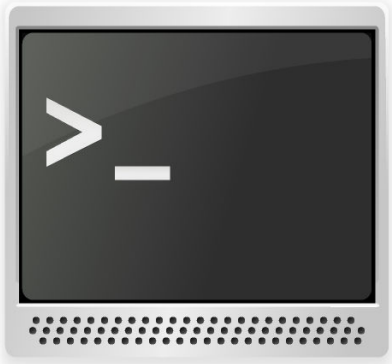
The Konsole Handbook

Jonathan Singer
Kurt Hindenburg
Ahmad Samir
Robert Knight
Kurt Hindenburg
Waldo Bastian
Mike McBride



Contents

1 Introduction **6**
- 1.1 What is a terminal? . . . 6
- 1.2 Scrollback . . . 6
- 1.3 Selection Mode . . . 6
- 1.4 Profiles . . . 7
- 1.5 Mouse Buttons . . . 7
- 1.6 Drag and Drop . . . 8
- 1.7 Semantic Shell Integration . . . 9
- 1.8 Complex Text Layout . . . 10
- 1.9 Visual Hints . . . 11

2 Command Reference **12**
- 2.1 The Menubar . . . 12
  - 2.1.1 File Menu . . . 12
  - 2.1.2 Edit Menu . . . 13
  - 2.1.3 View Menu . . . 14
  - 2.1.4 Bookmarks Menu . . . 15
  - 2.1.5 Plugins Menu . . . 16
  - 2.1.6 Settings Menu . . . 16
  - 2.1.7 Help Menu . . . 17
- 2.2 Konsole Dialogs . . . 18
  - 2.2.1 Configure Tab Settings Dialog . . . 18
  - 2.2.2 Copy Input Dialog . . . 18
  - 2.2.3 Adjust Scrollback Dialog . . . 18

3 Command-line Options **19**

4 Scripting Konsole **21**

5 Terminal Key Bindings **22**
- 5.1 How Konsole Uses Key Bindings . . . 22
  - 5.1.1 Introduction . . . 22
  - 5.1.2 Key Combinations and Modes . . . 22
  - 5.1.3 The Output Field . . . 24
  - 5.1.4 Other System Resources . . . 25
  - 5.1.5 Further Reading . . . 25

6 Using Style Sheet for the Tab Bar **26**

7 Did You Know?, Common Issues and More **27**

7.1 Did You Know? 27

7.2 Common Issues 27

8 Credits and Copyright **29**

A Links **30**

Abstract

Konsole is KDE's terminal emulator.

Chapter 1

Introduction

1.1 What is a terminal?

Konsole is an X terminal emulator, often referred to as a terminal or a shell. It emulates a command line interface in a text only window.

Konsole typically runs a command shell, an application that executes commands that you type. The shell the Konsole runs depends on your account settings. Consult your operating system documentation to know what the shell is, how to configure it and how to use it.

1.2 Scrollback

Konsole uses the notion of scrollback to allow users to view previously displayed output. By default, scrollback is on and set to save 1000 lines of output in addition to what is currently displayed on the screen.

As lines of text scroll off the top of the screen, they can be reviewed by moving the scroll bar upwards, scrolling with a mouse wheel or through the use of the **Shift**+**Page Up** (to move back), **Shift**+**Page Down** (to move forward), **Shift**+**Up Arrow** (to move up a line) and **Shift**+**Down Arrow** (to move down a line) keys.

The amount of scrolling using **Shift**+**Page Up/Down** can be switched between half and full page in the **Scrolling** tab of the profile configuration window (use **Settings** → **Edit Current Profile...** to open this window).

1.3 Selection Mode

Konsole has a selection by keyboard mode. In this mode it is possible to move around the scrollback and select text without the mouse.

Enter and leave this mode by using the keyboard shortcut (**Ctrl**+**Shift**+**D** by default).

Esc also leaves the keyboard selection mode.

Moving the cursor: Arrows, **PageUp**, **PageDown**, **Home**, **End**.

Moving the cursor vi style: h,j,k,l, to move one character, **Ctrl**+b,f,u,d for page up/down or half page up/down.

Select text by using **Ctrl** or **Shift** with arrows, or by using **V** to start selection, moving the cursor and then **V** again to end selection. **Shift**-**V** selects whole lines, instead of characters.

1.4 Profiles

Profiles allow the user to quickly and easily automate the running of common commands. Examples could include:

- ssh into another machine
- starting an irc session
- use tail to watch a file

All new and changed profiles are saved in the user's local home folder in `$XDG_DATA_HOME /konsole`.

Procedure to create a new profile:

1. Click on the menu entry **Settings** → **Manage Profiles...**
2. Switch to the **Profiles** page.
3. Click on the button **New Profile...**.
4. Fill in the first entry with a name. This is the name that will show in the menu, and will be the default label instead of **Shell** when you start a session of this type.
5. Enter a command just as you normally would if you opened a new shell and were going to issue that command. For our first example above, you might type **`ssh administration`**.
6. On the other tabs of the dialog, configure this session's appearance. You can configure a different font, color scheme, `$TERM` type and many other settings for each session.
7. Press the **OK** button. The new session is now available in the **Manage Profiles...** dialog.

