NAME

mintty - Cygwin terminal emulator

SYNOPSIS

mintty [*OPTION*]... [**-** | *PROGRAM* [*ARG*]...]

DESCRIPTION

Mintty is a terminal emulator for Cygwin with a native Windows user interface and minimalist design. Its terminal emulation is largely compatible with **xterm**, but it does not require an X server.

INVOCATION

If a program name is supplied on the command line, this is executed with any additional arguments given. Otherwise, mintty looks for a shell to execute in the *SHELL* environment variable. If that is not set, it reads the user's default shell setting from */etc/passwd*. As a last resort, it falls back to */bin/sh*. If a single dash is specified instead of a program name, the shell is invoked as a login shell.

OPTIONS

The standard GNU option formats are accepted, with single dashes introducing short options and double dashes introducing long options.

-c, **--config** *FILENAME*

Read settings from the specified configuration file, in addition to */etc/minttyrc* and *˜/.minttyrc*.

-e, **--exec** *PROGRAM* [*ARG* ...]

Execute the specified program in the terminal session and pass on any additional arguments.

This option is present for compatibility with other terminal emulators only. It can be omitted, in which case the first non-option argument, if any, is taken as the name of the program to execute.

-h, **--hold never**|**start**|**error**|**always**

Determine whether to keep the terminal window open when the command has finished and no more processes are connected to the terminal. The argument can be abbreviated to a single letter.

By default, the window is closed immediately, except if the child process has exited with status 255, which is used to indicate failure to execute the shell command. (Exit status 255 is also used by **ssh** to indicate connection errors.)

Alternatively, the window can be set to never stay open, to always stay open, or to stay open only if the child process terminates with an error, i.e. with a non-zero exit status or due to a signal indicating a runtime error.

-i, **--icon** *FILE*[**,***INDEX*]

Load the window icon from an executable, DLL, or icon file. The optional comma-separated index can be used to select a particular icon in a file with multiple icons.

-l, **--log** *FILE*
Copy all output into the specified log file. (See also *script*(1) for a more flexible logging tool.)

-o, **--option** *NAME*=*VALUE*
Override the named config file option with the given value, e.g. `-o Scrollback-Lines=1000`.

-p, **--position** *X*,*Y*
Open the window with its top left corner at the specified coordinates.

-s, **--size** *COLS*,*ROWS*
Set the default size of the window in character columns and rows.

-t, **--title** *TITLE*
Use *TITLE* as the initial window title. By default, the title is set to the executed command.

-u, **--utmp**
Create a utmp entry.

-w, **--window normal**|**min**|**max**|**full**|**hide**
Set the initial window state: normal, minimised, maximised, full screen, or hidden.

--class *CLASS*
Use *CLASS* as the window class name of the main window. This allows scripting tools to distinguish different mintty instances. The default is "mintty".

-H, **--help**
Display a brief help message and exit.

-V, **--version**
Print version information and exit.

USAGE

Mintty tries to adhere to both Windows and Unix usage conventions. Where they conflict, an option is usually provided. This section primarily describes the default configuration; see the **CONFIGURATION** section on how it can be customised.

Menus

The context menu can be opened by right-clicking the mouse or by pressing the **Menu** key that is normally located next to the right Ctrl key.

Mintty also adds a couple of items to the window menu, which can be accessed by clicking on the program icon or pressing **Alt+Space**.

Both menus have an entry that leads to the options dialog for changing mintty's configuration.

Copy & paste

Screen contents can be selected by holding down the left mouse button and dragging the mouse. If Alt is held down before the left mouse button, a rectangular block instead of whole lines will be selected. The selection can be extended by holding down **Shift** while left-clicking. Double-clicking or triple-clicking selects a whole word or line, whereby word selection includes special characters that commonly appear in file names and URLs.

By default, selected text is automatically copied to the clipboard. This can be disabled on the **Mouse** page of the options dialog. Selected text can also be copied manually using either the **Copy** menu command, the **Ctrl+Ins** keyboard shortcut, or the middle mouse button combined with **Shift**.

The selected region is copied as "rich text" as well as normal text, which means it can be pasted with colours and formatting into applications that support it, e.g. word processors.

The window title can be copied using the **Copy Title** command in the window menu.

The clipboard contents can be pasted using either the **Paste** menu command, the **Shift+Ins** keyboard shortcut, or the middle mouse button. Not only text but also files and directories can be pasted, whereby the latter are inserted as Cygwin filenames. Shell quoting is added to filenames that contain spaces or special characters.

