

term

A terminal program for Amiga computers

Version 2.3
23 August 1992

by Olaf Barthel

Copyright © 1990-1992 Olaf Barthel & MXM

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

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

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

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

'term' may be included with commercial programs as long as no money is made by including it in the distribution (this also applies to inclusion in AmigaDOS Enhancer kits, on Workbench/Extras disks or A3x00 hard disk drives).

You *must not* make any money by distributing 'term'. A small copy fee for data media (US\$ 5 maximum) or the inclusion of 'term' in Public-Domain collections such as Fred Fish's Amiga Library of Freely Distributable Software is permitted though.

1 Important information

I have placed the following information at the beginning of this document so nobody can claim not to have found them.

1.1 Name changes

For compliance with existing design guidelines, the names of configuration files and the preferred storage directory have changed. 'term' will now store and read configuration files in the 'TERM:config' drawer (well, unless there is a 'TERMPATH' variable which lead to a different path... see below). If no 'TERM:' device is available, 'term' will assign 'TERM:' to the directory it was started from. If no 'TERM:config' drawer is available, it will be created.

The following names have changed:

'Preferences.term'

'term_preferences.iff'

The new name is 'term.prefs'

'Phonebook.term'

'term_phonebook.iff'

The new name is 'phonebook.prefs'

'Hotkeys.term'

'term_hotkeys.iff'

The new name is 'hotkeys.prefs'

'Speech.term'

'term_speech.iff'

The new name is 'speech.prefs'

'Macros.term'

'term_macros.iff'

The new name is 'macros.prefs'

'Fast!Macros.term'

'term_fastmacros.iff'

The new name is 'fastmacros.prefs'

1.2 The carrier signal

Starting with revision 2.3 the ‘Check carrier’ switch in the Section 10.2 [Modem panel], page 26 has a new function. If enabled, the carrier signal will be checked during a file transfer (upload, download). If the signal gets lost, the transfer will be aborted. If you wish to do a null-modem transfer make sure that the switch is disabled or the transfer will be finished before it has even started.

1.3 High-speed mode

Enabling the so-called ‘High-speed mode’ in the Section 10.1 [Serial panel], page 25 always had the effect of defaulting the serial parameters to eight bits per char, one stop bit, no parity. Up to v2.3 ‘term’ would not display the changes. Please do not wonder that this switch also affects other serial parameters.

2 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. I used the following tools to create ‘term’: SAS/C 5.10b, WShell, CygnusEd Professional Release 2 & ARexx.

3 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, Markus Stoll, Martin Berndt, Matthias Zepf, Michael Vaeth, Michael Wolfgang Hohmann,, Oliver Wagner, Peter Fischer, Ralf Thanner,, 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.

The xpr-libraries were created by Kenneth Osterberg & Markus Pietz (‘jmodem’), Marco Papa & Stephen Walton (‘kermit’), Jack Rouse (‘quickb’), Marc Boucher (‘xmodem’), Ueli Kaufmann (‘ascii’, ‘ymodem’ & ‘vms’) and Rick Huebner & William M. Perkins (‘zmodem’).

The xpr-standard was created by Willy Langeveld, I borrowed a single routine (‘ahtoi’) from Matthew Dillon’s ‘DMouse’ program, the quicksort routine (‘QuickSort.asm’) was written by David Jones, the touch-tone dialing routines are based on the article *DTMF – A Program for Generating Touch-Tone Signals* by Adam Levin, the touch tone period and cycle values are Copyright © 1989 by Commodore-Amiga, Inc.

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.

Special thanks go to Leo Schwab who discovered means to use interleaved bitmaps in a system-integrated manner.

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 termRexx documentation – thanks to all of you!

4 Source code

Since there are still only very few well-documented examples (or general programming examples) for Kickstart 2.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.

5 Program updates

Whenever a new release of 'term' becomes available I will try to make it known in the telecommunications networks. To order a copy send a self addressed envelope, an international mail reply coupon and two 3.5" disks to the author.

The most current 'term' release will be available through ftp from `ftp.informatik.uni-oldenburg.de` (134.106.1.9), look into the `'/pub/amiga/term'` directory.

6 Operating the program

'term' can be controlled both by mouse and keyboard. With very few exceptions each operation requiring mouse control can also be executed using keyboard shortcuts.

The program can be started both from Workbench and from Shell. Kickstart 2.x is required to run 'term'. The program will refuse to work if executed under control of Kickstart 1.3 and below!

At least Kickstart 2.04 (revision 37.175) and Workbench 2.04 (revision 37.67) are required to execute 'term' successfully. Earlier developer Kickstart revisions or the Kickstart release installed on the older A3000 models will probably not work (try it if you want – but don't complain!).

If called from Shell 'term' will detach itself immediately allowing the Shell window to be closed afterwards. This effect can be avoided if `KEEPPIO` is entered in the command line ('`term keepio`').

Usually, running 'term' twice will cause the screen of the other program to be popped to the front instead of creating a second 'term' process. To avoid this effect, enter `DONTPOP` as a calling parameter (e.g. '`term dontpop`'). If called from Workbench, each program will run as a separate process.

If 'term' is to read its configuration files from a special directory on startup, you can pass it the `SETTINGS` keyword and the name of the directory to use (e.g. '`term settings work:term`').

7 Keyboard control

Starting with revision 2.2 almost any action can be executed both by mouse and by keyboard. Each element of the user interface in whose title an underlined character can be found can be controlled or activated by pressing the corresponding key. If the object in question is a list view or scroll bar, pressing the key along with a **Shift** key will scroll the list/bar back and forth. Pressing the **Tab** key will activate the first string gadget on the screen.

8 Screen

The ‘term’ main screen always opens to the full text-overscan size (which can be set using the ‘Overscan’ tool in the ‘Prefs’ drawer).

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

- ‘Status’ The current program operating status. This includes ‘Ready’, ‘Holding’ (Control+S was pressed), ‘Dialing’ (the dialing function is at work), ‘Upload’ (data is being sent), ‘Download’ (data is being received), ‘Breaking’ (a break signal is transmitted across the serial line) and ‘Hanging up’ (connection is being cancelled).
- ‘Font’ The currently active screen font used for text/graphics rendering.
- ‘Protocol’ The currently set data transfer protocol.
- ‘Emulation’ The currently active terminal emulation mode.
- ‘Baud rate’ The data transfer rate in bits per second.
- ‘Parameters’ The current serial parameters (Data bits-Parity-Stop bits).
- ‘Time’ The current time of day.
- ‘Online’ The time elapsed after a connection was successfully established. This counter will be stopped as soon as the connection is cancelled (e.g. by hanging up) and is reset to 00:00:00 as soon as a new connection is made.