1.5 Mouse Buttons

This section details the use of the mouse buttons for the common right handed mouse button order. For the left handed mouse button order, swap left and right in the text below.

Left

All left mouse button clicks will be sent to a mouse-aware application running in Konsole. If an application will react on mouse clicks, Konsole indicates this by showing an arrow cursor. If not, an I-beam (bar) cursor is shown.

Holding the left mouse button down and dragging the mouse over the screen with a mouse-unaware application running will mark a region of the text. While dragging the mouse, the marked text is displayed in reversed color for visual feedback. Select **Copy** from the **Edit** menu to copy the marked text to the clipboard for further use within Konsole or another application. The selected text can also be dragged and dropped into compatible applications. Hold the **Ctrl** key and drag the selected text to the desired location.

Normally, new-line characters are inserted at the end of each line selected. This is best for cut and paste of source code, or the output of a particular command. For ordinary text, the line breaks are often not important. One might prefer, however, for the text to be a stream of characters that will be automatically re-formatted when pasted into another application. To select in text-stream mode, hold down the **Ctrl** key while selecting normally.

Pressing the **Ctrl** and **Alt** keys along with the left mouse button will select text in columns.

Double-click with the left mouse button to select a word; triple-click to select an entire line.

If the upper or lower edge of the text area is touched while marking, Konsole scrolls up or down, eventually exposing text within the history buffer. The scrolling stops when the mouse stops moving.

After the mouse is released, Konsole attempts to keep the text in the clipboard visible by holding the marked area reversed. The marked area reverts back to normal as soon as the contents of the clipboard change, the text within the marked area is altered or the left mouse button is clicked.
To mark text in a mouse-aware application (Midnight Commander, for example) the **Shift** key has to be pressed when clicking.

Middle

Pressing the middle mouse button pastes text currently in the clipboard. Holding down the **Ctrl** key as you press the middle mouse button pastes the text and appends a new-line. That is convenient for executing pasted command quickly, but it can be dangerous so use it with caution.

> NOTE
> If you have a mouse with only two buttons, pressing both the left mouse button and right mouse button together emulates the middle mouse button of a three button mouse.

If you have a wheel as the middle button, rolling it in a mouse-unaware program will move Konsole's scrollbar.

Right

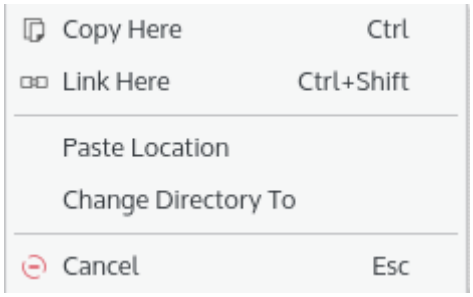
These items appear in the menu when the right mouse button is pressed:

- **Copy**
- **Paste**
- With a text selection a submenu **Search for** with a list of the preferred Web Shortcuts and an option to configure web shortcuts.
- **Open File Manager**
- **Set Encoding**
- **Clear Scrollback**
- **Adjust Scrollback...**
- **Show Menu Bar**, only when the menubar is hidden
- **Switch Profile**
- **Edit Current Profile...**
- **Close Tab**

In a mouse aware application, press the **Shift** key along with the right mouse button to get the popup menu.

1.6 Drag and Drop

If you drop a file, folder or URL on a Konsole window, a context menu appears with these actions:



Move Here (Shift)

Move the dropped item into the current folder. This item only appears in the context menu, if you have the rights to delete the dropped file or folder.

Copy Here (Ctrl)

Copy the dropped item into the current folder.

Link Here (Ctrl-Shift)

Insert a symbolic link to the dropped item.

Paste Location

Insert the full file path of the dropped item at the cursor.

Change Directory To

If a folder is dropped, this action appears in the context menu and allows you to change the working folder of the Konsole session.

Cancel (Esc)

Break the drag and drop action.

If you press the shortcuts before releasing the left mouse button during drag and drop, no context menu appears and the actions will be executed immediately.

If you want to use the **Ctrl** key for drag and drop or disable the context menu to insert URLs as text by default, enable the corresponding options on the **Mouse** tab in the profile settings dialog.

1.7 Semantic Shell Integration

A shell program running in Konsole may emit escape sequences that partition the text displayed into three types: shell prompt, user input and command output. Using this semantic information enables various enhancements in Konsole.

- **Ctrl**+**Shift**+**PgUp** and **Ctrl**+**Shift**+**PgDown** scroll up/down to previous/next command prompt.
- Visual hints:
  - A line is displayed above each prompt, prompt colors are less intense, and output colors are more intense.
  - A red bar is displayed to the left of input and output lines of commands that resulted in error
  - A red background for input and output lines of commands that resulted in error
  - A gray bar is displayed to the left of the input and output lines of every other command.
  - A gray background for the input and output lines of every other command.

