week. Use the buttons of the control panel to select the days the current settings will be valid for.

'Import' Much work can be saved if the rate settings for the current phonebook entry are imported (or copied) from a different phonebook entry. To do so, select this button. The control panel to be opened will display the list of phonebook entries available and three buttons:

'Replace rates'
The rate settings of the current phonebook entry will be replaced by the settings of the selected entry.

'Append rates'
The rate settings of the selected entry will be appended to the current phonebook entry.

'Cancel' Will abort the selection.

Whenever a rate entry is selected, the corresponding parameters ('Pay/unit' and 'Seconds/unit') can be edited. If the entry refers to a certain date or a specific day of week three additional buttons are made available:

'Edit' Just as the labels says, will allow you to modify an entry after it has been created.

'Clone' Will duplicate the current rate entry and append it to the list.

'Remove' Removes an entry from the list.

20.10 Copy panel

This control panel allows you to select which parts of the global configuration to copy into the currently selected phonebook entry.

'All' Selects all parts.

'Clear' Clears the current selection.

'Copy to all'
: The selected parts will be copied to all phone book entries. If any phone book entries are selected when this action is to be performed, only the selected entries will be affected.

'Copy' This is where you select from which source the configuration information will be copied:

- 'Global configuration'
  : Parts of the currently active global configuration will be copied.
- 'Defaults'
  : When going online, instead of overriding the currently active global configuration with the supplied local phone book configuration the corresponding global configuration will be left unchanged.

20.11 Dial panel

The following information about the dialing process is displayed:

'Calling'
: The name of the telephone book entry belonging to the number being dialled. If it is just a telephone number the text '-- Unknown --' is shown, indicatinging that the name of the BBS is unknown.

'Comment'
: This is where the comment corresponding to the current dialing list entry is displayed.

'Number' The telephone number being dialed or just dialed.

'Next' The name of the telephone book entry which will be processed next if no connection is established. If no further entry exists, '-- None --' will be displayed.

'Timeout'
: A counter which is decreased every second and which reflects the time remaining to establish a connection or to cycle through the dial queue again.

'Attempt'
: This field shows the number of unsuccessful cycles made through the dialing queue to establish a connection.

'Message'
: A message to the user. This can be:

  'Dialing...'
  : A dial is in process.

  'Line is busy.'
  : The dialed number is engaged.

  'Incoming call!'
  : The modem has been called from another modem.

  'Incoming voice call!'
  : The modem is receiving a call which was not originated by another modem.

  'No dialtone detected!'
  : The modem was unable to detect any dialing tone on the line, it may possibly be not connected.

  'Connection established.'
  : Just as the name says...

  'Maximum number of dial retries reached!'
  : Just as the name says...

  'Dial attempt timeout.'
  : The time available to establish a connection has been reached or exceeded.

  'Redial Delay...'
  : Pause until the next cycle through the dialing queue.

Additionally, the following controls are available:

'Skip call'
: With this function the current dialing attempt is cancelled and the next number is processed. If no succeeding telephone number exists 'term' waits for the next cycle through the dial queue or until 'Skip call' is pressed again.

  There also is a hotkey combination available to accomplish the same task.

'Remove'
: This button works in part similar to the 'Skip call' button. Additionally, it removes the current phonebook entry from the dialing list.

'Go to online'
: If the line is very noisy, the connection to a mailbox may have been made, but the CONNECT text may be got lost. Pressing this button will cause 'term' to assume that the modem is in fact online now, start the rates accounting and return you to the main window.

'Abort dialing'
: Operation of this button exits the dial queue (leaving the the dial queue intact) and ends the dialing process.

'Start recording on connection'
: As soon as the connection is establish 'term' will start recording incoming text and your responses to it, thus making it possible to create auto-login scripts and such. For more information on this topic see Chapter 28 [Script recording], page 107.

If a connection is successfully made the corresponding entry in the dial queue will be removed.

Selecting the close gadget will close the window and cause the phone book panel to be reopened.

20.12 Printing panel

This control panel is part of the phone book. It is opened whenever the 'Print' button is selected and allows for setting the output options.

'Output file or device'
: This is where you enter the name of the file or device (such as 'PRT:') the phonebook printout is to be sent to.

'Plain text'
: If enabled only the plain and bare information text will be printed, else text attribute control sequences will be sent as well.

'Include...'
: Each switch determines whether the corresponding phonebook entry information will be included in the printout.

'Use' Will start printing the phonebook contents.

'Cancel' Returns to the phonebook.

21 Configuration hints

Admittely, 'term' has more configuration options and settings than you can shake a stick at. I have received a number of request to explain where to start after installing the program:

1. Start with the serial settings (see Section 19.1 [Serial panel], page 45). 'term' will usually copy your current system preferences settings. If you happen to know that they are correct and worked fine for you in the past you probably don't need to make any changes. But if you never were quite happy with the setup this is your chance to make it fit.

   As the lucky owner of a high speed modem to support all those nifty compressing transfer protocols nobody knows how to pronounce correctly (v.32/v.32bis/MNP/etc.) you will probably want to run it at baud rates around 9,600-19,200 bps. If you choose to do so make sure that the '`Handshaking`' switch is set to '`RTS/CTS`' or data is easily lost during transmissions. Note: some modems will lock up if the '`RTS/CTS`' handshaking protocol is enabled although they should support it. In most cases the modem behaviour can be changed, I recommend to consult the manual (good luck!), to turn the '`RTS/CTS`' handshaking off, to find the modem command to change the handshaking behaviour, to save the modem setup back to its nonvolatile RAM, to turn '`RTS/CTS`' handshaking back on and to restart.

   Older modem hardware usually supports only a fixed number of baud rates, mostly up to 2,400 bps. Do not enable '`RTS/CTS`' handshaking, leave it turned off. In fact if you don't turn it off 'term' will have trouble sending and receiving data.

   Make sure that the baud rate fits and your modem supports it. Modern modem hardware usually can adjust to the baud rate you choose, older modems will send & receive illegible gibberish if addressed at the wrong baud rate. Not unheard of are modems which can communicate with the terminal program only at fixed baud rates: while they are happy with 9,600 bps they might find 14,400 bps not at all worth responding to. I recommend that you try several baud rate settings until one is found to fit.

   If you don't want to use the built-in Amiga serial port hardware you will want to change the device name and unit number settings. Your I/O expansion hardware manual will tell you which name to choose and which device unit numbers are valid.

   The serial panel (see Section 19.1 [Serial panel], page 45) sports a number of additional options. *Do not change them right now!* In particular stay away from that sexy '`High-speed mode`' button and don't let the '`Buffer size`' slider tempt you. Return from the serial settings to the main menu by clicking on the '`Use`' button and save your current setup back to disk using the '`Save settings`' menu item.

2. Proceed to the modem settings (see Section 19.2 [Modem panel], page 47) and take a look at text gadget labeled '`Dial prefix`'. A modem usually dials phone numbers either using a technique called 'tone' or 'pulse' dialing. Technically, tone dialing requires your local phone net operator (some kind of computer) to listen to a sequence of sounds which represent the single digits of the phone

number dialled. Pulse dialing involves getting a number of electric pulses, each of which represents a digit of the phone number, transmitted across the line. Tone dialing is usually much faster than pulse dialing, but it isn't supported all over the world. If the receiver of your phone reports a number of beeping sounds when you dial a number you can use tone dialing. If you hear rattling sounds it's probably pulse dialing for you. Let's get back to the '`Dial prefix`' entry: you will probably see something like `~~ATDP` here. Arrogant as I am, this is my personal default setup. The letter `P` indicates that pulse dialing will be used. If you can, change this to `T` for tone dialing.