Drag & drop

Text, files and directories can be dropped into the mintty window. They are inserted in the same way as if they were pasted from the clipboard.

Opening files, directories and URLs

Files, directories and URLs can be opened either by holding **Ctrl** while left-clicking on them, or by selecting them and choosing the **Open** command from the context menu. Please note that opening a file or directory with a relative path only works correctly if the path refers to the current working directory of the process invoked by mintty.

Font zoom

The font size can be increased or decreased using the keyboard shortcuts **Ctrl+plus** and **Ctrl+minus**, or by holding **Ctrl** while rolling the mousewheel. **Ctrl+zero** returns the font size to the default.

Full screen

Full screen mode can be toggled using either the **Full Screen** command in the menu or either of the **Alt+Enter** and **Alt+F11** keyboard shortcuts.

Default size

If the window has been resized, it can be returned to the default size set in the Window pane of the options using the **Default size** command in the menu or the **Alt+F10** shortcut.

Reset

Sometimes a faulty application or printing a binary file will leave the terminal in an unusable state. In that case, resetting the terminal's state via the **Reset** command in the menu or the **Alt+F8** keyboard shortcut may help.

Scrolling

Mintty has a scrollback buffer that can hold up to 10000 lines in the default configuration. It can be accessed using the scrollbar, the mouse wheel, or the keyboard. Hold the **Shift** key while pressing the **Up** and **Down** arrow keys to scroll line-by-line or the **PageUp** and **PageDown** keys to scroll page-by-page.

Flip screen

Applications such as editors and file viewers normally use a terminal feature called the alternate screen, which is a second screen buffer without scrollback. When they exit, they switch back to the primary screen to restore the command line as it was before invoking the application.

The **Flip Screen** menu command and **Alt+F12** shortcut allow looking at the primary screen while the alternate screen is active, and vice versa. For example, this allows to refer to past commands while editing a file.

Switching session

The **Ctrl+Tab** and **Ctrl+Shift+Tab** shortcuts can be used to switch between mintty windows. Minimised windows are skipped.

Closing a session

Clicking the window's close button, pressing **Alt+F4**, or choosing **Close** from the window menu sends a *SIGHUP* signal to the process running in mintty, which normally causes it to exit.

That signal can be ignored, though, in which case the program might have to be forced to terminate by sending a *SIGKILL* signal instead. This can be done by holding down **Shift** when using the close button, shortcut or menu item.

Mouse tracking

When an application activates mouse tracking, mouse events are sent to the application rather than being treated as window events. This is indicated by the mouse pointer changing from an **I** shape to an arrow. Holding down **Shift** overrides mouse tracking mode and sends mouse events to the window instead, so that e.g. text can be selected and the context menu can be accessed.

Alt codes

The Windows Alt+Numpad method for entering character codes is supported, whereby the Alt key has to be held while entering the character code. Only the first key has to be on the numpad; subsequent digits can be entered both on the numpad or the main part of the keyboard.

If the first key is the **'+'** on the numpad, the code is interpreted as hexadecimal, whereby digits A through F can be entered using the letter keys. If the first key is a zero, the code is interpreted as octal. If the first key is any other digit from 1 to 9, the code is interpreted as decimal.

For UTF-8 and other Unicode encodings such as GB18030, the entered code is interpreted as a Unicode codepoint and encoded accordingly before it is sent. For other encodings, the entered code is sent as is. If it doesn't fit into one byte, it is sent as multiple bytes, with the the most significant non-zero byte first.

Shortcuts

An overview of all the keyboard shortcuts.

Scrollback

- **Shift+Up**: Line up
- **Shift+Down**: Line down
- **Shift+PgUp**: Page up
- **Shift+PgDn**: Page down
- **Shift+Home**: Top
- **Shift+End**: Bottom

Copy and paste

- **Ctrl+Ins**: Copy
- **Shift+Ins**: Paste
- **Ctrl+Shift+Ins**: Copy and paste

Window commands

- **Alt+F2**: New
- **Alt+F4**: Close
- **Alt+F8**: Reset
- **Alt+F10**: Default size
- **Alt+F11** or **Alt+Enter**: Full screen
- **Alt+F12**: Flip screen
- **Alt+Space**: Window menu
- **Ctrl+Tab**: Next window
- **Ctrl+Shift+Tab**: Previous window

Font zoom

- **Ctrl+plus**: Zoom in
- **Ctrl+minus**: Zoom out
- **Ctrl+zero**: Back to configured font size

Ctrl+Shift+letter shortcuts

An alternative set of shortcuts for clipboard and window commands using **Ctrl+Shift+letter** combinations is available. These can be enabled on the Keys pane of the options dialog.