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

The screen itself is opened as a public screen (called **TERM**) which is available to other programs for their purposes. If more than one ‘term’ process is running, the public screen name will change according to the number of the program (i.e. the first ‘term’ to be started will call the screen **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.

9 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.

9.1 Project

‘Save screen as Picture/Text’

This menu serves to save the current terminal window contents either as plain ASCII text file or as a picture file.

‘Print Screen/Clipboard’

These menu entries are to output text on the printer. You can either print the contents of the main screen or the contents of the clipboard.

‘Capture to File/Printer’

Selecting one of these menu entries will toggle capturing incoming text to the printer and/or a file on disk.

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

While ‘term’ is iconified, all incoming synchronous ARexx-commands will be queued and the corresponding ARexx programs will appear to ‘hang’. As soon as ‘term’ is ‘awake’ again pending commands will be processed again.

‘About’ Shows some information on the program.

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

After selecting the ‘Quit’ menu item the program may take a few seconds to exit since the contents of the Chapter 11 [Text buffer], page 51 are freed first.

9.2 Edit

‘Paste’ Pastes the contents of the clipboard at the current cursor position provided that the clipboard contains text data.

9.3 Commands

‘Execute AmigaDOS command’

Enter the command you want to execute and its command line arguments here.

‘Execute ARexx command’

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

The ARexx command set supported by ‘term’ is described in the file ‘**termRexx.doc**’.

‘Set 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:’.

The menu functions ‘Execute AmigaDOS command’ and ‘Execute ARexx command’ allow you to select the name of the file to be executed by clicking on the ‘Load file’ gadget.

9.4 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 10.14 [Phonebook], page 42).

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

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

‘Play phone number’

Outputs a telephone number using touch-tone dial coding on the Amiga audio hardware.

‘Send break’

Sends a ‘break’ signal across the serial line.

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

‘Flush receive buffer’

Tells the serial driver to drop its input buffer contents and resets the state of the internal serial buffers.

‘Release serial device’

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

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

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

9.5 Transfer

‘Upload ASCII file(s)’

This is a pure ASCII-file upload implemented through ‘xprascii.library’. It was added to allow poor BBS programs to receive text files (especially Paragon doesn’t seem to like ZModem text file upload).

‘Download ASCII file(s)’

This is a pure ASCII-file download implemented through ‘xprascii.library’. Refer to ‘Upload ASCII file(s)’ for features/options of this mode.

‘Upload text file(s)’

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

‘Download text file(s)’

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

‘Edit & upload text file’

Invokes the currently selected (see Section 10.7 [Path panel], page 35) text editor on a file to be selected using a file requester.

‘term’ will block and wait until the editor has returned.

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

‘Upload binary file(s)’

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

‘Download binary file(s)’

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

The transfer routines open an information window in which a number of transfer parameters are displayed (see Section 10.13 [Transfer panel], page 40).

9.6 Buffer

‘Clear buffer’

Clears the contents of the Chapter 11 [Text buffer], page 51. Any text will be discarded and *cannot* be recovered. Capture files are not affected by this command.

‘Display buffer’

Opens the Chapter 11 [Text buffer], page 51 screen.

‘Close buffer’

Closes the Chapter 11 [Text buffer], page 51 screen but does not free the contents.

‘Freeze buffer’

This menu entry will, if enabled, stop the Chapter 11 [Text buffer], page 51 from filling up with new text.

‘Open buffer...’

Loads the contents of the Chapter 11 [Text buffer], page 51 from a file. If there are still text lines in the Chapter 11 [Text buffer], page 51 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 Chapter 11 [Text buffer], page 51 to a file. You will be notified if the file to save to exists already (you may discard the old file, append the new data or cancel the action).

More detailed information on the Chapter 11 [Text buffer], page 51 will be given later in this document.

9.7 Terminal

‘Clear screen’

Clears the whole ‘term’ screen and moves the cursor to the top left home position.

‘Reset font’

Will change the screen font back to the default screen font.

‘Reset styles’

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

‘Reset terminal’

Use this menu item to reset the state of the entire terminal emulation.

‘Change emulation parameters’

Will open a control panel which allows setting some of the internal parameters of the currently active terminal emulation. Depending on whether an external or the internal terminal emulation is currently active, different control panels will appear. For more information on the internal terminal emulation see Section 10.12 [Emulation panel], page 39.

9.8 Settings

This is where you adjust the program preferences settings. There is a difference between local and global settings: local settings can also be set by making a connection to a mailbox while global settings are kept even while local settings are active. They have to be changed explicitly.

‘Serial settings’

See Section 10.1 [Serial panel], page 25

‘Modem settings’

See Section 10.2 [Modem panel], page 26.

‘Screen settings’

See Section 10.3 [Screen panel], page 28.

'Terminal settings'

See Section 10.4 [Terminal panel], page 29.

'Commands'

See Section 10.5 [Command panel], page 33.

'Miscellaneous'

See Section 10.6 [Miscellaneous panel], page 34.

'Paths' See Section 10.7 [Path panel], page 35.**'Transfer protocol'**

This command allows you to specify the transfer protocol library to be employed for up-/downloads.

'Transfer'

The transfer options can be changed with this command. If possible a control panel will pop up which allows you to change the current parameters by mouse/keyboard. If the selected transfer protocol does not provide these option hooks, a simple string requester will prompt for input. Consult the documentation of your favourite transfer protocol for legal options and the values to which they can be set.

'Macro settings'

See Section 10.8 [Macro panel], page 36.

'Fast! macro settings'

See Section 10.9 [Fast macro panel], page 37.

'Hotkeys' See Section 10.10 [Hotkey panel], page 37.**'Speech'** See Section 10.11 [Speech panel], page 38.

9.9 Windows

'Status' This function opens a window to display program status information. Click the window to update the information:

'Session start'

When was the program started?

'Bytes received'

The number of bytes received.

'Bytes sent'

The number of bytes sent.

'Connection message'

The string returned by the modem when a connection was made.

'Screen size'

The size of the terminal output window in characters (columns / rows).

'Buffer size (bytes)'

The size of the Chapter 11 [Text buffer], page 51.

'Free memory (bytes)'

The amount of free system memory.

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

'Packet' Opens an input window in which a single line of characters to be transferred across the serial line can be entered. More on this feature is explained under Chapter 15 [Packet window], page 59 below.

