

## 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. Note that setting ShortLongOpts enables single-dash long options.

-c, --config FILENAME

Read settings from the specified configuration file, in addition to the default config files. Configuration changes are saved to the last file thus specified.

-C, --loadconfig FILENAME

Read settings from the specified configuration file, in addition to the default config files. The file is not taken into account for saving configuration changes. This is useful to mix-in partial configuration variants, particularly colour schemes. However, -o ThemeFile=FILENAME may be preferable.

--configdir DIRNAME

Use the given directory to check for resource subdirectories (themes, sounds, lang); also read settings from the configuration file DIRNAME/config, in addition to the default config files, and save configuration changes here.

--dir directory

Change initial directory to start in. This is especially useful for invocation of mintty from a Windows context menu via registry entry.

-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.

Note: About interaction problems of icon, shortcut, and the Windows taskbar: In a Windows desktop shortcut, to achieve consistent icon behaviour, the same icon should be specified in the shortcut properties (Change Icon...) and the mintty command line (Target:), or (beginning 2.2.3) no icon should be specified on the command line as mintty will then take the icon from the invoking shortcut, also resolving a leading Windows environment variable (like %SystemRoot%).

`-l, --log FILE|-`

Copy all output into the specified log file, or standard output if a dash is given instead of a file name. (Implies `-o Logging=yes.`)

If FILE contains %d it will be substituted with the process ID. See description of equivalent option "Log file" (Log=) below for further formatting options and hints.

Note that logging can be toggled from the extended context menu.

`--logfile FILE|-`

Like `--log` but with logging initially disabled, so just specifying a potential log file name in case logging is enabled from the extended context menu. (Equivalent to combining `--log` with `-o Logging=no.`)

`-o, --option NAME=VALUE`

Override the named config file option with the given value, e.g. `-o ScrollbackLines=1000.`

`-p, --position X,Y`

Open the window with its top left corner at the specified coordinates. Instead of coordinates, "centre" or "center" can be specified to place the window in the screen centre, and "right" or "bottom" can be specified to align the right or bottom window border with the right or bottom screen border (together with another option `-p` to specify an offset).

Option value "@N" where N is a number places the window on monitor N.

Multiple `-p` options can be combined; coordinates have a different meaning depending on other options:

- With "left", "top", or "@N", related coordinates are relative to the monitor.
- With "right" or "bottom", related coordinates adjust the right or bottom window border relative to the monitor.
- Otherwise, coordinates are absolute and address the common multi-monitor address space as provided by Windows.

Note: For another option to select the monitor for a new mintty window, see the description of Alt+F2.

- s, --size COLS,ROWS  
Set the default size of the window in character columns and rows. (The xterm-like syntax COLSxROWS is accepted too.) Instead of coordinates, "maxwidth" or "maxheight" can be specified; this can be combined with another parameter -s for the other dimension. The dimension for which "max" is applied is ignored in further -s or -p parameters. For example, mintty -s maxwidth -p 0,0 -s 0,10 will start a window at full screen width, positioned at the top of the screen, with 10 lines.
  
- nobidi, --nortl  
Disable bidi display (right-to-left support). Same as -o Bidi=0.
  
- t, --title TITLE  
Use TITLE as the initial window title. By default, the title is set to the executed command.
  
- T, --Title TITLE  
Use TITLE as the permanent window title. The title is not changeable by control sequences. This feature is only available on the command line.
  
- B, --Border frame|void  
Suppress window title, display only a frame or no border. This feature is only available on the command line.
  
- 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".
  
- d, --nodaemon  
Do not apply "daemonizing". By default, mintty tries to detach itself from the invoking terminal when started from a Cygwin Console in order to avoid disabled signal reception, and when cloning the window (Alt+F2) in order to avoid a remaining zombie process.
  
- D, --daemon  
Enforce "daemonizing". By default, mintty tries to detach itself from the invoking terminal only as described above. With this option, it tries to detach always. This makes a difference if a Windows "Shortcut key" is configured in a Windows desktop shortcut for starting mintty. Without daemonizing, the shortcut key will focus an already running instance of mintty, with daemonizing it always starts a new instance.
  
- R, --Report info/mode  
Report requested information.  
  
With values "s" or "o", mintty reports the position and size of the window when it exits. This can be used to manage last window positions and reopen mintty windows accordingly. Reporting mode is "s" or "o" to choose short or long option syntax for the

restored (i.e. neither maximised nor minimised) geometry;  
min/max/fullscreen information is added.

With value "m", mintty reports the system's monitor configuration (listing all connected monitors and their geometry and position in Windows' virtual monitor coordinate system), and exits.

With value "f", mintty reports the monospace fonts installed on the system as determined by mintty, and exits.

- store-taskbar-properties  
Enable persistent storage of Windows taskbar properties together with options AppName and AppLaunchCmd.
- nopin  
Prevent pinning of the mintty window to the Windows taskbar.
- wsl Adjust to WSL (the Windows Subsystem for Linux, or Bash/Ubuntu on Windows):
  - When dragging a Windows file or folder into mintty, it will be pasted using the Linux path name.
  - When Ctrl+clicking a file name, it will be interpreted in the Linux namespace and converted before opening in Windows.
  - Options DropCommands and setting MINTTY\_PROG for UserCommands are disabled.
  - The working directory of the current foreground process (for click-opening pathnames) cannot be detected.
  - Locale modification (@cjk...) is disabled.
- WSL WSL DISTRIBUTION NAME  
Run a WSL session, setting other parameters as appropriate and involving the wslbridge gateway implicitly (which should be installed in /bin for this purpose). If the distribution name is empty, the default WSL installation is run; otherwise, it refers to the installed WSL packages as listed by the Windows tool wslconfig /l. Implies --wsl -o Locale=C -o Charset=UTF-8, --rootfs=..., and --icon=... if a respective icon file exists for the distribution.
- rootfs ROOTFOLDER  
Provide the root filesystem folder to adjust path conversion properly for the respective WSL installation.
- Start in the user's home directory. Affects also WSL sessions.
- H, --help  
Display a brief help message and exit.
- V, --version  
Print version information and exit.