- **Ctrl+Shift+C**: Copy
- **Ctrl+Shift+V**: Paste
- **Ctrl+Shift+N**: New
- **Ctrl+Shift+W**: Close
- **Ctrl+Shift+R**: Reset
- **Ctrl+Shift+D**: Default size
- **Ctrl+Shift+F**: Full screen
- **Ctrl+Shift+S**: Flip screen

CONFIGURATION

Mintty has a graphical options dialog that can be reached via the context menu or the window menu. As usual, both the **Apply** and **OK** buttons apply any changes made, but **OK** also closes the dialog. **Cancel** discards changes.

In configuration files, settings are stored as *NAME*=*VALUE* pairs, with one per line. By default, they are read from */etc/minttyrc* and *~/.minttyrc*. Additional configuration files can be specified using the **--config** command line option. These are read in order, with settings in later files overriding those in earlier ones. Configuration changes are saved to the last file specified, or *~/.minttyrc* if none is given. Settings can also be specified on the command line using **--option**.

The following sections explain the settings on each pane of the options dialog, followed by settings that do not appear in the dialog. For each setting, its name in the config file is shown in parentheses, along with its default value.

Looks

Settings affecting mintty's appearance.

Colours

Clicking on one of the buttons here opens the colour selection dialog. In the config file, colours are represented as comma-separated RGB triples with decimal 8-bit values (i.e. ranging from 0 to 255).

− **Foreground** (ForegroundColour=191,191,191)
− **Background** (BackgroundColour=0,0,0)
− **Cursor** (CursorColour=191,191,191)

Transparency (Transparency=off)

Window transparency level, with the following choices:

− **Off**
− **Low**
− **Medium**
− **High**
− **Glass**

The **Glass** option is only available on Vista and above with desktop compositing enabled. To make this reasonably usable, the glass colour needs to be set to be as dark as possible in the Windows control panel: choose *Personalize* from the desktop context menu, click on *Window Color*, turn the colour intensity up to the maximum, show the colour mixer, and turn the brightness down to black.

Numeric transparency values ranging from 4 to 254 can be specified in config files or on the command line. (Values below 4 are multiplied by 16, for backward compatibility reasons.)

Opaque when focused (OpaqueWhenFocused=no)

Enable to make the window opaque when it is active (to avoid background distractions when working in it).

Cursor (CursorType=line)

The following cursor types are available:

− **Line**
− **Block**
− **Underscore**

The line cursor is displayed with the width set in the Accessibility Options control panel.

Cursor blink (CursorBlinks=yes)

If enabled, the cursor blinks at the rate set in the Keyboard control panel.

Text

Settings controlling text display.

Font selection

Clicking on the **Select** button opens a dialog where the font and its properties can be chosen. Font styles other than **Bold** are ignored. In the config file, this corresponds to the following entries:

− **Font** (Font=Lucida Console)
− **Style** (FontIsBold=no)
− **Size** (FontHeight=9)

Show bold as colour (BoldAsFont=no)

When this option is enabled, the ANSI bold (or 'intense') text attribute is shown as a bold-style font. Where a bold variant of the selected font that has the same width as the base font is available, that is used; otherwise, the bolding is simulated by rendering the text twice with a one-pixel offset.

Show bold as colour (BoldAsColour=yes)

By default, text with the ANSI bold attribute set is displayed with a different colour, usually with increased brightness. This can be disabled here.

Note that when **BoldAsFont** is enabled, only bold text in one of the eight ANSI colours has its colour changed, i.e. bold text without an explicitly spefified colour is shown with a bold font only. This matches **xterm** behaviour.

This option also controls how the 'half-bright' (or 'dim') text attribute is displayed: if it is on, half-bright text is shown with halved foreground colour brightness; otherwise, it is shown by blending the foreground colour with the background colour.

Allow blinking (AllowBlinking=no)

When text blinking is disabled, as it is by default, the blink attribute is displayed as a bold background colour instead.

Font smoothing (FontSmoothing=default)

Select the amount of font smoothing from the following choices:

− **Default**: Use Windows setting.
− **None**: With all the jaggies.
− **Partial**: Greyscale anti-aliasing.

