term

A terminal program for Amiga computers

Version 4.4
7 July 1995

by Olaf Barthel

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 phonebook 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.

One of the preferred methods of 'payment' would be to order the 'Olsen Opals' CD-ROM. This CD contains almost every program I have written for the Amiga in the last seven years. The CD 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:

- An old Cinemaware game (*Rocket Ranger*, *Sinbad and the Throne of the Falcon*, *SDI*, *It came from the Desert II*)
- An old game by Activision (*Portal*, *Tass Times in Tonetown*)
- An old Electronic Arts program (*Seven Cities of Gold*, *Marble Madness*, *Deluxe Paint*, *Deluxe Print*, *Deluxe Music Construction Set*, *Deluxe Video*)
- An old Epyx game (*Rogue*)
- An old Rainbird or Firebird game (*Starglider*, *Jewels of Darkness*, *Black Lamp*)
- 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.

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. If you send a check to cover my expenses, please make sure that the fee I have to pay for cashing it (which currently is DM 3,-) does not 'eat up' the mailing costs.

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 all Aminet sites, such as '`ftp.wustl.edu [128.252.135.4]`'. Look into the `pub/aminet/comm/term` directory.

'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 35 [PGP key], page 141).

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.

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.

- As of Kickstart 2.04 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.
- 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 '`Stop entire transfer`', '`Skip current file`' and '`Stop 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 '`Stop transfer batch`' command. All other XPR libraries either treat all buttons the same way, i.e. they stop the transfer no matter which button is pressed, or just pay attention to the '`Stop 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 59).
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 smaller in which interrupt processing may be affected. You can change the default file buffer size in the miscellaneous settings (see Section 19.11 [Miscellaneous panel], page 69). If this still does not have the desired effect, turn on the switch '`Simple file I/O`' (see Section 19.11 [Miscellaneous panel], page 69) and change the buffer size of the file transfer protocol you are using. Start with a small buffer size, such as 4,096 bytes and gradually increase it as long as transfers still work correctly.

'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 49).

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 49).

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 need to be rewritten.

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

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.

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, try to find an 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, v4.3 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`’), 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, Daniel M. Makovec, Dean S. Pemberton, Eric W. Sommer, Florian Hinzmann, Frank Duerring, Gary B. Standen, Gregory A. Chance, Holger Heinrich, Holger Lubitz,

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'`

14 Foreign language support

'term' supports foreign language text catalog tables as introduced with Workbench 2.1 '`locale.library`'. As of this writing there are Danish, Dutch, French, German, Italian, Spanish and Swedish translations 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!*.

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 the minimum 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`

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 'stop' 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.

In control panels featuring a '`Page`' button, pressing the `Tab` key will flip the pages.

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. This screen will usually be larger than the visible region. Move the mouse towards the borders of the screen to reveal more of it.

17 Screen

At the bottom of the 'term'~screen or window a small display shows 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 selected data transfer protocol.

'Emulation'
The currently active terminal emulation mode.

'Rate' The data transfer rate in bits per second (= Baud).

'Params.'
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.

This display will show the online time, the online cost or both (toggled every five seconds) depending on your current settings.

Unless configured different, 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 (as text)/Clipboard`'
: These menu entries are to output plain text on the printer. You can either print the contents of the main screen or the contents of the clipboard.

'`Print Screen (as graphics)`'
: Select this menu entry to make a hardcopy of the screen contents. Printing will respect screen colours and follow the on-screen bit image.

'`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, if this feature is enabled, 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.

the corresponding command sequence (see Chapter 27 [Command sequences], page 125) 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 94).

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

'`Dial phone number`'
: 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'.

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 88).

18.6 Buffer

'`Clear buffer`'
: Clears the contents of the text buffer (see Chapter 25 [Text buffer], page 121). 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 25 [Text buffer], page 121).

'`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.

'`Load 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.

documentation of the chosen transfer protocol for legal options and the values to which they can be set.

As an example, see Section 19.14 [XPR options sample], page 78.

'`Translation tables`'
: See Section 19.15 [Translation panel], page 81.

'`Function keys`'
: See Section 19.16 [Function key panel], page 82.

'`Fast! macros`'
: See Section 20.1 [Fast macro panel], page 85.

'`Hotkeys`'
: See Section 20.2 [Hotkey panel], page 86.

'`Speech`' See Section 20.3 [Speech panel], page 87.

'`Sound`' See Section 20.4 [Sound panel], page 87.

'`Area codes`'
: See Section 20.15 [Area code panel], page 104.

'`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 immediately following the `CONNECT` message.

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'.

For more information on the Upload queue see Section 20.14 [File upload panel], page 103.

18.10 The quick dialing menu

Note: this menu is available only in case any phonebook 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`'
: The transfer speed in bits per second at which 'term' communicates with the serial hardware. The minimum value allowed is 110 baud. You may enter any value you like but keep in mind that not all modems will respond to all possible settings. Clicking on the arrows pointing to the left and to the right will cycle through a set of predefined rates.

  As of this writing the standard Amiga serial hardware is capable of running baud rates from 110 up to and including 1,000,000 bits per second. These are the basic hardware limits, but in practice reliable transfer speeds are far lower at 57,600 baud and below. Custom serial hardware may support higher transfer rates. Do not underestimate the overhead caused by the terminal program & interrupt processing and the effects of connecting cable lengths.

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

  With data transfer speed greater than 2,400 baud most modems require flow control to be used in order to guarantee reliable data transfer. In such cases it is recommended to set the '`Protocol`' switch to '`RTS/CTS`'. If you fail to do so you may lose data.

'`Bits/char`'
: Number of bits per transferred character (7 or 8). As of this writing the Amiga serial hardware does not support all possible bits/char, parity and stop bits combinations.

'`Parity`'
: Serial parity (none, odd, even, mark or space). As of this writing the Amiga serial hardware does not support all possible bits/char, parity and stop bits combinations.

'`Stop bits`'
: Number of stop bits (1 or 2). As of this writing the Amiga serial hardware does not support all possible bits/char, parity and stop bits combinations.

'`Handshaking`'
: Serial handshaking mode (RTS/CTS 7 wire hardware handshaking, RTS/CTS with DSR signal check, none). The `RTS/CTS (DSR)` mode will make 'term' check the 'data set ready' signal first before turning on RTS/CTS handshaking. If no signal is present 'term' will turn off RTS/CTS 7 wire hardware handshaking.

the signal is detected will start the online timer. If offline, 'term' will check the carrier signal periodically and go into online state if the signal is detected.

Note: Be sure to switch the carrier detect check off in case you wish to transfer files using a null-modem cable!

'`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 character, no parity and 1 stop bit.* Do not expect dramatic speed increases.

'`Direct connection`'

You can directly connect the Amiga to other computers, such as by using a nullmodem cable. In such a case it does not make good sense to enforce a carrier signal check or to warn you if the transmission speed requires that you enable RTS/CTS handshaking. This switch disables both features.

