

The nodetree package

Josef Friedrich

josef@friedrich.rocks

github.com/Josef-Friedrich/nodetree

with contributions by Werner Lemberg

v2.3.0 from 2023/09/10

```
before callback post_linebreak_filter
-----
├─GLUE (baselineskip) wd 5.06pt
├─HLIST (line) wd 345pt, dp 0.11pt, ht 6.94pt
│  └─head
│     └─LOCAL_PAR
│        └─HLIST (indent) wd 15pt
│           ├──GLYPH (glyph) 'n', font 15, wd 5.56pt, ht 4.42pt
│           ├──GLYPH (glyph) 'o', font 15, wd 5pt, ht 4.48pt, dp 0.11pt
│           ├──KERN (fontkern) 0.28pt
│           ├──GLYPH (glyph) 'd', font 15, wd 5.56pt, ht 6.94pt, dp 0.11pt
│           │   props {['injections'] = {['leftkern'] = 18350.08}}
│           ├──GLYPH (glyph) 'e', font 15, wd 4.44pt, ht 4.48pt, dp 0.11pt
│           ├──DISC (regular) penalty 50
│           │   └─pre
│           │      ├──GLYPH (glyph) '-', font 15, wd 3.33pt, ht 2.45pt
│           │      ├──GLYPH (glyph) 't', font 15, wd 3.89pt, ht 6.15pt, dp 0.11pt
│           │      ├──GLYPH (glyph) 'r', font 15, wd 3.92pt, ht 4.42pt
│           │      ├──GLYPH (glyph) 'e', font 15, wd 4.44pt, ht 4.48pt, dp 0.11pt
│           │      └─GLYPH (glyph) 'e', font 15, wd 4.44pt, ht 4.48pt, dp 0.11pt
│           ├──PENALTY (linepenalty) 10000
│           ├──GLUE (parfillskip) plus +1fil
│           └─GLUE (rightskip)
```

Contents

1	Abstract	5
2	Usage	5
2.1	As a plain Lua _T _E X package	6
2.1.1	Available macros	6
2.1.2	Available options	6
2.2	As a Lua ^A _T _E X package	7
2.2.1	Available macros	7
2.2.2	Available options	7
2.3	As a Lua module	8
2.4	The package <code>nodetree-embed</code>	10
2.4.1	Available macros	11
2.4.2	Available environment	11
2.4.3	Available options	11
3	Macros	12
3.1	<code>\NodetreeRegisterCallback</code>	12
3.2	<code>\NodetreeUnregisterCallback</code>	12
3.3	<code>\NodetreeSetOption</code>	12
3.4	<code>\NodetreeResetOption</code>	12
3.5	<code>\NodetreeSet</code>	12
3.6	<code>\NodetreeReset</code>	12
3.7	<code>\NodetreeEmbedCmd</code>	12
3.8	<code>\NodetreeEmbedInput</code>	12
4	Environments	13
4.1	<code>NodetreeEmbedEnv</code>	13
5	Options	14
5.1	Option <code>callback</code>	14
5.2	Option <code>channel</code>	14
5.3	Option <code>verbosity</code>	15
5.3.1	Example: <code>verbosity=0</code>	15
5.3.2	Example: <code>verbosity=1</code>	15
5.3.3	Example: <code>verbosity=2</code>	15
5.3.4	Example: <code>verbosity=3</code>	16
5.4	Option <code>color</code>	16
5.5	Option <code>unit</code>	16
5.5.1	Example: <code>unit=pt</code>	16
5.5.2	Example: <code>unit=sp</code>	17
5.5.3	Example: <code>unit=cm</code>	17
5.6	Option <code>decimalplaces</code>	17
5.6.1	Example: <code>decimalplaces=0</code>	17
5.6.2	Example: <code>decimalplaces=2</code>	18
5.6.3	Example: <code>decimalplaces=5</code>	18
5.7	Option <code>theme</code> and <code>thememode</code>	18
5.7.1	Example: <code>theme=bwdark thememode=dark</code>	18
5.7.2	Example: <code>theme=bwlight thememode=light</code>	18

5.7.3	Example: <code>theme=monokaisoda thememode=dark</code>	19
5.7.4	Example: <code>theme=monokaisoda thememode=light</code>	19
5.8	Option <code>font</code>	19
5.8.1	Example: <code>font={Liberation Mono}</code>	19
5.8.2	Example: <code>font={Ubuntu Mono}</code>	19
5.9	Option <code>fontsize</code>	20
5.9.1	Example: <code>\small</code>	20
5.9.2	Example: <code>\tiny</code>	20
5.10	Options <code>firstline</code> and <code>lastline</code>	20
6	Visual tree structure	21
6.1	Two different connections	21
6.2	Unicode characters to show the tree view	21
7	Examples	22
7.1	The node list of the package name	22
7.2	The node list of a mathematical formula	22
7.3	The node list of the word <i>Office</i>	22
8	Node types	23
8.1	Type <code>hlist(0)</code> , subtype <code>line(1)</code>	23
8.2	Type <code>hlist(0)</code> , subtype <code>box(2)</code>	23
8.3	Type <code>hlist(0)</code> , subtype <code>indent(3)</code>	24
8.4	Type <code>vlist(1)</code>	24
8.5	Type <code>rule(2)</code>	25
8.6	Type <code>mark(4)</code>	25
8.7	Type <code>disc(7)</code> , subtype <code>discretionary(0)</code>	26
8.8	Type <code>disc(7)</code> , subtype <code>explicit(1)</code>	26
8.9	Type <code>disc(7)</code> , subtype <code>regular(3)</code>	26
8.10	Type <code>whatsit(8)</code> , subtype <code>pdf_action(22)</code>	27
8.11	Type <code>whatsit(8)</code> , subtype <code>pdf_colorstack(28)</code>	28
8.12	Type <code>dir(10)</code>	28
8.13	Type <code>glue(12)</code> , subtype <code>baselineskip(2)</code>	29
8.14	Type <code>glue(12)</code> , subtype <code>parskip(3)</code>	29
8.15	Type <code>glue(12)</code> , subtype <code>spaceskip(13)</code>	30
8.16	Type <code>glue(12)</code> , subtype <code>leaders(100)</code>	31
8.17	Type <code>glue(12)</code> , subtype <code>cleaders(101)</code>	31
8.18	Type <code>glue(12)</code> , subtype <code>xleaders(102)</code>	31
8.19	Type <code>glue(12)</code> , subtype <code>gleaders(102)</code>	32
8.20	Type <code>kern(13)</code> , subtype <code>userkern(0)</code>	32
8.21	Type <code>kern(13)</code> , subtype <code>fontkern(1)</code>	33
8.22	Type <code>kern(13)</code> , subtype <code>accentkern(2)</code>	33
8.23	Type <code>kern(13)</code> , subtype <code>italiccorrection(3)</code>	33
8.24	Type <code>penalty(14)</code>	34
8.25	Type <code>glyph(29)</code>	34
8.26	Type <code>attributelist(40)</code>	34