   Leave the rest of the modem setup as it is, do not change the '`Connect auto baud`' switch.

3. Next, take a look at the screen settings (see Section 19.3 [Screen panel], page 49). This is where you choose the terminal screen/window look and colours. By default 'term' is configured to open a plain four colour screen using the Amiga default font. This should be sufficient unless you plan to spend most of your modeming time in PC-driven BBSes which keep throwing lots of colours at you.

   Choose how many colours the terminal should use, the switch labeled '`Colour`' will let you choose between '`4 Colours (Amiga)`', '`8 Colours (ANSI)`', '`16 Colours (EGA)`' and '`2 Colours (Monochrome)`'. Each of these settings has a particular default palette attached. The '`Amiga`' mode will use your current system default colours. '`ANSI`' represents the choice of colours the ANSI committee responsible for standardizing a certain terminal command protocol to be the best given the constraints they had. '`EGA`' reflects whatever the engineers who designed the first Enhanced Graphics Adaptor card for the PC considered to be an enhanced colour palette. '`Monochrome`' is my idea how an extremely simplistic, while still readable colour choice could look like. Choose what you find appropriate, but keep in mind that the more colours to use the slower screen updates, scrolling and text output will get. Also, a 16 colour high resolution screen will put your system under additional stress if you are running an older Amiga model which is not equipped with the AGA chip set. Careful please, any changes you make will affect the performance of the program!

   You might want to change the screen mode or the user interface font. When you are satisfied with the setup, return to the main menu.

4. Now it's time to edit the terminal settings (see Section 19.5 [Terminal panel], page 53). This is where you control the basic behaviour of the terminal emulation. If you wish to use an IBM-PC style font for the terminal display you can do so by changing the '`Font`' switch to '`IBM PC style`'. Alternatively, you might find it worth changing the '`Text font`' instead which is the font to be used for terminal text output. Note that if the '`Font`' switch is set to anything else but '`Standard`' your '`Text font`' settings will be ignored. Well, actually they will not be entirely ignored, but the IBM PC style font will be opened in the point size you selected.

   Don't touch any other controls, return to the main menu when you are finished.

5. If you are likely to visit a lot of PC BBSes, edit the emulation settings now (see Section 19.6 [Emulation panel], page 55). You might want to turn on the switch labeled '`'CLS' resets cursor position`', otherwise the terminal screen might not get cleared properly when the BBS sends the control codes it considers appropriate for this purpose.

   Leave the rest of the setup as it is and return to the main menu.

6. The next step involves changing the path settings (see Section 19.11 [Path panel], page 64). When receiving files on your machine you might want to have them stored in a special drawer. You can do this by editing the default download paths. Most important is the '`Default binary download path`', I suggest to create a drawer called '`Downloads`' within the drawer 'term' resides in. Once this is done simply type the name `PROGDIR:Downloads` and return to the main menu. The next binary file downloaded will go into the '`Downloads`' drawer.

If you followed these steps 'term' should be configured for the first session. Save the current settings to disk now so you can always return to this working configuration later in case the changes you made to the current setup did not have the desired effect. You can try to fine-tune your 'term' setup now and change some of the options not covered in this brief introduction, but please remember to keep your original configuration file in a safe place, you will be glad you did.

22 Built-in terminal emulation

The 'term' built-in terminal emulation implements the VT-220 command set with a few exceptions. There are no country specific character sets, no down-line-loadable character sets, no user defined keys, no keyboard language support and only ten function keys, not twenty (many of these features are supported through the Amiga operating system). Most VT-102 and VT-52 commands should be supported as well, but since my documentation on these command sets is rather incomplete I cannot be entirely sure all the features are covered.

The numeric keypad and the four cursor keys can be switched into applications mode if requested by the remote. The four programmable function keys (also known as PF keys) are mapped to the top row of the numeric keypad. When in applications mode these keys will generate the codes produced by the PF keys on a VT-102 terminal. If in standard mode, you will need to hold down the `Control` key in order to make theses keys generate the correct PF key codes.

The `Tab` and `Space` keys receive special treatment if a qualifier key is held down when they are pressed. `Shift` + `Tab` will generate two `Escape` + `Tab` characters. `Control` + `Space` generates the ASCII `NUL` byte.

23 Text buffer

The text buffer implements a service which continually stores text displayed on 'term's main screen, so the user can refer to it lateron.

23.1 General characteristics

The size of the text buffer is managed dynamically so that for every new line which is read new memory must be allocated. So the size of the text buffer is limited only by the amount of the available memory. It is recommended that the text buffer is emptied periodically to avoid using the entire free memory.

If there is insufficient memory to place a new line into the text buffer, the first line will be deleted to make room for the new line.

23.2 Operation

The contents of the text buffer can be paged through using the keys for moving of the cursor (`Shift + Cursor` keys moves page by page, `Control + Cursor` key jumps to the beginning or end of the text buffer). Additionally, the numeric keypad keys are overlaid with jump and paging functions (corresponding to the inscriptions/graphics on the front of the keys).

There also is a pull-down menu available which is briefly described below:

'`Search`'
: A search function is called which scans from the topmost line on the screen downwards for the search text entered. Unless specified otherwise, the search does not distinguish between upper and lower case letters. If the search text is found it is displayed and highlighted.

  'term' remembers search strings entered. You can use the `Cursor up` and `Cursor down` keys to recall previous input.

'`Repeat search`'
: Continues the search process started with '`Search`'. The previously entered search text is carried over.

'`Go to main screen`'
: Switches to the main screen of 'term'.

'`Clear buffer`'
: Clears the contents of the text buffer.

'`Close buffer`'
: Closes the text buffer screen but leaves the contents unchanged.

24 Clipboard

Cut & paste functions are available on the main screen, the buffer screen and the review buffer. Here is how to use them:

'`Buffer screen`'
: Use the mouse to point to the first character you wish to send to the clipboard, hold down the select button, drag the mouse to the last character you wish to copy and release the button. The text marked will be transferred to the clipboard.

  Holding down the `Control` key while clicking on a character will feed the single character into the input stream, it will not be buffered in the clipboard.

'`Main screen`'
: Use the mouse to point to the first character you wish to send to the clipboard, hold down the select button, drag the mouse to the last character you wish to select and release the button. Select the '`Copy`' menu item (see Section 18.2 [Edit], page 38) to transfer the text to the clipboard. Instead of dragging the mouse you may also double-click on a single word to select it.

  Holding down the `Control` key while clicking on a character will feed the single character into the input stream, it will not be buffered in the clipboard.

'`Review buffer`'
: Use the mouse to point to the first character you wish to send to the clipboard, hold down the select button, drag the mouse to the last character you wish to select and release the button. Press `Amiga + C` to copy the selected text to the clipboard.

To paste the clipboard contents, i.e. feed them into the terminal input stream, either select the '`Paste`' menu item (see Section 18.2 [Edit], page 38) or press `Amiga + V`. In order to send the clipboard contents along with a '`Paste prefix`' and '`Paste suffix`' hold down any `Shift` key when selecting the '`Paste`' menu entry or when selecting text with the mouse (this works both with the main screen and the text buffer screen).

Hold down one of the `Alt` keys and press the left mouse button to make 'term' emit a number of cursor move sequences which will position the on-screen cursor at the spot where you clicked the mouse.

In standard text gadgets a solution had to be found to preserve the line editing functions while still supporting menu shortcuts. To undo any changes made press `Amiga + Q`, to clear the text gadgets press `Amiga + X`. Menus associated with the shortcuts `Amiga + Q/X` are called by holding down any `Shift` key along with the `Amiga` keys (i.e. `Shift + Amiga + Q` will select the '`Quit`' menu item if available).