'Fast! macros'

Opens or closes the so-called fast! macro panel. The role of the Chapter 14 [Fast! macros], page 57 will be discussed later in this text.

10 Control panels

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

10.1 Serial panel

This is where the serial parameters may be changed.

‘Baud rate’

Transfer speed in bits per second.

‘Bits/char’

Number of bits per transferred character (7 or 8).

‘Parity’ Serial parity (none, odd, even, mark or space).

‘Stop bits’

Number of stop bits (1 or 2).

‘Handshaking’

Serial handshaking mode (RTS/CTS, RTS/CTS with DRS signal check, none).

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

‘Buffer size’

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

‘High-speed mode’

Activates a special mode of the serial driver which is to skip a couple of internal parity and stop bits checks resulting in higher data throughput rate. *If this switch is turned on, the serial parameters will be reset to 8 bits per char, no parity and 1 stop bit.*

‘Shared access’

Requested by many (two to be accurate) users, this switch allows you to run the serial driver in shared access mode. Not all driver types will allow this to happen.

‘xON/xOFF’

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

'Break length'

Length of the break signal given in microseconds.

'Serial device'

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

'Select new device'

Press this button to select the serial device driver using a file requester.

'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.

'Use' Accept the current settings.

'Cancel' Don't use the current settings.

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

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

'term' will take advantage of Christopher Whichura's `OwnDevUnit.library` if installed in your system, allowing controlled access to the serial driver's resources.

10.2 Modem panel

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

'Modem init string'

The string to send to the modem after successful program initialization. This string is optional and does not need to be entered.

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

'Modem exit string'

The string to be sent to the modem shortly before the program terminates. Just like the modem init string this string is optional and does not need to be entered.

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

‘Modem hangup string’

The string to be sent to the modem when asked to hang up the line. This string is not optional, it has to be present for the program to work properly.

‘Dial prefix’

The string to be used to prefix each dialing command. This is usually a variant of ATDP or ATDT.

‘Dial suffix’

The string to be used to append to each dialing command. This is usually the carriage-return character `\r`.

‘No carrier string’

The string the modem emits if the data carrier is lost. The program uses this to determine the length of the connection and to calculate the how much the user is to pay for it.

‘Connect string’

The string the modem emits after detecting a carrier signal. ‘term’ uses this input to determine successful telephone connection, to reconfigure itself and to start the online timer.

‘No dialtone string’

The string to be returned by the modem in case it does not encounter any dialing tone on the phone line.

‘Voice string’

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

‘Ring string’

The string the modem emits if it receives a call, same effects as with the ‘Voice string’.

‘Busy string’

The string the modem returns if the number which has just been dialed is busy.

‘Redial Delay’

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

‘Number of dial retries’

The number of times the dialer walks through the dialing list trying to make a successful connection before giving up.

‘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.

‘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.

‘Connect auto-baud’

Most modems echo the baud rate upon successful connection. If enabled the baud rate will be read and set for the serial driver.

‘Log actions’

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

‘Check carrier’

‘term’ will recognize the **NO CARRIER** message a modem emits when the carrier line signal drops back to low. For maximum safety, ‘term’ will also check the carrier signal line after receiving the **NO CARRIER** message if this switch is enabled. This is to make sure that no accidentally appearing string causes confusion.

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

‘Drop DTR on hangup’

Some modems will track the **data terminal ready** line in order to make sure that the terminal program is listening, once the line goes back to low potential, these modems will drop the line and hang up. Use this button to enable this feature.

‘Pass xON/xOFF through’

If this switch is enabled, ‘term’ will pass the **xON/xOFF** characters through to the modem rather than swallowing them.

‘Create call info’

If this switch is enabled, ‘term’ will create call-log files in a format compatible with the **NComm** terminal program in the log-file directory specified under ‘Paths’ settings (see Section 10.7 [Path panel], page 35). The name of the log file will always be **term-call.log**.

‘Use’ Use the current settings.

‘Cancel’ Keep original settings.

10.3 Screen panel

‘Display modes’

All display modes available for the main screen are displayed here.

The A2024-modes should only be used in connection with an A2024/Hedley-monitor. Otherwise the screen may ‘crash’ and working with ‘term’ will not be possible any longer.

‘Edit screen palette’

These gadgets are used to select a colour of the screen palette that is to be changed.

‘Red/Green/Blue’

Use these sliders to modify the red, green and blue components of the currently active colour.

‘Use default colours’

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

‘Screen font’

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

‘Select new screen font’

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

‘Make screen public’

As dealt with before, the ‘term’ mainscreen is opened as ‘public’. With this gadget the user may define if other applications are allowed to open up their windows on the ‘term’ screen.

‘‘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’ Use the current settings.

‘Cancel’ Keep the old settings.

The new ECS display modes (Super-Hires, Productivity, etc.) restrict the number of screen colours. Instead of 16 bit portions for red, green and blue only 2-bit portions are used which leads to a maximum width of the palette of at all 64 colours. This is not a ‘feature’ of ‘term’, but a hardware restriction.

10.4 Terminal panel

All settings that may be done here determine the behaviour of the terminal itself, i.e. how command sequences are interpreted and displayed.

‘Capture filter’

If selected, command sequences are filtered out before the incoming characters are captured to disk or printer.

‘Destructive BS’

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.

‘Swap DEL <-> BS’

For the hard core VT100 user: if this switch is enabled the roles of the backspace key and the delete key are reversed. *This also applies to sequences such as Control+H which will produce a delete character instead of a backspace character.*

‘Strip bit 8’

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

‘Stop blinking’

If selected the VT100-blink option is suppressed. If running in eight colour mode, ‘term’ will only use three bitplanes instead of four for text display.

‘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.

‘Screen title’

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

‘Status line’

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

‘Disabled’

No status line is displayed.

‘Standard’

The standard two status lines are displayed.

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

1. Status
2. Terminal type
3. Transfer protocol

4. Baud rate
5. Serial parameters
6. Time of day
7. Online time

‘Answerback message’

The string to send across the serial line whenever an ENQ character is received. The string is – as usual – a command sequence.

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

‘Standard’

The standard text font.

‘IBM’ A font similar to the IBM-PC text font will be used. *This requires the ‘IBM.font’ to be present in the ‘Fonts:’ directory.*

‘Text font’

Here the name of the standard terminal window text font is displayed.

‘Select new text font’

Press this button to select a new terminal window text font.