'`Quantum`'

The number of bytes 'term' tries to read in one chunk at a time. 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 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.

'`Satisfy requests`'

When a program requests access to the serial device driver, 'term' is supposed to let go of the driver if possible. This switch controls what 'term' will do in this case:

'`Release device`'

The driver is released, just as if you would select the main menu item of the same name.

'`Release device, attempt to reown`'

The device driver is released, but 'term' will retry every four seconds to reopen and thus 'reown' it again.

'`Keep device open`'

'term' always ignores requests to release the device driver.

'`Serial device`'

The name of the serial driver to be used by 'term'. This is usually '`serial.device`' ('`modem0.device`' for the internal Supra modem, '`sxbios.device`' for ASDG's serial IO card, etc.; consult your hardware manual for more information).

'`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.

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

‘‘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.

‘‘Voice’ message’
: The message ‘term’ is expected to receive if the modem detects a voice call. If in dialing mode, ‘term’ will stop 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.

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

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

‘Dial mode’
: This switch affects whether dialing commands will use touch tone or pulse dialing. Touch tone dialing usually is quite a bit faster than pulse dialing, but not all phone networks support it. This switch requires that either the dial prefix or dial suffix commands include the `\\W` command sequence (see Chapter 27 [Command sequences], page 125).

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

‘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.

'Limit macro'
: A command 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.

'Use' Use the current settings.

'Default (phonebook 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 mode'
: Choose the display mode 'term' is to use for the main screen here.

'Screen font'
: This is where you choose the user interface font 'term' is to use for the main screen.

'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.

'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.

'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!

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

`Colour' This button determines the colour mode the terminal emulation is going to use. Until now, four 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 (Mono.)'
: Monochrome, two colours.

`Blinking'
: If selected the VT-100 blinking option is enabled. This may require to allocate more colours for a specific colour mode than with blinking disabled, so do not be surprised if display performance suddenly drops like a brick.

`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.

`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 58) which permits editing the drawing pens to use for this colour mode.

`Use' Use the current settings.

`Default (phonebook 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.

19.5 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 61).

`Emulation`
: This is where you select the terminal emulation. Choose one of the following:

  `ANSI/VT-220`
  : This emulation is a 'melange' of three terminal command sets which themselves are supersets or subsets of one another. Most of the VT-220 command set is supported, including some additions made in the ANSI X3.64 specifications. With VT-220 its subsets VT-100 and VT-102 are supported. However, not all the VT-52 commands are supported.

  `Atomic`
  : A plain text-only terminal mode which filters out terminal commands and special control characters.

  `TTY`
  : Also a text-only terminal mode but which displays all control codes and commands it cannot handle on-screen, great for debugging.

  `Hex`
  : Another debugging mode which displays all incoming data in hexadecimal notation. If possible the corresponding glyphs will be displayed as well.

  `External`
  : This enables the use of external terminal emulation libraries following the XEM v2.0 specifications. In this mode you need to specify the library to use, otherwise 'term' will return to `ANSI/VT-220` mode.

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

`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.

These buttons determine the sequences that are sent to the remote if a carriage return (CR) or line feed (LF) character is to be transmitted. Both characters serve as end-of-line indicators.

'`-`'
: The character is suppressed.

'`<<CR>>`'
: A carriage return character is sent.

'`<<LF>>`'
: A line feed character is sent.

'`<<CR>><<LF>>`'
: A sequence of two characters (carriage return followed by line feed) is sent.

'`<<LF>><<CR>>`'
: A sequence of two characters (line feed followed by carriage return) is sent.

'`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.

'`Use`'
: Use the current settings.

'`Default (phonebook 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 in 'applications mode' the cursor keys will cause a command sequence to be transmitted instead of a cursor move event. This mode is usually activated by special applications on the remote side.

'`Lock cursor key mode`'
: Certain applications may excessively change the cursor key mode from standard to applications mode. If you do not want this to happen you can forbid it by using this switch.

'`Printer control enabled`'
: 'term' supports the standard VT-220 printer control commands. If you do not want the remote application to play with the printer the corresponding support commands can be disabled with this switch. If disabled 'term' will act like a VT-220 terminal with no printer attached.

'`Lock text style`'
: If this switch is enabled, any requests to change the text rendering attributes (underlined, highlight, blinking, inverse) will be rejected.

'`Lock text colour`'
: If this switch is enabled, any requests to change the text rendering colour will be rejected.

'`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.

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

'`Maximum prescroll lines`'
: 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.*

'`Maximum scroll jump`'
: 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 standard pens`'
: The built-in terminal emulation makes use of up to sixteen text rendering pens and four text rendering attributes. For each colour mode supported (see Section 19.3 [Screen panel], page 55) a specific pen order and text attribute assignment is predefined. You can select your own pen order and attribute assignment if this switch is not enabled by pressing the button labeled '`Edit pens...`'.

'`Edit pens...`'
: Pressing this button will open the text pen panel (see Section 19.7 [Textpen panel], page 64) which permits editing the drawing pens to use for this colour mode.

'`Use`'
: Use the current settings.

'Paste suffix'
: If enabled, the text to send after the clipboard contents are fed into the input stream, see Chapter 26 [Clipboard], page 123.

'Convert LF to CR'
: On the Amiga, new lines end with the LF (line feed) character. To simulate typed text, these characters should be converted to CR (carriage return characters) when pasting the contents of the clipboard. If this switch is enabled, the conversion will take place.

'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 modes are 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 BBS 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(s) to wait for the receiver to issue after a line of text is send. This text may include command 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.

'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 entries.

'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.

'Convert characters'
: This switch works in conjunction with the 'Filter enabled' option. When using the 'IBM PC style' terminal font, 'term' receives characters which normally do not have a place in the standard Amiga character set. If the 'Convert characters' switch is enabled, these characters will be converted into their Amiga equivalents, if there are any, before they go into the capture buffer. Note that this character conversion is always enabled for text captured to the printer.

'Creation date'
: By default 'term' will append the date of the call made to the name of the auto-capture file created ('Add to name'). Alternatively, 'term' will leave the name untouched and store the creation date within the file ('Write to file').

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

'Open 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 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 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.

'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 (phonebook 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 Miscellaneous panel

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

'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 restore the previous global configuration.

'Show fast! macros'
: If this switch is enabled, the fast macro window (see Section 20.1 [Fast macro panel], page 85) will be opened whenever the current configuration becomes active (e.g. at startup time). Please note that if this switch is not enabled, this does not cause the fast macro window to be closed.

'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.

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

'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'

'Ignore' The default position and size won't be touched.

'Centre' The requester will be centered within the limits of the main window.

'Relative'
The requester will appear at a specific position relative to the top left corner of the main window.

'Edit...'
Pressing this button opens a standard file requester, prompting you to resize and move the window to a position where you want to have it appear the next time it is opened. Click on the 'Use' button when finished.

'Use' Use the current settings.

'Default (phonebook 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 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 34 [Environment variables], page 139).

'Default 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 [Transf.], page 42).