A number of xterm-style convenience options are also available:

- fg Sets ForegroundColour.
- bg Sets BackgroundColour.
- cr Sets CursorColour.
- selfg  
Sets HighlightForegroundColour.
- selbg

Sets HighlightBackgroundColour.

--fn, --font  
Sets Font.

--fs Sets FontSize.

--geometry COLSxROWS[[-+]X[-+]Y][@MONITOR]  
Sets size and position, extending xterm syntax by an optional monitor number.

--en Sets Charset within the current locale.

--lf Sets Log, the log file name. Use -l to both set the log file name and enable loggin.

--sl Sets ScrollbackLines.

## 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.

### Font rendering

Mintty uses Windows Uniscribe font rendering to display a wider range of characters; the TextOut API is automatically used instead if suitable.

### Menus

The context menu can be opened by right-clicking the mouse (with Shift in case right-click has been redefined or redirected to the application) or by pressing the Menu key that is normally located next to the right Ctrl key. If invoked while the Ctrl key is held down, an extended context menu will be opened, with some additional entries.

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 or Ctrl+Shift+C keyboard shortcuts (Ctrl+C with option CtrlExchangeShift=yes), 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 ("true colour" attributes are not supported).

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 or Ctrl+Shift+V keyboard shortcuts (Ctrl+V with option CtrlExchangeShift=yes), or the middle mouse button. Not only text but also files and directories can be pasted, whereby the latter are inserted as Cygwin file names. Shell quoting is added to file names 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, URLs and web addresses beginning with "www." can be opened either by holding Ctrl while left-clicking on them (or double-clicking, if and as enabled by option OpeningClicks), or by selecting them and choosing the Open command from the context menu. Embedded spaces are considered if escaped with a backslash; for selected pathnames, also embedding quote marks are considered.

A relative pathname is interpreted as relative to the current working directory of the terminal foreground process if that can be determined, overridden by the working directory interactively communicated by the respective control sequence.

#### Font zoom

The font size can be increased or decreased using the keyboard shortcuts Ctrl+(keypad-)plus and Ctrl+(keypad-)minus, or by holding Ctrl while rolling the mousewheel. Ctrl+zero or Ctrl+middle-mouse click returns the font size to the default.

Shift-coupled window-with-font zooming: If Shift is also held while zooming, the window will be resized to scale together with the font, keeping the terminal character size if possible. This is not applied to the shifted numeric keypad "0" (which has other meaning) and to the shifted normal (non-keypad) "-" and "+" keys (because the shifted key could have a valid mapping, e.g. Ctrl+\_, or the "+" key could be shifted itself already).

Zooming by keyboard or mouse can be disabled, respectively, with options ZoomShortcuts=no or ZoomMouse=no.

#### Drag resize

The usual windows function to drag on the window border resizes the terminal.

Shift-coupled font-with-window zooming: If Shift is also held while resizing, the font will be scaled along with the resizing, unless disabled with ZoomFontWithWindow=false (which would help to avoid interference with certain shifted hotkeys configured to resize the window). Note that due to the different height/width factors, coupled font zooming is not a precise operation.

#### DPI change

When DPI setting changes (by reconfiguration of display properties "what's on your screen ... smaller/medium/larger" or moving the mintty window between monitors with different DPI settings), mintty adapts its screen size to avoid windows blurred auto-adaptation. If Shift is also held during the change, the font will be scaled too, roughly maintaining the screen dimensions.

#### 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, or the generic window title functions.

## 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. Shift+Alt+F10 also restores the font size to its default.

## 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.

## Searching in the text and scrollback buffer

Alt-F3 opens a search bar with an input field for a search string. Matches are highlighted in the scrollback buffer. Enter/Shift+Enter find the next/previous position of the match and scrolls the scrollback buffer accordingly. The appearance of the search bar and the matching highlight colours can be customized. Matching is case-insensitive and ignores combining characters.

Shift+cursor-left/right offers another scrolling feature. If prompt lines are marked with scroll markers they navigate to the previous/next prompt, to provide a better orientation among the output of previously invoked commands. See the Control Sequences wiki page <https://github.com/mintty/mintty/wiki/CtrlSeqs#scroll-markers> for details.

## 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.

## Virtual Tabs

The Virtual Tabs feature provides a list of all running mintty sessions (session switcher) as well as configurable launch parameters for new sessions (session launcher). By default, the list is shown in the extended context menu (Ctrl+right-click), the mouse button 5 menu, and the menus opened with the Ctrl+Menu key and the Ctrl+Shift+I shortcut (if enabled). (Menu contents for the various context menu invocations is configurable.) For configuration, see settings SessionCommands, Menu\*, and SessionGeomSync. Distinct sets of sessions can be set up with the setting -o Class=....

## 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.

#### Terminal Break

A traditional BRK event on a serial terminal connection can be simulated. The Break is available in the extended context menu and it can be mapped to the Break key by configuration. Note, however, that a BRK can be ignored by configuration of the terminal device (pty) or can be ignored by an application by catching the SIGINT signal. For more forceful interruption of the terminal client application, see the Tips wiki page <https://github.com/mintty/mintty/wiki/Tips#terminating-the-foreground-program>.

Note that Ctrl+C is often configured to raise a SIGINT signal. However, this is not a terminal feature and can also be reconfigured (stty), so in fact BRK and Ctrl+C are inherently different functions.

#### 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.

Mintty supports 5-button mice, handling mouse buttons 4 / 5 like Alt+click-left / right in most mouse modes.

#### 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 most significant non-zero byte first.

#### Shortcuts

An overview of all the keyboard shortcuts.