− **Full**: Subpixel anti-aliasing ("ClearType").

Locale (Locale=)

The locale setting consists of a lowercase two-letter or three-letter language code followed by a two-letter country code, for instance **en_US** or **zh_CN**. The Windows default system and user locales are shown in the drop-down list for this setting. Alternatively, the language-neutral "C" locale can be selected.

If no locale is set here, which is the default, mintty uses the locale and character set specified via the environment variables *LC_ALL*, *LC_CTYPE* or *LANG*.

If the locale option is set, however, it will override any environment variable setting: *LC_ALL* and the *LC_** variables for specific locale categories are cleared, while *LANG* is set according to the selected locale and character set.

Character set (Charset=)

The character set to be used for encoding input and decoding output. If no locale is set, this setting is ignored.

While changing the character set takes effect immediately for text input and ouput, it does not affect the processes already running in mintty. This is because the environment variables of a running process cannot be changed from outside that process. Therefore mintty needs to be restarted for a character set change to take full effect.

Keys

Settings controlling keyboard behaviour.

Backspace sends ˆH (BackspaceSendsBS=no)

By default, mintty sends **ˆ?** as the keycode for the backspace key. If this option is enabled, **ˆH** is sent instead. This also changes the **Ctrl+Backspace** code from **ˆ_** to **ˆ?**.

Ctrl+LeftAlt is AltGr (CtrlAltIsAltGr=no)

The AltGr key on non-US Windows systems is a strange beast: pressing it is synonymous with pressing the left Ctrl key and the right Alt key at the same time, and Windows programs usually treat any Ctrl+Alt combination as AltGr.

Some programs, however, chief among them Microsoft’s very own Office, do not treat Ctrl+LeftAlt as AltGr, so that Ctrl+LeftAlt combinations can be used in command shortcuts even when a key has an AltGr character binding.

By default, mintty follows Office’s approach, because a number of terminal programs make use of Ctrl+Alt shortcuts. The "standard" Windows behaviour can be restored by ticking the checkbox here.

The setting makes no difference for keys without AltGr key bindings (e.g. any key on the standard US layout).

Copy and Paste shortcuts (ClipShortcuts=yes)

Checkbox for enabling the clipboard shortcuts **Ctrl+Ins** for copying and **Shift+Ins** for pasting.

Menu and Full Screen shortcuts (WindowShortcuts=yes)

Checkbox for enabling the **Alt+Space** and **Alt+Enter** shortcuts for showing the window menu and toggling full screen mode.

Switch window shortcuts (SwitchShortcuts=yes)

Checkbox for enabling the **Ctrl+Tab** and **Ctrl+Shift+Tab** shortcuts for switching between mintty windows.

Zoom shortcuts (ZoomShortcuts=yes)

Checkbox for enabling the font zooming shortcuts **Ctrl+plus/minus/zero**.

Alt+Fn shortcuts (AltFnShortcuts=yes)

Checkbox for enabling the use of combinations of Alt and functions keys as shortcuts, for example **Alt+F4** for closing the window or **Alt+F11** fortoggling full screen mode. Disable to have **Alt+Fn** combinations sent to applications instead.

Ctrl+Shift+letter shortcuts (CtrlShiftShortcuts=no)

Checkbox for enabling alternative clipboard and window command shortcuts using **Ctrl+Shift+letter** combinations such as **Ctrl+Shift+V** for paste or **Ctrl+Shift+N** for starting a new session.

These can replace the **Ctrl/Shift+Ins** and **Alt+Fn** shortcuts, whereby they show up in menus only if the corresponding default shortcuts are disabled.

See the shortcuts section above for the list of shortcuts controlled by this option. When it is disabled, Ctrl+Shift+letter combinations are sent to applications as C1 control characters instead.

Mouse

Settings controlling mouse support.

Copy on select (CopyOnSelect=yes)

If enabled, the region selected with the mouse is copied to the clipboard as soon as the mouse button is released, thus emulating X Window behaviour.

Copy as rich text (CopyAsRTF=yes)

If this option is enabled, which it is by default, text is copied to the clipboard in rich text format (RTF) in addition to plain text format. RTF preserves colours and styles when pasting text into applications that support it, e.g. word processors.

Clicks place command line cursor (ClicksPlaceCursor=no)

If enabled, the command line cursor can be placed by pressing the left mouse button. This works by sending the number of cursor keycodes needed to get to the destination.