‘Send CR’ This gadget determines the sequence that is sent to the remote if a carriage-return is issued.

‘As CR’ Without any change

‘As CR+LF’ Carriage-return and linefeed

‘Ignore’ Suppress any carriage-return

‘Send LF’ This gadget determines the sequence that is sent to the remote if a linefeed is issued.

‘As LF’ Without any change

‘As LF+CR’ Linefeed and carriage-return

‘Ignore’ Suppress any linefeed

‘Receive CR’

‘Receive LF’

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

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

‘Visual’ The screen will flash.

‘Audible’ An audible signal will be generated.

‘Visual & audible’

A combination of both effects.

‘Ignore’ Nothing will happen.

‘System default’

The system beep routines will be used.

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

‘Bell’ A bell signal will be given.

‘Screen’ The ‘term’ screen will be brought to the front.

‘Bell/Screen’

A combination of the two actions above.

‘None’ Nothing will happen.

‘Display size’

Usually, the number of characters to be displayed on screen is determined by the physical dimensions of the main screen. But now and then a BBS requires the screen to be of a fixed size, here is what this switch allows you to set it to:

‘Screen Size’

Will fit as many characters on the screen as the physical screen size allows.

‘80 x 24 (VT100)’

As required by VT100, this will limit the number of characters to be displayed on screen to the typical 80 x 24 rectangle.

‘80 x 25 (ANSI)’

Similar to VT100, ANSI uses a fixed screen width but supports one more display line than VT100.

In each case, using this switch will not change the physical dimensions of the main screen but the limits in which the cursor can be moved.

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

‘Amiga’ Four colours and blinking.

‘8 Colours (Blink)’

Eight colours and blinking.

‘16 Colours’

Sixteen colours, as the EGA-palette.

‘Mono’ Monochrome, two colours.

‘Emulation’

At this point the user may select the terminal emulation. First there is a solid VT102/VT220/ANSI emulation which supports all common commandsequences, then

there is a so called ‘Atomic’ emulation which does only support the very basic command-sequences (CR, LF, Backspace, etc.), all VT102- and ANSI-command-sequences are suppressed, and, last but not least, a ‘TTY’-emulation which does not suppress all command-sequences, as ‘Atomic’ does, but rather echoes them to the screen.

In addition to the built-in terminal emulation, ‘term’ also supports external terminal emulations. To select one, set the terminal emulation type to ‘External’ and enter the name of the emulation in the string gadget below (the file may alternatively be selected by hitting the ‘Select new emulation’ button). In case anything goes wrong loading or initializing the terminal emulation library, ‘term’ will switch back to the built-in ‘ANSI/VT’ emulation.

‘Emu. name’

The name of an external terminal emulation library to be used by ‘term’ instead of the built-in emulation code.

‘Select new emulation’

Press this button to invoke a file requester allowing you to select the external terminal emulation file to be used by term instead of the built-in terminal emulation.

‘KeyMap name’

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

At the time of this writing the program will not support custom keymap layouts with the Chapter 15 [Packet window], page 59 due to operating system limitations.

‘Use’ Use the current settings.

‘Cancel’ Keep the old settings.

10.5 Command panel

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

‘Startup/Login macro’

At the beginning of every session with ‘term’ and after a connection has been established by the dialing routine a command-sequence is executed which may execute command-files or perform a automatical ‘login’.

‘Logoff macro’

The macro to execute when the line is hung up or the carrier signal is lost.

‘Upload macro’

‘term’ will execute this macro after a successful upload has been made.

‘Download macro’

‘term’ will execute this macro after a successful download has been made.

10.6 Miscellaneous panel

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

‘Program priority’

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

‘Override transfer path’

Each batch file transfer protocol allows you to specify the name of the directory to place the files it receives in. By default ‘term’ will redirect the files to a directory to be specified in the Section 10.7 [Path panel], page 35. If this switch is disabled, the internal settings of the current transfer protocol will be used.

‘Backup configuration’

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

‘Use auto upload panel’

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

‘Set ‘archived’ bit’

If enabled, this switch will cause ‘term’ to set the **Archived** bit on files transferred using the XPR-interface.

‘Clipboard unit’

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

‘Clipboard character delay’

When pasting the clipboard contents this number determines how many seconds to wait before sending the next character.

‘Clipboard line delay’

When pasting the clipboard contents this number determines how many seconds to wait before sending the line-termination character (carriage return).

‘Maximum buffer size’

To save memory, a high-water mark concerning the maximum amount of memory the Chapter 11 [Text buffer], page 51 will allocate for text may be specified. The minimum value to be entered here is 2,000 bytes which are roughly equivalent to two Chapter 11 [Text buffer], page 51 pages of text. A value of 0 will cause the Chapter 11 [Text buffer], page 51 to always allocate as much memory as required to buffer all the incoming text.

‘Buffer enabled’

This switch works in conjunction with the ‘Freeze buffer’ menu entry (freezing the Chapter 11 [Text buffer], page 51 contents). In fact, the menu entry is adjusted according to the configuration settings whenever a new configuration is invoked (that is, whenever a new connection is made through the dialing panel or at program startup time). *‘term’ will only freeze the buffer if this switch is set, it will leave the buffer state (frozen or not) untouched if this switch is not enabled.*

‘Show fast! macros’

If this switch is enabled, the fast! macro panel will be opened whenever the current configuration becomes active (e.g. at startup time).

10.7 Path panel

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

‘Config. storage dir’

The directory that will contain all configuration files (Phonebook, Makrokeys, etc.). The default configuration file is called ‘**term.prefs**’ and will be searched in the path defined by the environment variable ‘**TERMPATH**’ (see Chapter 16 [Environment variables], page 61).

‘??? Up/Downloadpath’

The directories in which the functions contained in the ‘Transfer’ menu will search and create files.

‘Capture path’

This gadgets contain the path in which the the capture files will be created if ‘Connect auto-capture’ is enabled.

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

If the ‘Create call info’ feature is enabled, ‘term’ will create the ‘`term-call.log`’ file in the directory given here.

‘Text editor’

Contains the name and search path of the editor used by ‘Edit & upload text file’ in the Section 9.5 [Transfer], page 19 menu.

‘Terminal ‘beep’ sound’

The name of the IFF-sound-file that will be played if a BEL-signal is detected. This file may for technical reasons not be larger than 102,400 bytes.

‘Use’ Use the current settings.

‘Cancel’ Keep the old settings.

If any ‘Select’ button is pressed, the path or filename beside can be selected by the file-requester.