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

- White line

  This line displays the smallest transfer performance (should be constant during the transfer).

'Hide upload icon'
: The file upload panel can be invoked by double-clicking on the corresponding icon placed in the Workbench window (see Section 20.14 [File upload panel], page 103). If you do not want the icon to apear, turn on the 'Hide upload icon' switch.

'Notify user after <n> errors have occured'
: Here you select after how many file transfer errors you want to be notified. The errors are counted separately for each file. When the given number of errors have accumulated, an error notification sound will be played. Setting this value to 0 suppresses this feature.

'Notify user'
: This switch controls when the file transfer window should be brought the front and a notification sound should be played:

  'only when an error occurs'
  : The 'term' screen and the file transfer window will stay in the background until a serious problem occurs.

  'when transfer begins/ends'
  : Screen and window will be brought to the front at the beginning and at the end of a transfer.

  'when transfer begins'
  : Screen and window will be brought to the front at the beginning of a transfer.

  'when transfer ends'
  : Screen and window will be brought to the front at the end of a transfer.

'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.

'Page'
: This control panel holds more settings than are visible at first glance. Right below the Page button you will find the invididual settings for the default file transfer protocol, the ASCII transfer protocol, the text transfer protocol and the binary transfer protocol. You can flip the pages by pressing the Page button:

‘`Edit settings...`’

Press this button to edit the settings of an XPR library. For an example of how these settings can look like, see Section 19.14 [XPR options sample], page 78.

‘`ASCII transfer protocol`’

This is where you set up the transfer protocol that is invoked when you select the `Upload ASCII file(s)` and `Download ASCII file(s)` menu items.

‘`Type`’

You can either select `XPR library`, `External program`, `<< Default >>` or `Internal` here.

`XPR library` will use an external transfer protocol library, such as ‘`xprzmodem.library`’. This library will be kept open all the time and may for example handle download session automatically. For more information on how to use XPR libraries see Section 22.1 [Data transfer via XPR library], page 109.

`External program` will invoke a program when necessary. This program has to temporarily take over serial I/O processing. Special facilities are available to pass parameters such as the name of the device driver ‘term’ uses to the program. For more information on how to use external programs see Section 22.2 [Data transfer via external program], page 109.

`<< Default >>` will use the default file transfer protocol.

`Internal` will use the built-in ASCII transfer protocol. For more information see Section 20.7 [ASCII-transfer settings], page 92.

‘`Send`’

This text entry field either holds the name of the XPR library to use or the name of the external program for sending ASCII data. Clicking on the select button at the right hand side of the text entry field will bring up either a file requester or another control panel to select the program name and to edit the program parameters.

‘`Receive`’

This text entry field either holds the name of the XPR library to use or the name of the external program for receiving ASCII data.

‘`Signature`’

Many file transfer protocols transmit characteristic data to the remote at the beginning of a transmission. This data is called a signature and when found in the incoming data stream ‘term’ will automatically invoke the protocol in question. This is particularly useful with external programs. This text entry field holds the signature which will when received start a transfer using the current ASCII will when received start a transfer using the current text transfer protocol. For more information on protocol signatures see Section 22.3 [Protocol signatures], page 110.

'Edit settings...'
: Press this button to edit the settings of an XPR library. For an example of how these settings can look like, see Section 19.14 [XPR options sample], page 78.

'Binary transfer protocol'
: This is where you set up the transfer protocol that is invoked when you select the `Upload binary file(s)` and `Download binary file(s)` menu items.

'Type'
: You can either select `XPR library`, `External program` or `<< Default >>` here.

`XPR library` will use an external transfer protocol library, such as '`xprzmodem.library`'. This library will be kept open all the time and may for example handle download session automatically. For more information on how to use XPR libraries see Section 22.1 [Data transfer via XPR library], page 109.

`External program` will invoke a program when necessary. This program has to temporarily take over serial I/O processing. Special facilities are available to pass parameters such as the name of the device driver 'term' uses to the program. For more information on how to use external programs see Section 22.2 [Data transfer via external program], page 109.

`<< Default >>` will use the default file transfer protocol.

'Send'
: This text entry field either holds the name of the XPR library to use or the name of the external program for sending binary data. Clicking on the select button at the right hand side of the text entry field will bring up either a file requester or another control panel to select the program name and to edit the program parameters.

'Receive'
: This text entry field either holds the name of the XPR library to use or the name of the external program for receiving binary data.

'Signature'
: Many file transfer protocols transmit characteristic data to the remote at the beginning of a transmission. This data is called a signature and when found in the incoming data stream 'term' will automatically invoke the protocol in question. This is particularly useful with external programs. This text entry field holds the signature which will when received start a transfer using the current binary

'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 19.11 [Miscellaneous panel], page 69).*

'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.

> *Additional note for 'term' users: the default received path option is ignored if the* 'Override transfer path' *switch in the miscellaneous panel (see Section 19.11 [Miscellaneous panel], page 69) 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.

19.15 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. Alternatively, you can press the key combination corresponding to the character whose translation you wish to change.

Both receive and send translation texts can consist of standard command sequences (see Chapter 27 [Command sequences], page 125), 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.

'`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.

The traditional VT-100 PF-keys (programmable function keys) are mapped to the top row of the numeric keypad. Hold down the `Control` key and press a top row key to produce the corresponding PF key code.

20 Cursor key panel

This control panel both works and looks similar to the function key panel (see Section 19.16 [Function key panel], page 82), 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 19.16 [Function key panel], page 82) 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 28 [Fast! macros], page 127).

'`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 text`'
: The command sequence (see Chapter 27 [Command sequences], page 125) 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.

'`Remove`' Removes the currently selected macro from the list.

'`Clear`' Removes all the macros from list, clearing it.

'`Load`' Loads the macro list from a file.

'`Save`' Saves the macro list to a file.

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 (words/minute)'
: Speaking speed in words per minute.

'Pitch (Hz)'
: The greater this value, the higher the voice appears to be speaking.

'Frequency (Hz)'
: Voice frequency in Hertz.

'Volume'
: The volume of the voice in percent.

'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.

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. Error messages are printed in a special colour.

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

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

'`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 xfered`'
: Number of bytes transferred yet.

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

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

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

'`Blocks xfered`'
: 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.

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 19.13 [Transfer panel], page 72) 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 xfered`'
: The number of bytes sent/received.

'`Lines xfered`'
: 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.

'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 modes are 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 command 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'

'Send LF'

These buttons determine the sequences that are sent to the remote if a carriage return (CR) or line feed (LF) character is to be transmitted. Both characters serve as end-of-line indicators.

'_'
: The character is suppressed.

'<<CR>>'
: A carriage return character is sent.

'<<LF>>'
: A line feed character is sent.

'<<CR>><<LF>>'
: A sequence of two characters (carriage return followed by line feed) is sent.

'<<LF>><<CR>>'
: A sequence of two characters (line feed followed by carriage return) is sent.

'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.

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').

Note: if you do not enter a phone number you will be unable to use the entry for dialing.