9	Implementation	36
9.1	The file <code>nodetree.tex</code>	36
9.2	The file <code>nodetree.sty</code>	37
9.3	The file <code>nodetree-embed.sty</code>	37
9.4	The file <code>nodetree.lua</code>	41

1 Abstract

`nodetree` is a LuaTeX development package for both plain TeX and LaTeX that visualizes the structure of node lists while compiling with the TeX engine. It uses a visual representation of node lists similar to the UNIX `tree` command's output for folder trees. The processed document isn't changed.

The tree view can be emitted to the console, to a log file, or as a LaTeX input file. Its appearance is highly customizable; multiple color and B/W themes together with various levels of verbosity are provided.

Node lists are the main building blocks of the TeX engine, which LuaTeX allows to inspect and modify. `nodetree` is inspired by a [gist from Patrick Gundlach](#).

2 Usage

The package `nodetree` has four usage scenarios. It can be used as a standalone Lua module, as a plain LuaTeX, a LuaLaTeX package or as package to embed `nodetree` views in a LuaLaTeX document.

2.1 As a plain LuaTeX package

Run `luatex luatex-test.tex` for example to list the nodes using LuaTeX.

```
\input{nodetree.tex}
\NodetreeRegisterCallback{postline}

Lorem ipsum dolor.
\bye
```

2.1.1 Available macros

Macro name	Reference
<code>\NodetreeRegisterCallback{<callbacks>}</code>	Page 12, Section 3.1
<code>\NodetreeUnregisterCallback{<callbacks>}</code>	Page 12, Section 3.2
<code>\NodetreeSetOption[<option>]{<value>}</code>	Page 12, Section 3.3
<code>\NodetreeResetOption{<option>}</code>	Page 12, Section 3.4
<code>\NodetreeReset</code>	Page 12, Section 3.6

2.1.2 Available options

Option name	Reference
<code>callback</code>	Page 14, Section 5.1
<code>verbosity</code>	Page 15, Section 5.3
<code>color</code>	Page 16, Section 5.4
<code>unit</code>	Page 16, Section 5.5
<code>decimalplaces</code>	Page 17, Section 5.6
<code>channel</code>	Page 14, Section 5.2

2.2 As a Lua^AT_EX package

Run `luaAtex luaAtex-test.tex` to show a node tree using Lua^AT_EX. In Lua^AT_EX you can omit a call to `\NodetreeRegisterCallback{postline}`, since `\usepackage{nodetree}` registers the `post_linebreak_filter` by default. Use `\NodetreeUnregisterCallback{postline}` if you don't want to debug the `post_linebreak_filter`.

```
\documentclass{article}
\usepackage{nodetree}

\begin{document}
Lorem ipsum dolor.
\end{document}
```

2.2.1 Available macros

Macro name	Reference
<code>\NodetreeRegisterCallback{<callbacks>}</code>	Page 12, Section 3.1
<code>\NodetreeUnregisterCallback{<callbacks>}</code>	Page 12, Section 3.2
<code>\NodetreeSetOption[<option>]{<value>}</code>	Page 12, Section 3.3
<code>\NodetreeResetOption{<option>}</code>	Page 12, Section 3.4
<code>\NodetreeReset</code>	Page 12, Section 3.6
<code>\NodetreeSet{<kv-options>}</code>	Page 12, Section 3.5

2.2.2 Available options

Option name	Reference
<code>callback</code>	Page 14, Section 5.1
<code>verbosity</code>	Page 15, Section 5.3
<code>color</code>	Page 16, Section 5.4
<code>unit</code>	Page 16, Section 5.5
<code>decimalplaces</code>	Page 17, Section 5.6
<code>channel</code>	Page 14, Section 5.2

2.3 As a Lua module

Import the Lua module of the package inside `\directlua{}` with this command: `local nodetree = require('nodetree')`. Then use the Lua function `nodetree.print(head, options)` to debug nodes inside your Lua code.

```
local nodetree = require('nodetree')

local rule1 = node.new('rule')
rule1.width = 20 * 65536
rule1.height = 10 * 65536
rule1.depth = 10 * 65536
nodetree.print(vbox)
```

The function `nodetree.print()` takes as a second argument a Lua table to configure the output.

```
nodetree.print(vbox, { verbosity = 2, unit = 'cm' })
```

These are the default options:

```
options = {
  callback = 'post_linebreak_filter',
  channel = 'term',
  color = 'colored',
  decimalplaces = 2,
  unit = 'pt',
  verbosity = 0,
  firstline = 1,
  lastline = -1,
}
```

Options `firstline` and `lastline` only have an effect on function `nodetree.input(filename)`, which is used to implement `\NodetreeEmbedInput` (→ 3.8).

The following code snippet demonstrates the usage in Lua_{TeX}. `head` is the current node.

```
\directlua{
  local nodetree = require('nodetree')
  local test = function (head)
    nodetree.print(head)
  end
  callback.register('post_linebreak_filter', test)
}

Lorem ipsum dolor.
\bye
```

This example illustrates how the function has to be applied in Lua_{TeX}.

```
\documentclass{article}
\usepackage{nodetree}

\begin{document}

\directlua{
  local nodetree = require('nodetree')
  local test = function (head)
    nodetree.print(head)
  end
}
```



```
end
luatexbase.add_to_callback('post_linebreak_filter', test, 'test')
}

Lorem ipsum dolor.
\end{document}
```

2.4 The package `nodetree-embed`

The single purpose of this auxiliary package is to provide a view similar to a terminal (console) output. This view mimics the output of `nodetree` in a terminal. The view can be embedded in a Lua \LaTeX file. You have to compile documents using this embedded view with the option `--shell-escape`. The main environment of this package is `NodetreeEmbed`. Markup inside this environment is written into a temporary \LaTeX file. This file is compiled in the background by `latexmk` and the `nodetree` output is embedded into this view. The following list shows each intermediate step:

