

# Package ‘weaver’

July 2, 2025

**Title** Tools and extensions for processing Sweave documents

**Version** 1.75.0

**Author** Seth Falcon

**Description** This package provides enhancements on the Sweave() function in the base package. In particular a facility for caching code chunk results is included.

**Maintainer** Seth Falcon <seth@userprimary.net>

**License** GPL-2

**Depends** R (>= 2.5.0), digest, tools, utils, codetools

**Suggests** codetools

**biocViews** Infrastructure

**LazyLoad** yes

**git\_url** <https://git.bioconductor.org/packages/weaver>

**git\_branch** devel

**git\_last\_commit** dcce163

**git\_last\_commit\_date** 2025-04-15

**Repository** Bioconductor 3.22

**Date/Publication** 2025-07-01

## Contents

internal_use_only . . . . .	1
weaver . . . . .	2
<b>Index</b>	<b>4</b>

---

internal_use_only	<i>Internal functions for the weaver package</i>
-------------------	--

---

## Description

These functions are not part of the public API of the package.

---

weaver

*A modified R/LaTeX Driver for Sweave*


---

## Description

A modified driver for [Sweave](#) that translates R code chunks in LaTeX files. Presently, the primary modification is that this version provides chunk caching. In the chunk options, you can specify `cache=TRUE` and the results of that chunk will be cached.

## Usage

```
weaver()
```

```
weaverLatexSetup(file, syntax, output = NULL, quiet = FALSE, debug = FALSE, echo = TRUE, eval = TRUE,
```

## Arguments

<code>file</code>	Name of Sweave source file.
<code>syntax</code>	An object of class <code>SweaveSyntax</code> .
<code>output</code>	Name of output file, default is to remove extension <code>‘.nw’</code> , <code>‘.Rnw’</code> or <code>‘.Snw’</code> and to add extension <code>‘.tex’</code> . Any directory names in <code>file</code> are also removed such that the output is created in the current working directory.
<code>quiet</code>	If <code>TRUE</code> all progress messages are suppressed.
<code>debug</code>	If <code>TRUE</code> , input and output of all code chunks is copied to the console.
<code>stylepath</code>	If <code>TRUE</code> , a hard path to the file <code>‘Sweave.sty’</code> installed with this package is set, if <code>FALSE</code> , only <code>\usepackage{Sweave}</code> is written. The hard path makes the TeX file less portable, but avoids the problem of installing the current version of <code>‘Sweave.sty’</code> to some place in your TeX input path. The argument is ignored if a <code>\usepackage{Sweave}</code> is already present in the Sweave source file.
<code>echo</code>	set default for option <code>echo</code> , see details below.
<code>eval</code>	set default for option <code>eval</code> , see details below.
<code>keep.source</code>	set default for option <code>‘keep.source’</code> , see details below.
<code>split</code>	set default for option <code>split</code> , see details below.
<code>pdf</code>	set default for option <code>pdf</code> , see details below.
<code>eps</code>	set default for option <code>eps</code> , see details below.
<code>use.cache</code>	a logical indicating whether or not to use the caching system. If <code>FALSE</code> , no caching is performed and all code chunks are recomputed.

## Supported Options

weaver supports the following options for code chunks (the values in parentheses show the default values):

**echo:** logical (`TRUE`). Include S code in the output file?

**eval:** logical (`TRUE`). If `FALSE`, the code chunk is not evaluated, and hence no text or graphical output produced.

**keep.source:** logical (FALSE). When echoing, if `keep.source == TRUE` the original source is copied to the file. Otherwise, deparsed source is echoed.

**results:** character string (verbatim). If `verbatim`, the output of S commands is included in the verbatim-like Soutput environment. If `tex`, the output is taken to be already proper latex markup and included as is. If `hide` then all output is completely suppressed (but the code executed during the weave).

**print:** logical (FALSE). If TRUE, each expression in the code chunk is wrapped into a `print()` statement before evaluation, such that the values of all expressions become visible.

**term:** logical (TRUE). If TRUE, visibility of values emulates an interactive R session: values of assignments are not printed, values of single objects are printed. If FALSE, output comes only from explicit `print` or `cat` statements.

**split:** logical (FALSE). If TRUE, text output is written to separate files for each code chunk.

**strip.white:** character string (false). If true, blank lines at the beginning and end of output are removed. If `all`, then all blank lines are removed from the output.

**prefix:** logical (TRUE). If TRUE generated filenames of figures and output have a common prefix.

**prefix.string:** a character string, default is the name of the `' .Snw'` source file.

**include:** logical (TRUE), indicating whether input statements for text output and `includegraphics` statements for figures should be auto-generated. Use `include = FALSE` if the output should appear in a different place than the code chunk (by placing the input line manually).

**fig:** logical (FALSE), indicating whether the code chunk produces graphical output. Note that only one figure per code chunk can be processed this way.

**eps:** logical (TRUE), indicating whether EPS figures shall be generated. Ignored if `fig = FALSE`.

**pdf:** logical (TRUE), indicating whether PDF figures shall be generated. Ignored if `fig = FALSE`.

**width:** numeric (6), width of figures in inch.

**height:** numeric (6), height of figures in inch.

**cache** If TRUE and `weave` was called with `use.cache=TRUE`, then the calculations in the code chunk will be cached. Subsequent invocations will restore from the cache if the code chunk has not changed. Side effects are not captured. Do not use for chunks defining S4 classes or methods.

### Author(s)

Seth Falcon (borrowing heavily from code by Friedrich Leisch)

### References

Friedrich Leisch: Sweave User Manual, 2002  
<http://www.ci.tuwien.ac.at/~leisch/Sweave>

### See Also

[Sweave](#), [Rtangle](#)

### Examples

```
## Not run:
  Sweave("yourfile.Rnw", driver=weaver())

## End(Not run)
```

# Index

## \* **utilities**

internal\_use\_only, [1](#)  
weaver, [2](#)

CACHE\_DIR (internal\_use\_only), [1](#)  
cache\_expr (internal\_use\_only), [1](#)  
CACHE\_EXT (internal\_use\_only), [1](#)  
cat, [3](#)

deps\_changed (internal\_use\_only), [1](#)

eval\_and\_cache (internal\_use\_only), [1](#)  
expr\_printer (internal\_use\_only), [1](#)

findDeps (internal\_use\_only), [1](#)

get\_cache\_dir (internal\_use\_only), [1](#)  
get\_chunk\_id (internal\_use\_only), [1](#)  
get\_expr\_hash (internal\_use\_only), [1](#)

internal\_use\_only, [1](#)

load\_from\_cache\_env  
(internal\_use\_only), [1](#)  
log\_debug (internal\_use\_only), [1](#)

make\_cache\_expr (internal\_use\_only), [1](#)

print, [3](#)

resetStorage (internal\_use\_only), [1](#)  
Rtangle, [3](#)

Sweave, [2](#), [3](#)

weaver, [2](#)  
weaver\_opts (internal\_use\_only), [1](#)  
weaverEvalWithOpt (internal\_use\_only), [1](#)  
weaverLatexFinish (internal\_use\_only), [1](#)  
weaverLatexSetup (weaver), [2](#)  
weaverRemoveOrphans  
(internal\_use\_only), [1](#)  
weaverRuncode (internal\_use\_only), [1](#)