'Quick menu'
: If this switch is in effect, the corresponding phonebook 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'
: Generates a new telephonebook entry with standard settings and places it at the end of the telephonebook.

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

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

'Copy cfg.'
: 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 99 for more information.

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

'Tag'
: Tags the currently selected phonebook entry for inclusion in the dialing list. Pressing the Space key toggles the selection.

'Untag'
: Removes the currently selected phonebook entry from the dialing list. Pressing the Space key toggles the selection. Press the Del key to untag the currently selected phonebook entry.

'Tag all'
: Includes all phonebook entries in the dialing list.

'Toggle all'
: Adds all phonebook entries that are not in the dialing list to the dialing list and removes all entries from it that are already in it.

'Untag all'
: Removes all phonebook entries from the dialing list. Press Shift+Del to untag all entries.

'Load'
: Loads the contents of a telephonebook from a file.

With 'Translations', 'Function keys', 'Cursor keys' and 'Fast! macros' the data will be loaded from the corresponding files allowing you to edit it. 'term' will remember the names of the settings files data is read from or is written to. You can change the name directly by holding down a shift key when clicking on 'Translations', 'Function keys', 'Cursor keys' or 'Fast! macros'.

'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 98).

If you are still online, the 'Dial' button 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' 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' and 'Modem exit' command entries of the phonebook can also be used for initialization.

'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 'Sec./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.

'To all entries'
The selected parts will be copied to all phonebook entries. If any phonebook 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 phonebook configuration the corresponding global configuration will be left unchanged.

'Select all'
Selects all parts.

'`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`' 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.

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

'`Start script 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 31 [Script recording], page 133.

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.

‘Command’
: This text entry field contains the command sequence (see Chapter 27 [Command sequences], page 125) to be executed when the corresponding trap sequence is found.

‘|<’ Move the currently selected entry to the beginning of the list.

‘<’ Move the currently selected entry up in the list.

‘>’ Move the currently selected entry down in the list.

‘>|’ Move the currently selected entry to the end of the list.

‘New’ A new trap list entry is added, prompting you to edit it.

‘Remove’ Removes the currently selected list entry

‘Clear’ Removes all entries from the list, clearing it.

‘Use’ Closes the window, using the current trap settings.

‘Load’ Loads the trap settings from a file.

‘Save’ Stores the trap settings in a file. *Note: ‘term’ reads the default settings from the file trap.prefs, so make sure your trap settings are named accordingly if you wish to use them upon startup.*

20.14 File upload panel

‘term’ permits building a list of files to upload before the upload is started. This list can be built in many ways, such as by dropping the icons of the files to send on the icon labeled ‘term Upload queue’, by dropping the icons on the upload panel window, by entering the names of the files in the upload panel window or by using the file requester.

There are two ways to open the file upload panel. You can double-click on the ‘term Upload queue’ icon or use the main menu entry ‘Upload queue’. It includes the following controls:

‘Files to upload’
: This is the list of files to be sent. The text entry field below serves to add new file names or to edit the currently selected file name.

‘Add files’
: Clicking on this button brings up a file requester to add new files to the list. You can select files from one directory at a time. The file requester will pop up over and over again asking you to add more files until you press the ‘Done’ button.

‘Add’ Click on this button to add another file name to the list, you will be prompted to type in its name.

'>|' Moves the currently selected entry to the end of the list.

'New' Creates a new area code entry and prompts you to edit it.

'Remove' Removes the currently selected area code entry from the list.

'Clear' Removes all area code entries from the list, clearing it.

'Edit' Brings up the rates editing window for the currently selected entry. See Section 20.9 [Rate panel], page 98 for more information.

'Use' Closes the window, keeps the current settings.

'Load' Loads the area code & rates accounting information from a file.

'Save' Saves the area code & rates accounting information to a file. Upon startup 'term' will read the default area code & rates accounting information from a file named '`rates.prefs`', so make sure that your settings file is named correctly for 'term' to find it.

20.16 Parameter panel

When 'term' invokes an external program which is to handle the job of transferring files it can pass special parameters to the program on the command line, such as drawer names. This control panel helps you to build a command line for the program in question.

'Command'
This is where you enter the command to invoke, such as '`run hydracom`'.

'1 File' This adds `%f` to the command line. When the program is invoked a file requester will prompt you to select one single file. Its name will appear in place of the `%f` characters in the list of arguments passed to the program.

'Files' This adds `%m` to the command line. When the program is invoked a file requester will prompt you to select a list of files. Their names will appear in place of the `%m` characters in the list of arguments passed to the program.

'Port' This adds `%p` to the command line. When the program is invoked the name of the ARexx port 'term' uses will appear in place of the `%p` characters in the list of arguments passed to the program.

'Device' This adds `%d` to the command line. When the program is invoked the name of the serial device driver 'term' uses (see Section 19.1 [Serial panel], page 49) will appear in place of the `%d` characters in the list of arguments passed to the program.

'Unit' This adds `%u` to the command line. When the program is invoked the unit number of the serial device driver 'term' uses (see Section 19.1 [Serial panel], page 49) will appear in place of the `%u` characters in the list of arguments passed to the program.

21 Signature panel

‘term’ will let you choose from a number of predefined signatures for use with file transfer protocols. Just pick the signature you need. Please note that different signatures will be presented for upload and download protocols. For more information on signatures, see Section 22.3 [Protocol signatures], page 110.

22 Data transfer

One of the important features 'term' offers are means to transfer data from one computer to another conveniently. This is accomplished by using so-called XPR libraries and external programs which 'term' will invoke when necessary.

22.1 Data transfer via XPR library

The so-called XPR libraries implement one or more file transfer protocols in the form of an Amiga shared library. They offer a standardized interface for settings their protocol options and for transferring data. Some XPR libraries will handle file transfers all on their own, e.g. if the remote initiates an upload the XPR library will respond by automatically starting a download.

Of particular importance is the '`Default protocol`' (see Section 19.13 [Transfer panel], page 72). If you have selected an XPR library for this protocol, the library will remain open during the entire 'term' session. For the Z-Modem protocol as implemented through '`xprzmodem.library`' this means that the XPR library will automatically handle downloads when initiated by the remote.

22.2 Data transfer via external program

'term' can make use of external programs for the purpose of transferring data. Whenever the corresponding file transfer function is invoked, 'term' will try to run the selected program. While the program is running 'term' will temporarily halt its serial I/O processing, so programs which permit sharing the serial device driver with 'term' can immediately pick up the ball and start transferring data. Please note that this feature requires 'term' to open the serial device driver in shared access mode (see Section 19.1 [Serial panel], page 49).

Almost every external program will need a few command line options to know its whereabouts, such as the serial device driver to use or which files to transfer. You can provide this information by editing the command line (see Section 20.16 [Parameter panel], page 105) to include special escape sequences 'term' will expand into data. The following line could be put into the binary `Receive` text entry field:

```
run hydracom device %p speed %b line %c nocarrier rec %> get
```

A signature usually consists of a unique sequence of characters, some of which may not be printable or visible on the screen. This is why the standard command sequence syntax is employed for entering signature text (see Chapter 27 [Command sequences], page 125).