10.8 Macro panel

This menu is used to set user definable strings for all ten function keys. All strings are considered Chapter 13 [Command sequences], page 55, a topic which will be covered later in this document.

‘Modifier’

All in all 40 Keys may be covered with user defined Chapter 13 [Command sequences], page 55. As the Amiga keyboard only has ten function-keys this gadget switches between the modifier keys (**Shift**, **Control**, **Alt**) which, if pressed in addition which a function key, will execute one of the 40 command sequences.

‘Load’ Load the macro-keys from a file.

‘Save’ Save the macro-keys to a file.

‘Use’ Use the current settings.

‘Cancel’ Keep the old settings.

As the definition of the function keys with command sequences contradicts the standard definition of the four functions keys of a VT-100-terminal, the keys F1-F4, which may be executed by

pressing the **Shift** key and the appropriate 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.

10.9 Fast macro panel

The design and implementation of the settings to be configured in this menu are closely related to the menu entry ‘Section 10.8 [Macro panel], page 36’ 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 14 [Fast! macros], page 57).

‘Macro list’

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

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

‘Macro code’

The command sequence (see Chapter 13 [Command sequences], page 55) associated with a fast! macro. Command sequences are discussed later in this document.

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

‘Delete’ Removes the currently selected macro from the list.

‘Load’ Loads the macro list from a file.

‘Save’ Saves the macro list to a file.

‘Top’ Places the currently selected macro at the top of the list.

‘Up’ Moves the currently selected macro one entry up.

‘Down’ Moves the currently selected macro one entry down.

‘End’ Places the currently selected macro at the end of the list.

10.10 Hotkey panel

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

'term screen to front'

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

'Buffer screen to front'

The keys to press to bring the screen of the Chapter 11 [Text buffer], page 51 process to the front.

'Skip dial entry'

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

'Abort ARexx command'

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

'Commodity priority'

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

'Hotkeys enabled'

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

'Load' Loads the hotkey settings from a file.

'Save' Saves the hotkey settings to a file.

'Use' Use the current settings.

'Cancel' Keep the original settings.

If any changes made here do not seem to take effect immediately, the user has probably typed a keyword wrong (causing the entire hotkey setup to fail). I'm sorry about this, but the current implementation of hotkey support does not sport error checking.

10.11 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.

'Rate' Speaking speed in words per minute.

'Pitch' The greater this value, the higher the voice appears to speak.

‘Frequency’	Voice frequency in Hertz.
‘Volume’	The volume of the voice.
‘Sex’	Enabled female or male voice.
‘Speech enabled’	Toggles the activity of the speech synthesizer.
‘Speak!’	Speaks a small sample text, note that speech must be enabled for this function to work.
‘Load’	Loads the speech settings from a file.
‘Save’	Saves the speech settings to a file.
‘Use’	Use the current settings.
‘Cancel’	Keep the original settings.

10.12 Emulation panel

The gadgets 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 supported by the external terminal emulation module.

‘Smooth Scrolling’	If active, the each carriage-return code will cause the contents of the screen to scroll up smoothly.
‘Wrap characters’	This function activates the automatical carriage return function which is triggered as soon as the cursor crosses the right screen margin. To avoid unpleasant side-effects, this gadget should be activated all the time.
‘Wrap cursor’	According to the VT100-specification the cursor movements have to stop at the edges of the screen. In spite of this the cursor may leave these borders, especially in ANSI-mode, and may appear at the other side of the screen. This gadget activates a more ‘tolerant’ mode.
‘Insert mode’	Normally, ‘term’ is in overwrite-mode (characters entered overwrite the contents of the screen). If this gadget is activated, typed characters are inserted by pushing all the characters right of the cursor towards the right margin.

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

‘New line mode’

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

‘Numeric keypad’

If this mode is activated the keys of the numeric keypad transmit a command-sequence instead of a number. If the remote computer needs these sequences, this mode – applications mode – can be activated with this gadget.

‘Cursor keys’

If this mode is active the cursor keys transmit a command-sequence instead of a cursor move event. If the remote computer needs these sequences, this mode – applications mode – can be activated with this gadget.

‘Font scale’

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

‘Normal’ The normal height of the font.

‘Half width’

Half width of font.

‘Use’ Use the current settings.

‘Cancel’ Keep the old settings.

10.13 Transfer panel

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

The following information is displayed in the transfer window:

‘Protocol’

The name of the transfer protocol currently running.

‘File’

The name of the file being transferred.

'File size'

If available, the size of the file.

'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 propable 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.

'Bytes transferred'

Number of bytes transferred yet.

'Blocks transferred'

Number of data blocks transferred yet.

'Block check type'

The method employed to verify the integrity of the data blocks being transferred (this usually is a form of cyclic redundancy checking).

'Block size'

Size of a data block in bytes.

'Estimated time'

The time the transfer protocol expects the transfer will take.

'Elapsed time'

The time elapsed during transfer.

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

'Information'

A list to contain the last error occured or the name of the last file transferred.

'Number of errors'

The number of errors occured during file transfer.

'Number of timeouts'

The number of timeouts occured during file transfer.

'Packet type'

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

'Packet delay'

The delay between two packets being sent.

‘Characters/second’

The effective transfer speed in characters per second.

‘Character delay’

The delay between two character being sent.

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

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

Some transfer protocols will allow you to enter a default receive path the library is supposed to create files it receives in. On request (see Section 10.6 [Miscellaneous panel], page 34) ‘term’ will ignore these settings and use the settings to be changed in the ‘Settings/Paths’ (see Section 10.7 [Path panel], page 35) menu instead.

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

10.14 Phonebook

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

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

‘Name list’

The names of all phonebook entries are displayed here.

‘BBS name’ Name of the last selected telephone book entry.

'Phone number'

The telephone number of the last selected telephone book entry.

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

The '|' character also works for the modem init, modem exit and dial prefix sequences. Whenever the dialing routine dials another phone number from a list separated by bars, it will try to find a matching init/exit/dial prefix sequence. If more phone numbers are specified than sequences are available, it will use the last sequence given (an example: a phone number may be given as '123456|654321|12345', the dial prefix string 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').

'New entry'

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

'Clone entry'

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

'Remove entry'

Removes the last selected telephone book entry from the telephone book and frees the memory allocated for this entry.

'Copy config'

A lot of time can be saved by copying selected parts of the global configuration to a local configuration which is part of a phonebook entry. Selecting this button will invoke a control panel which allows to select which parts of the global configuration should be copied. The control panel also remembers which parts were copied when it was invoked the last time.