  Each of those may be configured to never show, always show, or only when URL hints are shown. The configuration is in the **Semantic Integration** tab of the **General** page of the profile configuration window.
- Context menu options **Copy user input**, **Copy command output**, and **Copy except prompt** may be used to filter selection when it is copied to the clipboard.
- When selection is empty, copy to clipboard action copies the current input line if it is not empty, or the last output if there is no current input.
- Pressing Up/Down arrow when editing a long input, will instead place the cursor one line up/down by sending the appropriate number of Left/Right key events to the shell. Configurable in the profile settings.

- Clicking the mouse on text input will place the cursor in the clicked location. Configurable in the profile settings.
- Pressing the **Ctrl** key while triple clicking the mouse on the output of a command, selects the whole output of that command.

Semantic shell integration needs to be setup in the shell. Pressing **Ctrl-Alt-]** will paste the necessary commands needed in bash. For other shells, such as fish, zsh, python, etc. consult the relevant program's documentation.

1.8 Complex Text Layout

In the **Complex Text Layout** tab of the **Appearance** page in the **Edit Profile** dialog, you will find options that control the rendering of text.

- **Word mode** - in this mode (some) strings are displayed in the screen as a whole, instead of a character at a time. This allows Qt to render text correctly when the shape of a character depends on the characters before or after it. This might result in incorrect position of some characters.
  Spaces always break strings, so they are always in the correct positions. This ensures that characters are never too far from their correct position.
- **Use the same attributes for each whole word** - When this is enabled, words are rendered with the same attributes (text color, bold, italic, etc.). If an attribute changes mid-word, it will only take effect after the end of the word. When this is disabled, a new word starts when attributes change. This results in characters changing shape and position when moving the cursor or selecting text.
- **ASCII characters** - Group ASCII characters into words as described above. The most noticeable effect of this option is that enabling this shows programming ligatures (for fonts that support them). E.g. the string <= may be displayed as ⩽.
- **Brahmic scripts characters** - Group Brahmic characters as described above. Without this option (depending on the font) some words may not be connected as they should. With this enabled Brahmic characters may appear out of position. E.g. the third character in the second line might not appear directly under the third character in the first line.
- **Font hinting:** - This enables applying hinting to the glyphs when rendering them. It may improve the overall legibility of characters when using certain fonts and some displays. It can also help resolve some situations where the characters are not rendered with the correct width or height, especially when word mode is enabled. If set, full hinting is used. If not set, the default level of hinting is used.
- **Emoji Font:** - This allows specifying the font to use for Unicode Emoji characters. If not set, the default profile font will be used, or some fallback font may be used by the system if the glyphs are missing from this font.
- **Bi-Directional text rendering** - Reorder right to left characters so Arabic and Hebrew texts appear correctly.
- **force LTR line direction** - Lines are always Left to right. Without this, each line's direction is determined by the first character with strong directionality.
- **Table characters BiDi mode override** - Consider graphic table characters as strong LTR characters. This allows table containing RTL characters to display correctly, but might cause incorrect order if those characters are used in RTL texts.
- **Override wcwidth** - Problematic characters follow Unicode standard, rather than glibc's wcwidth(). Currently only soft hyphen (Unicode 0x00AD) which has wcwidth of 1 and Unicode width of 0 is affected by this option. Generally, this option should be disabled when mainly using those characters on the command line, and enabled when they are only displayed.

1.9 Visual Hints

In addition to the various visual hints described in Section 1.7, Konsole has other visual hints:

- A vertical line at column 80 (or another). This is configured in the **Miscellaneous** tab of the **Appearance** page in the **Edit Profile** dialog.
- Line numbers may be displayed as an overlay of the terminal text. Line numbers appear in red on the right end of each line. The line are numbered consecutively from the first (top) line in the scrollback. Displaying the line numbers can be configured in the **Advanced** page in the **Edit Profile** dialog. Cycling between the three displayed modes can also be done by a keyboard shortcut. The default shortcut is **Ctrl**-**Alt**-****

Chapter 2

Command Reference

2.1 The Menubar

The menubar is at the top of the Konsole window. If the menubar is hidden, **Show Menu Bar** can be reached by right clicking in the window (as long as no full screen application is running in that window such as vi, minicom, etc.). The default shortcut is listed after each menu item.

Alternatively you can use the shortcut **Ctrl**+**Shift**+**M** to show or hide the menubar.

2.1.1 File Menu

File → **New Window (Ctrl+Shift+N)**
Opens a new separate Konsole window with the default profile

File → **New Tab (Ctrl+Shift+T)**
Opens a new tab with the default profile

> NOTE
> The first profile in the submenu will always be "Default" which is the built-in profile. All other profiles will be listed below that in alphabetical order. The user specified default profile will be in bold type.

File → **Clone Tab**
Attempts to clone the current tab in a new tab

File → **Save Output As... (Ctrl+Shift+S)**
Saves the current scrollback as a text or html file

File → **Print Screen ... (Ctrl+Shift+P)**
Print the current screen. By default the output is scaled to fit the size of the paper being printed on with black text color and no background. In the print dialog these options can be changed on the **Output Options** tab.