You should avoid using a single signature for more than one protocol. As ‘term’ scans the input data stream it will always invoke the first protocol which sports a matching signature. Signatures are scanned in the following order:

`Default protocol (upload)`

`Default protocol (download)`

`ASCII upload`

`ASCII download`

`Text upload`

`Text download`

`Binary upload`

`Binary download`

Most transfer protocols use different signatures for uploads and downloads. Hydra for example is an exception as it uses the same signature for both purposes. Take care, it is recommended to use the Hydra signature only for downloads. Some signatures, such as the CompuServe Quick B protocol, use very simple signatures which consist only of a single character. In the case of the Quick B protocol this would be the `ENQ` character which is easily generated by spurious line noise. In this case the protocol may start up expecting a file transfer and find out rather soon that none is taking place. Although single character signatures are supported it is recommended not to use them.

Some XPR libraries implement auto-upload and auto-download functions all on their own. A common feature is that the signatures that trigger these functions will not turn up in the input data stream ‘term’ receives as the protocols will filter them out. Consequently, the ‘term’ supplied protocol auto-invocation may not work. Be prepared to handle this.

download path (see Section 19.12 [Path panel], page 71 and Section 19.13 [Transfer panel], page 72).

`%s (Screen name)`
: Inserts the name of the public screen 'term' is using.

Note: This may be an empty string. Be prepared to handle this.

`%b (Baud rate)`
: Inserts the baud rate 'term' is currently using (see Section 19.1 [Serial panel], page 49).

`%c (Connection rate)`
: Inserts the baud rate the modem made the connection with.

Note: This value may be the same as given by %b if the modem is not currently online.

`%% (Percent sign)`
: Inserts the percent sign.

22.5 How to set up Hydracom?

In case you don't know already what Hydracom is: it is a bidirectional file transfer protocol which also sports a chat option. It permits to send and receive data at the same time. So far, Hydracom versions exist for the IBM-PC, the Atari ST and the Amiga of course.

With the introduction of 'term' v4.0 an interface was added to the Hydracom Amiga port to allow it to take over the serial I/O processing from 'term'. Note that this requires the Hydracom Amiga port revision 2 or higher to work.

'term' v4.3 will let you choose external programs for use as file transfer protocols. Hydracom falls into this cathegory.

Please open the transfer settings editor (see Section 19.13 [Transfer panel], page 72) now and press the button labeled `Page` three times until the page `Binary transfer protocol` becomes visible. This page is divided into two parts. The top half controls the upload protocol and the other half controls the download protocol. To use the Hydracom external protocol, now do the following: there are two buttons labeled `Type`. Press them both twice until they show `External program`. This will make the `Send` and `Receive` text entry fields available. In the `Send` field enter the following line:

```
run hydracom device %p speed %b line %c nocarrier rec %> send %m
```

In the `Receive` field enter the following line:

23 Configuration hints

Admittedly, '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 49). '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. First turn off* '`RTS/CTS`' *handshaking (set it to* '`None`'*), then enter* `AT&S0` *and press return, and finally turn on* '`RTS/CTS`' *Handshaking on again. Your modem should now respond properly to your commands. To make this change permanent, enter* `AT&W` *and press return to store the current modem profile in its nonvolatile RAM.*

   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 49) 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 52) and take a look at the switch labeled '`Dial mode`'. 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

6. The next step involves changing the path settings (see Section 19.12 [Path panel], page 71). 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.

24 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.

25 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.

25.1 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.

25.2 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 for the search text entered. 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.

  In addition to the search text there are a number of options which may be specified when searching:

  '`Search forward`'
  : If this switch is enabled 'term' search from the topmost line on the screen downward to the end of the buffer, otherwise it searches upward to the beginning of the buffer.

26 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 40) 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 40) 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).

27 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.

27.1 Backslash

‘\\’ Generates a single backslash.

‘\0’ Resets the text pacing mode (see Section 19.8 [Clipboard panel], page 64) 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.

28 Fast! macros

In implementation and design the fast! macros are closely related to the function key macros (see Section 19.16 [Function key panel], page 82). 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 27 [Command sequences], page 125) 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.

29 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'.

30 Chat line

The chat line is roughly functionally equivalent to the packet window (see Chapter 29 [Packet window], page 129). However, there is no special pull-down menu and no option to save or load the command history. Unlike the packet window the command history is kept between invocations.

The chat line is, as the name says, a text entry field which allows one single line of text to be entered. Except for the optical appearance and the handling of control characters (the text entry field appears as a single line above the status line, it's also a tad smaller than the packet window) it is virtually identical in handling with the packet window. The only exception is the special key combination to use when clearing the entire past command history. To clear the history, hold down either `Amiga` key and then press either the `Del` or the `Backspace` key.

The chat line always passes control characters, such as `Control + C` and `Tab` straight through to the modem.

31 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 phonebook entry.

To make sure that the script associated with the phonebook entry will be called the next time you dial it the phonebook file must be saved to disk before you quit 'term'.

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 27 [Command sequences], page 125. 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.

32 term and Emplant

You need to keep a few things in mind before you actually try to use 'term' with Emplant, the Apple Macintosh emulation and the on-board serial ports:

1. Both the emulation and 'term' are very demanding programs in terms of memory usage. The Macintosh emulation will allocate a fixed memory area for itself which normally should be as large as possible. 'term' has to use the amount of memory that remains, which may not be much. It is recommended that at least 3-4 MBytes of memory should be available when you start 'term'. Although the program will show an error message if it cannot allocate enough memory the external modules (terminal emulation libraries, file transfer libraries, serial device driver, etc.) may not work properly under low memory conditions and thus can cause software failures. You should reduce the sizes of the many memory buffers 'term' uses to perform its functions, such as transfering files and capturing text. For example, the text buffer will keep growing until all available memory is exhausted unless you set a maximum limit for its size (see Section 19.9 [Capture panel], page 66).
2. 'term' can share the device driver selected for the I/O ports with Emplant. For example, if you select '`serial.device`' as the driver to use for `Port A` Emplant will open the driver in shared mode. In 'term' you would select the '`Shared access`' switch in the serial settings (see Section 19.1 [Serial panel], page 49). When both programs are up and running you must make sure that only one program at a time will access the serial device driver, or data may be lost. For example, if you have '`ZTerm`' and 'term' running and wish to use '`ZTerm`' for communications you **must** make 'term' release the serial device driver (use the '`Modem`' menu item '`Release serial device`' for this purpose). Likewise, if you wish to use 'term' instead of '`ZTerm`' or some other terminal program on the Macintosh side, make sure you quit the Macintosh terminal program first.