25 Command sequences

Each text sent directly to the modem is a command sequence. This includes telephone numbers, modem initialisation strings, function key assignments, etc. In addition to the normal text strings various other commands are supported which will be described in the following section.

25.1 Backslash

'\\' Generates a single backslash.

'\0' Resets the text pacing mode (see Section 19.7 [Clipboard panel], page 58) to the settings defaults. Any changes of the text pacing mode affect only the line to be sent. The next following line will be sent using the default text pacing mode.

'\1' Sets the text pacing mode to 'Direct'.

'\2' Sets the text pacing mode to 'Wait for echo'.

'\3' Sets the text pacing mode to 'Wait for any echo'.

'\4' Sets the text pacing mode to 'Wait for line prompt'.

'\5' Sets the text pacing mode to 'Character/line delay'.

'\6' Sets the text pacing mode to 'Keyboard delay'.

'\a' Executes an ARexx command (all text to follow this character).

'\b' Generates a backspace (deletes the character to the left of the cursor).

'\c' Calls a main menu entry, the menu entry to be called is determined by the argument to follow; this is either a six digit number (example: '\c 010203' would call subitem 1, item 2, menu 3) or the name of the menu entry enclosed in single quotes to call (example: '\c 'about'' would call the 'About...' menu entry, the search is case-insensitive and only compares the characters given).

'\d' Executes an AmigaDOS command (all text to follow this character).

'\e' Generates the escape character (ASCII code 27).

'\f' Generates a form feed (skip to beginning of the next page or clear the screen).

'\g' Places the text to follow this character in the clipboard.

'\h' Appends the text to follow this character to the current clipboard contents.

'\i' Feeds the contents of the clipboard into the input stream.

'\n' Generates a line feed.

'\p' Feeds the password of the currently active telephone book entry into the input stream. *The password is automatically cleared for security reasons when the connection is lost.*

'\r' Generates a carriage return.

'\t' Generates a tab jump.

'\u' Similar to the \p command, the \u command will feed the current user name into the input stream.

'\x' Generates a break signal (as with the 'Send break' menu entry).

'\^' Generates a caret character.

'\~' Generates a tilde character.

'*' The code to follow the asterisk determines the character to produce. This can be any three digit number or a symbolic name from the following list[1]:

'NUL', 'SOH', 'STX', 'ETX', 'EOT', 'ENQ', 'ACK', 'BEL', 'BS', 'HT', 'LF', 'VT', 'FF', 'CR', 'SO', 'SI', 'DLE', 'DC1', 'DC2', 'DC3', 'DC4', 'NAK', 'SYN', 'ETB', 'CAN', 'EM', 'SUB', 'ESC', 'FS', 'GS', 'RS', 'US', 'SP', 'DEL', 'SS2', 'SS3', 'DCS', 'CSI', 'ST', 'OSC', 'PM', 'APC', 'NBS' and 'SHY'

If none of the mentioned combinations is recognized the character which follows the \ will be fed into the input stream without any changes.

25.2 Caret

This character is used to change the following character to a 'control character'. So the sequence '^J' will become a Line feed and '^I' becomes a tab jump. The character which follows the ^ has to be located between @ and [, otherwise it is fed into the input stream without changes.

25.3 Tilde

This character causes the program to pause for exactly half a second before it continues to process the following commands.

[1] 'EOU' may be implemented in a future release

26 Fast! macros

In implementation and design the fast! macros are closely related to the function key macros (see Section 19.15 [Function key panel], page 71). If invoked by selecting the corresponding menu entry, a window will open on the right hand side of the screen sporting a scrollable list of macros (the contents of this list can be edited using the fast! macro panel). When a list entry is selected, the associated command sequence (see Chapter 25 [Command sequences], page 101) will be executed.

By using the fast! macros it is theoretically possible to control a BBS just by mouse, provided that you have the approriate macros in your fast! macro list.

The fast! macro panel can be resized and acts just like the main ‘term’ window: menu items can be selected and characters entered are sent to the serial driver.

27 Packet window

In this window a line can be edited before it is sent. All the usual editing functions known from standard input fields are available (`Shift + cursor left/right` jumps to the start/end of the line).

Additionally, some extended functions exist which are performed by pressing a cursor key together with the `Shift` or `Control` key:

‘`Control + Cursor left`’
: Jumps to the next word.

‘`Control + Cursor right`’
: Jumps to the previous word.

‘`Cursor up`’
: Shows the last entered command in the input line.

‘`Shift + Cursor up`’
: Shows the very first command entered so far.

‘`Cursor down`’
: Shows the next entered command (if you moved back for some commands before).

‘`Shift + Cursor down`’
: Shows the very last command entered so far.

This text gadget has a buffer where all previously entered commands are stored (‘Command history’). You can page through this buffer, load and save it and individual lines can be recalled. As with the text buffer this buffer is managed dynamically. The same memory restrictions that apply to the text buffer are valid for this buffer.

The input line also has a menu which offers the following functions:

‘`Load history`’
: Loads the contents of the input line buffer from a file. Each stored line in this file can be recalled and sent.

‘`Save history as...`’
: Saves the contents of the input line buffer to a file.

‘`Clear history`’
: Simply releases all previously stored commands and the memory used by them.

‘`Other window`’
: Switches to the main screen of ‘term’.

'`Show output`'
: If not enabled, this causes the input line not to be echoed in the terminal window.

'`Quit`' Closes the window (corresponds to clicking the close gadget of the window).

Every character entered into this window is shown immediately so that those things where it is better that they should not appear on the screen (like passwords for a mailbox) should be entered in another way.

The contents of every input line are interpreted as a command sequence and therefore can also contain control characters.

If a line taken from the input buffer is sent without change it is *not* stored in the buffer again ('true history' as known from 'ConMan').

The contents of the input buffer are cleared automatically after the window is closed. *Under no circumstances are the contents maintained until the next call!*

Provided that the packet window is large enough, a list to contain the command line history will be displayed.

28 Script recording

'term' offers a feature called 'Script recording' which lets you record incoming data sent by a BBS or a remote host and your response to it, i.e. the text you typed, such as login name and password. The recorded data can then be saved to an ARexx script file which can be used as an auto-login script. In order to record a script you can either use the dialing panel button labeled '`Start recording on connection`' or the menu item '`Record`'.

Once 'term' is recording terminal output and your input the status display will show '`Recording`' or '`Rec.line`', depending on the text entry mode. By default 'term' will only record single keystrokes, which makes it difficult to enter whole words. If you want 'term' to remember the entire line of text you are about to enter either use the '`Record line`' menu item or press the `shift+return` key combination: the status will change to '`Rec.line`'. To return to keystroke recording just press the `return` key or select the '`Record line`' menu item/press `shift+return` again.

'term' only remembers the last ten characters sent and a maximum of 256 characters you can enter per line. If you enter more than this number of characters older keystrokes will be discarded.

When you are finished recording the script select the '`Record line`' menu item. A file requester will ask you for the file name to save the script under. If the file is successfully saved you may be asked whether you want the script file to be used as a login script for the currently active phone book entry.

The script file generated will consist of ARexx commands 'term' understands, mostly `TIMEOUT`, `WAIT` and `SEND`. The text to be waited for and to be sent is given in standard 'term' command sequence notation. For more information consult the chapter entitled Chapter 25 [Command sequences], page 101. The 'term' ARexx interface documentation provides the necessary background to explain how the script commands work, it should also give you hints how to customize the recorded scripts.