##### Scrollbar

- Shift+Up: Line up
- Shift+Down: Line down
- Shift+PgUp: Page up
- Shift+PgDn: Page down
- Shift+Home: Top
- Shift+End: Bottom
- Alt+F3: Search
- Shift+cursor-left: Go to previous scroll marker (e.g. in prompt)
- Shift+cursor-right: Go to next scroll marker (e.g. in prompt)



## Copy and paste

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

## Window commands

- Alt+F2: New (clone window at current size); see notes below
- Shift+Alt+F2: New (clone window at configured size); see notes below
- Alt+F3: Search (in scrollback buffer)
- Alt+F4: Close
- Alt+F8: Reset
- Alt+F10: Default terminal size (rows/columns)
- Shift+Alt+F10: Default terminal size (rows/columns) and font size
- Alt+F11 or Alt+Enter: Toggle full screen
- Shift+Alt+F11 or Shift+Alt+Enter: Toggle full screen and zoom font (Note that due to the different height/width factors, this is not a precise operation)
- Alt+F12: Flip screen
- Alt+Space: Window menu
- Ctrl+Tab: Next window
- Ctrl+Shift+Tab: Previous window

Multi-monitor selection support: Alt+F2 will only spawn a new window after F2 has been released. While F2 is being held, the target monitor can be selected with a sequence of numeric keypad keys:

- cursor-up/down/left/right (8/2/4/6) navigate the target focus to the respective neighbour in the monitor grid;
- the diagonal keys (7/9/1/3) combine two directions respectively;
- the central key (5) sets the target focus to the Windows "primary" monitor;
- the Ins key (0) or Del resets the focus to the current monitor.

These navigation controls can be applied repeatedly to select a monitor further away.

Note that a heuristic algorithm is used, based on the size of the smallest monitor attached to the system, so the target may not always be selected as expected if multiple monitors of different size are available or monitors are not arranged in a regular grid.

## Font zoom

- Ctrl+(keypad-)plus: Zoom font in
- Ctrl+(keypad-)minus: Zoom font out
- Ctrl+Shift+(keypad-)plus: Zoom font and window in
- Ctrl+Shift+(keypad-)minus: Zoom font and window 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+A: Select all
- Ctrl+Shift+C: Copy
- Ctrl+Shift+V: Paste
- Ctrl+Shift+N: New
- Ctrl+Shift+H: Search scrollback buffer
- Ctrl+Shift+W: Close

- Ctrl+Shift+R: Reset
- Ctrl+Shift+D: Default terminal size (rows/columns)
- Ctrl+Shift+F: Full screen (not zooming font despite Shift)
- Ctrl+Shift+S: Flip screen
- Ctrl+Shift+O: Toggle scrollbar
- Ctrl+Shift+P: Cycle pointer styles
- Ctrl+Shift+T: Cycle or tune transparency levels

Ctrl+Shift+T cycles through transparency levels in steps, whenever Ctrl+Shift+T is released. Alternatively, while Ctrl+Shift+T is held down, the navigation keys on the numeric keypad can be used for further fine-tuning:

Up/Dn to increase/decrease, PgUp/PgDn for steps, Del/Ins for no/max transparency, End for highest preconfigured transparency, Home for previous value, Clear ("5") for glass.

If OpaqueWhenFocused is set, opaqueness is temporarily disabled to provide visible feedback for the changes.

#### Embedding graphics in terminal output

The new support of the SIXEL feature facilitates a range of applications that integrate graphic images in the terminal, animated graphics, and even video and interactive gaming applications.

An example of the benefit of this feature is the output of `gnuplot` with the command

```
GNUTERM=sixel gnuplot -e "splot [x=-3:3] [y=-3:3] sin(x) * cos(y)"
```

#### Diagnostic support

##### Screen logging

A couple of options are available to enable logging initially (Log=... or -l ... on the command line), or to specify a log file name for later logging (Log=... combined with Logging=no, or --logfile ... on the command line). In either case, logging can be toggled from the extended context menu.

##### Character information display

Diagnostic display of current character information can be toggled from the extended context menu.

Unicode character codes at the current cursor position will then be displayed in the window title bar. (Note that mintty may pre-compose a combining character sequence into a combined character which is then displayed.)

Unicode character names will be included in the display if the unicode-ucd package is installed in /usr/share (or the file charnames.txt generated by the mintty script src/mknames is installed in the mintty resource subfolder info).

Note that the "normal" window title setting sequence and the character information output simply overwrite each other.

## CONFIGURATION

Mintty has a graphical options dialog that can be reached via the context menu or the window menu. It has the following action buttons:

- Cancel: discards changes.
- Save: applies and saves changes and closes the dialog.
- Apply: applies changes to the current instance of mintty but does not save them to the configuration file. So using Apply then Cancel, local changes can be applied (and tested) without affecting further instances of mintty.

In configuration files, settings are stored as NAME=VALUE pairs, with one per line. By default, they are read from any file of /etc/minttyrc, \$APPDATA/mintty/config, ~/.config/mintty/config, ~/.minttyrc, in this order. Additional configuration files can be specified using the -c/--config or -C/--loadconfig command line options. These are read in

order after the default config files, with settings in later files overriding those in earlier ones. Configuration changes are saved to the last writable file read by default or `~/.minttyrc` if none is given, or (with precedence) to a configuration file specified with `-c/--config` or `--configdir`. Individual settings can also be specified on the command line using the `-o/--option`.

Note: Many string values in the config files, especially those referring to file names or Windows items, are Unicode-enabled, meaning they are expected to be UTF-8-encoded in the configuration file independently of the encoding the terminal runs in; as a fallback, if the configuration value is not valid UTF-8, it is interpreted in the system ANSI encoding. (This does not apply to the same configuration settings when given on the command-line.)

Unicode-enabled settings: `BellFile`, `ThemeFile`, `Title`, `ExitTitle`, `Icon`, `Log`, `Language`, `Font`, `Font1...`, `FontSample`, `Printer`, `Answerback`, `SixelClipChars`, `Class`, `AppID`, `AppName`, `AppLaunchCmd`, `DropCommands`, `UserCommands`, `SessionCommands`.

Be careful when running multiple instances of mintty. If options are saved from different instances, or the config file is edited manually, options can obviously be overwritten; if older mintty versions are run (e.g. from cygwin and msys sharing the same home directory), options may even get dropped from the configuration file; mintty versions since 261 preserve unknown options and comment lines.