1. `jobname.tex`

```
\begin{NodetreeEmbedEnv}
nodetree
\end{NodetreeEmbedEnv}
```

2. `_nodetree-jobname/1.tex`

```
%!TEX program = lualatex
\documentclass{article}
\usepackage{nodetree}
\NodetreeSetOption[channel]{tex}
\NodetreeSetOption[verbosity]{0}
\NodetreeSetOption[unit]{pt}
\NodetreeSetOption[decimalplaces]{2}
\NodetreeUnregisterCallback{post_linebreak_filter}
\NodetreeRegisterCallback{post_linebreak_filter}
\begin{document}
nodetree
\end{document}
```

3. `_nodetree-jobname/1.nttex`: This temporary Lua \LaTeX file is compiled using `latexmk` and embedded in the environment `NodetreeEmbed` (the trailing `\` character indicates line continuation).

```
Callback: \textcolor{NTEred}{post\_linebreak\_filter}\par
-----\par
\mbox{|\textcolor{NTEmagentabright}{GLUE}\hspace{0.5em}(baselineskip)\
\textcolor{NTEyellow}{wd} 5.06\textcolor{NTEwhite}{pt}}\par
...
```

4. Finally the result:

```
before callback post_linebreak_filter
-----
--GLUE (baselineskip) wd 5.06pt
--HLIST (line) wd 345pt, dp 0.11pt, ht 6.94pt
  |--head
    |--LOCAL_PAR
      |--HLIST (indent) wd 15pt
        |--GLYPH (glyph) 'n', font 15, wd 5.56pt, ht 4.42pt
        |--GLYPH (glyph) 'o', font 15, wd 5pt, ht 4.48pt, dp 0.11pt
        |--KERN (fontkern) 0.28pt
        |--GLYPH (glyph) 'd', font 15, wd 5.56pt, ht 6.94pt, dp 0.11pt
          props {['injections'] = {['leftkern'] = 18350.08}}
        |--GLYPH (glyph) 'e', font 15, wd 4.44pt, ht 4.48pt, dp 0.11pt
        |--DISC (regular) penalty 50
          |--pre
```

```

└─GLYPH (glyph) '-', font 15, wd 3.33pt, ht 2.45pt
└─GLYPH (glyph) 't', font 15, wd 3.89pt, ht 6.15pt, dp 0.11pt
└─GLYPH (glyph) 'r', font 15, wd 3.92pt, ht 4.42pt
└─GLYPH (glyph) 'e', font 15, wd 4.44pt, ht 4.48pt, dp 0.11pt
└─GLYPH (glyph) 'e', font 15, wd 4.44pt, ht 4.48pt, dp 0.11pt
└─PENALTY (linepenalty) 10000
└─GLUE (parfillskip) plus +1fil
└─GLUE (rightskip)
-----

```

2.4.1 Available macros

Macro name	Reference
<code>\NodetreeRegisterCallback{<callbacks>}</code>	Page 12, Section 3.1
<code>\NodetreeUnregisterCallback{<callbacks>}</code>	Page 12, Section 3.2
<code>\NodetreeSetOption[<option>]{<value>}</code>	Page 12, Section 3.3
<code>\NodetreeResetOption{<option>}</code>	Page 12, Section 3.4
<code>\NodetreeReset</code>	Page 12, Section 3.6
<code>\NodetreeSet{<kv-options>}</code>	Page 12, Section 3.5
<code>\NodetreeEmbedCmd[<kv-options>]{<tex-markup>}</code>	Page 12, Section 3.7
<code>\NodetreeEmbedInput[<kv-options>]{<nttex-file>}</code>	Page 12, Section 3.8

2.4.2 Available environment

Environment name	Reference
<code>\begin{NodetreeEmbedEnv}[<kv-options>]</code>	Page 13, Section 4.1

2.4.3 Available options

Option name	Reference
<code>callback</code>	Page 14, Section 5.1
<code>verbosity</code>	Page 15, Section 5.3
<code>color</code>	Page 16, Section 5.4
<code>unit</code>	Page 16, Section 5.5
<code>decimalplaces</code>	Page 17, Section 5.6
<code>theme</code>	Page 18, Section 5.7
<code>thememode</code>	Page 18, Section 5.7
<code>font</code>	Page 19, Section 5.8
<code>fontsize</code>	Page 20, Section 5.9
<code>firstline</code>	Page 20, Section 5.10
<code>lastline</code>	Page 20, Section 5.10

3 Macros

3.1 `\NodetreeRegisterCallback`

`\NodetreeRegisterCallback` `\NodetreeRegisterCallback{<callbacks>}`: Globally register `{<callbacks>}`, which is a comma-separated list of callback aliases (→ 5.1).

3.2 `\NodetreeUnregisterCallback`

`\NodetreeUnregisterCallback` `\NodetreeUnregisterCallback{<callbacks>}`: Globally unregister `{<callbacks>}`, which is a separated list of callback aliases (→ 5.1).

3.3 `\NodetreeSetOption`

`\NodetreeSetOption` `\NodetreeSetOption[<option>]{<value>}`: Globally set a single `[<option>]` to `{<value>}` (→ 5).

3.4 `\NodetreeResetOption`

`\NodetreeResetOption` `\NodetreeResetOption{<option>}`: Globally reset a single `{<option>}` to its default value (→ 5).

3.5 `\NodetreeSet`

`\NodetreeSet` `\NodetreeSet{<kv-options>}`: Globally set multiple options at once. It can only be used along with Lua^AT_EX. `{<kv-options>}` are key-value pairs.

`\NodetreeSet{color=no, callbacks={hpack, vpack}, verbosity=2}`

3.6 `\NodetreeReset`

`\NodetreeReset` `\NodetreeReset`: Globally reset multiple options to their default values.

3.7 `\NodetreeEmbedCmd`

`\NodetreeEmbedCmd` `\NodetreeEmbedCmd[<kv-options>]{<tex-markup>}`:

Main macro (cmd) to evaluate some T_EX markup and generate a node tree from it. See environment version (→ 3.7). Uses `xparse`'s `+v` option to grab the verbatim content. `{<kv-options>}` are key-value pairs and set locally only.

Only available in package `nodetree-embed`; you need option `--shell-escape`.