Right click action (RightClickAction=menu)

Action to take when the right mouse button is pressed. If this is set to **Paste**, the middle button extends the selection region.

− **Paste**: Paste the clipboard contents.
− **Extend**: Extend the selected region.
− **Show menu**: Show the context menu.

Default click target (ClicksTargetApp=yes)

This applies to application mouse mode, i.e. when the application activates xterm-style mouse reporting. In that mode, mouse clicks can be sent either to the application to process as it sees fit, or to the window for the usual actions such as select and paste.

− **Window**
− **Application**

Modifier key for overriding default (ClickTargetMod=shift)

The modifier key selected here can be used to override the click target in application mouse mode. With the default settings, clicks are sent to the application and Shift needs to be held to trigger window actions instead.

The **Off** setting disables overriding.

− **Shift**
− **Ctrl**
− **Alt**
− **Off**

Window

Window properties.

Columns (Columns=80)

Default width of the window, in character cells.

Rows (Rows=24)

Default height of the window, in character cells.

Current size

Pressing this button sets the default width and height to the window's current size.

Scrollback lines (ScrollbackLines=10000)

The maximum number of lines to keep in the scrollback buffer.

Scrollbar (Scrollbar=right)

The scrollbar can be shown on either side of the window or just hidden. By default, it is shown on the right-hand side.

− **Left**
− **None**
− **Right**

Modifier for scrolling (ScrollMod=shift)

The modifier key that needs to be pressed together with the arrow up/down, PgUp/PgDn, or Home/End keys to access the scrollback buffer. The default is **Shift**. The **Off** setting disables scrolling with keyboard shortcuts.

− **Shift**
− **Ctrl**

− **Alt**
− **Off**

PgUp and PgDn scroll without modifier (PgUpDnScroll=no)

If this is enabled, the scrollback buffer can be accessed by just pressing PgUp or PgDn, without the ’modifier for scrolling’ selected above. If the modifier is pressed anyway, plain PgUp/PgDn keycodes are sent to the application. This option does not affect the arrow keys or Home/End keys.

Terminal

Terminal emulation settings.

Type (Term=xterm)

The terminal type. This determines the setting of the TERM environment variable at mintty startup. Choices available from the dropdown list are **xterm**, **xterm-256color**, **xterm-vt220**, **vt220**, and **vt100**.

If the setting contains "vt220", xterm VT220-style function key mode is enabled instead of the default PC-style function key mode. (This can otherwise be set with the DECSET 1061 control sequence.)

Apart from that, this setting has no effect on mintty’s terminal emulation, i.e. all the features are always available. However, the TERM setting does tell applications what features they can use.

The **xterm-256color** setting enables 256-color mode in some applications, but may not be recognised at all by others, which is why plain **xterm** is the default.

Answerback (Answerback=)

The answerback string is sent in response to the **ˆE** (ENQ) character. By default, this is empty.

Bell The three checkboxes here determine what effects the bell character **ˆG** has. Only taskbar highlighting is enabled by default.

− **Sound** (BellSound=no): Play the system’s default beep sound.

− **Flash** (BellFlash=no): Briefly invert the foreground and background colours.

− **Highlight in taskbar** (BellTaskbar=yes): Change the colour of mintty’s taskbar entry if the mintty window is not active.

Printer (Printer=)

The ANSI standard defines control sequences for sending text to a printer, which are used by some terminal applications such as the mail reader **pine**. The Windows printer to send such text to can be selected here. By default, printing is disabled.

Prompt about running processes on close (ConfirmExit=yes)

If enabled, ask for confirmation when the close button or *Alt+F4* is pressed and the command invoked by mintty still has child processes. This is intended to help avoid closing programs accidentally.

Command line

The settings here are config file versions of command line options described in the OPTIONS section. They do not appear in the options dialog.

Holding the window open (Hold=start)

The **Hold** setting determines whether to keep the terminal window open when the command has finished and no more processes are connected to the terminal. It takes the following values:

− **never**: Don't keep the window open.

− **start**: Only keep the window open if the command exited with status 255, which is used to indicate failure to start the command. This is the default.

− **error**: Keep the window open if the command exited with a non-zero status or it was terminated by a signal indicating a runtime error.

− **always**: Always keep the window open.

Window icon (Icon=)

The **Icon** setting with format *FILE*[**,***INDEX*] allows to load the window icon from an executable, DLL, or icon file. The optional comma-separated index can be used to select a particular icon in a file with multiple icons. If the setting is empty, as it is by default, mintty's program icon is used.