If the 'Copy to all' button is selected, the configuration data to be copied will be copied to all phonebook entries.

'Dial list'

Will pass the list of currently marked phonebook entries to the dialing routine.

'Clear list'

Will clear the marks set for each telephone entry in the dialing list.

'Play number'

Plays the currently selected telephone number using touch-tone signal coding.

‘Use entry’

Takes over the local configuration settings saved with the currently selected phone book entry.

‘Load list’

Loads the contents of a telephone book from a file.

‘Sort list’

If any phonebook entries have been selected to be dialed, the phonebook entries will be sorted in the order of dialing. The remaining phonebook entries will be sorted in ascending alphabetical order.

‘Save list’

Saves the contents of a telephone book to a file.

‘Password’

Press this button if you wish to save a special access password with the currently active telephone book file. You will then be asked to enter the password (*what you type will appear on the screen!*), only the first 20 characters will be used.

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

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

Whenever an encrypted phonebook file is loaded and the password saved with it does not match the last password you entered, you will be asked to enter the access password. What you type will not appear on the screen, press the <Return> key when done, the <Backspace> key will delete a character, the <Escape> key will abort the process, as will closing the window. The input process will be terminated automatically when 20 characters have been entered.

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

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

‘Serial’

‘Modem’

‘Transfer’

‘Screen’

‘Terminal’

‘Commands’

‘Miscellaneous’

‘Paths’

‘Keyboard macros’

‘Fast! Macros’

These entries refer directly to the settings main menu entries of the same name.

‘Password/User’

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 10.15 [Rate panel], page 45).

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

To put a phonebook entry into the dialing list, double-click the mouse on 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, double-click it again. Instead of double-clicking on an entry, the space bar may be pressed as well.

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

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

10.15 Rate panel

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

‘Pay/unit’

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

‘Seconds/unit’

This is where you enter how many seconds each time unit lasts.

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

‘Days and dates’

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

‘Day(s)’ Settings for certain days of the week

‘12. Jan (example)’

Settings for a specific date

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

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

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

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

‘Add day(s)’

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

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

‘Replace units’

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

‘Append units’

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

‘Cancel’ Will abort the selection.

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

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

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

‘Remove’ Removes an entry from the list.

10.16 Copy panel

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

‘All’ Selects all parts.

‘Clear’ Clears the current selection.

‘Copy to all’

The selected parts will be copied to all phone book entries.

10.17 Dial panel

The following information about the dialing process is displayed:

‘Calling’ The name of the telephone book entry belonging to the number being dialled. If it is just a telephone number the string ‘-- Unknown --’ is shown, meaning that the name of the mailbox is unknown.

‘Number’ The telephone number being dialed or just dialed.

'Next'	The name of the telephone book entry which will be processed next if no connection is established. If no further entry exists, '-- None --' will be displayed.
'Timeout'	A counter which is decreased every second and which reflects the time remaining to establish a connection or to cycle through the dial queue again.
'Attempt'	This field shows the number of unsuccessful cycles made through the dialing queue to establish a connection.
'Message'	A message to the user. This can be: 'Dialing...' A dial is in process. 'Line is busy.' The dialed number is engaged. 'Incoming call!' The modem has been called from another modem. 'Incoming voice call!' The modem is receiving a call which was not originated by another modem. 'No dialtone detected!' The modem was unable to detect any dialing tone on the line, it may possibly be not connected. 'Connection established.' Just as the name says... 'Maximum number of dial retries reached!' Just as the name says... 'Dial attempt timeout.' The time available to establish a connection has been reached or exceeded. 'Redial Delay...' Pause until the next cycle through the dialing queue.

Additionally three buttons can be found in the input field which initiate the following functions.

'Skip call'

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

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

‘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 has been made but the **CONNECT** signal has been lost, you can use this button to switch directly to the online mode of the modem. This will also start the pay unit counter and the dial queue will be exited.

‘Abort dialing’

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

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.

11 Text buffer

The text buffer is a function which continually stores text displayed on ‘term’s main screen, so that the user can inform himself about the received text at any time.

11.1 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.

11.2 Operation

The contents of the text buffer can be paged through using the keys for moving of the cursor (**Cursor keys+Shift** keys moves page by page, **Cursor keys + Control** 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).

Additionally there is another menu whose functions should be explained briefly:

‘Search’ A search function is called which searches from the topmost line on the screen downwards for an entered search string. Upper- and lowercase are ignored by this function. If the search string is found it is displayed and highlighted.

‘Repeat search’

Continues the search process started with ‘Search’. The previously entered search string is carried over.

‘Go to main screen’

Switches to the main screen of ‘term’.

‘Clear buffer’

Clears the contents of the Chapter 11 [Text buffer], page 51.

'Close buffer, Quit'

Closes the Chapter 11 [Text buffer], page 51 screen but leaves the contents unchanged.

12 Clipboard

Cut & paste functions are available on the main screen and the buffer screen. Use the mouse to point to the first character you wish to send to the clipboard, hold down the select button, move the mouse to the last character you wish to cut and release the button. The characters you have just marked will have been transferred to the clipboard, use the key combination **Amiga+V** to paste the contents of the clipboard and to feed them back into the input stream. 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. Hold down any **Shift** key to have the contents of the clipboard fed into the input stream right after the mouse button is released, a carriage return will follow the data (this also works on the Chapter 11 [Text buffer], page 51 screen). The middle mouse button has the same function as if the left mouse button when pressed along with a **Shift** key.

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 string editing 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 string editing 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).

Clipboard functions are also available within string gadgets: press **Amiga+C** to copy the contents of the gadget to the clipboard and press **Amiga+V** to insert the current clipboard contents at the current cursor position.

13 Command sequences

Each string 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.

13.1 Backslash

'\'	Generates a single backslash.
'\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 string to follow this character in the clipboard.
'\i'	Feeds the contents of the clipboard into the input stream.
'\n'	Generates a line feed.
'\p'	Feeds the password of the telephone book entry into the input stream of the last made connection. <i>The password is automatically cleared for security reasons when the connection is broken.</i>
'\r'	Generates a carriage return.
'\t'	Generates a tab jump.
'\u'	Similar to the \p command, the \u command will feed the current user name into the input stream.
'\x'	Generates a break signal (as with the 'Send break' menu entry).
'\^'	Generates a caret character.

If none of the mentioned combinations is recognized the character which follows the \ will be fed into the input stream without any changes.