3.8 `\NodetreeEmbedInput`

`\NodetreeEmbedInput` `\NodetreeEmbedInput[<kv-options>]{<nttex-file>}`: The path or file name of the `*.nttex` file without the extension. `{<kv-options>}` are key-value pairs and set locally only.

Only available in package `nodetree-embed`. This command works without option `--shell-escape`.

4 Environments

4.1 NodetreeEmbedEnv

```
NodetreeEmbedEnv (env.) \begin{NodetreeEmbedEnv}[\langlekv-options\rangle]  
... TeX markup for evaluation ...  
\end{NodetreeEmbedEnv}
```

Main environment (*env*) to evaluate some \TeX markup and generate a node tree from it. See command version (\rightarrow 3.7). Uses the `\detokenize` command to grab the verbatim content. $\{\langle i>kv-options\rangle\}$ are key-value pairs and set locally only.

Only available in package `nodetree-embed`; you need option `--shell-escape`.

5 Options

5.1 Option callback

The option `callback` is the most important setting of the package. It is possible to specify an alias to select the callback. Take a look at the overview of callbacks (→ Figure 1). `nodetree` supports all node-related callbacks as listed in the LuaTeX reference manual.

These macros process callback options:

```
\NodetreeRegisterCallback{<callbacks>}
\NodetreeUnregisterCallback{<callbacks>}
\NodetreeSet{<callback=<callbacks>}
\usepackage[<callback=<callbacks>]{<nodetree>}
```

The `nodetree` package can watch the node tree before and after the functions of a callback are executed: It is possible to prepend and/or append a colon (`:`) to indicate the desired watchpoint position, which defaults to 'before' if no colon is used.

Use commas to specify multiple callbacks; trailing and leading whitespace is ignored. For example, this call

```
\NodetreeRegisterCallback{:preline, line, :postline:}
```

watches the node tree before the `preline` callback functions, before the `line` callback functions, and before and after the `postline` callback functions. In case there are no callback functions registered for one of the `hyphenate`, `kerning`, `ligaturing`, and `mlist_to_hlist` callbacks, LuaTeX executes some internal code instead. It thus makes sense to watch the node tree before and after these (empty) callbacks even in this case.

Wrap your callback aliases in curly braces for the macro `\NodetreeSet`. Note that no whitespace between `=` and `{` is allowed.

```
\NodetreeSet{callback={:preline, line, :postline:}}
```

The same applies for the macro `\usepackage`:

```
\usepackage{callback={:preline, line, :postline:}}
```

The callbacks in Figure 1 are listed in the same order as in the LuaTeX reference manual. Note that the `ligaturing` and `kerning` callbacks only have an effect on ligatures and kernings, respectively, if the `luaotfload` package (which is the default for LuaL^ATeX, and an optional package for LuaTeX) handles the affected font with `mode=base` (see the [documentation](#) for more details).

5.2 Option channel

You can select the debug output channel with this option. The default value for the option `channel` is `term`, which displays the node tree in the current terminal. Specify `log` and the package creates a log file named `<jobname>.ntlog`. Specify

Callback	Alias	Alias (longer)
contribute_filter	contribute	contributefilter
buildpage_filter	buildfilter	buildpagefilter
build_page_insert	buildinsert	buildpageinsert
pre_linebreak_filter	preline	prelinebreakfilter
linebreak_filter	line	linebreakfilter
append_to_vlist_filter	append	appendtovlistfilter
post_linebreak_filter	postline	postlinebreakfilter
hpack_filter	hpack	hpackfilter
vpack_filter	vpack	vpackfilter
hpack_quality	hpackq	hpackquality
vpack_quality	vpackq	vpackquality
process_rule	process	processrule
pre_output_filter	preout	preoutputfilter
hyphenate	hyph	
ligaturing	liga	
kerning	kern	
insert_local_par	insert	insertlocalpar
mlist_to_hlist	mhlist	mlisttohlist

Figure 1: The callback aliases

tex and a log file named `<jobname>.nttex` is created. `nt...` stands for `nodetree`. `<jobname>` is the basename of your file you want to debug. The debug channel is only useful for the auxiliary package `nodetree-embed`. Paste the markup in the environment `NodetreeEmbedView` and you get a terminal-like view in your document.

5.3 Option verbosity

Higher integer values result in a more verbose output. The default value for this option is 0. At the moment verbosity levels 0 to 3 are implemented.

5.3.1 Example: verbosity=0

```
before callback pre_linebreak_filter
-----
--LOCAL_PAR
--HLIST (indent) wd 15pt
--GLYPH (glyph) '.', font 15, wd 2.78pt, ht 1.06pt
--PENALTY (linepenalty) 10000
--GLUE (parfillskip) plus +ifil
-----
```

5.3.2 Example: verbosity=1

```
before callback pre_linebreak_filter
-----
--LOCAL_PAR
--HLIST subtype: indent, width: 15pt
--GLYPH subtype: glyph, char: 46 ('.'), font: 15, width: 2.78pt, height: 1.06pt
--PENALTY subtype: linepenalty, penalty: 10000
--GLUE subtype: parfillskip, stretch: +ifil
-----
```

5.3.3 Example: verbosity=2

Unit	Description
pt	Point 1/72.27 inch. The conversion to metric units, to two decimal places, is 1 point = 2.85 mm = 28.45 cm.
pc	Pica, 12 pt
in	Inch, 72.27 pt
bp	Big point, 1/72 inch. This length is the definition of a point in PostScript and many desktop publishing systems.
cm	Centimeter
mm	Millimeter
dd	Didot point, 1.07 pt
cc	Cicero, 12 dd
sp	Scaled point, 1/65536 pt

Figure 2: Fixed units

Unit	Description
ex	x-height of the current font
em	Width of the capital letter M

Figure 3: Relative units

```
before callback pre_linebreak_filter
-----
|LOCAL_PAR[9] no: 464
|HLIST[0] no: 417, subtype: indent[3], width: 15pt
|GLYPH[29] no: 398, subtype: glyph[0], char: 46 ('.'), font: 15, left: 2, width: 2.78pt, height: 1.06pt
|PENALTY[14] no: 198, subtype: linepenalty[2], penalty: 10000
|GLUE[12] no: 470, subtype: parfillskip[15], stretch: +1fil
-----
```

5.3.4 Example: verbosity=3

```
before callback pre_linebreak_filter
-----
|LOCAL_PAR[9] no: 464
|HLIST[0] no: 417, subtype: indent[3], width: 15pt, prev: 464
|GLYPH[29] no: 398, subtype: glyph[0], char: 46 (0x2e, '.'), font: 15, left: 2, width: 2.78pt, height: 1.06pt, prev: 417
|PENALTY[14] no: 198, subtype: linepenalty[2], penalty: 10000, prev: 398
|GLUE[12] no: 470, subtype: parfillskip[15], stretch: +1fil, prev: 198
-----
```

5.4 Option color

The default option for `color` is `colored`. Use any other string (for example `none` or `no`) to disable the colored terminal output of the package.

```
\usepackage[color=no]{nodetree}
```

5.5 Option unit

The option `unit` sets the length unit to display all length values of the nodes. The default option for `unit` is `pt`. See figures 2 and 3 for possible values.