File → **Open File Manager**
Opens KDE's file manager at the current directory. By default, that is Dolphin.

File → **Close Session (Ctrl+Shift+W)**
Closes the current session

File → Close Window (Ctrl+Shift+Q)

Quits Konsole

> NOTE
> Konsole will display a confirmation dialog if there is more than one session open or if there are certain programs running in any session. These dialogs can be disabled by clicking on the **Do not ask again** checkbox.

2.1.2 Edit Menu

Edit → Copy (Ctrl+Shift+C)

Copies the selected text to the clipboard

Edit → Paste (Ctrl+Shift+V)

Pastes text from the clipboard at the cursor location

Edit → Select All

Selects all the text in current window

Edit → Copy Input To → All Tabs in Current Window

Allows input from the current session to be sent simultaneously to all sessions in current window

Edit → Copy Input To → Select Tabs... (Ctrl+Shift+.)

Allows input from the current session to be sent simultaneously to sessions picked by user

Edit → Copy Input To → None (Ctrl+Shift+/)

Stop sending input from current session into other sessions

Edit → Send Signal

Send the specified signal to the shell process, or other process, that was launched when the new session was started.
Currently available signals are:

STOP	to stop process
CONT	continue if stopped
HUP	hangup detected on controlling terminal, or death of controlling process
INT	interrupt from keyboard
TERM	termination signal
KILL	kill signal
USR1	user signal 1
USR2	user signal 2

Refer to your system manual pages for further details by giving the command **`man 7 signal`**.

Edit → Configure Tab Settings... (Ctrl+Alt+S)

Opens a dialog box allowing you to change the name format, the remote tab title format and the color of the current tab (more info)

Edit → ZModem Upload... (Ctrl+Alt+U)

Opens up a dialog to select a file to be uploaded if the required software is installed

Edit → Find... (Ctrl+Shift+F)

Opens a search bar at the bottom of Konsole's window

This allows for case sensitive, forward or backwards, and regular expressions searches.

Edit → Find Next (F3)

Moves to the next search instance . If the search bar has the focus, you can use the shortcut **Enter** as well.

Edit → Find Previous (Shift+F3)

Moves to the previous search instance . If the search bar has the focus, you can use the shortcut **Shift**+**Enter** as well.

2.1.3 View Menu

View → Split View → Split View Left/Right (Ctrl+()

Splits all the tabs into left and right views

Any output on one view is duplicated in the other view.

View → Split View → Split View Top/Bottom (Ctrl+))

Splits all the tabs into top and bottom views

Any output on one view is duplicated in the other view.

View → Split View → Expand View (Ctrl+Shift+])

Makes the current view larger

View → Split View → Shrink View (Ctrl+Shift+[)

Makes the current view smaller

View → Split View → Toggle maximize current view (Ctrl+Shift+E)

Toggles the current view between current size and the maximize size

View → Split View → Equal size to all views (Ctrl+Shift+\)

Sets the same size for all views

View → Detach Current Tab (Ctrl+Shift+L)

Opens the current tab in a separate window

Quitting the previous Konsole window will not affect the newly created window.

View → Detach Current View (Ctrl+Shift+H)

Opens the current split view in a separate window

View → Save tab layout to file

Allows you to save the tab layout of the current view in a specialized Konsole layout file which can then be loaded to restore one of your favorite layouts.

View → Load tab layout from file

Allows you to load one of your favorite view layouts from the layout file that has been saved using the **View → Save tab layout to file** menu item earlier. The default layouts (2x2, 2x1, and 1x2) can be loaded via toolbar.

View → **One-shot monitors**

The following monitors only notify once, and are then disabled.

View → **Monitor for Prompt (Ctrl+Shift+R)**

Toggles the monitoring of the current tab for a shell prompt.

When a shell prompt is displayed, Konsole will show a notification. This option is shown only when semantic integration is enabled in the shell.

View → **Monitor for Silence (Ctrl+Shift+I)**

Toggles the monitoring of the current tab for lack of activity

By default, after 10 seconds of inactivity, an info icon will appear on the session's tab. The type of alerts can be changed through **Settings** → **Configure Notifications** → **Silence in monitored session**.

View → **Monitor for Activity (Ctrl+Shift+A)**

Toggles the monitoring of the current tab for activity

Upon any activity, an info icon will appear on the session's tab. The type of alerts can be changed through **Settings** → **Configure Notifications** → **Activity in monitored session**.

View → **Monitor for Process Finishing**

Toggles the monitoring of the current tab for the process finishing.

If checked, upon finishing the current process, Konsole will show a notification **The process '*`name of the process`*' has finished running in session '*`name of the session`*'**.

View → **Read-only**

Toggles the session to be read-only: no input is accepted, drag and drop is disabled.