Caution: scripts recorded by 'term' usually need additional editing, don't expect a script to work right away. You may want to change the timeout values, remove extra characters and input.

29 Environment variables

Information which is to be available the next time the program is run is placed as AmigaDOS variables in the directories '`ENV:`' and '`ENVARC:`' by 'term'.

The variables used by 'term' can be used and manipulated by other programs transparently. In detail these variables are:

'`TERMCONFIGPATH`'
: The name of the directory in which all information used by 'term' is placed (standard configuration, phonebook, etc.). The default settings path name is '`TERM:config`'.

'`TERMWINDOW`'
: The window definition which can also be entered in the program via the menu item Section 18.8 [Settings], page 41.

'`xpr...`'
: The standard settings used for the corresponding transfer protocol ('`xprzmodem`', '`xprkermit`', etc.).

'`xem...`'
: The standard settings used for the corresponding external terminal emulation library ('`xemvt340`', '`xemascii`', etc.).

30 PGP key

Below you will find my signed public key. Save it to a disk file and enter PGP <file name> to decode it. This will produce a file called 'public_key'. To add my key to your PGP keyring now enter PGP -ka public_key and follow the instructions on the screen. To verify my signature, now enter PGP <file name> again. It's probably pretty paranoid to rely upon PGP signatures and keys, but then again you might want to have a somewhat unambiguous proof that the distribution archives you have downloaded are intact. Security can still be compromised, it you don't trust the key below you can still contact me to ask for an official key.

Note: this document does not include the public key, look for an ASCII text file, such as 'term.doc' or 'term.guide' instead.

31 Revision history

Changes introduced with v4.0:

- The window status line is no longer ‘misplaced’ under Kickstart v2.x.
- Font, file and screen requesters are no longer quite so tiny.
- Different font scales as by the VT100 specs (half width, double width, top double size, bottom double size) work again.
- All memory allocations now go through memory pools, reducing memory fragmentation greatly. Note: Kickstart 3.x owners *should* have SetPatch 40.16 installed as the memory pool code might have trouble freeing empty memory pools until the program exits. This memory allocation scheme should also help to get ‘term’ to work with certain virtual memory system extensions.
- In the XPR transfer window the error/message list will no longer hold more than 100 entries in order to save memory. I have received reports of users who ran large file transfers overnight and when getting up in the next morning the transfer error/message list had accumulated so much memory it was no longer possible to move the mouse: Intuition was unable to allocate enough memory to create new input events. If 100 messages have accumulated and a new one is about to be added the first and oldest entry will be removed.
- The status line display now properly reflects the name of the current file transfer protocol.
- The quick dial menu checkmarks would get cleared only on some rare occasions, i.e. if the first phonebook entry had the ‘Quick dial menu’ feature set. Now it gets the job done no matter which phonebook entry is the first one in the quick dial menu.
- The main window menus used to have the command shortcut ‘W’ assigned twice.
- If running on the Workbench dropping icons on the ‘term’ window would ask for the type of file transfer (text or binary), but it would get the selection wrong, i.e. if you chose text you got a binary upload and the other way round. This has been fixed.
- When in zoomed state the XPR transfer window will display the name of the file currently being transferred and how much of it has already been transferred (if available). This display will be updated about once a second.
- If icons are to be created for files downloaded it is no longer necessary to turn on file type identification to actually get the icons attached.
- Freezing the text buffer contents now properly updates the text processing routine variables.
- Calling the ‘Print clipboard’ function twice will no longer result in a general system lockup.
- The double-buffered I/O routines now let you configure the buffer size to use. The memory allocation also is a lot more ‘forgiving’ than it used to be: if necessary it will shrink the buffer size until it can allocate enough space.

- To keep naughty applications from switching the cursor key and numeric keymap into applications mode you can lock both key sets now, so they will not to change their current modes.
- The old AmigaGuide release (v34 to be accurate) is supported now, but with limited functionality. The help text is not context sensitive and you need to shut down the AmigaGuide server manually (by closing the AmigaGuide window) if the screen its window resides upon is to be closed.
- 'RING' and 'CONNECT' messages from the modem are now reported along with the time when they came in.
- The program now complains loudly about outdated catalog files and incorrectly installed locale.library.
- The hotkey settings panel now checks each hotkey description text after it is entered and complains if it is unuseable.
- The 'QUIET' command line option (makes 'term' start up iconified) no longer crashes if Workbench isn't running.
- The program now features an all-new user interface.
- There is a new command line option called 'BEHIND' which causes the main screen to stay in the background and the main window not to become active upon startup.
- If 'term' fails to open a screen and finds out that the requested screen display mode is unavailable it will copy the screen mode the default public screen is in and retry.
- No more trouble with mixed-case device and library names. As you pick them from the list the files are validated, i.e. 'term' tries to load the file in question and hunts for the library/device resident tag included. If the tag is found the 'real' device/library name is copied from it, replacing the original name the file was opened with. This means that you can select 'XPRZModem.Library' using the file requester and 'term' will look into the file to find out that the library wants to be opened under the name of 'xprzmodem.library'. Also included are a type check (i.e. if a library is to be opened only files with a library type resident tag are included in the list) and a brief name comparison (i.e. only name case differences are allowed, so 'XPRZModem.Library' = 'xprzmodem.library', but 'foo.device' != 'bar.device').
- The external emulation and the external protocol support routines no longer share the same code and the same set of error messages/options texts.
- A transfer protocol such as xprkermit.library will no longer leave 'term' in a 'half-dead' state if it opens a new window in the protocol setup phase. In previous releases this window was never closed.
- The label text of XPR/XEM command options which accept a parameter and thus cause the settings window to be closed after text is entered is now drawn in the current highlight colour. This helps to distinguish regular string gadgets and command option parameters.
- If 'term' fails to set up the XPR protocol properly, i.e. the setup routine does not flag success, the library is closed right away. This is how it has always been in previous 'term' releases. But this time the main menu is also updated to keep you from starting a file transfer or changing the transfer options while the library base pointer is invalid.
- In order to support external protocols which write data to or read data from the serial line serial I/O processing is temporarily disabled while the corresponding setup routines are running. Although

this behaviour isn't quite that nice it should avoid serious trouble with xprkermit.library which could otherwise disrupt the serial device request queue.

- There are now three buttons in the file transfer panel which correspond to different abort levels:

  * Skip current file Skips just the file currently being in transfer (level 2)

  * Abort transfer batch Cancels the entire batch transfer (level 1)

  * Abort entire transfer Emergency abort (level -1)

  These abort levels are supported by xprkermit.library and a number of other file transfer protocols.
- Opening the transfer protocol settings editor will cause the currently selected default XPR protocol to be reopened in case it is not open yet.
- Holding down the control key and clicking with the mouse on a space character will now correctly send it.
- Double-clicking on a phonebook entry will no longer start dialing if the program is still online, i.e. if the regular 'Dial' button is disabled.
- If the line is hung up or the carrier is lost the call log file (human readable, not the one that is intended for postprocessing using call log analyzers) will include the costs for the call. In previous releases 'term' would only look for the 'NO CARRIER' message.
- The destructive backspace option now removes the character to the left of the cursor but does not move the rest of the line one step to the left. This makes the terminal emulation behaviour more consistent with how real video terminals handle this job.
- The translation table settings panel now features a 'Default' button which resets the current translation settings to defaults.
- Just like the text buffer screen the review buffer window is handled by a coprocess now.
- The packet window now supports function keys.
- For technical reasons I dropped the string gadget clipboard support. I suggest that you use StringClip or a similar program instead.
- Clipboard pasting from the review buffer window no longer requires that you activate the main window.
- In the phonebook the edit list (right hand side listview display) indicates whether an entry uses default settings or whether it actually uses custom settings. Custom settings are indicated by asterisks ('*').
- The packet window now uses the current terminal text font.
- Pressing Amiga+- in the packet window no longer transmits the current string but changes to the main window while leaving the current string intact.
- Fixed a potential bug in the generic list management module.
- With some file requester patches installed, such as old ReqTools or MFR, the file requester code could fail to notice if a single file was selected in multiselect mode.
- The phone rates are now reported according to the current locale settings, i.e. they take the grouping and special attributes of the local currency into account.