5.5.1 Example: unit=pt

```
before callback pre_linebreak_filter
-----
|LOCAL_PAR
```



```

-HLIST (indent) wd 15pt
-GLYPH (glyph) 'L', font 15, wd 6.25pt, ht 6.83pt
-GLYPH (glyph) 'o', font 15, wd 5pt, ht 4.48pt, dp 0.11pt
-GLYPH (glyph) 'r', font 15, wd 3.92pt, ht 4.42pt
-GLYPH (glyph) 'e', font 15, wd 4.44pt, ht 4.48pt, dp 0.11pt
-GLYPH (glyph) 'm', font 15, wd 8.33pt, ht 4.42pt
-GLYPH (glyph) '.', font 15, wd 2.78pt, ht 1.06pt
-PENALTY (linepenalty) 10000
-GLUE (parfillskip) plus +1fil
-----

```

5.5.2 Example: unit=sp

```

before callback pre_linebreak_filter
-----
-LOCAL_PAR
-HLIST (indent) wd 983040sp
-GLYPH (glyph) 'L', font 15, wd 409600sp, ht 447611sp
-GLYPH (glyph) 'o', font 15, wd 327680sp, ht 293601sp, dp 7209sp
-GLYPH (glyph) 'r', font 15, wd 256901sp, ht 289669sp
-GLYPH (glyph) 'e', font 15, wd 290980sp, ht 293601sp, dp 7209sp
-GLYPH (glyph) 'm', font 15, wd 545915sp, ht 289669sp
-GLYPH (glyph) '.', font 15, wd 182190sp, ht 69468sp
-PENALTY (linepenalty) 10000
-GLUE (parfillskip) plus +1fil
-----

```

5.5.3 Example: unit=cm

```

before callback pre_linebreak_filter
-----
-LOCAL_PAR
-HLIST (indent) wd 0.53cm
-GLYPH (glyph) 'L', font 15, wd 0.22cm, ht 0.24cm
-GLYPH (glyph) 'o', font 15, wd 0.18cm, ht 0.16cm, dp 0cm
-GLYPH (glyph) 'r', font 15, wd 0.14cm, ht 0.16cm
-GLYPH (glyph) 'e', font 15, wd 0.16cm, ht 0.16cm, dp 0cm
-GLYPH (glyph) 'm', font 15, wd 0.29cm, ht 0.16cm
-GLYPH (glyph) '.', font 15, wd 0.1cm, ht 0.04cm
-PENALTY (linepenalty) 10000
-GLUE (parfillskip) plus +1fil
-----

```

5.6 Option decimalplaces

The options `decimalplaces` sets the number of decimal places for some node fields. If `decimalplaces` is set to 0 only integer values are shown.

```
\NodetreeSetOption[decimalplaces]{4}
```

5.6.1 Example: decimalplaces=0

```

before callback pre_linebreak_filter
-----
-LOCAL_PAR
-HLIST (indent) wd 1cc
-GLYPH (glyph) 'L', font 15, wd 0cc, ht 1cc
-GLYPH (glyph) 'o', font 15, wd 0cc, ht 0cc, dp 0cc
-GLYPH (glyph) 'r', font 15, wd 0cc, ht 0cc
-GLYPH (glyph) 'e', font 15, wd 0cc, ht 0cc, dp 0cc
-GLYPH (glyph) 'm', font 15, wd 1cc, ht 0cc

```

```

└GLYPH (glyph) '.', font 15, wd 0cc, ht 0cc
└PENALTY (linepenalty) 10000
└GLUE (parfillskip) plus +1fil
-----

```

5.6.2 Example: decimalplaces=2

```

before callback pre_linebreak_filter
-----
└LOCAL_PAR
└HLIST (indent) wd 1.17cc
└GLYPH (glyph) 'L', font 15, wd 0.49cc, ht 0.53cc
└GLYPH (glyph) 'o', font 15, wd 0.39cc, ht 0.35cc, dp 0.01cc
└GLYPH (glyph) 'r', font 15, wd 0.31cc, ht 0.34cc
└GLYPH (glyph) 'e', font 15, wd 0.35cc, ht 0.35cc, dp 0.01cc
└GLYPH (glyph) 'm', font 15, wd 0.65cc, ht 0.34cc
└GLYPH (glyph) '.', font 15, wd 0.22cc, ht 0.08cc
└PENALTY (linepenalty) 10000
└GLUE (parfillskip) plus +1fil
-----

```

5.6.3 Example: decimalplaces=5

```

before callback pre_linebreak_filter
-----
└LOCAL_PAR
└HLIST (indent) wd 1.16821cc
└GLYPH (glyph) 'L', font 15, wd 0.48676cc, ht 0.53193cc
└GLYPH (glyph) 'o', font 15, wd 0.3894cc, ht 0.34891cc, dp 0.00857cc
└GLYPH (glyph) 'r', font 15, wd 0.30529cc, ht 0.34423cc
└GLYPH (glyph) 'e', font 15, wd 0.34579cc, ht 0.34891cc, dp 0.00857cc
└GLYPH (glyph) 'm', font 15, wd 0.64875cc, ht 0.34423cc
└GLYPH (glyph) '.', font 15, wd 0.21651cc, ht 0.08255cc
└PENALTY (linepenalty) 10000
└GLUE (parfillskip) plus +1fil
-----