   Take care, LocalTalk can have a negative effect on the serial data transfer performance.
3. If you connect one of the Emplant serial ports to your modem you should know whether the connector pins that are used for 7 wire hardware handshaking (RTS/CTS handshaking) are properly connected or not. Some cables that are sold for use with Hayes modems or the Apple ImageWriter do not have the necessary pins connected. If you wish to use the RTS/CTS handshaking protocol (see Section 19.1 [Serial panel], page 49), set the handshaking mode to '`RTS/CTS (Check DSR)`'. This insures that 'term' will run properly even if your cable cannot be used for RTS/CTS handshaking. Your Amiga may lock up if the cable does not support RTS/CTS handshaking and you have '`RTS/CTS`' selected as the handshaking protocol.
4. It is unwise to use '`empser.device`' while the Macintosh emulation is running. Since the Macintosh drivers are unaware of the Amiga side trying to access the serial port hardware conflicts are not to be avoided. Either use 'term' with '`empser.device`' or run the Macintosh emulation, you cannot do both at the same time.

33 term and SLIP

If you are brave enough to use 'term' to dial into your SLIP account, you should make sure that your modem does not hang up when you leave 'term' and hand control over to your SLIP software. Typically, closing the serial device driver causes the DTR signal to drop which some modems interprete as an immediate command to abort the connection. This signal is automatically dropped when 'term' is terminated.

To avoid this problem, consult your modem manual for information on a command that controls how the modem reacts when the DTR signal is dropped. For a ZyXEL modem this would be `AT&D0`. Put the command `AT&D0\\r` into the modem init command field (see Section 19.2 [Modem panel], page 52) of the phonebook entry you use to dial into your SLIP account.

34 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 44.

'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.).

35 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, if 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.

PGP-encoded mail is not welcome at my internet site!

36 Revision history

Changes introduced with v4.4:

- Threw out the optimized scrolling code. It could slow things down on slow computers and wasn't working particularly well either.
- Removed the code that did the jump scrolling by taking a look at how many line feeds were received in one go. This could slow things down and interfere with on-screen rendering, such as with on-line games like "Perihelion".
- Now uses gtlayout.library v13 to layout the menus; now many more menu items and menus fit onto small screens.
- Screens and windows now open large enough for the status line display to fit.
- Put the optimized scrolling code back in. It should now be able to behave itself in an orderly fashion.
- The screen saving/printing code is now smart enough not to include the status line in the bitmap that it is going to be output.
- The phonebook now uses a fixed-width font when running under Kickstart 2.04.
- The status line display boxes are a bit wider now, so hopefully no data will be cut off any more.
- Simplified the jump scrolling code a bit, eliminating another level of data filtering. This should remove the odd slowdown that would occur when the modem delivered a large chunk of data.
- Changed the way the download procedure queries how much space is still left on the destination filing system. This time it should get it right.
- Fixed a weird memory alignment problem in the double-buffered file handling code.
- Saving/loading the fast macro settings is now properly remembered, so 'term' will no longer tell you that the settings have been changed upon exit unless they have really been changed and not saved to disk yet.
- In the paged settings editors for screen, terminal and capture prefs the displays for screen mode and font no longer auto expand with their contents.
- Insert mode now works properly again [Andreas Kirchwitz].
- Fixed an Enforcer hit in the emulation settings when called from the phonebook [Matthias Scheler].
- Fixed locale string lookup bug in file identification code, also added PNG file signature.
- 'term' no longer refuses to dial out if the modem is still online and the 'protective mode' is enabled. In this case, a requester will pop up and ask you if you want to hang up the line before dialing out.
- Small fixes for deep CyberGfx screens (15 bits and up).
- The check to see if the serial device driver you picked from the list did really exist was looking at the wrong buffer.

Changes introduced with v4.3:

- Fixed an Enforcer hit in the code that would open the file transfer window in case of error.
- Colour palettes are now 24 bits wide (in reality even 96 bits, but the user interface does not support this precision).
- Added support code for AmigaUW terminal window resizing.
- Fixed another bug in gtlayout.library which would cause trouble with palette editor gadgets using only two colours.
- Fixed another Enforcer hit in the XEM settings editor.
- Updated the screen settings editor. It no longer displays options that cannot be changed.
- The `"Dial number"` function would use the wrong temporary buffer when prompting to enter the number to dial.
- Opening the status window no longer causes crashes. It was the call to DateStamp() and DateToStr() which caused the Task to handle the status window to crash. It's a process now.
- Cut & paste while the chat line is active now works properly.
- The chat line is now unavailable if an external terminal emulation is active.
- The dialing menu items now get properly disabled if the program starts up in online state.
- Made the only (!) call in gtlayout.library which could cause AmigaDOS to be called an option for Processes only. It will no longer crash when called by a Task.
- When running under Kickstart 2.04, the screen overscan mode is by default set to the text overscan size.
- The built-in ASCII transfer windows now also get size- adjusted in order to avoid overlapping the status line.
- Added a startup notice to explain that *this* really is a beta test release.
- The program now consists only of load hunks smaller than 100000 bytes each. This should make it possible to load the program even if the system memory is greatly fragmented.
- Finally discovered why the rate panel editor would swallow the first cost entry. Turning off the SAS/C global optimizer did the trick.
- The chat line text could become unreadable with some text pen choices. According to the BOOPSI documentation my original code was correct, but I discovered that the ROM code actually expects a different data format.
- The colour palette management code would not work properly under Kickstart 2.04.
- Fixed three long standing bugs in the terminal emulation code. If a command would erase/clear more lines/characters than the screen would hold memory trashing was not to be avoided. This has been fixed.
- An uninitialized variable in the colour palette setup code could cause real trouble, crashing the machine almost instantly.

gadgets will let you choose between 2..4 possible settings for each protocol. `"XPR library"` uses the good old XPR interface, the text entry field holds the name of the library to use. `"Internal"` uses the built-in code. `"Default"` uses the default protocol. `"External program"` selects an external program to handle the file transfer, the text entry field holds the name of the program and possible program parameters. If using the `"External program"` mode clicking on the downward pointing 'select' button will bring up another editor. Here you can choose the program to use and you can edit the command line options it should use. Pressing a button will append the corresponding escape sequence:

'`1 File (= %f)`'
: Inserts a single file name when the program is executed. A file requester will pop up if necessary.

  NOTE: Case matters; %f inserts the file name along with its complete path %F inserts the plain file name only, omitting the path.

'`Files (= %m)`'
: Inserts a list of file names when the program is executed. A file requester will pop up if necessary.

  NOTE: Case matters; %m inserts the file names along with their complete paths, %M inserts the plain file names only, omitting their paths.

'`Port (= %p)`'
: Inserts the ARexx port name 'term' is currently using. Very useful in conjunction with HydraCom.

'`Device (= %d)`'
: Inserts the name of the serial device driver 'term' is currently using. This comes in handy with external programs which permit sharing a device driver with other programs.

'`Unit (= %u)`'
: Inserts the serial device driver unit number 'term' is currently using. This comes in handy with external programs which permit sharing a device driver with other programs.