- The IFF-8SVX sound file player code now handles stereo and compressed sound files gracefully (all flavours including uncompressed stereo, compressed stereo, uncompressed mono and compressed mono).
- The ARexx 'WAIT' command could fail to report how many characters it pulled from the data stream when a matching string was found. This could result in random characters showing up in the terminal text output.
- The routine to attach the wait mouse pointer to windows and to block input to them was easily losing track of its nesting count. In theory the nesting count could have wrapped around, locking you out. The harmless side-effect was that sometimes windows would not get blocked.
- The ARexx 'WAIT' command argument and the wait list may include control sequences now. These are expanded as soon as they go into the list. This feature makes it possible to wait for sequences such as 'login:\rpassword\r' but will return result strings which include control characters, so watch out!
- The ARexx 'READ' command now allows you to combine the 'CR' and 'NUM' options. Also, the maximum number of characters to read with the 'CR' option is no longer limited to 255 characters.
- The file transfer window will at startup display the name of the currently selected transfer protocol. It displays a default value, leaving the protocol identification to the XPR library.
- The ARexx 'GETATTR' command would run into serious trouble if told to put information into a stem variable. The result would be Enforcer hits or crashes.
- If you put 'term' into iconified state and press the hotkey combination to bring its screen to the front you will no longer get an Enforcer hit. Instead 'term' will exit its iconified state and return to normal action.
- The status line display now coexists much nicer with MagicMenu and the like.
- The status line no longer displays what text mode the terminal window is in (this was rather a silly feature) but rather if the text buffer is currently recording or if it's frozen.
- The status line looks a bit different now (there is a proper separation bar now instead of the hair line).
- Just like the v3.0 preferences palette editor 'term' now permits to select the screen rendering pens. This will *not* work under v2.04 since gadtools.library v37 cannot handle it (actually it can, but the default glyphs do not support it). This feature permits you to change the screen colours while you can still keep the window new look.
- The ARexx 'READ' command could return random characters since the return buffer was not set up correctly.
- Both the ARexx 'READ' and 'WAIT' commands now allow you type text on the local console and have it sent across the line. As for the 'WAIT' command this is of great help if the script 'hangs'. Note: halting scripts and and such still requires pressing the 'Shift + Shift + Esc' key combination.
- The ARexx 'READ' command would, if used with the 'NUM' option, always return a NULL-terminated string, no matter what kind of data came in. It now returns the entire amount of data transferred, including NULL-bytes.

- The phonebook list now clearly shows which entries are selected for dialing and which are not (it flips the background and text colours). Note: this works only with Kickstart 3.0 and above.
- I increased the possible number of dial retries in the modem panel to 1000. In addition to that you can set the number of dial retries to 'unlimited' now.
- For those nasty MS-DOS based file transfer protocols which choke on file names longer than 12 characters (eight for the name, one for the dot, three for the extension) the file transfer options now permit to have filenames shrunken before they are handed to the transfer protocol. Internally, the XPR interface will still refer to the file under its original name. The routine responsible for shrinking the file names also takes care of the extension separator dot. If there is more than one dot in the file name all the others get replaced by underscore characters. Also, if there is no dot extension it will be added.
- The packet window is handled by a coprocess now.
- If you're bold and daring you can make 'term' handle the terminal output on the schedule of a terminal emulation task. Please note that this requires additional memory and will slow down the emulation if your memory is fragmented. But on the other hand the coprocess will stop serial input getting munged before it arrives in the terminal emulation output buffer. Such things can happen with systems which experience heavy DMA bus or task loading
- Turning on the 'faster layout feature' no longer drops the screen & window newlook.
- Due to an oversight the terminal emulation process could get enabled even when using an external emulation, this has been fixed.
- The ARexx command 'SENDFILE' would never remove the names of files transferred from the upload list.
- Yet another new option: by default the dialer sends a '\r' string when skipping an entry and when hanging up the line. You can change this behaviour via the 'Dialer abort hangs up' option now. If enabled the dialer will go through the routine hang up procedure (dropping the DTR signal, sending the hangup string, you name it). This should convince even the most stubborn modems to stop doing what they are currently thinking to be fun and to return to normal operation.
- Clicking on the main window in order to activate it will no longer trigger the character snapping function.
- Rewrote the ARexx 'WAIT' routine to a great deal. It could easily forget to turn serial input processing back on for the main program. After a script would exit you would get stuck with data coming in from the serial line, but none of it would be displayed or worked upon.
- Clicking on the text buffer screen window in order to activate it will no longer trigger the character snapping function. Since there is no safety catch, i.e. the clipboard contents are immediately replaced by what you selected after you let go of the mouse button, this will reduce the chance of losing your current clipboard contents.
- Subtle change in the dialer procedure: if an entry would use the default serial settings they did not replace the current serial settings, even if the previous dial list entry had altered them. However, the original purpose of the default settings was to use the unmodified global settings. The dialer behaviour

now respects this, changing the current serial settings back to the global settings, not keeping the changes the previous dial list entry had made.

- In previous program releases trying to make certain phone book entries not use the default settings was somewhat difficult: you had to change the corresponding settings entries to something different from the global defaults. Things are much easier now, just open the settings editor and click on the 'Use' button.
- The 'Startup/Login macro' has been split into a startup macro and a login macro. The dialing routine will first invoke the login macro and then the startup macro. Only the dialing routine makes use of the login macro.
- Small cosmetic changes to the user interface code: cycle gadgets are a few pixels wider now in order to keep 'CycleToMenu' happy.
- When hanging up the line the logoff macro was never executed.
- Finally discovered why the serial read quantum and all the scheduled events were never processed again once they reported that no further data was available. Now the event response loop updates the signal mask again when it reaches the bottom of the loop to see if any new data came in.
- You can now configure the screen depth, permitting to use the Picasso II chunky display mode with 'term'. Note: asl.library v38-v40 fails to handle nonstandard background pen colours correctly, i.e. the depth slider text may be illegible.
- External emulation libraries, namely xemvt340.library, should work again. I changed the memory allocation call for the XEM support interface, but I have no idea why it did the trick.
- The status window is handled by a coprocess now.
- Some ARexx interface commands now run asynchronously.
- Yet another visual gimmick (sorry, couldn't resist): menu checkmarks and Amiga keys are now scaled according to the current screen display ratio. Note: not really compatible with utilities such as MagicMenu or Silicon Menus. Although the programs will run the menu layout may look odd.
- The phonebook list can be scrolled with cursor keys now. You also get a visual feedback if running under Kickstart 3.x.
- Fixed a nasty bug in the user interface code to pick the gadget shortcuts: it would prefer to pick the last letter of gadget labels and ignore any preceding letters.
- Scrolling lists such as the phonebook list will size-adapt to the screen 'real estate' available.
- The screen settings now give you full access over display overscan mode and screen dimensions. Note that asl.library v38 or higher is required to use these features.
- Changed the cursor key control in listviews, making it possible to use the Shift/Alt/Control qualifier keys in Style Guide compatible fashion.
- Rewrote certain dos.library related parts of the user interface code, permitting plain tasks to call the routines. Consequently, a number of coprocess-driven routines was rewritten to run on the schedule of a task. Note: this may conflict with the ChangeScreen utility included in the Picasso II distribution.