```

5.7 Option theme and thememode

5.7.1 Example: theme=bwdark thememode=dark

```

before callback pre_linebreak_filter
-----
└LOCAL_PAR
└HLIST (indent) wd 15pt
└GLYPH (glyph) '.', font 15, wd 2.78pt, ht 1.06pt
└PENALTY (linepenalty) 10000
└GLUE (parfillskip) plus +1fil
-----

```

5.7.2 Example: theme=bwlight thememode=light

```

before callback pre_linebreak_filter
-----
└LOCAL_PAR
└HLIST (indent) wd 15
└GLYPH (glyph) '.', font 15, wd 2.78 , ht 1.06
└PENALTY (linepenalty) 10000
└GLUE (parfillskip) plus +1
-----

```

5.7.3 Example: theme=monokaisoda thememode=dark

```
before callback pre_linebreak_filter
-----
|LOCAL_PAR
|HLIST (indent) wd 15pt
|GLYPH (glyph) '.', font 15, wd 2.78pt, ht 1.06pt
|PENALTY (linepenalty) 10000
|GLUE (parfillskip) plus +1fil
-----
```

5.7.4 Example: theme=monokaisoda thememode=light

```
before callback pre_linebreak_filter
-----
|LOCAL_PAR
|HLIST (indent) wd 15pt
|GLYPH (glyph) '.', font 15, wd 2.78pt, ht 1.06pt
|PENALTY (linepenalty) 10000
|GLUE (parfillskip) plus +1fil
-----
```

5.8 Option font

nodetree-embed passes the option `font` down to the command `\setmonofont{}` of the `fontspec` package. The used font should be monospaced and have some box drawing glyphs (see table 4).

5.8.1 Example: font={Liberation Mono}

```
before callback post_linebreak_filter
-----
|GLUE (baselineskip) wd 10.94pt
|HLIST (line) wd 345pt, ht 1.06pt
|head
| |LOCAL_PAR
| |HLIST (indent) wd 15pt
| |GLYPH (glyph) '.', font 15, wd 2.78pt, ht 1.06pt
| |PENALTY (linepenalty) 10000
| |GLUE (parfillskip) plus +1fil
| |GLUE (rightskip)
|-----
```

5.8.2 Example: font={Ubuntu Mono}

```
before callback post_linebreak_filter
-----
|GLUE (baselineskip) wd 10.94pt
|HLIST (line) wd 345pt, ht 1.06pt
|head
| |LOCAL_PAR
| |HLIST (indent) wd 15pt
| |GLYPH (glyph) '.', font 15, wd 2.78pt, ht 1.06pt
| |PENALTY (linepenalty) 10000
| |GLUE (parfillskip) plus +1fil
| |GLUE (rightskip)
|-----
```

5.9 Option fontsize

5.9.1 Example: \small

```
before callback pre_linebreak_filter
-----
-LOCAL_PAR
-HLIST (indent) wd 15pt
-GLYPH (glyph) '.', font 15, wd 2.78pt, ht 1.06pt
-PENALTY (linepenalty) 10000
-GLUE (parfillskip) plus +1fil
-----
```

5.9.2 Example: \tiny

```
before callback pre_linebreak_filter
-----
-LOCAL_PAR
-HLIST (indent) wd 15pt
-GLYPH (glyph) '.', font 15, wd 2.78pt, ht 1.06pt
-PENALTY (linepenalty) 10000
-GLUE (parfillskip) plus +1fil
-----
```

5.10 Options firstline and lastline

These two options are for function `\ModetreeEmbedInput` only (→ 3.8). They specify the first and last shown line of the read `*.nttex` file. Values 1, 2, ..., corresponds to the first line, second, line, etc. Values -1, -2, ..., correspond to the last line, the line before the last line, etc. The default values are `firstline = 1` and `lastline = -1` to display the whole file.

Code	Character	Name
U+2500	–	BOX DRAWINGS LIGHT HORIZONTAL
U+2502		BOX DRAWINGS LIGHT VERTICAL
U+2514	└	BOX DRAWINGS LIGHT UP AND RIGHT
U+251C	├	BOX DRAWINGS LIGHT VERTICAL AND RIGHT
U+2550	=	BOX DRAWINGS DOUBLE HORIZONTAL
U+2551		BOX DRAWINGS DOUBLE VERTICAL
U+255A	└┐	BOX DRAWINGS DOUBLE UP AND RIGHT
U+2560	├┤	BOX DRAWINGS DOUBLE VERTICAL AND RIGHT

Figure 4: The Unicode box drawings glyphs

6 Visual tree structure

6.1 Two different connections

Nodes in LuaTeX are connected. The `nodetree` package distinguishes between *list* and *field* connections.

- *list*: Nodes that are doubly connected by `next` and `previous` fields.
- *field*: Connections to nodes by other fields than `next` and `previous`, for example, using `head` and `pre`.

6.2 Unicode characters to show the tree view

The package `nodetree` uses the unicode box drawing symbols. Your default terminal font should contain this characters to obtain the tree view. Eight box drawing characters are necessary.

For *list* connections *light* characters are shown.



field connections are visualized by *Double* characters.



7 Examples

This section lists some examples of the `nodetree` output.

7.1 The node list of the package name

`nodetree`

```
before callback post_linebreak_filter
-----
├─GLUE (baselineskip) wd 5.06pt
├─HLIST (line) wd 345pt, dp 0.11pt, ht 6.94pt
├─head
├─LOCAL_PAR
├─HLIST (indent) wd 15pt
├─GLYPH (glyph) 'n', font 15, wd 5.56pt, ht 4.42pt
├─GLYPH (glyph) 'o', font 15, wd 5pt, ht 4.48pt, dp 0.11pt
├─KERN (fontkern) 0.28pt
├─GLYPH (glyph) 'd', font 15, wd 5.56pt, ht 6.94pt, dp 0.11pt
├─props {'injections'} = {'leftkern'} = 18350.08}}
├─GLYPH (glyph) 'e', font 15, wd 4.44pt, ht 4.48pt, dp 0.11pt
├─DISC (regular) penalty 50
├─pre
├─GLYPH (glyph) '-', font 15, wd 3.33pt, ht 2.45pt
├─GLYPH (glyph) 't', font 15, wd 3.89pt, ht 6.15pt, dp 0.11pt
├─GLYPH (glyph) 'r', font 15, wd 3.92pt, ht 4.42pt
├─GLYPH (glyph) 'e', font 15, wd 4.44pt, ht 4.48pt, dp 0.11pt
├─GLYPH (glyph) 'e', font 15, wd 4.44pt, ht 4.48pt, dp 0.11pt
├─PENALTY (linepenalty) 10000
├─GLUE (parfillskip) plus +1fil
├─GLUE (rightskip)
-----
```

