

The `l3sort` package

Sorting lists*

The L^AT_EX3 Project[†]

Released 2015/10/14

1 `l3sort` documentation

L^AT_EX3 comes with a facility to sort list variables (sequences, token lists, or comma-lists) according to some user-defined comparison. For instance,

```
\clist_set:Nn \l_foo_clist { 3 , 01 , -2 , 5 , +1 }
\clist_sort:Nn \l_foo_clist
{
  \int_compare:nNnTF { #1 } > { #2 }
  { \sort_reversed: }
  { \sort_ordered: }
}
```

will result in `\l_foo_clist` holding the values `{ -2 , 01 , +1 , 3 , 5 }` sorted in non-decreasing order.

The code defining the comparison should perform `\sort_reversed:` if the two items given as `#1` and `#2` are not in the correct order, and otherwise it should call `\sort_ordered:` to indicate that the order of this pair of items should not be changed.

For instance, a *comparison code* consisting only of `\sort_ordered:` with no test will yield a trivial sort: the final order is identical to the original order. Conversely, using a *comparison code* consisting only of `\sort_reversed:` will reverse the list (in a fairly inefficient way).

T_EXhackers note: Internally, the code from `l3sort` stores items in `\toks`. Thus, the *comparison code* should not alter the contents of any `\toks`, nor assume that they hold a given value.

<code>\seq_sort:Nn</code>	<code>\seq_sort:Nn <sequence> {<comparison code>}</code>
---------------------------	--

<code>\seq_gsort:Nn</code>	Sorts the items in the <i><sequence></i> according to the <i><comparison code></i> , and assigns the result to <i><sequence></i> .
----------------------------	--

*This file describes v6210, last revised 2015/10/14.

[†]E-mail: latex-team@latex-project.org

<hr/> <code>\tl_sort:Nn</code> <hr/>	<code>\tl_sort:Nn <tl var> {<comparison code>}</code>
<code>\tl_gsort:Nn</code>	Sorts the items in the <code><tl var></code> according to the <code><comparison code></code> , and assigns the result to <code><tl var></code> .
<hr/> <code>\clist_sort:Nn</code> <hr/>	<code>\clist_sort:Nn <clist var> {<comparison code>}</code>
<code>\clist_gsort:Nn</code>	Sorts the items in the <code><clist var></code> according to the <code><comparison code></code> , and assigns the result to <code><clist var></code> .
<hr/> <code>\tl_sort:nN</code> ★ <hr/>	<code>\tl_sort:nN {<token list>} <conditional></code>
	Sorts the items in the <code><token list></code> , using the <code><conditional></code> to compare items, and leaves the result in the input stream. The <code><conditional></code> should have signature <code>:nnTF</code> , and return true if the two items being compared should be left in the same order, and false if the items should be swapped.
<p>T_EXhackers note: The result is returned within <code>\exp_not:n</code>, which means that the token list will not expand further when appearing in an <code>x</code>-type argument expansion.</p>	

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.

	C		<code>\seq_sort:Nn</code> <i>1, 1</i>
clist commands:		sort commands:	
<code>\clist_gsort:Nn</code> <i>2</i>		<code>\sort_ordered:</code> <i>1, 1</i>	
<code>\clist_sort:Nn</code> <i>2, 2</i>		<code>\sort_reversed:</code> <i>1, 1</i>	
	E		
exp commands:			T
<code>\exp_not:n</code> <i>2</i>		T _E X and L ^A T _E X 2 _ε commands:	
	F	<code>\toks</code> <i>1, 1</i>	
foo commands:		tl commands:	
<code>\l_foo_clist</code> <i>1</i>		<code>\tl_gsort:Nn</code> <i>2</i>	
	S	<code>\tl_sort:Nn</code> <i>2, 2</i>	
seq commands:		<code>\tl_sort:nN</code> <i>2, 2</i>	
<code>\seq_gsort:Nn</code> <i>1</i>			