- Old style XPR settings entered using the old style prefs interface (i.e. those that would pop up a text entry requester) were never saved.
- The transfer library selection panel now allows you change the settings of all selected protocols, you no longer need to select all individual protocols as the default protocol and invoke the 'Protocol settings...' menu function on them.
- The text and review buffer search requesters maintain backlogs of the previous search patterns now (use the cursor keys to scroll through the patterns). Use the capture settings to change the number of patterns to keep.
- The text buffer search function no longer enforces a case-insensitive search.
- Most time and date displays now use the current locale settings. It did not make sense to make all such displays use this text formatting scheme. All remaining displays will use the common dos date/time formatting parameters.
- Modified OwnDevUnit.library support: if another task wants to gain access to the locked device 'term' will release it unless the modem is still online. The device is released by calling the 'Release serial device...' menu entry.
- Oops... the XPR options editing code did not flag changes in numeric arguments to the main program. Also, old style XPR options were always reported as 'changed'.
- Not all memory allocations went through memory pools, this has been fixed.
- The 'Printer control enabled' switch was omitted from the emulation panel, sorry about that.
- Some windows would not be moved into the foreground when opened, so you had to play 'hide and seek' to see on which screen they appeared.
- With display aspect ratios that aren't even remotely square incrementer arrows for integer gadgets are no longer larger than their container boxes.
- The program no longer crashes if the XPR protocol feels about displaying a message before any other window is open.
- By public demand the highlighting scheme in the phone book window was changed to yield better contrast.
- The speech volume is given in percent now, but the ARexx interface 'GETATTR' command did not reflect this.
- The sound settings now sport a volume slider which affects all sounds played. Suppose a sound is to be played at maximum volume, i.e. 64 for the current Amiga hardware and the volume slider is set to 25%, then the sound will be played at volume level 16. Setting the volume slider to 0 will cause 'term' not to produce any sound.
- The transfer panel message list dimensions are now auto-adjusted to the screen size. The layout code also tries hard not to make the window overlap the status line.
- Added some bells & whistles to the phone book window. I hope it still works.
- Moved the user interface code into a shared library in order to make things more complicated.

- The OwnDevUnit feature is no longer linked to the 'Shared access' option.
- All sounds are now replayed in the same fashion as the bell sound, i.e. if a sound is currently being played a request to play another sound will not be satisfied.
- Important file transfer notification messages are now printed in the current highlight colour.
- The destructive backspace mode now offers three choices: off, overstrike and shift. Overstrike mode clears the character below the cursor and shift mode will shift the line contents to follow the cursor to the left.
- The screen panel now features some more options which permit to open the main window on a custom screen as though it were a public screen and an option to split the status line from the main window.
- Tried to squeeze some space out of the control panels with mixed results.
- The text buffer screen no longer uses the main screen display mode by default, it is possible to select the display mode now.
- The built-in ASCII file transfer routines no longer draw upon the current clipboard settings to determine how to send and receive text. There is an all-new preferences editor for this purpose now.
- Added an upload list editor, permitting to collect the files to be transmitted in a list before the transfer is started. You can enter the file names, drop icons on the editor window or on the AppIcon. When you are finished, just press the upload button.
- The packet window string gadget now has room for more than 1000 characters. Note: the 'Load history' command only supports 255 characters per line.
- Added a prescroll option to the emulation settings. The system is not very smart but should get the job done rather quickly; if the cursor is positioned on the last terminal line the number of line feeds in the input data stream is counted. The number of line feeds or the max.prescroll number (whatever is smaller) will determine how many lines to scroll the screen contents up.
- With an empty phonebook loading a new phonebook file would not enable the phonebook list, this has been fixed.
- Added another option to complement the prescroll settings. Testing revealed that the conditions leading to the prescroll feature to be used were met only rarely. The 'max. jump' option will give you roughly the same functionality as the 'max. prescroll' option, the difference is in the handling of the input data stream: the prescroll option counts the number of line feeds, the jump option only looks for a single line feed and then scrolls up the screen the given number of lines.
- The program now remembers the window position and size before going into iconified state. When the window is reopened it will reappear at the position and in the size remembered
- The ARexx `ADDITEM` command supports a new keyword `RESPONSE` to be used for entries added to the wait list. Suppose you want all `More (Y/n)?` prompts of the BBS login procedure to be skipped during your auto-login script. Then you would call `ADDITEM TO wait NAME "More (Y/n)?" RESPONSE "\r"`, add the other keywords you would want to wait for and then finally call `WAIT`. Whenever the `More (Y/n)` prompt shows up the `WAIT` command will all by itself send the `\r` string (or in other words, the carriage return character) without exiting until the login string it was watching

for is found. Please note that this feature only makes sense with the wait list but not with the `WAIT` command itself.

- Added login learn mode controls. If the option is enabled in the dial panel incoming text and your responses to it will be recorded. When you are finished with your usual login procedure, call the 'Record script' menu item in order to save the recorded data to an ARexx script. By default the recorder only watches for single keystrokes. If you wish to enter an entire line use the 'Record line' menu entry, calling this item a second time will transfer the line you entered into the recording buffer. Pressing the return key has the same effect. Pressing Shift+Return will also put you into line recording mode, no need to use the cryptic menu shortcut. Note: the 'term' main menu now uses all available printable 7-bit ASCII characters for keyboard shortcuts :-(
- Reworked the error reporting scheme, one of the weak spots of 'term'. In previous releases you would be told that a file could not be saved or loaded, but no explanation would be given why the action failed. Where possible 'term' will now tell you the cause of the error.
- The printer support code was not particularly sensible to printer trouble, such as reported by the infamous printer.device requester. Even if this requester was cancelled 'term' would happily continue to print, causing the printer trouble requester to pop up over and over again. This has been fixed.
- The ARexx `DIAL` command is no longer synchronous by default, there is a new `SYNC` option which causes the ARexx script to wait until the dialer has made a connection/has failed to make a connection. The `RC` (result code variable) will be set to 5 if no connection was made, it will be 0 if a connection was made. Now control can pass right through the `DIAL` command.
- ARexx support is no longer a compile-time option but an integral part of the program.
- Fixed a couple of bugs in the user interface library which dealt with odd alignment of slider level strings under v39. Some few strings still break the rules, but the results are less devastating than they used to be. Thankfully, most of the gadtools.library bugs that caused such trouble were fixed in v40.
- Added the 'trap' feature which implements the functionality of the ARexx wait list in an asynchronous fashion. For each sequence found a command sequence is executed. This should make it much easier to write UUCP scripts, BBS programs and such. Please note that the trap list is not identical to the wait list.
- Rewrote the ARexx/AmigaDOS command execution routines which now execute their commands in truly asynchronous fashion, i.e. once their processes are spawned they keep to themselves and notify the main program when they are finished. This made it possible to remove the odd command handling workarounds and the special ARexx interface code that would take care of ARexx/AmigaDOS command execution.
- The upload queue window code did not get the upload buttons enabled if icons were dropped on it.
- The trap list loading code mixed up the order of commands and sequences, this has been fixed.
- Fixed a monetary quantity formatting bug in the locale support code.
- A malformed text formatting string was causing Enforcer hits in the action logging routine.
- Added a few more 'safety catches' to the screen opening code in order to make sure it does not open screens 'too deep'.