7.2 The node list of a mathematical formula

$\$1+2\$$

```
before callback post_linebreak_filter
-----
├─GLUE (baselineskip) wd 5.56pt
├─HLIST (line) wd 345pt, dp 0.83pt, ht 6.44pt
├─head
├─LOCAL_PAR
├─HLIST (indent) wd 15pt
├─MATH (beginmath)
├─GLYPH (glyph) '1', font 16, wd 5pt, ht 6.44pt
├─GLUE (medmuskip) wd 2.22pt, plus 1.11pt, minus 2.22pt
├─GLYPH (glyph) '+', font 16, wd 7.78pt, ht 5.83pt, dp 0.83pt
├─PENALTY (noadpenalty) 700
├─GLUE (medmuskip) wd 2.22pt, plus 1.11pt, minus 2.22pt
├─GLYPH (glyph) '2', font 16, wd 5pt, ht 6.44pt
├─MATH (endmath)
├─PENALTY (linepenalty) 10000
├─GLUE (parfillskip) plus +1fil
├─GLUE (rightskip)
-----
```

7.3 The node list of the word *Office*

The characters *ffi* are deeply nested in a discretionary node.

Office

```

before callback post_linebreak_filter
-----
├─GLUE (baselineskip) wd 5pt
├─HLIST (line) wd 345pt, dp 0pt, ht 7pt
├─head
│   └─LOCAL_PAR
│       └─HLIST (indent) wd 15pt
│           └─GLYPH (glyph) 'o', font 15, wd 8pt, ht 7pt, dp 0pt
│               └─DISC (regular) penalty 50
│                   └─pre
│                       └─GLYPH (glyph) 'f', font 15, wd 3pt, ht 7pt
│                           └─GLYPH (glyph) '-', font 15, wd 3pt, ht 2pt
│                               └─replace
│                                   └─GLYPH (glyph ligature) 'fi', font 15, wd 8pt, ht 7pt
│                                       └─components
│                                           └─GLYPH (glyph ligature) 'fi', font 15, wd 6pt, ht 7pt
│                                               └─components
│                                                   └─GLYPH (glyph) 'f', font 15, wd 3pt, ht 7pt
│                                                       └─GLYPH (glyph) 'i', font 15, wd 3pt, ht 7pt
│                                                           └─post
│                                                               └─GLYPH (glyph ligature) 'fi', font 15, wd 6pt, ht 7pt
│                                                                   └─components
│                                                                       └─GLYPH (glyph) 'f', font 15, wd 3pt, ht 7pt
│                                                                           └─GLYPH (glyph) 'i', font 15, wd 3pt, ht 7pt
├─GLYPH (glyph) 'c', font 15, wd 4pt, ht 4pt, dp 0pt
├─GLYPH (glyph) 'e', font 15, wd 4pt, ht 4pt, dp 0pt
├─PENALTY (linepenalty) 10000
├─GLUE (parfillskip) plus +1fil
└─GLUE (rightskip)
-----

```

8 Node types

This section shows some node types in a nodetree view.

8.1 Type hlist(0), subtype line(1)

Lorem

```

before callback post_linebreak_filter
-----
├─GLUE (baselineskip) wd 5.17pt
├─HLIST (line) wd 345pt, dp 0.11pt, ht 6.83pt
├─head
│   └─LOCAL_PAR
│       └─HLIST (indent) wd 15pt
│           └─GLYPH (glyph) 'L', font 15, wd 6.25pt, ht 6.83pt
│               └─GLYPH (glyph) 'o', font 15, wd 5pt, ht 4.48pt, dp 0.11pt
│                   └─GLYPH (glyph) 'r', font 15, wd 3.92pt, ht 4.42pt
│                       └─GLYPH (glyph) 'e', font 15, wd 4.44pt, ht 4.48pt, dp 0.11pt
│                           └─GLYPH (glyph) 'm', font 15, wd 8.33pt, ht 4.42pt
├─PENALTY (linepenalty) 10000
├─GLUE (parfillskip) plus +1fil
└─GLUE (rightskip)
-----

```

8.2 Type hlist(0), subtype box(2)

L\hbox to 40pt{ore}m

```

before callback post_linebreak_filter
-----
├─GLUE (baselineskip) wd 5.17pt
├─HLIST (line) wd 345pt, dp 0.11pt, ht 6.83pt
└─head
  └─LOCAL_PAR
    ├─HLIST (indent) wd 15pt
    ├─GLYPH (glyph) 'L', font 15, wd 6.25pt, ht 6.83pt
    └─HLIST (box) wd 40pt, dp 0.11pt, ht 4.48pt
      └─head
        ├─GLYPH (glyph) 'o', font 15, wd 5pt, ht 4.48pt, dp 0.11pt
        ├─GLYPH (glyph) 'r', font 15, wd 3.92pt, ht 4.42pt
        └─GLYPH (glyph) 'e', font 15, wd 4.44pt, ht 4.48pt, dp 0.11pt
      └─GLYPH (glyph) 'm', font 15, wd 8.33pt, ht 4.42pt
    └─PENALTY (linepenalty) 10000
  └─GLUE (parfillskip) plus +1fil
  └─GLUE (rightskip)

```

8.3 Type hlist(0), subtype indent(3)

`\setlength {\parindent }{5cm} I`

```

before callback post_linebreak_filter
-----
├─GLUE (baselineskip) wd 0.18cm
├─HLIST (line) wd 12.13cm, ht 0.24cm
└─head
  └─LOCAL_PAR
    ├─HLIST (indent) wd 5cm
    └─GLYPH (glyph) 'I', font 15, wd 0.13cm, ht 0.24cm
  └─PENALTY (linepenalty) 10000
  └─GLUE (parfillskip) plus +1fil
  └─GLUE (rightskip)

```

8.4 Type vlist(1)

`L\vbox to 40pt{0}L`

```

before callback post_linebreak_filter
- groupcode: vbox
-----
├─HLIST (line) wd 345pt, dp 0.2pt, ht 7.1pt
└─head
  └─LOCAL_PAR
    ├─HLIST (indent) wd 15pt
    ├─GLYPH (glyph) 'O', font 15, wd 7.8pt, ht 7.1pt, dp 0.2pt
    └─PENALTY (linepenalty) 10000
  └─GLUE (parfillskip) plus +1fil
  └─GLUE (rightskip)

