

saveenv — Save environment content verbatim *

user202729

Released 2022/07/08

Abstract

Allow users to collect environment content verbatim.

1 Motivation

This package provides tools to create your own verbatim environments, and works in all values of `\endlinechar`.

2 Comparison to existing similar packages

- `scontents` package is similar; however it does not allow programmer to programmatically retrieve the macro content, only do a limited number of actions with them (execute, typeset as verbatim code, etc.).
- `fancyvrb` and `verbatimcopy` provides some internal macro for defining similar environments, however they're internal.
- `filecontentsdef` and `newverbs` provides very similar facilities (`filecontentsdefmacro` environment and `\Collectverbenv` command respectively); however it requires `\endlinechar=13` and does not support `\ExplSyntaxOn` environment.
- `verbatim` provides facilities to define custom verbatim environment that processes the content line-by-line, however the interface is more complicated.

For comparison, the following code

```
1 \ExplSyntaxOn
2 \makeatletter
3 \cs_generate_variant:Nn \tl_build_gput_right:Nn {No}
4 \NewDocumentEnvironment{}{}{
5   \@bsphack
6
7   \endlinechar='\^^M ~ \let\do\@makeother \dospecials
8   \@makeother \^^I
9   %\cctab_select:N \c_other_cctab
10  % the above cannot be used, as \verbatim@ relies on
11  % the environment name having letter catcode
```

*This file describes version v0.0.0, last revised 2022/07/08.

```

12      \catcode '\^M \active
13
14      \tl_build_gbegin:N \__all_data
15      \def\verbatim@processline{
16          \tl_build_gput_right:No \__all_data{\the\verbatim@line ^^J}
17      }
18      \verbatim@
19  }{
20      \tl_build_gend:N \__all_data
21      \str_gset:NV \__all_data \__all_data
22
23      % can do something else with \__all_data here
24
25      \@esphack
26  }
27  \makeatother
28  \ExplSyntaxOff

```

defines an environment that saves the data similar to `saveenv` environment described below (with the overhead of `\tl_build_*` functions), but inside `\ExplSyntaxOn` environment it generates a spurious space at the beginning of the string.

3 Main environment

`saveenv` Environment that saves its body.
For example, the following code

```

1 \begin{saveenv}{\data}
2 123
3 456
4 \end{saveenv}

```

will save the string `123`*(newline)*`456` globally into `\data`.

The braces around `{\data}` is optional; however, in the unlikely case if `\endlinechar` has the “letter” catcode, it might be absorbed and gives unexpected result.

A newline is represented as a token with charcode 10 (`^^J`) and catcode other.

Note that this is unusual, for comparison `filecontentsdef` stores it as an active `^^M`.

In the current implementation, it can be used in functions such as `\iow_now:Nn` and newline characters will appear as `newline`. (the number 10 is hard coded in the implementation of `\iow_now:Nn`, nevertheless I’m not sure if this behavior is guaranteed. Manually replacing them with `\iow_newline:` tokens and x-expand the result would be guaranteed to work)

The assignment is global, and done before the macro `\endsaveenv` is executed.

The data is represented as an `expl3`-string, that is, a sequence of tokens with catcode 12 (for non-space characters) or 10 (for space character).

In other words, the token list is equal to its own `\detokenize`.

`saveenvghost` Similar to above; however the content inside is still typesetted/executed, and the

SyncTeX information is preserved.

Note that this environment is implemented by reading the whole file from the beginning, therefore there might be some performance hit for large files and multiple usages of the environment. Use sparingly.

`saveenvkeeplast`

Similar to above; however the final newline and the space characters before `\end{environment name}` are preserved.

For example, the example above with `saveenv` replaced with `saveenvkeeplast` will save the string `123<newline>456<newline>` into `\data` instead.

4 Extending the environments

All of the environments are extensible.

Follow the instruction in <https://tex.stackexchange.com/q/14683/250119> to define an environment based on an existing one.

For example, the following definition

```
1 \ExplSyntaxOn
2 \NewDocumentEnvironment{custom}{}{
3   \saveenv \data
4 }{
5   \endsaveenv
6   \tl_show:N \data
7 }
```

will define an environment `custom` that prints out the content inside the environment using `\tl_show:N`.

5 Limitation

- This package does not process the content line-by-line. Therefore:
 - The memory consumption might be larger than approaches that process the content line-by-line, in case you only need to do something with the line. Nevertheless, in modern computers, the overhead is negligible.
 - In case it's desired to typeset the content afterwards, it's difficult to preserve the SyncTeX data.
- Currently nested environments with the same name are not supported (unlike `scontents` package).
- Note that there must be nothing after the `\begin{environment name}` or the `\end{environment name}` line.
- Note that `\end{environment name}` must not be in the middle of any line.
- The `\endlinechar` must not be tokenized in advance (or if it is, its catcode must be 12/other). This might happen when for example your environment looks ahead for optional argument. See <https://tex.stackexchange.com/q/649331/250119> for an example and one way how to fix it.

Alternatively, you can

- manually remove the end line token,
- set `\endlinechar=-1\relax` before calling `\saveenv`
- (optional) reset the value of `\endlinechar` afterwards.

Index

The italic numbers denote the pages where the corresponding entry is described, numbers underlined point to the definition, all others indicate the places where it is used.

B		<code>\ExplSyntaxOn</code> <i>1, 2</i>
<code>\begin</code>	<i>3</i>	
D		I
<code>\data</code>	<i>2, 3</i>	iow commands:
<code>\detokenize</code>	<i>2</i>	<code>\iow_now:Nn</code> <i>2</i>
E		S
<code>\end</code>	<i>3</i>	<code>saveenv</code> (environment) <i>2</i>
<code>\endlinechar</code>	<i>1–4</i>	<code>\saveenv</code> <i>4</i>
<code>\endsaveenv</code>	<i>2</i>	<code>saveenvghost</code> (environment) <i>2</i>
environments:		<code>saveenvkeeplast</code> (environment) <i>3</i>
<code>saveenv</code>	<i>2</i>	T
<code>saveenvghost</code>	<i>2</i>	TeX and L ^A T _E X 2 _ε commands:
<code>saveenvkeeplast</code>	<i>3</i>	<code>\Collectverbenv</code> <i>1</i>