View → **Enlarge Font (Ctrl++)**

Increases the text font size

View → **Reset Font Size (Ctrl+0)**

Reset the text font size to the profile default

View → **Shrink Font (Ctrl+-)**

Decreases the text font size

View → **Set Encoding**

Sets the character encoding

View → **Clear Scrollback**

Clears the text in the scrollback

View → **Clear Scrollback and Reset (Ctrl+Shift+K)**

Clears the text in the current tab and scrollback and resets the terminal

View → **Full Screen Mode (F11)**

Toggles Konsole filling the entire screen

2.1.4 Bookmarks Menu

Bookmarks → **Add Bookmark (Ctrl+Shift+B)**

Adds the current location

Bookmarks → **Bookmark Tabs as Folder...**

Adds all tabs to a bookmark folder
A dialog will open for the bookmark folder name.

Bookmarks → **New Bookmark Folder...**
Adds a new folder to the bookmark list
A dialog will open for the bookmark folder name.

Bookmarks → **Edit Bookmarks**
Opens the bookmark editor

> NOTE
> The keditbookmarks program must be installed for this menu item to appear.
> You can use the bookmark editor to manually add URLs. Currently, Konsole accepts the following:
>
> - ssh://user@host:port
> - telnet://user@host:port

2.1.5 Plugins Menu

Any installed plugins will be listed or ″No plugins available″

2.1.6 Settings Menu

Settings → **Edit Current Profile...**
Opens a dialog to configure current profile

Settings → **Switch Profile**
Switch current profile to a listed profile

Settings → **Manage Profiles...**
Opens an editor for managing profiles

Settings → **Window Color Scheme**
Change Konsole's GUI to the specified scheme

Settings → **Show Menu Bar (Ctrl+Shift+M)**
Toggles the menubar being visible

Settings → **Toolbars Shown**
Allows visibility toggling of the Konsole toolbars

Settings → **Show Statusbar**
Toggles the statusbar being visible

Settings → **Configure Language...**
Opens window to choose an interface translation of Konsole.

Settings → **Configure Keyboard Shortcuts...**
Opens the keyboard shortcut editor. More on shortcuts configuration can be found in the KDE Fundamentals.
Additionally Konsole has a few special shortcuts with no corresponding menu item:

Shortcut	Description
Shift+Right	Next Tab
Shift+Left	Previous Tab
Ctrl+Alt+Left	Move Tab Left
Ctrl+Alt+Right	Move Tab Right
Ctrl+Shift+Ins	Paste Selection

Settings → **Configure Toolbars...**

Opens the toolbar configuration window

Settings → **Configure Notifications...**

Opens the notifications editor

Settings → **Configure Konsole...**

Opens the Konsole settings editor

This dialog has options influencing the appearance and behaviour of the Konsole window.

- The **General** page allows configuring the menubar visibility, remembering the Konsole window size, running all Konsole windows in a single process, enabling menu accelerators, showing window title on the titlebar, removing window titlebar and frame, and focusing terminals when the mouse pointer is moved over them. It is also possible to configure the search case sensitivity, usage of the regular expression, highlighting all matches, and the search direction (**Search backwards** is the default). The **General** page is also the place where you can **Enable all "Don't Ask Again" messages** again if they were switched off before.
- The **Profiles** page is meant for creating and handling profiles.
- Using the **Tab Bar/Splitters** page you can configure the visibility and placement of tab bar, define tabs behavior and fine-tune the tab buttons options. It is possible to configure if you want to **Show 'New Tab' button** and **Expand individual tab widths to full window** or configure **Use user-defined stylesheet**. The **Behavior** tab can be used to define the place for the new tabs (**At the end** or **After current tab**) and closing of tabs with a middle mouse button click.
  It is also possible to configure visibility of the split headers (**When needed** (default), **Always**, or **Never**) and define the drag handle size for splits (**Small** (default), **Medium**, or **Large**) using the **Splits** tab of this configuration page.
- The **Temporary Files** page is used to define the scrollback file location.
- The **Thumbnails** page can be used to define the thumbnail size and activation options (you can choose the activation control key from **Shift**, **Alt**, and **Ctrl**).

> NOTE
> To use the thumbnails feature showing image thumbnails in popups when hovering image items with the mouse pointer, you should enable file underlining for your current profile: **Settings** → **Edit Current Profile...** → **Mouse** → **Miscellaneous** → **Underline files**.

2.1.7 Help Menu

Konsole has the some of the common KDE **Help** menu items, for more information read the section about the Help Menu of the KDE Fundamentals.

2.2 Konsole Dialogs

2.2.1 Configure Tab Settings Dialog

The name format, the remote tab title format, and the color of the current tab can be changed from this dialog. The dialog can be displayed via the menu, the shortcut **Ctrl**+**Alt**+**S** or by double-clicking on the tab in the tab bar. These changes are temporary and can be made permanent by editing the current profile.