Additional resource files are used for colour schemes (option `ThemeFile`, subdirectory `themes`), wave files (option `BellFile`, subdirectory `sounds`), and localization translation files (option `Language`, subdirectory `lang`) within the mintty resource directories `/usr/share/mintty`, `$APPDATA/mintty`, `~/.config/mintty`, `~/.mintty`, or as specified with command line option `--configdir`.

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.

If there is only a name in parentheses, there is currently no GUI configuration facility for that option (see also Hidden settings below).

## Looks

Settings affecting mintty's appearance.

### Colours

Clicking on one of the buttons here opens the colour selection dialog.

In the settings (config file or command-line options), colours are represented as comma-separated RGB triples with decimal 8-bit values ranging from 0 to 255. X-style hexadecimal colour specifications such as `#RRGGBB`, `rgb:RR/GG/BB` or `rgb:RRRR/GGGG/BBBB` can be used as well. Also X11 color names are supported.

- Foreground (`ForegroundColour=191,191,191`)

- Background (`BackgroundColour=0,0,0`)

- Cursor (`CursorColour=191,191,191`)

- Underline, Strikeout, Overline (`UnderlineColour=-1`)

- Theme (`ThemeFile=`): The popup menu offers theme files as stored in a resource subdirectory `themes` for selection as a colour scheme. The option can also be set to a filename (like `D:/.../solarized-light.minttyrc`).

The field can also be used as a drag-and-drop target for colour

schemes downloaded from the Color Scheme Configurator, or for theme files from the web. See the Tips wiki page <https://github.com/mintty/mintty/wiki/Tips#using-colour-schemes-themes> about this mechanism.

Note: Mintty also provides the command-line script `mintheme` which can display the themes available in the mintty configuration directories or activate one of them in the current mintty window.

#### 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 / Ease of Access Center, mouse panel or Optimize visual display.

#### 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); only monospace fonts are listed
- Font style (FontWeight=400, FontIsBold=no)
- Size (FontHeight=9)

The font selection dialog also offers an Apply button for convenient testing how the selected font looks. Its function is the same as the Apply button of the options dialog.

Further settings can be given in the config file:

- Font boldness (FontWeight=400): This is an implicit value after selecting a font in the font selection menu, or can be specified in the config file or on the command line for font selection. Typical weights are Normal/Regular (FontWeight=400) and Bold (FontWeight=700 or FontIsBold=yes) but if a font family has a different scheme or more than 2 font weights, the weight value can be used for more specific selection. If a font family has no bold weight but boldness was requested, mintty does not adhere to this scheme but enforces bold font selection; however, in this case the bold attribute may not be effective.
- Alternative fonts (Font1= ... Font10= , Font1Weight= ... Font10Weight=): With these settings, up to 10 alternative fonts (and optionally weights) can be configured which would then be selectable via ECMA-48 SGR character attributes (see Tips wiki page <https://github.com/mintty/mintty/wiki/Tips#text-attributes-and-rendering>). Note: Font10 has a special preference property; if it is not configured, mintty will try to activate it anyway, looking for an installed Fraktur or Blackletter font (ECMA-48 "Gothic"). Note: The control sequence for alternative font 1 overrides the identical control sequence to select the VGA character set, which would thus be disabled. Configuring alternative font 1 is therefore discouraged.
- Font sample text (FontSample=): This setting overrides the text for the "Sample" box in the Font chooser dialog.
- Show "hidden" fonts (ShowHiddenFonts=no): This hidden setting enables display of monospace fonts in the font selection menu even if they are marked to Hide in the Windows Font settings (from the Control Panel – Fonts folder).
- Configure font chooser (FontMenu=-1): This hidden setting selects and tunes the font chooser dialog element. Value 1 selects the Windows system font chooser unmodified; value 2 enables font chooser localization, adding value 4 (to 6 or 14) enables horizontal item scaling (making space for localized labels), adding value 8 (to 10 or 14) enables item and size adjustments, value -1 enables all tuning; value 0 selects a built-in inline font chooser.

#### Text lines (UnderlineManual=false)

By setting this true, text attributes underline, doubly underline, strikeout and overline are enforced to be drawn manually. The default is to use Windows font variants for strikeout and for underline, unless mintty detects that the underlined font would not display properly. Note that font smoothing may be affected by Windows-generated underline modes.

#### Show bold as font (BoldAsFont=no)

This option sets the preferred rendering of the ANSI bold (or 'intense') text attribute to use a bold-style font; where a suitable 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 ('overstrike').  
(Corresponds roughly to the xterm resource allowBoldFonts.)

This option is not fully independent. If both BoldAsFont and BoldAsColour are true, both display methods are combined where applicable. If both are false, xterm default behaviour is applied. See Bold Behaviour for an overview.

#### Show bold as colour (BoldAsColour=yes)

This option sets the preferred rendering of the ANSI bold (or 'intense') text attribute to use a different colour, usually with increased brightness; it maps ANSI colours 0..7 (unless selected with the palette colour escape sequences) to their

bright variants 8..15, and the default colour to a brightened variant. Rendering of other colours is not affected. (Corresponds largely to the xterm resource boldColors.)

This option is not fully independent. If both BoldAsColour and BoldAsFont are true, both display methods are combined where applicable. If both are false, xterm default behaviour is applied. See Bold Behaviour for an overview.

**Bold substitution colour (BoldColour=)**

This hidden option sets a colour to be used to render the bold attribute of text that would otherwise have the default foreground colour, overriding other bold rendering; it is only applied if option BoldAsColour is true. The bold substitution colour can also be set, modified, enabled or disabled with the respective xterm OSC control sequences. (Corresponds to the xterm resources colorBD and colorBDMode.)

**Bold as special background (BoldAsRainbowSparkles=false)**

This hidden option displays the bold attribute by underlaying special background. Overrides BoldAsFont. This is a fun option, use at your own risk.

**Note: Bold Behaviour**

When the bold text attribute is set, mintty distinguishes three classes of colours:

- Default: The default terminal foreground colour.
- ANSI-8: The ANSI colours 0..7 (used for SGR 30..37).
- Extended: True colours and the rest of the 256 colours palette.