Log file (Log=)

The **Log** setting can be used to specify a log file that all output is copied into. If it is empty, as it is by default, no logging is done. See also the *script*(1) utility for a more flexible logging solution.

Window title (Title=)

The **Title** setting can be used to determine the initial window title. If it is empty, as it is by default, the title is set to the command being run.

Utmp record (Utmp=no)

If enabled, an entry for the session is written into the system's *utmp* file for recording logins, so that the session appears for example in the output of the *who*(1) utility.

Initial window state (Window=normal)

This setting determines how the terminal window should be shown at startup:

− **normal** (default)
− **min** (minimised)
− **max** (maximised)
− **full** (full screen)
− **hide** (invisible)

Window position (X=, Y=)

X and **Y** are integer settings that can be used to determine the initial coordinates of the top left corner of the terminal window. By default, these are unset, which means that the position suggested by the window manager is used.

Window class name (Class=mintty)

The **Class** setting determines the name of the window class of the terminal window. This can be used to help Windows scripting tools such as AutoHotKey to distinguish different mintty windows. The name defaults to "mintty".

Hidden settings

The following settings appear neither in the options dialog nor as command line options, which means they can only be set in config files or using the **--option** command line option.

Word selection characters (WordChars=)

By default, this string setting is empty, in which case double-click word selection uses the default algorithm that is geared towards picking out filenames and URLs.

If a string is specified here, word selection only picks out characters in the string along with alphanumeric characters. For example, specifying just the underscore character (WordChars=_) would allow selecting identifiers in many programming languages.

Use system colours (UseSystemColours=no)

If this is set, the Windows-wide colour settings are used instead of the foreground, background, and cursor colours chosen on the Looks page of the options dialog.

IME cursor colour (IMECursorColour=)

The cursor colour can be set to change when the Input Method Editor (IME) for entering characters not available directly on the keyboard is active. The setting is a RGB triplet such as 255,0,0 for bright red.

By default, this is unset, which means that the cursor colour does not change. The colour can also be changed using xterm's OSC 4 control sequence with colour number 262.

ANSI colours

These are the 16 ANSI colour settings along with their default values:

- **Black**=0,0,0
- **Red**=191,0,0
- **Green**=0,191,0
- **Yellow**=191,191,0
- **Blue**=0,0,191
- **Magenta**=191,0,191
- **Cyan**=0,191,191
- **White**=191,191,191
- **BoldBlack**=64,64,64
- **BoldRed**=255,64,64
- **BoldGreen**=64,255,64
- **BoldYellow**=255,255,64
- **BoldBlue**=64,64,255
- **BoldMagenta**=255,64,255
- **BoldCyan**=64,255,255
- **BoldWhite**=255,255,255

LIMITATIONS

Console issue

Mintty is not a full replacement for the Windows console window that Cygwin uses by default. Like xterm and rxvt, mintty communicates with the child process through a pseudo terminal device, which Cygwin emulates using Windows pipes. This means that native Windows command line programs started in mintty see a pipe rather than a console device. As a consequence, such programs often disable interactive input. Also, direct calls to low-level Win32 console functions will fail. Programs that access the console as a file should be fine though.

Termcap/terminfo

Mintty does not have its own *termcap* or *terminfo* entries; instead, it simply pretends to be an xterm.

Missing xterm features

Mintty is nowhere near as configurable as xterm. Of xterm's keyboard modes, only the default PC-style and VT220-style are available. 8-bit control characters are not supported, nor are ISO2022 character sets. There is no Tektronix 4014 emulation. Mouse highlighting mode is not implemeted.

The majority of xterm's many control sequences is implemented, but quite a few are missing. Significant omissions should be reported as bugs.

SEE ALSO

Additional information can be found on the wiki on the mintty project page, for example:

− Keycodes: *http://code.google.com/p/mintty/wiki/Keycodes*
− Control sequences: *http://code.google.com/p/mintty/wiki/CtrlSeqs*
− Tips: *http://code.google.com/p/mintty/wiki/Tips*
− Contributors: *http://code.google.com/p/mintty/wiki/Credits*
− Changelog: *http://code.google.com/p/mintty/wiki/Changes*

LICENSE

CONTACT

Please report bugs or suggest enhancements via the issue tracker at *http://code.google.com/p/mintty/issues*. Questions can be sent to the discussion group at *http://groups.google.com/group/mintty-discuss* or the Cygwin mailing list at *cygwin@cygwin.com*.