13.2 Caret

This character is used to change the following character to a 'control character'. So the sequence '^J' will become a Line feed and '^H' becomes a tab jump. The character which follows the '^' has to be located between @ and [, otherwise it is fed into the input stream without changes.

13.3 Tilde

This character causes the program to pause for exactly half a second before it continues to process the following commands.

14 Fast! macros

In implementation and design the fast! macros are closely related to the function key macros (see Section 10.8 [Macro panel], page 36). 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 Section 10.9 [Fast macro panel], page 37). When a list entry is selected, the associated command sequence (see Chapter 13 [Command sequences], page 55) will be executed.

By using the fast! macros it is theoretically possible to control a BBS just by mouse, provided that you have the appropriate 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.

15 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:

'Left+Control'

Jumps to the next word.

'Right+Control'

Jumps to the previous word.

'Up' Shows the last entered command in the input line.

'Up+Shift'

Shows the very first command entered so far.

'Down' Shows the next entered command (if you moved back for some commands before).

'Down+Shift'

Shows the very last command entered so far.

This string 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 Chapter 11 [Text buffer], page 51 this buffer is managed dynamically. The same memory restrictions that apply to the Chapter 11 [Text buffer], page 51 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'.

'Quit' Closes the window (corresponds to clicking the close gadget of the window).

Every character entered into this window is shown immediately so that those things where it is better that they should not appear on the screen (like passwords for a mailbox) should be entered in another way.

The content of every input line is interpreted as a command sequence and therefore can also contain control characters.

If a line taken from the input buffer is sent without change it is *not* stored in the buffer again ('true history' as known from 'ConMan').

The contents of the input buffer are cleared automatically after the window is closed. *Under no circumstances are the contents maintained until the next call!*

Provided that the packet window is large enough, a list to contain the command line history will be displayed.

16 Environment variables

Information which is to be available the next time the program is run is placed as AmigaDOS variables in the directory 'ENV:' and 'ENVARC:' by 'term'.

The variables used by 'term' can be used and manipulated by other programs transparently. In detail these variables are:

'TERMPATH'

The name of the directory in which all information used by 'term' is placed (standard configuration, telephone book, etc.). The default settings file name is 'ENVARC:term'.

'TERMWINDOW'

The window definition which can also be entered in the program via the menu function 'Set console window'.

'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.).

17 Gift-Ware fee

The development of ‘term’ required considerable expense. The effort to find out by trial and error and with a little help from the DevCon-Disks and the Includes & Autodocs the operating principles of some routines would alone be worth a remuneration to the author (it’s simply bad luck if the Atlanta-DevCon records arrive with considerable delay). The author is susceptible to all kinds of donation and gifts so long as it is assured that they are useful to him (these nice little sacks with elephant food are better sent to the Hanover zoo). Therefore anyone pleased by ‘term’ or who works with the source code is welcome to remunerate my programming efforts as he/she thinks that it matches my efforts (If you’ve got no ideas yet: I am looking for the following Infocom games: *Arthur and the quest for Excalibur*, *Border zone*, *Bureaucracy*, *Cutthroats*, *Leather Goddesses of Phobos*, *Plundered Hearts*, *Seastalker*, *Sherlock - The Riddle of the Crown Jewels*, and *Trinity*; any formats are welcome (Amiga, Atari ST, MS-DOS, C64, etc.) – old CinemaWare games – with the exception of *Defender of the Crown* and *Sinbad* – the source code to ‘ARJ’, a Modula-2 compiler, the complete ‘GNU Emacs’ or the complete ‘GNU C’ compiler) - a lot of thanks!

My address:

Olaf Barthel

Brabeckstrasse 35

D-3000 Hannover 71

Federal Republic of Germany

Z-Netz: O.BARTHEL@A-LINK-H

Internet: olsen@sourcery.mxm.sub.org, o.barthel@a-link-h.comlink.de

Would those people looking at the source code to extend it or do some modifications here and there please contact me before publication so that our efforts can be coordinated.

Index

?

??? Up/Downloadpath 35

‘

‘Shanghai’ windows 29

\

\^ 55

\\ 55

\a 55

\b 55

\c 55

\d 55

\e 55

\f 55

\g 55

\i 55

\n 55

\p 55

\r 55

\t 55

\u 55

\x 55

1

12. Jan (example) 46

16 Colours 32

8

8 Colours (Blink) 32

80 x 24 (VT100) 32

80 x 25 (ANSI) 32

A

Abort ARexx command 38

Abort dialing 49

About 17

Add date 46

Add day(s) 46

Alert 32

All 47

Amiga 32

Answerback message 31

Append units 47

As CR 31

As CR+LF 31

As LF 31

As LF+CR 31

Attempt 48

Audible 31

B

Backup configuration 34

Baud rate 15, 25

BBS name 42

Bell 31, 32

Bell/Screen 32

Bits/char 25

Block check type 41

Block size 41

Blocks transferred 41

Break length 26

Buffer enabled 35

Buffer screen to front 38

Buffer size 25

Buffer size (bytes) 23

Busy string 27

Bytes received 22

Bytes sent 22

Bytes transferred 41

C

Calling 47

Cancel 26, 28, 29, 33, 36, 38, 39, 40, 47

Capture filter 30

Capture path 36

Capture to File/Printer 17

Change emulation parameters 21

Character delay	42	Disabled	30
Characters/second	42	Display buffer	20
Check carrier	28	Display modes	29
Clear	47	Display size	32
Clear buffer	20, 51	Down	37, 59
Clear history	59	Down+Shift	59
Clear list	43	Download ASCII file(s)	19
Clear screen	21	Download binary file(s)	20
Clipboard character delay	34	Download macro	34
Clipboard line delay	35	Download text file(s)	19
Clipboard unit	34	Drop DTR on hangup	28
Clone	47	Duplex	25
Clone entry	43		
Close buffer	20	E	
Close buffer, Quit	52	Edit	47
Colour	32	Edit & upload text file	19
Commands	22, 44	Edit screen palette	29
Commodity priority	38	Elapsed time	41
Compact	30	Emu. name	33
Config. storage dir	35	Emulation	15, 32
Connect auto-baud	28	End	37
Connect auto-capture	28	Estimated time	41
Connect string	27	Execute AmigaDOS command	18
Connection established	48	Execute ARexx command	18
Connection message	22		
Copy config	43	F	
Copy to all	47	Fast! macro settings	22
Create call info	28	Fast! macros	23
Cursor keys	40	Fast! Macros	45
		Faster layout	30
D		File	40
Day(s)	46	File size	41
Days and dates	46	Flush receive buffer	19
Delete	37	Font	15, 31
Destructive BS	30	Font scale	40
Device unit number	26	Free memory (bytes)	23
Dial	18	Freeze buffer	20
Dial attempt timeout	48	Frequency	39
Dial list	43		
Dial prefix	27	G	
Dial suffix	27	Go to main screen	51
Dial timeout	27	Go to online	49
Dialing	48		