- Changing the screen size in the screen panel did not cause the screen to be reopened, this has been fixed.
- Realized that it was not such a good idea to make most windows simple refresh and changed them to smart refresh. I've probably seen too many Macintosh programs forced to redraw their window contents over and over again.
- The auto-upload panel now sports a button labeled 'Upload from queue' which if enabled will cause the current upload queue to be transferred. Thanks to Bob Maple for pointing me into the right direction.
- Fixed a truly nasty bug in the user interface code: integer type gadgets did not remember their original values when created, they would forget about them when queried and only would report them correctly when the values were changed. There is something I forgot to mention: integer gadgets understand hexadecimal (both 0x.. and $.. notation), binary (%..) and octal (&..) notation. The idea came from Martin Taillefer.
- The date and time entries in log files are now printed in DOS format again.
- Recorded scripts and such no longer include `\*SP` codes instead of plain spaces. This should improve readability.
- With the 'shared screen' feature enabled 'term' would not respect the screen depth settings. This has been fixed.
- The XPR routines no longer queue more than one IORequest, this should help to avoid trouble with some few device drivers.
- The ARexx `QUIT` command really works now.
- Removed the fixed-width font dependencies for Kickstart 2.04. Although windows may look funny if you use a proportional-spaced font, the program should deliver the same functionality as if it were using a fixed-width font.
- Triggering the iconification function with the 'Release serial device when iconified' option enabled will ask you for confirmation if the modem is still online.
- Just for the fun of it added datatypes support. The IFF-ILBM saving routines of picture.datatype are used when saving the window contents to a file.
- If the console output window specifier includes the `%s` string formatting parameter it will be replaced with the name of the public screen 'term' resides upon.
- New ARexx command `PROCESSIO` permits to turn off serial I/O processing by the main program. This will let ARexx programs receive & process all incoming data, without having 'term' pull single strings from the input data stream.
- Keymap strings to include null-bytes should be properly processed now, previous 'term'~releases would stop at the null-bytes and ignore the rest of the input lines.
- Changes in the serial buffer size now cause the serial driver data to be updated as well.
- Added facilities for external programs to rendezvous with 'term', allowing them to take over the serial driver. This was added primarily to support my HydraCom Amiga port (please note that you need

revision 2 of my HydraCom port to interface to ‘term’), but it may also help to support other file transfer protocols, such as the original rz/sz programs.

- Whether trap list processing is enabled or not is now saved along with the trap list itself.
- You can now lock the cursor keys and the numeric keypad separately.

Changes introduced with v3.4:

- Raised the maximum number of quick dial menu entries to 50.
- Fixed a small bug connected with the text display screen.
- XPR protocols which do not support batch transfers now work again as they did in v2.4.
- Corrected a spelling mistake in the call logfile creation routine.
- Fixed a memory allocation error in the review buffer code.
- Fixed a major bug in the file transfer routines. In previous releases (say 3.1 and above) ‘term’ used to throw invalid file locks around (seems I will have to use less ambiguous identifier names in the future...).
- Fixed an Enforcer hit in the quick-dial menu creation.
- Rarely, the program could fail to rebuild the quick-dial menu after leaving the phonebook.
- As required by the specs, the XPR interface now switches baud rates and serial parameters.
- Enhanced the capture functions.
- Changed the user interface layout for some windows, such as the modem settings window.
- When running on a public screen will no longer cause lockups and trouble with Workbench. Rendering errors are still possible.
- The very first characters to follow a modem connect message are no longer quietly discarded.
- Finally solved the truly mysterious case of the missing screen line.
- Fixed yet another Enforcer hit in the packet window code.
- The standard beep routine would not work.
- Libraries and devices can now be selected using simple menus rather than the file requester.
- The program now flushes capture files to disk every minute in order to insure that at least fragments of the captured session will be safe when disaster strikes.
- Finally got the cross-hatch pattern to draw ghosted list views right.
- The terminal hex mode would write too many bytes per line.
- The program would not pay attention to any icon tool type settings.
- Upon making a connection the dialing routine will no longer drop the line by accident.
- The review buffer window will filter out unprintable characters before displaying text lines.

- The program no longer knocks itself out if an initialization error occurs during the startup phase.
- You can now select the name of the public screen the 'term' window is to be opened upon using a menu.
- If using a public screen the program will make use of the corresponding screen font and no longer try to modify it in any weird way.
- The ARexx command `READ` did not pay attention to the translation table settings.
- A configuration file to be loaded upon startup as specified using tooltypes or command line options will no longer be ignored.
- The positions and sizes of certain program windows will be stored in the main configuration file.
- Initially, when a file was received the information window would not display the space left on the output device.
- Introduced the 'time to connect' interval.
- Added the serial read quantum.
- The phone book window will now be opened large enough to hold all its gadgets inside.
- Blinking text display is no longer restricted to eight colour screens, provided the display architecture permits high-resolution screens in more than 4 bit planes.
- The program now exists cleanly if vital resources such as the output screen cannot be opened.
- The file transfer routines should run a tiny little bit faster now.
- Added the transfer performance meter.
- The clear screen control sequence now optionally resets the cursor position.
- Rarely, the text buffer screen would not update the first text line properly. Also, the screen display mode will no longer fall back to HIRES/HIRES_INTERLACED.
- Rarely, the review buffer window would render text in the wrong colours.
- The pen/palette sharing code was submitting incorrect colour codes.
- The review buffer window now responds to the same keypresses as the text buffer screen and the main input window.
- Not all packet menu items were to be invoked using menu shortcuts.
- Added 'Completion time' display to the transfer progress window.
- Introduced special transfer library settings. You can now select which library to use for ASCII, text or binary transfers.
- Added built-in ASCII transfer routines.
- You can now search forwards or backwards through the text buffer.
- Text stored in the buffer no longer gets trailing spaces stripped in order to keep weird uuencoded data intact.
- The menu items featured in the 'Edit' menu now properly reflect the state of the clipboard.
- A menu shortcut would be used twice in the main window menus.

- Rarely, the colour choices for the bar charts in the transfer progress window would be wrong.
- The quick dial menu would not be rebuilt if the phonebook was sorted or new entries were added.
- The ARexx commands `DELAY` and `SEND` would not turn off the text cursor prior to text processing.
- Unless it is absolutely necessary, the text display scope is no longer moved automatically.
- As for the text display screen and window the 'Alt' keys now have the same effect as the 'Control' key.
- If new text is added to the text buffer the buffer screen and the review window are properly updated now.
- A dialing list built using the quick dial menu was not enumerated properly.
- Building a dialing list using the quick dial menu via extended selection, then selecting the phonebook control panel, clearing the dialing list and leaving the phonebook will no longer start dialing the list just cleared.
- Checkmarks now indicate which entries in the quick dial menu are to be dialed.
- The bar charts displayed in the transfer progress window are now updated in a more efficient manner (the text colour has changed, too). Also, unzooming the window will properly update the bar charts now.
- A hardware buffer overrun error is no longer considered a fatal error.
- With some requesters and windows string gadgets are auto-activated now. Pressing the `Return` key will cycle through all the available strings gadgets, holding down either `Shift` key will break the cycle.
- Trying to quit the program will no longer ask for confirmation if there is no reasons to ask for.
- The bar charts in the transfer progress window will be omitted if the remaining screen space would not permit the entire window to be displayed.
- The fast! macro window contents would be rendered in the wrong colours.
- The phonebook window did not support any online-help features.

Changes introduced with v3.3:

- Added the quick dialing menu.
- Enhanced the modem and screen settings
- Fixed a few bugs in the ARexx interface commands `REQUESTFILE` and `REQUESTRESPONSE`
- Unfortunately a single line was missing in the XPR option setup, causing them program not to save the transfer protocol options.
- No longer forces the creation date of an auto-capture file to be included in the file. As an option, will use the naming convention used in previous program releases.
- The program now optionally creates icons for files.