Konsole will substitute these tokens for local tabs:

- %n : program name
- %d : current directory (short)
- %D : current directory (long)
- %h : local host (short)
- %u : user name
- %B : user's Bourne prompt sigil ($ = normal user, # = superuser)
- %w : window title set by shell
- %# : session number

Konsole will substitute these tokens for remote tabs:

- %c : current program
- %h : remote host (short)
- %H : remote host (long)
- %u : user name
- %U : user name@ (if given)
- %w : window title set by shell
- %# : session number

Examples:

- **`%d : %n`** with /usr/src as current directory and running bash will display **src : bash**
- **`%D : %n`** with /usr/src as current directory and running top will display **/usr/src : top**
- **`%w (%#)`** with ~ as current directory and running vim in the first tab will display **[No Name] (~) - VIM(1)**

2.2.2 Copy Input Dialog

The text entered in one tab can simultaneously be sent to other tabs. This dialog allows you to select which tabs will get that input. The current tab will be greyed out.

2.2.3 Adjust Scrollback Dialog

The scrollback options for the history size can be changed in this dialog. Any changes are for the current tab only and will not be saved to the profile.

Chapter 3

Command-line Options

When Konsole is started from the command line, various options can be specified to modify its behavior.

`--help`
: List various options.

`--profile` ***`file`***
: Start Konsole using the specified profile instead of the default profile.

`--layout` ***`file`***
: Start Konsole using a saved JSON layout file.

`--builtin-profile`
: Use the built-in profile instead of the current default profile.

`--workdir` ***`dir`***
: Open with *`dir`* as the initial working directory.

`--hold, --noclose`
: Do not close the initial session automatically when it ends.

`--new-tab`
: Create a new tab in an existing window rather than creating a new window.

`--tabs-from-file` ***`file`***
: Create tabs as specified in the given tabs configuration file.

> NOTE
> The file has one tab per line in the following format:
> Each line specifies a tab to open using up to 4 fields specifying how it is to open. Fields are delimited with **`;;`** and a field name must have a **`:`** appended. Empty lines or lines with **#** at the beginning are ignored, so you can use line beginning with **#** to add comments.
> **`title:`** a name for this tab, tab default if blank or not specified
> **`workdir:`** working directory, ~ if blank or not specified
> **`profile:`** a Konsole profile to use, the default if blank or not specified
> **`command:`** a command to run
> Each line should contain at least one of **`command`** or **`profile`** field.
> Example: **`title: %n;; command: /usr/bin/top ;; profile: Shell`**

--background-mode
: Start Konsole in the background and bring to the front when **Ctrl**+**Shift**+**F12** (by default) is pressed.

--separate, --nofork
: Run the new instance of Konsole in a separate process.

--show-menubar
: Show the menubar, overriding the default behavior.

--hide-menubar
: Hide the menubar, overriding the default behavior.

--show-tabbar
: Show the tabbar, overriding the default behavior.

--hide-tabbar
: Hide the tabbar, overriding the default behavior.

--fullscreen
: Start Konsole in fullscreen mode.

--notransparency
: Disable transparent backgrounds, even if the system supports them.

--list-profiles
: List all available profiles.

--list-profile-properties
: List all possible properties with name and type. See option `-p`.

  For more information, please visit Konsole API Reference.

-p *property=value*
: Change the value of a profile property.

-e *command*
: Execute *command* instead of the normal shell.

> **NOTE**
> This option will catch all following arguments passed to Konsole, and execute it as *command*. So this option should always be used as the last option.

Konsole also accepts generic Qt™ and KDE Frameworks 6 options, see man pages qt6options and kf6options.

Chapter 4

Scripting Konsole

Konsole does support numerous methods that can be used with D-Bus.

There are two ways to use the D-Bus interface: Qt™'s GUI qdbusviewer and the command line qdbus.

Examples:

- `%` **qdbus** will display all services available.
- `%` **qdbus** `org.kde.konsole` will display the D-Bus interface for Konsole.
- `%` **qdbus** `org.kde.konsole /Windows/1` will display methods for controlling window 1.
- `%` **qdbus** `org.kde.konsole $KONSOLE_DBUS_WINDOW` will display methods for controlling the current window.
- `%` **qdbus** `org.kde.konsole /Sessions/1` will display methods for controlling session 1.
- `%` **qdbus** `org.kde.konsole $KONSOLE_DBUS_SESSION` will display methods for controlling the current session.
- `%` **qdbus** `$KONSOLE_DBUS_SERVICE $KONSOLE_DBUS_SESSION` will display methods for controlling the current Konsole's session.

If any of the above commands outputs: Service 'org.kde.konsole' does not exist, change `org.kde.konsole` to one of the following:

- `org.kde.konsole-`` `pidof -s konsole` `` (will select first pid)
- `$KONSOLE_DBUS_SERVICE` (this can be used from the current Konsole)
- `select one from the output of 'qdbus | grep konsole'`

For more information, please visit D-Bus tutorial.

Chapter 5

Terminal Key Bindings

5.1 How Konsole Uses Key Bindings

5.1.1 Introduction

Konsole uses *.keytab files to translate key combinations into control characters and escape sequences that are sent to the shell or to interactive programs (typically programs that use the Alternate Screen buffer, e.g. vim, less, screen) running in the shell.

Users can customize the key bindings settings in Konsole using the Key Bindings Editor. A key combination can be configured to send a specific control or escape sequence to the terminal.