The colour classes are affected by the bold text attribute as follows:

- Extended colours are always shown with a boldened font only.
- When both BoldAsFont and BoldAsColour are disabled, mintty engages a mode similar to the xterm default behaviour: ANSI-8 is displayed with a bold font and a different colour, Default colour only uses a bold font.
- Otherwise, Default and ANSI-8 colours are affected by BoldAsFont and BoldAsColour independently, such that it is possible to choose only bold font, or only different colour, or both.
- Note that Default bold display can be overridden by the BoldColour mode.

**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 in font rendering from the following choices:

- Default: Use Windows setting.
- None: With all the jaggies.
- Partial: Greyscale anti-aliasing.
- Full: Subpixel anti-aliasing ("ClearType").

Note that font smoothing may be affected by some Windows-generated font attributes; see UnderlineManual.

**Font rendering (FontRender=uniscribe)**

Select the rendering system used for text display:

- textout: Use the Windows ExtTextOut API.
- uniscribe: Use the Windows Uniscribe API.

## 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.

Note: This means, while not strictly necessary, that also locale variables unrelated to the terminal character set (e.g. LC\_MESSAGES) are cleared to avoid confusion.

Note: If the locale option is set, mintty further checks whether the locale is a "wide" locale (i.e. ambiguous-width characters are wide in the locale) but the selected font is actually "ambiguous-narrow" (i.e. ambiguous-width characters are narrow in the font) in which case it appends the "@cjknnarrow" locale modifier.

By default, the selected locale also determines the character width assumptions used for screen rendering. For exceptions, see the setting Charwidth and the control sequences for special wide character handling (<https://github.com/mintty/mintty/wiki/CtrlSeqs#wide-characters>).

## Character set (Charset=)

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

Note: While changing the character set takes effect immediately for text input and output, 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 should be restarted for a character set change to take full effect, or the locale environment of the shell should be changed accordingly.

Note: The locale and character set can also be changed with an escape sequence, see the Control Sequences wiki page <https://github.com/mintty/mintty/wiki/CtrlSeqs#locale>. That setting takes precedence; changes from the Options menu will silently have no effect unless an escape sequence with empty locale is sent to the terminal to restore to "default".

## Keys

Settings controlling keyboard behaviour.

Backarrow sends ^H (BackspaceSendsBS=no)

By default, mintty sends ^? (ASCII DEL) as the keycode for the backspace key. If this option is enabled, ^H is sent instead. This also changes the Ctrl+Backspace code from ^\_ to ^?. (Corresponds to the xterm resource backarrowKey.)

Delete sends DEL(DeleteSendsDEL=no)

By default, mintty sends VT100 Remove as the keycode for the

keypad Del key. If this option is enabled, ^? (ASCII DEL) is sent instead.

(Corresponds to the xterm resource deleteIsDEL.)

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).

Old method of AltGr detection (OldAltGrDetection=no)

Setting this hidden option would disable a workaround for an incompatibility in the Windows on-screen keyboard, just in case it has any side effects.

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 function keys as shortcuts, for example Alt+F4 for closing the window or Alt+F11 for toggling 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.



#### Compose key selection (ComposeKey=off)

The modifier key selected here will have the function of a Compose key. Pressing and releasing the key, following by a sequence of composing keys, will enter a composition of them, according to X11 compose key data.

The Off setting disables the Compose key.

- Shift
- Ctrl
- Alt
- Off

#### 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 mouse button (RightClickAction=menu)

Action to take when the right mouse button is pressed.

- Paste: Paste the clipboard contents.
- Extend: Extend the selected region.
- Enter: Simulate Enter/Return key.
- Menu: Show the context menu.

If this is set to Paste, the middle button extends the selected region instead of pasting the clipboard. If it is set to Extend, a left click with Shift pressed pastes the clipboard instead of extending the selection.

#### Middle mouse button(MiddleClickAction=paste)

Action to take when the middle mouse button is pressed.

- Paste: Paste the clipboard contents.
- Extend: Extend the selected region.
- Enter: Simulate Enter/Return key.
- Void: Do nothing.

#### 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

(HideMouse=on)

By default, mintty automatically hides the cross-hair mouse cursor when keyboard input is being entered. Setting this option =false keeps the cursor.

## 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.

(RowSpacing=0)

Additional row padding.

Note: Mintty adjusts row spacing according to the font metrics, to compensate for tight or tall spacing of some fonts (e.g. Courier, Consolas, FreeMono, Monaco). The RowSpacing value is added to that.

(ColSpacing=0)

Additional column padding; ColSpacing=1 can avoid boldened glyphs being clipped.

(Padding=1)

Window padding; margin between text and window border. The effective value is limited by the character cell width (scaling with font zooming).

(Corresponds to the xterm resource internalBorder.)

A negative value indicates that always the character cell width shall be used, without fixed limit.

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, Home/End, or arrow-left/right 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 scrollbar 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.

UI localization language (Language=)

This selects the language or language/region code to use for localization of the mintty user interface, the options dialog, menus, message boxes, and terminal in-line error messages.

- (empty) an empty entry disables localization
- @ use the Windows user language setting
- \* use environment settings (variables LANGUAGE, LC\_ALL, LC\_MESSAGES, LANG)
- = use the same language as the Locale setting (section Text)
- (language[\_region]) use the given language or language/region code

See the Tips wiki page

<https://github.com/mintty/mintty/wiki/Tips#localization> about how to configure localization.

Note that Windows may already have localized the default entries of the system menu, which makes the system menu language inconsistent because mintty adds a few items here. Select Language=en to "reverse-localize" this.

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, vt100, vt220, vt340. The option vt340 facilitates a terminal ID indication corresponding to the Sixel graphics feature (but there is no mutual dependence between this setting and that feature).

If the setting contains "vt220" or higher, 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 ^E (ENQ) character. By default, this is empty.

Alternate screen (NoAltScreen=false)

With this setting, the alternate screen can be disabled.

