

THE ANSITERM PACKAGE

VO.2.1 2026-08-26

ANSI-aware terminal windows and executable shell transcripts

Matthias WERNER¹

<https://github.com/werner-matthias/osglecture>

The package provides two environments that display text in a box that resembles a terminal.

In this context, `ansitermexec` allows (when `--shell-escape` is enabled) the execution and logging of shell commands and their results, with ANSI control characters being correctly interpreted. If `\DocumentMetadata{tagging=on}` is set, the output is tagged.

This package is part of the `osglecture` bundle, but is largely independent of the other packages.

Table of Contents

1 Usage	1 2 Implementation	4
1.1 Examples	2	
1.2 Options	3 Index	5

1 Usage

Attention, Security Risk!

Environment `ansitermexec` requires the translation with `--shell-escape` to run. This poses a security risk, as it allows Lua^AT_EX to execute arbitrary code on your system. Make sure to write only non-critical code in the body of the environment.

Load the package in the usual way with `\usepackage{ansiterm}`. There are no package options. The package provides two environments and needs Lua^AT_EX.

`\begin{ansiterm}[\langle options \rangle]`

Displays the content within a terminal window.

1. matthias.werner@informatik.tu-chemnitz.de

`\begin{ansitermexec}[\langle options \rangle]`

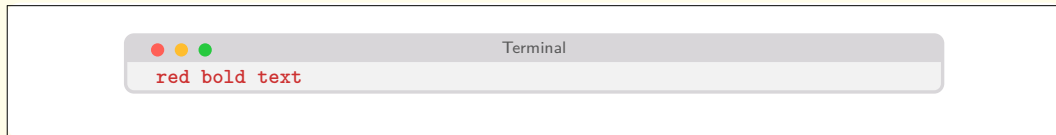
Passes the content line by line to a shell for execution and displays the input and output in a terminal window. Benötigt zur Übersetzung `--shell-escape`

1.1 Examples

```

1 \documentclass{article}
2 \usepackage{ansiterm}
3 \begin{document}
4   \begin{ansiterm}[escape-char=$]
5     ${1;31m}red bold text${0m}
6   \end{ansiterm}
7 \end{document}

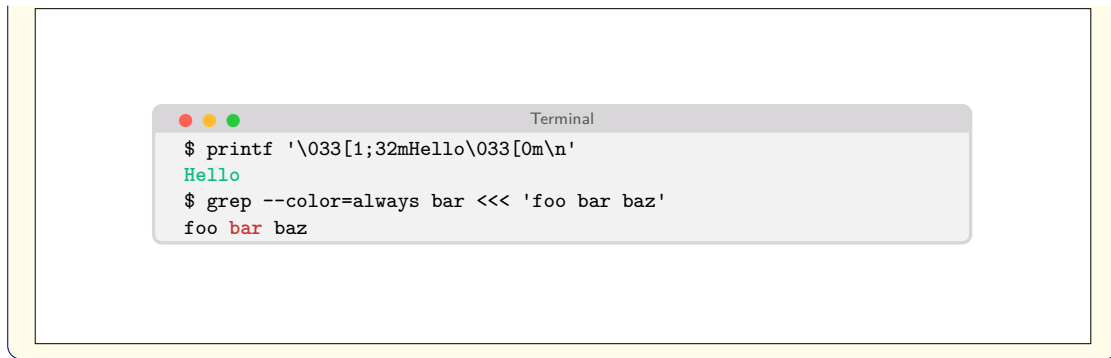
```



```

1 \documentclass{article}
2 \usepackage{ansiterm}
3 \begin{document}
4   \begin{ansitermexec}[
5     shell=/bin/sh,
6     cache=true,
7     sandbox=true,
8     show-input=true,
9   ]
10   printf '\033[1;32mHello\033[0m\n'
11   grep --color=always bar <<< 'foo bar baz'
12 \end{ansitermexec}
13 \end{document}

```



1.2 Options

title = {*<Terminal title>*} Default: terminal

fontsize = {**} Default: *small*

ansitermexec shell = {*<shell>*} Default: /bin/sh
shell for execution

ansitermexec cwd = {*<directory>*} Default: .
working directory for execution

ansitermexec cache = *true|false|refresh* Default: *true*
Specifies whether the result is reused when the translation is performed again. *refresh* renews the cache.

ansitermexec sandbox = *true|false* Default: *true*
If the key is *true*, the command is executed in a newly created directory, which is then deleted. The results are still cached, if applicable.

ansitermexec show-input = *true|false* Default: *true*
If *false*, only the output is displayed.

ansitermexec prompt = {*<prompt>*} Default: \$~

escape-char = {*<escape character>*} Default: \$

lines = {*<line range>*} Default: -
Range of lines to display.

ellipsis = *true|false* Default: *true*
Displays ellipsis “:” for skipped lines

ansitermexec fail-on-error = *true|false* Default: *true*
Generates an error message if the execution returns an error status.

2 Implementation

To view the implementation description, please uncomment `\OnlyDescription` in `ansitem.dtx`.

Index

Symbols

`--shell-escape`..... 2

A

`ansiterm` (environment) 1

`ansitermexec` (environment) 1 ff.

C

`cache`..... 3

`cwd`..... 3

D

`\DocumentMetadata`..... 1

E

`ellipsis` 3

`escape-char`..... 3

F

`fail-on-error`..... 3

`fontsize`..... 3

L

`lines`..... 3

O

`\OnlyDescription`..... 4

P

`prompt` 3

S

`sandbox` 3

`shell`..... 3

`show-input` 3

`\small` 3

T

`title`..... 3

U

`\usepackage` 1

`ansitermexec`

`ansitermexec`