You can open the Key Bindings Editor from the menu entry **Settings** → **Edit Current Profile**, and going to the **Keyboard** tab. Listed there are the Key Bindings schemas that come by default with Konsole.

5.1.2 Key Combinations and Modes

Key combinations follow the pattern:

```
Key (+|-) Modes
```

for example:

```
Up+Shift+AppScreen
Down+Shift-AppScreen
Space+Ctrl
```

Key names are defined in the qnamespace.h header file, with the 'Qt::Key_' prefix removed, for a list of key names check the Qt::Key enumeration in the Qt documentation.

A '+' preceding a Mode name means that mode is *set*; for a modifier key, that means it's pressed, whereas for all other modes it means that particular mode is in effect (i.e. active). For example '+Ctrl' means the key combination will work only if the **Ctrl** key is pressed.

A '-' preceding a Mode name means that mode is *reset*; basically this is the opposite of putting '+' before a Mode name, so for a modifier key that means the key isn't pressed, whereas for all other modes it means that particular mode is inactive. For example '-Ctrl' means the key combination will work only if the **Ctrl** key is *not* pressed.

> NOTE
> If a Mode name isn't present in a key combination, its state is ignored.

The supported Key Bindings modes are listed below:

Alt, Ctrl, Shift

One or more of these Modes can be used in a key combination, if any of them is set, the key combination uses that modifier key, respectively; and vice versa if it's reset

AnyModifier

If this mode is set, the key combination uses any modifier key (any of the previous three modifier keys); and vice versa if it's reset

Ansi

If this mode is set, Konsole will send ANSI escape and control sequences

If this mode is reset Konsole will send VT52 escape and control sequences

AppScreen

If this mode is set, the key combination will only affect interactive programs that use the Alternate Screen buffer
If this mode is reset the key combination will only affect the terminal when it's using the Normal Screen buffer

> NOTE
> Konsole makes use of two screen buffers:
>
> - The Normal Screen buffer (default): allows you to scroll back to view previous lines of output, this is the default buffer you usually use to execute commands... etc.
> - The Alternate Screen buffer: the terminal switches to this buffer when you run an interactive program (e.g. less, vim, screen, tmux... etc.)

KeyPad

If this mode is set, the key combination uses a key on the Keypad (Number Pad). This mode is useful to distinguish between keys on the keyboard and keys on the Keypad. For example when Num Lock is *on* you can configure two separate key combinations, one using the key labelled '1' on the keyboard (usually under the **F1** key) and the other using the key labelled '1' on the Keypad. The same concept applies when Num Lock is *off* for the End, Home, Cursor Keys ...etc on the Keypad

AppCursorKeys

This mode implements the VT100 Cursor Keys Mode (DECCKM). It controls the escape sequences each Cursor Key (**Up**, **Down**, **Right**, **Left**) sends, depending on whether this mode is set or reset
By default Konsole follows the XTerm behavior of treating the **Home** and **End** keys as cursor keys with respect to DECCKM

AppKeyPad

If this mode is set, the key combination will only work when the Keypad is in Application Mode (DECKPAM)

If this mode is reset, the key combination will only work when the Keypad is in Numeric Mode (DECKPNM)

NewLine

If this mode is set, the **Return** (Enter) key on the keyboard will send both Carriage Return ″\r″ and New Line ″\n″ control characters
If this mode is reset, the **Return** key will send only a Carriage Return ″\r″

The same applies to the **Enter** key on the Keypad

This mode emulates the LNM - Line Feed/New Line Mode

Note that each combination of Key and Modes (set/reset) must be unique. For example, consider the following two rules:

- A+Shift : 'A'
- a : 'a'

Konsole will *not* accept the small letter 'a' rule, you have to add a '-Shift' to that rule to make it work.

5.1.3 The Output Field

In the Output field you can add the escape sequences or control characters that you want Konsole to send to the terminal when the associated key combination is pressed.

You can also use any of the following keywords, each of which has a special meaning in Konsole:

- scrollUpLine : scroll up one line in the shell history scrollback buffer
- scrollUpPage : scroll up one page in the shell history scrollback buffer
- scrollDownLine : scroll down one line in the shell history scrollback buffer
- scrollDownPage : scroll down one page in the shell history scrollback buffer
- scrollUpToTop : scroll up to the begining of the shell history scrollback buffer
- scrollDownToBottom : scroll down to the end of the shell history scrollback buffer

You can also use strings with C-string syntax; you may use the following escapes sequences:

- \E : Escape
- \\: Backslash
- \″ : Double quote
- \t : Tab
- \r : Carriage Return
- \n : New line
- \b : Backspace
- \xHH : where HH are two hex digits

> TIP
> This can be used to send ASCII control characters, e.g. '\x00' which is the NUL character

5.1.4 Other System Resources

There are other system resources that can affect terminal Key Bindings:

- Consult the terminfo or termcap database for the expected escape sequences and control characters that each key combination is supposed to send.
- It is likely that your system has other keyboard databases which have to be in sync too, (e.g. /etc/inputrc and readline for the BASH shell) as they affect the operations (interactions) bound to key combinations.