Apply old wraparound behaviour (OldWrapModes=false)

Setting this compatibility option disables some tweaks and fixes

of mintty 2.7.5:

- Backspace after pending Wraparound goes to previous column
- Reverse Wraparound mode initially disabled (but switchable), complying with xterm default and terminfo

Allow control sequence to set selection (AllowSetSelection=false)

If enabled, the terminal control sequence OSC 52 is allowed to set the clipboard selection for pasting (using base64-encoded contents, like xterm).

Bell The options here determine what effects the bell character ^G has. Default beep and taskbar highlighting are enabled by default. Mintty can also play wave sounds or frequency beeps.

- Bell system sound (BellType=1): Preferred system sound, values:

- 1 : Simple Beep
- 0 : No Beep (overrides BellFile and BellFreq)
- 1 : Default Beep
- 2 : Critical Stop
- 3 : Question
- 4 : Exclamation
- 5 : Asterisk

- Wave (BellFile=): The popup menu offers wave files as stored in a resource subdirectory sounds for selection. The option can also be set to a filename (like D:/.../soundfile.wav); this can be achieved also by drag-and-drop from a local file. This setting overrides the Bell system sound except No Beep.

- Play: The button plays the selected sound for testing. The sound is also played when it is changed.

- Flash (BellFlash=no): Briefly flash the terminal or window. (Corresponds to the xterm resource visualBell.)

- Flash style (BellFlashStyle=4): Tune the style to flash the terminal or window; this is a bitmask composed of the following values:

- 1 : Flash the window frame (using Windows)
- 2 : Flash the outer character cells; not recommended; will look ragged with double-width characters
- 4 : Flash the whole terminal pane (all character cells)
- 8 : cell flash briefly inverts foreground and background; default is to moderately brighten the background colour
- 12 : (combining 4 and 8) classic bright full flash

- Highlight in taskbar (BellTaskbar=yes): Change the colour of mintty's taskbar entry if the mintty window is not active. (Corresponds to the xterm resource bellIsUrgent, switchable by an escape sequence.)

- Popup on bell (BellPopup=no): Popup mintty to desktop foreground. (Corresponds to the xterm resource popOnBell, switchable by an escape sequence.)

A simple frequency beep can be configured in the configuration file or on the command line:

- (BellFreq=0): Beep sound frequency (overrides system sounds).
- (BellLen=400): Beep sound length (applies to frequency beep).

Printer (Printer=)

The ANSI standard defines control sequences ("Media Copy") 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. If printing gets disabled in the Options menu, an active print connection will be ended; if the printer is changed, an active print connection will be continued with the previous printer.

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. If possible, mintty also displays a list of running child processes, using the procs command if installed, or the ps command.

#### 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 run-time 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, unless mintty was invoked from a desktop shortcut in which case it uses the shortcut icon.

For interaction problems of icon, shortcut, and the Windows taskbar, see the note for the -i option above.

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. If it contains %d it will be substituted with the process ID. If it contains % placeholders other than a single %d, the log file name will be constructed by calling strftime(3) on the pattern; note that this is likely to fail if a placeholder expands with "/" (%D).

Note: If the requested log file exists already, mintty does not overwrite it but reports an error. To configure logging in the config file, use some % placeholders to create distinct log files. Note that logging can be toggled from the extended context menu.

See also the script(1) utility for a more flexible logging solution.

Logging initially enabled (Logging=yes)

Disabling this setting disables logging initially. If a log file name is specified with Log=... and Logging=no, logging can be enabled (and toggled) from the extended context menu.

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 or -o command line option.

Single-dash long options (ShortLongOpts=false)

This settings enables names options ("long options") on the command line to be given with only one dash rather than a double dash.

Character width handling (Charwidth=locale)

With this option, locale-determined character width properties can be overridden:

- locale Use locale width properties.
- unicode Use built-in width properties, likely based on a more up-to-date Unicode version. (Corresponds to the xterm resource mkWidth.)
- ambig-wide Use built-in width properties, with ambiguous-width characters assumed to be wide. (Corresponds to the xterm resource cjkWidth.)

Bidirectional rendering (Bidi=2)

With this option, bidi rendering support can be disabled conditionally or completely. Note that this can also be changed by an escape sequence. Also there is another escape sequence to disable bidi per screen line.

- 0 Disable bidi completely.
- 1 Disable bidi on alternate screen, support it on normal screen.
- 2 Enable bidi (default).

Application ID (AppID=)

Windows 7 and above use the application ID for grouping taskbar items. By default this setting is empty, in which case Windows groups taskbar items automatically based on their icon and com-

mand line. This can be overridden by setting the AppID to a custom string, in which case windows with the same AppID are grouped together.

Warning: Using this option in a Windows desktop shortcut may cause trouble with taskbar grouping behaviour. If you need to do that, the shortcut itself should also get attached with the same AppID.

Explanation: Note that Windows shortcut files have their own AppID. Hence, if an AppID is specified in the mintty settings, but not on a taskbar-pinned shortcut for invoking mintty, clicking the pinned shortcut will result in a separate taskbar item for the new mintty window, rather than being grouped with the shortcut. To avoid this, the shortcut's AppID has to be set to the same string, which can be done using the Win7AppId utility available cloned in the mintty utils repository <https://github.com/mintty/utils>.

#### Application Taskbar Shortcut Title (AppName=)

The title of the Windows 7 taskbar shortcut. If the shortcut is pinned, the title is kept only if it was made persistent as described for AppLaunchCmd.

#### Application Taskbar Shortcut Launch Command (AppLaunchCmd=)

The command to use if a shortcut pinned to the Windows 7 taskbar is invoked. This is only effective if combined with an AppName option and the command-line option `--store-taskbar-properties` to make it persistent. It should also be combined with an explicit and unique AppID.

Note: The command must be given in Windows pathname syntax (e.g. `AppLaunchCmd='C:\cygwin\bin\mintty -T mytitle -'`).

Note: An explicit icon supplied with the `-i` option can also be stored with the persistent properties; note, however, that for this purpose, it must be given in Windows pathname syntax and it must include a resource index; also, for consistent appearance, another `-i` option referring to the same icon should be included in the AppLaunchCmd (if the mintty invocation does not include an icon parameter but was started from a desktop shortcut, the icon in the AppLaunchCmd should be consistent with that one, respectively).