'`Source (= %<)`'
: Inserts the name of the upload path for the current transfer mode (ASCII, text, binary).

'`Dest. (= %>)`'
: Inserts the name of the download path for the current transfer mode (ASCII, text, binary).

'`Screen (= %s)`'
: Inserts the name of the public screen 'term' is running on.

  NOTE: May be an empty string.

'`Baud rate (= %b)`'
: The currently selected transfer speed in bits/second (Baud).

'`Connect. rate (= %c)`'
: The transfer speed your modem made the connection with.

  NOTE: this will be the same value as given by %b if the modem is not currently connected.

For xprzmodem.library it only makes sense to use the upload signature. Starting with v2.0 the library will always filter out the download signature and start the download process all on its own.

Hydra is a bit of a problem as it uses the same signature both for uploads and downloads. Take care; if things don't work as they should it may be better to delete the Hydra signature.

And before I forget to mention it: the signatures are entered in the (hopefully) familiar command sequence syntax, e.g. ^X stands for Control+X and \\ stands for \.

- The "\c <Menu name>" control sequence now checks if the menu function it is about to call is enabled.
- Even more changes to the file transfer settings; I moved some data from the misc settings over here. However, this beta version does not move your misc settings values over into the transfer settings. You need to to this manually.
- Renamed "Overwrite warning" to "Protective mode". Now this is what it does: in every situation (overwriting files, clearing the buffer, quitting the program, releasing the serial device driver, choosing a file/drawer/program) 'term' now runs a test to see if either the settings are valid or asks if the user really wants to do what he is about to do (it's not that we don't trust you). Using the "Protective mode" switch you can turn off all those sanity checks.
- Added another two terminal emulation control sequences for "ANSI" compliance.
- The menu "Wait" command requester can now be closed with a single keystroke.
- The program now uses special magic to make sure that all AmigaDOS and ARexx programs started receive proper search paths.
- The button labels "|<", "<", ">" and ">|" have been replaced by proper glyphs.
- The code to build the command line for external transfer protocols did not handle empty strings correctly. It does now.
- The Hydracom example invocation commands listed in the previous section of this document were not correct. If you are using this protocol, please update the command lines as described in the previous section.
- Any requester that shows just a single "Continue" button can now be closed with a single keystroke.
- The status window now properly displays the name of the currently connected BBS.
- Slight changes and enhancements to the user interface code.
- String gadgets are now properly aligned in columns in the path and command settings editors.
- The prescrolling/jump scrolling code now gets 'out of the way' if the background colour is currently nonzero.
- Slightly improved low-memory stability, especially during the initial setup procedure.
- The "You don't have RTS/CTS handshaking enabled..." request now enables RTS/CTS handshaking with DSR checking if the user decides so.
- The emulation pen selection now supports public screens again. Please note that the implementation is not perfect (which was the primary reason for disabling it in v4.2) and may not work properly when using the keyboard to pick the colours, i.e. keystroke activation may produce unexpected results.

- 'term' now opens screens as large as possible if the screen settings indicate a specific screen size, but the user has no means to change them. In previous program releases you would get whatever was found in the screen settings, even if you didn't have asl.library v38 or higher handy to change the dimensions.
- The xON/xOFF handling code works a bit differently now. If you have the "Internal xON/xOFF handling" switch enabled in the serial settings 'term' will now go into 'holding' state when you press Control+S (= xON). To return to normal operation, press Control+Q (= xOFF). If the "Pass xON/xOFF through" switch is enabled, both xON/xOFF characters will be sent through to the remote, otherwise 'term' will swallow them. The big difference between this handling and the old style of doing things is that 'term' will no longer drop into 'holding' state when receiving an xON character from the remote. The only way to bring 'term' into 'holding' state is by pressing Control+S.
- The chat line now passes control characters and function key macros through to the main program as you type them, i.e. they will not show up in the text you type. The Tab key is special; if pressed, the tab character will be passed through to the main program. If you press Control+I the character will show up in the chat line.
- Added a shortcut to select between pulse dialing and tone dialing. This requires that your dial prefix or suffix command includes the special command sequence \W. This sequence will translate into P for pulse dialing and into T for tone dialing. So in order to take advantage of this feature, you should change your dialing command to ATD\W and select the dialing mode you want. Please note that the dial mode option will be disabled if there is no \W in the dial prefix and dial suffix.
- The default serial and modem setups are a bit different now. As always, the program first tries to read the global system serial settings and converts them if necessary. If RTS/CTS handshaking is enabled, it now enables the RTS/CTS handshaking mode with DSR test in order to avoid lockups. If the serial settings could not be read the default setup now is 19,200 baud, 8-N-1 and RTS/CTS handshaking with DSR checking. The modem settings no longer include "ATZ\r" as the modem init command, the dial prefix now reads "ATD\W" and the default dial mode is tone dialing.
- The default screen display mode settings are now taken from the default public screen.
- Added another friendly reminder that is displayed whenever you upgrade from an older program release or start the program for the first time. The reminder will be displayed every time you start the program until you save the program settings.
- Pasting the current clipboard contents now optionally converts line feed characters into carriage returns (there is a new option in the clipboard settings for this purpose).
- 'term' used to fake an immediate XPR abort by returning a read error in xpr_sread() in case the user had pressed the abort button. This really should not be necessary, I just rewrote the code to abort the read prematurely and to return whatever came in so far. This implies the hope that the protocol will eventually drop into xpr_chkabort() and find out what's cooking.
- The chat line now gets properly redrawn even if the status line is currently turned off or sitting in a separate window.
- When copying the contents of the screen to the clipboard 'term' now converts alien IBM characters into ISO characters. This is a) required for the IFF FTXT format in which text gets stored in the clipboard

- When starting up for the first time, 'term' no longer complains about missing DSR signals or notifies the user that RTS/CTS handshaking should be enabled. This is done in order to avoid confusion, the reminders and messages will follow later when the user makes the first changes and saves them to disk.
- The XPR transfer window no longer warns about files not fitting on filing system which look suspiciously like ram disks, i.e. are not clearly identified as block mapped filing systems.