5.1.5 Further Reading

For more information on escape sequences and control characters, check the following documentation:

- The VT100 user guide
- The VT102 user guide
- The comprehensive and indispensable XTerm Control Sequences documentation

Chapter 6

Using Style Sheet for the Tab Bar

The default style sheet for the tab bar sets the minimum and maximum tab widths. The user can create a `.css` file and have Konsole use that as the style sheet for the tab bar. In the `.css` file, the widget to use is **`QTabBar::tab`**.

For more information, consider reading Qt Style Sheets

Examples:

- Change the selected tab's background to a light gray

```
QTabBar::tab:selected {
    background: #999999
}
```

- Change the selected tab's text to red

```
QTabBar::tab:selected {
    color: red
}
```

- All tabs will be at least 200 pixels in width

```
QTabBar::tab {
    min-width: 200px
}
```

- Only the selected tab will be at least 200 pixels in width

```
QTabBar::tab::selected {
    min-width: 200px
}
```

- Any of these can be combined in one file

```
QTabBar::tab::selected {
    background: #999999;
    color: red;
    min-width: 200px;
}
QTabBar::tab {
    min-width: 100px
}
```

Chapter 7

Did You Know?, Common Issues and More

7.1 Did You Know?

- Pressing **Ctrl** while selecting text will cause lines breaks to be converted to spaces when pasted.
- Pressing the **Ctrl**+**Alt** keys while selecting text will select columns.
- The **Ctrl**+**Wheel** combination will zoom text size, like in konqueror and firefox.
- When a program evaluates either mouse button, pressing the **Shift** key will allow the popup menu to appear.
- The **Ctrl**+**Shift**+**F10** shortcut will activate the menu.
- The **Shift**+**Insert** keys will insert the clipboard.
- Double-clicking will select a whole word. Continuing to hold the mouse button and moving the mouse will extend the selection.
- Triple-clicking will select a whole line. Continuing to hold the mouse button and moving the mouse will extend the selection.
- There is a hidden feature for the "%d" formatter in tab title. You can tell Konsole to abbreviate a directory name into its first character. For example, "/path/to/konsole/src" can be abbreviated into "konsole/s". If you want to enable and control this hidden feature, open `konsolerc` in **`qtpaths --paths GenericConfigLocation`** and add following lines:

```
[ProcessInfo]
CommonDirNames=name1,name2,name3...
```

> NOTE
> If you are using Yakuake, you need to edit `yakuakerc` in **`qtpaths --paths GenericConfigLocation`** instead.

7.2 Common Issues

- Some fonts might be unavailable for usage in Konsole, although they are available in other applications. That doesn't mean there is a bug in Konsole. Konsole requires monospaced fonts to provide the best visual result, so it asks Qt™ to only list monospaced fonts.

Starting with version 16.08 (August 2016), Konsole can be configured to allow selecting any font with the caveat that the display may not be correct.

- Since KDE4 all the tabs use the same process ID. This has the side-effect that if one tab's process has issues, all the other tabs may experience issues as well.

  This is most noticeable when a command that connects to an external device or system (ssh, nfs) has issues.

- Konsole treats arguments after the `-e` option as one command and runs it directly, instead of parsing it and possibly dividing it into sub-commands for execution. This is different from xterm.

  – **konsole -e "command1 ; command2"** does not work
  – **konsole -e $SHELL -c "command1 ; command2"** works

- Konsole doesn't provide convenience for running login shell, because developers don't like the idea of running login shell in a terminal emulator.

  Of course, users still can run login shell in Konsole if they really need to. Edit the profile in use and modify its command to the form of starting a login shell explicitly, such as "**bash -l**" and "**zsh -l**".

- The `--new-tab` option sometimes behaves strangely. It may create new window, or it may create new tab in another existing Konsole window instead of the current Konsole window.

  Those behaviors feel strange, but they are not necessarily bugs. The `--new-tab` option tries to reuse existing Konsole windows, but not all Konsole windows are reusable. All Konsole windows opened through KRunner are reusable, while most Konsole windows opened from command line are not.

Chapter 8

Credits and Copyright

Konsole is currently maintained by Kurt Hindenburg kurt.hindenburg@gmail.com

Previous Konsole maintainers include: Robert Knight robertknight@gmail.com and Waldo Bastian bastian@kde.org

The application Konsole Copyright (c) 1997-2008 Lars Doelle lars.doelle@on-line.de

This document was originally written by Jonathan Singer jsinger@leeta.net

This document was updated for KDE 4.x by Kurt Hindenburg kurt.hindenburg@gmail.com

This document was updated for KDE 3.4 by Kurt Hindenburg kurt.hindenburg@gmail.com

Originally converted to DocBook SGML by Mike McBride and Lauri Watts

This documentation is licensed under the terms of the GNU Free Documentation License.

This program is licensed under the terms of the GNU General Public License.

Appendix A

Links

For more information please visit these websites:

- Konsole's homepage on KDE's UserBase
- Konsole's homepage
- Konsole's mailing list
- KDE on FreeBSD
- KDE on Solaris