Example: `mintty -o AppID=Mintty.PinTest.1 -o AppName=Mintty.PinTest -o AppLaunchCmd="C:\cygwin\bin\mintty -i /cygdrive/c/Windows/System32/calc.exe -" -i C:\Windows\System32\calc.exe,0 --store-taskbar-properties -`

Warning: Once made persistent, the stored properties associated with a specific AppID cannot be removed or even modified again with normal means. For this reason, it is advisable to use temporary AppIDs for testing (like MyMintty.1).

#### 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 file names 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.

#### Word selection exclusion characters (WordCharsExcl=)

This string can list characters that are to be excluded from word selection.

#### Special key remapping

These options can attach a specific string to some special keys. If the string is a number, a corresponding Escape sequence will be generated, applying Shift/Ctrl/Alt modifiers, following the pattern of function keys.

If the string value contains a single control character, some generic Alt or Shift modification is applied. For Pause and Break, control characters are the default values, they can be cleared with an empty assignment. For empty values, the default layout of the keyboard is applied. The Break key can also be assigned the traditional terminal line Break function.

- Key\_Pause=^] (Ctrl+])
- Key\_Break=^\ (Ctrl+\)
- Key\_Menu=
- Key\_ScrollLock=
- Key\_PrintScreen=

Examples:

Key\_Menu=29 would make the Menu key send the same escape sequence as in xterm

Key\_Break=2 would turn the Break key into an Insert key

Key\_Break=\_BRK\_ would assign the simulated terminal line Break function

#### 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. Colours are represented as comma-separated RGB triples with decimal 8-bit values ranging from 0 to 255. X-style hexadecimal colour specifications such as #RRGGBB, rgb:RR/GG/BB or rgb:RRRR/GGGG/BBBB can be used as well. Also X11 color names are supported.

- 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=96,96,255



- BoldMagenta=255,64,255
- BoldCyan=64,255,255
- BoldWhite=255,255,255

#### Downloaded colour scheme (ColourScheme=)

This setting is not intended for manual configuration. It can store a colour scheme as downloaded from the Color Scheme Configurator or from a theme file on the web via drag-and-drop to the Theme of the Options menu. After the colour scheme is stored to a colour scheme file, this setting is not used anymore.

See the Tips wiki page

<https://github.com/mintty/mintty/wiki/Tips#using-colour-schemes-themes> about this mechanism.

#### Selection highlight colours

The highlighting colours of selected text can be configured.

- HighlightBackgroundColour= selected text background (corresponds to the xterm resource highlightColor)
- HighlightForegroundColor= selected text colour (corresponds to the xterm resource highlightTextColor with highlightColor-Mode); only effective if both are configured

#### Scrollbar search colours

The highlighting colours of search matches can be configured.

- SearchForegroundColor=black
- SearchBackgroundColor=light yellow
- SearchCurrentColour=bright yellow

#### Scrollbar search bar (SearchBar=)

This string option can customize the order of items in the search bar. Use x (close button), </> (previous/next buttons), s (search string) to select the order of these fields in the search bar; missing fields will be appended in a default order.

#### Write if exited (ExitWrite=no)

Together with a hold option that keeps the terminal open after its child process terminated, this option always writes an exit indication to the screen. By default, only an error exit code is displayed.

#### Change title if exited (ExitTitle=)

Together with a hold option that keeps the terminal open after its child process terminated, this option prefixes the window title with its string, for example -o ExitTitle="TERMINATED: ".

#### Configure document opening by mouse click (OpeningClicks=1)

Enabling opening files, directories or URLs with mouse clicking, in addition to the context menu. Values 1, 2, or 3 require Ctrl+mouse-click, double-click, or triple-click, respectively, to invoke the document opening. Value 0 disables click-opening.

#### Control key and shortcut Shift exchange (CtrlExchangeShift=no)

Exchange the range of Control characters with the range of Ctrl+Shift shortcuts, so that for example Ctrl+V will paste and Ctrl+Shift+V will enter a Control+V character.

#### Mouse zooming (ZoomMouse=yes)

Enabling font zooming with Ctrl+mouse-wheel/middle-mouse-click.

#### Disable Shift-coupled implicit font zooming (ZoomFontWithWindow=yes)

If this option is set to false, implicit font zooming coupled with window zooming by the Shift key is disabled, except for the

keyboard zoom functions Shift+Alt+Enter/F11 and the Shift+menu function.

Handling of DPI changes (HandledDPI=true)

By default, in Windows 10, when the window is moved to another monitor that has a different DPI value ("scaling factor") configured, this is considered to compensate by appropriate scaling. This option can be set to false to suppress DPI adjustments.

Check availability of mintty version update (CheckVersionUpdate=900)

If non-zero, mintty checks whether there is a version update available whenever the Options dialog is opened and this was not checked within the last given number of seconds.

Sixel image clipboard substitution (SixelClipChars=space)

Characters to copy to clipboard as a substitute for Sixel image graphics, to indicate their positions. With an empty value, U+FFFC will be used. Double-width characters should not be used here.

Drag-and-drop application-targetted commands (DropCommands=)

With this setting, a set of string patterns can be configured for paste insertion, in dependence of the program running in the terminal foreground. The value is a series of semicolon-separated, colon-combined pairs of programname and drop pattern; the pattern is pasted with the actual clipboard contents replaced for a "%s". If a semicolon shall be embedded into any of the drop patterns, a non-whitespace control character (i.e. none of ^I^J^K^L^M) can be specified as an alternative separator by starting the whole setting with it.

Note: This is an experimental feature, with an experimental configuration format.

Note: This feature potentially makes mintty vulnerable against command injection. Be careful what commands you configure! For shell commands, it is advisable to embed the "%s" parameter placeholder in single quotation marks explicitly, both to avoid trouble with special characters, and to reduce the risk of injecting commands via tricky filenames.

Examples:

- DropCommands=bash:cd '%s'^M;mined:^[fo^M%s^M;vim:^[[:e %s^M
- DropCommands=^\_bash:cd '%s';echo \$PWD^M^\_vim:^[[:e %s^M^\_

Note: An "Enter" key has to be specified with the CR character code, control characters need to be embedded verbatim (indicated above as "^M" or "^["); there is no escape notation.