- Added a terminal hex-mode for debugging purposes
- Fonts are now opened using properly initialized DPI and aspect ratio values which results in much better scaled outline font rendering.
- The name of an external emulation library is no longer quietly suppressed.
- It is possible to disable the double-buffered file management routines now.
- Added the text pacing option.
- Apparently, external and internal terminal emulations tried to turn on/off each other's cursors.
- Copying text to the clipboard now permits to append the text to the current clipboard contents.

Changes introduced with v3.2:

- Could not enter phonebook passwords
- The XPR interface now states much more clearly what the likely cause of a serial I/O error might have been
- New sound support functions added
- Added 'OK' and 'ERROR' modem response codes
- The order of messages displayed in the file transfer window has changed (now works top down rather than bottom up).
- The ARexx interface routines did not check whether the serial device handles were available or not (boom!)
- Redid the fast! macro handling
- The text buffer screen did not notice when the buffer contents were cleared
- Retuned the phone book and dial list routines which could generate Enforcer hits and trash innocent memory
- Boldface/italics characters will no longer leak into the window borders
- The cursor image now reflects the state of the terminal window
- The program would generate an Enforcer hit when files were to be transferred by dropping their icons on the main window
- When opening a capture file the program could ask twice whether an already existing file should be replaced.
- Paste operations triggered from the review buffer window would not start any paste operation until the main window was reselected
- Reworked the character translation table format. Translation table files should be smaller now and should load a lot faster
- The screen/output window size now changes to the number of columns specified in the terminal settings

- Rarely, the file transfer window would be opened very narrow and tall. So tall in fact that display elements would overlap each other, no text could be displayed in the big listview and Enforcer hits were not to be avoided.
- The file transfer interface would produce Enforcer hits when to transfer files using an old-style XPR library.
- The phonebook dial list handling would select single entries when a shift key was pressed during keyboard selection.
- The review buffer window no longer loses track of the text area it ought to display after resizing the window.
- Rewrote much of the keyboard/mouse/peanut butter handling loop. No longer quite so complex.
- Thanks to user persistence the 'Password' text entry field available through the password/user panel no longer hides the actual password text.
- 'Hang up' menu item did not cause a backed-up configuration to be restored.
- Naughty XPR protocols which would issue error reports through xpr_update() during XPRSetup() would cause the transfer window to be opened not to close afterwards.
- No more odd happenings in the status line online time display. The code responsible would misinterprete a string index. This has been fixed.
- It is no longer possible to dial phone book entries which have no name or phone number attached.
- The terminal emulation code no longer leaves background colour artifacts when scrolling text with a background colour other than the default background colour enabled.
- Calls to BeginIO() rather than SendIO() were giving some – if not most – device drivers hard times. According to the few tests I have run so far the program now performs more reliable when doing file transfers. 'term' now works again in conjunction with isdn.device.
- The program no longer encodes the creation date into the name of an auto-capture file but rather writes creation time and date to the first new line in the file.
- New tooltype/command line option 'Quiet' will cause 'term' to start up iconified.

Changes introduced with v3.1:

- Previous releases of the phone book would not clone single phone book entries correctly.
- The configuration copying routine will now selectively copy config information to the currently selected phone book entries.
- The phone book encryption/decryption routines are a tad faster now.
- The program will now prompt twice for a phone book access password.
- Comment, phone number, user name and password can be much longer now than they used to be in previous releases.

- The phone book now only contains the configuration information to differ from the global configuration.
- Previous releases would not save any rate information along with encrypted phone book files, this has been fixed.
- The configuration copying routine now allows to 'drop' parts of the phone book configuration rather than replacing them with parts of the global configuration. This feature works in conjunction with the new phone book file format.
- A new feature has been added: as an option, the program will immediately start to redial the currently configured dialing list as soon as the line is hung up or the carrier is lost.
- The program finally includes context-sensitive online help implemented through amigaguide.library (note: Kickstart 3.0 required). Any window to offer a help text will display it when the 'Help' key is pressed.
- The ARexx interface documentation was entirely rewritten. The main program documentation was updated and slightly enhanced. Be sure to reread it!
- Both the text buffer screen and review buffer window now feature new and improved scrollers. Scrolling and screen refreshes are also quite a bit faster now.
- The text searching routine has been enhanced and now works much faster than in previous releases. It also happens to find multiple occurences of a string in a single line.
- Improved text buffer handling, reduced memory fragmentation if running under control of Kickstart 3.0.
- The text buffer will no longer quietly swallow single characters and will finally correctly wrap lines longer than 80 characters.
- The review buffer window now also sports a search function just like the text buffer screen, other useful menu items have been added as well.
- In previous releases the program would miscalculate the number of bytes left on a filing device when receiving a file. It would not take the number of bytes received into account, this has been fixed.
- In order to transfer files one can simply drag the corresponding icons on the 'term' main window (requires that the 'term' window is opened on the Workbench screen).
- Instead of identifying the type of a file received, the program will optionally attach a file comment to show the name of the BBS the file was received from and the time and date when the file was received.
- In case a file transfer is terminated due to errors the file transfer window will remain open until explicitely closed.
- The XPR interface will disable the xpr_unlink() routine if the 'override transfer path' option is in effect.
- The XPR interface will no longer sort files to be uploaded by size and name as most users found this feature annoying. It now sorts the files by name.
- The file transfer window now correctly displays the names of all files sent and received.
- The file transfer server was removed as it would not lead to a performance gain but a performance loss.
- 'term' now finally also runs as a window on the Workbench or any other public screen (that's what you always wanted, right?). If running under control of Kickstart 3.0 will attempt to share screen colours

with other applications, making it possible to run the terminal emulation in eight or sixteen colours if enough shareable pens are available.

- ‘term’ will refuse to run on a public screen if running under control of Kickstart 2.x and if the screen font happens to be proportional-spaced.
- Some minor and some major bugs in the built-in terminal emulation have been fixed.
- The program distribution now includes the 11 point IBM.font donated by Bernhard F. Muller.
- I was surprised to discover that in previous releases the terminal emulation server would never get activated as the conditions it would be enabled would very rarely be met. Since the file transfer server would actually lead to a performance loss rather than a performance gain I chose to remove the terminal server along with it. Both may be implemented in a future release.
- The terminal emulation should work a tad faster now.
- The program now requires diskfont.library to be installed in order to work correctly.
- The program will ask for a confirmation prior to resetting the serial device driver while the program is still online. Most seasoned JR-Comm had trouble to adjust to the fact that the JR-Comm command ‘send password’ would use the same menu shortcut as the ‘term’¿release serial device’ command.
- The name of the call log file can now be configured completely. Previous releases would use the call file path and use the name ‘term-call.log’.
- The status window now displays the name of the ARexx host port, the BBS name, phone number, comment and current user name.
- In order to confuse and annoy you, the menu layout has been changed once again.
- A new feature has been added: the ‘wait’ menu item will repeatedly send the sequence <Space><Backspace> in order to simulate terminal input.
- The program now requires a bit more memory than before (program size has climbed by about 50 KBytes).
- New command line options and tool types have been added, some have been renamed.
- The program is now much smarter in determining the sizes of the phone book and file transfer window.
- The lists used by the phone book and the date panel no longer look quite so weird if using a proportional-spaced font.
- The main screen text snipping scheme has changed a bit and now resembles the standard console device snipping. Also supported is double-clicking on single words in order to snip them.
- The ARexx interface has been entirely rewritten from scratch.

Index

B

C

G

H

I

K

L

M

Table of Contents