Changes introduced with v4.2:

- Did not set up serial parameter correctly (nasty, those typos!).
- HydraCom could cause 'term' to hang upon startup.
- Increased the width of all the integer gadgets in the rates settings panel.
- Incrementer arrows did not work properly in all settings panels, this was due to a bug in the SAS/C optimizer which caused gtlayout.library to run into trouble.
- When called from the phonebook, the `"Standard"` button as shown by the translation table settings panel will cause the translation settings to be reset to standard values.
- Text stored in the buffer did not get bit 7 stripped if this feature was enabled in the serial settings.
- New look slider gadgets (requires gtlayout.library v5).
- Numerous bug fixes in the user interface support library.
- Rewrote the text buffer capture routines, there should no longer be extra, unwanted data in the buffer.
- Fixed the overly wide incrementer arrow bug which caused so much trouble in previous releases.
- Reloading the fast macros when making a new connection did not update the fast macro window.
- In the area codes editor, creating a new entry and moving it around no longer leads to unexpected results.
- There was a typo in the source code which prevented the EOL translation settings from getting changed via ARexx.
- For a phonebook entry dialed, the startup and login macros are now executed in sequence rather than in parallel.
- New capture settings options `"Convert characters"`: if enabled along with the capture filter, text stored in the text buffer and the capture file will be converted into proper ISO characters. This effectively discards unprintable IBM font style characters. Note that this option will do nothing if you are using the standard text font rather than the IBM text font. Also keep in mind that this special text filter will always be enabled for printer captures in order to avoid nasty side-effects. The text and review buffers will no longer use the IBM PC style font if this option is in effect.
- In the emulation settings you will find a new switch labeled `"Lock wrapping"` which will let you lock the current line wrapping mode so that application software and terminal resets will no longer modify it.

deadlocks, no more sluggish mouse movements, no more CPU hogging. The display window is a bit larger, but this hopefully won't be a problem. After all, the window mode is usable now. The old BOOPSI code is gone and will probably never return. The new code is in many ways quite a bit nicer than the old code. For example, it is synchronized with the window size changes. As soon as the terminal adapts itself to the new window size, so does the status line display.

- The review buffer window text rendering colours would also get set to some value when opening the program on a custom screen. It now leaves the text colour untouched in this case.
- The main window position is now saved along with the main settings.
- Finally added the one-line chat text entry field which surely is no replacement for the packet window, but nevertheless I hope at least some folks will find it useful. No split-screen chat yet, sorry.
- The 'dial number' requester now remembers phone numbers between calls.
- The quick dial menu now gets disabled if the modem is online.
- The ASCII transfer menu items no longer get disabled if the internal transfer routines are selected and the XPR ascii transfer lib names are blank.
- Updated the font selection code for text and review buffer displays. The review buffer now runs as a Process, so it can open disk resident fonts if it needs to.
- Fixed a few bugs in the SETATTR ARexx command.
- Auto-expanding control panels, such as the phonebook and the file transfer window no longer obscure the status line display.
- The program now properly pays attention to the number of lines to use for the terminal display. Thanks go to Russ for his persistence ;-)
- In monochrome mode text is no longer printed in inverse video mode.
- The screen settings editor now allows you to change the colour mode even if 'term' is running on a public screen.

Changes introduced with v4.1:

- Fixed an Enforcer hit in gtlayout.library caused by the text gadget handling code.
- If possible child windows are now opened within the bounds of their parent windows.
- Fixed an Enforcer hit caused by the ARexx interface building dialing lists.
- Reordered the parameters of the "ADDITEM" ARexx interface command.
- Rewrote the entire data capturing process. I somewhat opened a can of worms, making it necessary to rewrite the code that handles the translations for the "Receive CR as..." and "Receive LF as..." options as well. Careful please, although I am sure the code works correctly I may have knocked over some china cups.

- Finally discovered why so many old phonebook files would cause trouble: the internal conversion routine was *never* called.
- Horrors! The sound.datatype saves invalid sound files with the playback size set to zero, causing 'term' to crash with a 'division by zero' error. The replay routine now handles such odd files.
- The sliders for redial delay and time to connect now finally sport a resolution of a single second rather than ten seconds.
- Shortened gadget labels & texts and rearranged the gadget layout to make sure all windows will fit on a plain 640 x 400 sized screen with topaz/8. Sorry folks, 640 x 200 is right out of the question. These are the days of miracle and wonder and all modern Amiga hardware is capable of displaying screens this size in non-interlaced modes. If things still don't seem to fit try a different font, preferably proportional-spaced, or a different screen resolution (change the overscan size if necessary).
- Whilst reworking the documentation discovered that I forgot to add the 'Alert' control to the terminal panel.
- With Workbench v2.04 the screen mode requester automatically resets the screen size and overscan values to defaults.
- The text buffer search requester now sports another option, "Whole words only".
- The jump scroll option code had a control switch set in the 'wrong direction'. Nothing serious, 'term' would only scroll too many lines.
- The jump scrolling routines did not take the size and position of the currently active scroll region into account. This could knock out the emulation since the cursor could cross the legal position limits.
- Added some more control key codes as per the VT 220 Programmer Pocket Guide, such as ^2, ^3, ^4, ^5, ^[, ^/, ^].
- Fixed another bug that would cause 'term' to busy loop if it stumbled upon a non-printable character in IBM PC style font mode.
- 'term' now supports context-sensitive help with AmigaGuide v34. I finally discovered a set of AmigaGuide commands that would not crash when told to change the currently displayed context.
- Changing XEM options will save them back to disk.
- Rewrote the hangup/carrier lost/online/offline handling procedure. The online/offline status tracking is protected by semaphores now, the code to modify the status was reduced to a great deal, it's only in termDial.c, termARexxCommands.c and in termMain.c. The hang up command and carrier lost actions now go through the same code, i.e. backup config & redial on logoff now work both for logoff & hangup.
- The phone rates accounting by area codes did not work since the routines were commented out. I removed the comments, they should work now.
- The buffer management no longer collapses if you try to clear the contents while there is still data coming in.
- Reread the "VT 220 Programmer Pocket Guide" and added most of the remaining unsupported control sequences. Not supported are the programmable function keys and data transfers bracketed

- Shortened the button labels for the phonebook and the rate panel, so they will finally fit on a 640 x 400 screen using topaz/8. Some button labels now look fairly obscure, sorry about that. I guess I'll rething the part about the "clarification" again...
- More weird & wonderful changes to the serial device interface code. I hope it still works.
- When hanging up the line using the corresponding menu command the online state is no longer reset to 'offline' if in the serial settings the "Check carrier" feature is enabled. This leaves the test for the carrier to the usual routines which will detect if the carrier is really gone.
- With multi-number phonebook entries the dialer now displays how many of these numbers have been dialed already.
- There was something really wrong with the way clipboard text was pasted as the 'end of line' character conversion would be applied twice.
- The "Time to connect" data was never used for phone rates accounting, it only played a minor role in the "Connect limit" settings.
- The pop-up 'About' window no longer is GimmeZeroZero, which is both sexy and uses less memory.
- Beep & action sounds are now loaded via datatypes if available. Under v39 this may not work well for large sound files due to a bug in sound.datatype. Plain IFF-8SVX format sound files will still be played using the built-in routines since they are more flexible than sound.datatype and can replay stereo sound.
- Non-standard-sized system imagery (sizing gadgets, arrow gadgets, etc.) is now supported whereever it is used.
- The phonebook window no longer sports a 'Close' button, all that's left is the 'Dial...' button.
- Phonebook entries which lack a phone number can no longer be used for the dialing list.

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 VT-100 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

- 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)

  * Stop transfer batch Cancels the entire batch transfer (level 1)

  * Stop entire transfer Emergency stop (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.

- 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 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.

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.

- 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.

- 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.

- 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 phonebook 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 phonebook 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.

- 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.

Index

D

E

T

Table of Contents