Note: If different actions for directories/folders or even different command invocations depending on file name pattern are desired, this should be handled by a suitable cooperating shell function.

User commands (UserCommands=)

This setting lists user-defined commands for the extended context menu. The value is a series of semicolon-separated, colon-combined pairs of menu item label and command pattern; the command is invoked and its standard output is pasted into the terminal, applying bracketed-paste mode if enabled. If a semicolon shall be embedded into any of the command patterns, a non-whitespace control character (i.e. none of ^I^J^K^L^M) can be specified as an alternative separator by starting the whole setting

with it.

Mintty provides useful information in environment variables:  
MINTTY\_SELECT for the current selection  
MINTTY\_BUFFER for the complete terminal contents including  
scrollback buffer  
MINTTY\_SCREEN for the current screen; if scrolled back, starting  
at the current scroll position  
MINTTY\_OUTPUT for the output of the previous command, if prompt  
lines are marked  
MINTTY\_TITLE for the window title text  
MINTTY\_PID for the program ID of the terminal foreground process  
MINTTY\_PROG for the program name of the terminal foreground  
process  
MINTTY\_CWD for the current working directory of the foreground  
process

Note: Menu item labels are subject to localization if they are  
added to the localization file of the selected UI localization  
language (in subdirectory lang of a mintty resource directory).

Note: The previous command can be detected if prompt lines are  
marked with the escape sequence `^[[?7711h`. See the Control  
Sequences wiki page  
<https://github.com/mintty/mintty/wiki/CtrlSeqs#scroll-markers>.

Note: This is an experimental feature, with an experimental con-  
figuration format.

Note: Normal terminal interaction continues after the invoked  
commands have terminated. Be careful not to configure commands  
that can stall or block!

Note: This feature potentially makes mintty vulnerable against  
command injection. Be careful what commands you configure!  
Especially do not embed environment variable parameters unquoted  
(like `echo $MINTTY_SELECT`), in order to avoid the risk of  
injecting commands via tricky selected text.

Examples:

```
- UserCommands=Paste capital:echo -n "$MINTTY_SELECT" | tr 'a-z'  
'A-Z';Paste small:echo -n "$MINTTY_SELECT" | tr 'A-Z' 'a-z'
```

Note: Control characters need to be embedded verbatim; there is  
no escape notation.

Session launcher commands (SessionCommands=)

This setting lists mintty invocation parameters for the session  
launcher. The value is a series of semicolon-separated, colon-  
combined pairs of session launch names (used as menu item  
labels) and configuration parameter lists; when selecting one of  
the launch names, mintty is invoked with the respective parame-  
ters. (If a semicolon shall be embedded into any of the command  
patterns, a non-whitespace control character (i.e. none of  
`^I^J^K^L^M`) can be specified as an alternative separator by  
starting the whole setting with it.)

Note: Session launch names are subject to localization if they  
are added to the localization file of the selected UI localiza-  
tion language (in subdirectory lang of a mintty resource direc-  
tory).

Note: This is an experimental feature, with an experimental con-  
figuration format.

Examples:

- SessionCommands=big:-w max;Ubuntu:--WSL=Ubuntu;mybox:ssh mybox
- SessionCommands=mycolours:-C ~/.minttyrc.mycolours

Note: Control characters need to be embedded verbatim; there is no escape notation.

#### Menu contents

These settings allow to customize the context menu as opened in various ways.

- Mouse (MenuMouse=b); the normal mouse-click context menu
- Ctrl+Mouse (MenuCtrlMouse=e|ls); the Ctrl+mouse-click menu
- Mouse button 5 (MenuMouse5=ls); the mouse button 5 menu
- Menu key (MenuMenu=bs); the Menu key (unless redefined)
- Ctrl+Menu key (MenuCtrlMenu=e|ls); the Ctrl+Menu-key

Menu contents can be configured by a sequence of characters with the following meaning:

- b: basic context menu
- e: extended context menu
- |: vertical separator, adding a new column
- l: session launcher
- s: session switcher (Virtual Tabs)
- T: the menu is opened at the text cursor position
- P: the menu is opened at the mouse pointer position
- W: show window icon for uniconized windows in session switcher

#### Session geometry synchronization / Virtual tabs (SessionGeomSync=0)

This setting defines the level of approximation of "tabbed" window operation within the Virtual Tabs feature:

- 0: no geometry handling; terminal session windows are separate
- 1: sync. position/size when switching/launching a session
- 2: sync. also when window is moved or resized
- 3: sync. also when window is minimized
- 4: sync. also when window is started

## LIMITATIONS

### Console issue

Mintty is not a full replacement for Cygwin Console (i.e. cygwin running in a Windows console window). 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.

See <https://github.com/mintty/mintty/wiki/Tips#inputoutput-interaction-with-alien-programs> for further hints, especially on the winpty wrapper.

### 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. There is no Tektronix 4014 emulation. Mouse highlighting mode is not implemented.

The majority of xterm's many control sequences is implemented, up to

the VT300 series level; VT400 rectangular area operations are not implemented. Significant omissions should be reported to the issue tracker.

#### SEE ALSO

Additional information can be found on the wiki on the mintty project page <https://github.com/mintty/mintty/wiki>.

#### LICENSE

Copyright (C) 2013 Andy Koppe (C) 2016 Thomas Wolff

Mintty is released under the terms of the the GNU General Public License version 3. See <http://gnu.org/licenses/gpl> for the license text.

There is NO WARRANTY, to the extent permitted by law.

#### CONTACT

Please report bugs or suggest enhancements via the issue tracker at <https://github.com/mintty/mintty/issues>. Questions can be sent to the Cygwin mailing list at [cygwin@cygwin.com](mailto:cygwin@cygwin.com), to the Mintty discussion issue <https://github.com/mintty/mintty/issues/500>, or to the Gitter chat (experimental) <https://gitter.im/mintty/mintty>.

mintty

mintty(1)