H

Half width.....	40
Handshaking.....	25
Hang up.....	19
High-speed mode.....	25
Hotkeys.....	22
Hotkeys enabled.....	38

I

IBM.....	31
Iconify.....	17
Ignore.....	31, 32
Import.....	46
Incoming call!.....	48
Incoming voice call!.....	48
Information.....	41
Insert mode.....	39

K

Keyboard macros.....	44
KeyMap name.....	33

L

Left+Control.....	59
Line is busy.....	48
Load.....	36, 37, 38, 39
Load history.....	59
Load list.....	44
Log actions.....	28
Log file.....	36
Logoff macro.....	33

M

Macro.....	37
Macro code.....	37
Macro list.....	37
Macro settings.....	22
Make screen public.....	29
Maximum buffer size.....	35
Maximum number of dial retries reached!.....	48
Message.....	41, 48
Miscellaneous.....	22, 44
Modem.....	44

Modem exit string.....	26
Modem hangup string.....	27
Modem init string.....	26
Modem settings.....	21
Modifier.....	36
Mono.....	32

N

Name list.....	42
New.....	37
New entry.....	43
New line mode.....	40
Next.....	48
No carrier string.....	27
No dialtone detected!.....	48
No dialtone string.....	27
None.....	32
Normal.....	40
Number.....	47
Number of dial retries.....	27
Number of errors.....	41
Number of timeouts.....	41
Numeric keypad.....	40

O

Online.....	15
Open buffer.....	20
Other window.....	59
Override transfer path.....	34

P

Packet.....	23
Packet delay.....	41
Packet type.....	41
Parameters.....	15
Parity.....	25
Pass xON/xOFF through.....	28
Password.....	44
Password/User.....	45
Paste.....	18
Paths.....	22, 44
Pay/unit.....	45
Phone number.....	43

Phonebook.....	18	Screen title.....	30
Pitch.....	38	Search.....	51
Play number.....	43	Seconds/unit.....	46
Play phone number.....	18	Select new device.....	26
Print Screen/Clipboard.....	17	Select new emulation.....	33
Program priority.....	34	Select new screen font.....	29
Protocol.....	15, 40	Select new text font.....	31
 		Send break.....	18
Q		Send CR.....	31
Quit.....	17, 60	Send LF.....	31
 		Serial.....	44
R		Serial device.....	26
Rate.....	38	Serial settings.....	21
Rates.....	45	Session start.....	22
Receive CR.....	31	Set 'archived' bit.....	34
Receive LF.....	31	Set console window.....	18
Red/Green/Blue.....	29	Sex.....	39
Redial.....	18	Shared access.....	25
Redial Delay.....	27	Show fast! macros.....	35
Redial Delay... ..	48	Skip call.....	48
Release serial device.....	19	Skip dial entry.....	38
Remove.....	47, 49	Smooth Scrolling.....	39
Remove entry.....	43	Sort list.....	44
Repeat search.....	51	Space left.....	41
Replace units.....	47	Speak!.....	39
Reset font.....	21	Speech.....	22
Reset styles.....	21	Speech enabled.....	39
Reset terminal.....	21	Standard.....	30, 31
Review.....	23	Startup/Login macro.....	33
Right+Control.....	59	Status.....	15, 22
Ring string.....	27	Status line.....	30
 		Stop bits.....	25
S		Stop blinking.....	30
Save.....	36, 37, 38, 39	Strip bit 8.....	30
Save buffer as.....	20	Swap DEL <-> BS.....	30
Save history as.....	59	System default.....	32
Save list.....	44	 	
Save screen as Picture/Text.....	17	T	
Screen.....	32, 44	term screen to front.....	38
Screen font.....	29	Terminal.....	44
Screen settings.....	21	Terminal 'beep' sound.....	36
Screen size.....	23	Terminal settings.....	22
Screen Size.....	32	TERMPATH.....	61

TERMWINDOW.....	61	Use default colours.....	29
Text editor.....	36	Use entry.....	44
Text font.....	31		
Time.....	15	V	
Timeout.....	48	Visual.....	31
Top.....	37	Visual & audible.....	31
Transfer.....	22, 44	Voice string.....	27
Transfer protocol.....	22	Volume.....	39
U		W	
Up.....	37, 59	Wrap characters.....	39
Up+Shift.....	59	Wrap cursor.....	39
Upload ASCII file(s).....	19		
Upload binary file(s).....	20	X	
Upload macro.....	33	xem.....	61
Upload text file(s).....	19	xON/xOFF.....	25
Use.....	26, 28, 29, 33, 36, 38, 39, 40	xpr.....	61
Use auto upload panel.....	34		

Table of Contents

1	Important information	1
1.1	Name changes	1
1.2	The carrier signal	2
1.3	High-speed mode	2
2	Background	3
3	Acknowledgements	5
4	Source code	7
5	Program updates	9
6	Operating the program	11
7	Keyboard control	13
8	Screen	15
9	Menus	17
9.1	Project	17
9.2	Edit	17
9.3	Commands	18
9.4	Phone	18
9.5	Transfer	19
9.6	Buffer	20
9.7	Terminal	21
9.8	Settings	21
9.9	Windows	22
10	Control panels	25
10.1	Serial panel	25
10.2	Modem panel	26
10.3	Screen panel	28
10.4	Terminal panel	29

10.5	Command panel	33
10.6	Miscellaneous panel	34
10.7	Path panel	35
10.8	Macro panel	36
10.9	Fast macro panel	37
10.10	Hotkey panel	37
10.11	Speech panel	38
10.12	Emulation panel	39
10.13	Transfer panel	40
10.14	Phonebook	42
10.15	Rate panel	45
10.16	Copy panel	47
10.17	Dial panel	47
11	Text buffer	51
11.1	General characteristics	51
11.2	Operation	51
12	Clipboard	53
13	Command sequences	55
13.1	Backslash	55
13.2	Caret	56
13.3	Tilde	56
14	Fast! macros	57
15	Packet window	59
16	Environment variables	61
17	Gift-Ware fee	63
	Index	65