```

```

before callback post_linebreak_filter
-----
├─GLUE (lineskip) wd 1pt
├─HLIST (line) wd 345pt, dp 0.2pt, ht 40pt
└─head
  └─LOCAL_PAR
    ├─HLIST (indent) wd 15pt
    ├─GLYPH (glyph) 'L', font 15, wd 6.3pt, ht 6.8pt
    └─VLIST (unknown) wd 345pt, dp 0.2pt, ht 40pt
      └─head
        └─HLIST (line) wd 345pt, dp 0.2pt, ht 7.1pt
          └─head
            └─LOCAL_PAR

```



```

before callback post_linebreak_filter
-----
-GLUE (baselineskip) wd 5pt
-HLIST (line) wd 345pt, dp 0pt, ht 7pt
  -head
    -LOCAL_PAR
      -HLIST (indent) wd 15pt
      -GLYPH (glyph) 'o', font 15, wd 8pt, ht 7pt, dp 0pt
      -DISC (regular) penalty 50
      -pre
        -GLYPH (glyph) 'f', font 15, wd 3pt, ht 7pt
        -GLYPH (glyph) '-', font 15, wd 3pt, ht 2pt
      -post
        -GLYPH (glyph ligature) 'fi', font 15, wd 6pt, ht 7pt
        -components
          -GLYPH (glyph) 'f', font 15, wd 3pt, ht 7pt
          -GLYPH (glyph) 'i', font 15, wd 3pt, ht 7pt
      -replace
        -GLYPH (glyph ligature) '[]', font 15, wd 8pt, ht 7pt
        -components
          -GLYPH (glyph ligature) '[]', font 15, wd 6pt, ht 7pt
          -components
            -GLYPH (glyph) 'f', font 15, wd 3pt, ht 7pt
            -GLYPH (glyph) 'f', font 15, wd 3pt, ht 7pt
            -GLYPH (glyph) 'i', font 15, wd 3pt, ht 7pt
      -GLYPH (glyph) 'c', font 15, wd 4pt, ht 4pt, dp 0pt
      -GLYPH (glyph) 'e', font 15, wd 4pt, ht 4pt, dp 0pt
      -PENALTY (linepenalty) 10000
      -GLUE (parfillskip) plus +1fil
      -GLUE (rightskip)
  -----

```

8.10 Type whatsit(8), subtype pdf_action(22)

```

\usepackage{hyperref}
\begin{document}
\url{https://luatex.org}
\end{document}

```

```

before callback post_linebreak_filter
-----
-GLUE (baselineskip) wd 5.06pt
-HLIST (line) wd 345pt, dp 2.29pt, ht 6.94pt
  -head
    -LOCAL_PAR
      -HLIST (indent) wd 15pt
      -WHATSIT (pdf_start_link) wd -16384pt, dp -16384pt, ht -16384pt, objnum 4, link_attr '/Border[0 0 1]/H/I/C[0 1 1]'
        -action
          -WHATSIT (pdf_action) action_type 3, file '', data '/Subtype/Link/A<</Type/Action/S/URI/URI(https://luatex.org)>>'
      -MATH (beginmath)
        -GLYPH (glyph) 'h', font 16, wd 5.25pt, ht 6.11pt
        -GLYPH (glyph) 't', font 16, wd 5.25pt, ht 5.54pt, dp 0.06pt
        -GLYPH (glyph) 't', font 16, wd 5.25pt, ht 5.54pt, dp 0.06pt
        -GLYPH (glyph) 'p', font 16, wd 5.25pt, ht 4.37pt, dp 2.22pt
        -GLYPH (glyph) 's', font 16, wd 5.25pt, ht 4.4pt, dp 0.06pt
        -GLUE (thickmuskip)
        -GLYPH (glyph) ':', font 16, wd 5.25pt, ht 4.31pt
        -PENALTY (noadpenalty) 500
        -GLUE (thickmuskip)
        -GLYPH (glyph) '/', font 16, wd 5.25pt, ht 6.94pt, dp 0.83pt
        -GLUE (medmuskip)
        -GLYPH (glyph) '/', font 16, wd 5.25pt, ht 6.94pt, dp 0.83pt
        -PENALTY (noadpenalty) 700
        -GLUE (medmuskip)
        -GLYPH (glyph) 'l', font 16, wd 5.25pt, ht 6.11pt
        -GLYPH (glyph) 'u', font 16, wd 5.25pt, ht 4.31pt, dp 0.06pt
        -GLYPH (glyph) 'a', font 16, wd 5.25pt, ht 4.4pt, dp 0.06pt
        -GLYPH (glyph) 't', font 16, wd 5.25pt, ht 5.54pt, dp 0.06pt
        -GLYPH (glyph) 'e', font 16, wd 5.25pt, ht 4.4pt, dp 0.06pt
        -GLYPH (glyph) 'x', font 16, wd 5.25pt, ht 4.31pt
        -GLUE (medmuskip)
        -GLYPH (glyph) ':', font 16, wd 5.25pt, ht 1.25pt
        -PENALTY (noadpenalty) 700
        -GLUE (medmuskip)
        -GLYPH (glyph) 'o', font 16, wd 5.25pt, ht 4.4pt, dp 0.06pt
        -GLYPH (glyph) 'r', font 16, wd 5.25pt, ht 4.37pt
        -GLYPH (glyph) 'g', font 16, wd 5.25pt, ht 4.42pt, dp 2.29pt
      -----

```

```

-MATH (endmath)
-WHATSIT (pdf_end_link)
-PENALTY (linepenalty) 10000
-GLUE (parfillskip) plus +ifil
-GLUE (rightskip)
-----

```

8.11 Type whatsit(8), subtype pdf_colorstack(28)

```

\usepackage{color}
\begin{document}
Lo\textcolor{red}{re}m.
\end{document}

```

```

before callback post_linebreak_filter
-----
-GLUE (baselineskip) wd 5.17pt
-HLIST (line) wd 345pt, dp 0.11pt, ht 6.83pt
  └─head
    └─LOCAL_PAR
      -HLIST (indent) wd 15pt
      -GLYPH (glyph) 'L', font 15, wd 6.25pt, ht 6.83pt
      -GLYPH (glyph) 'o', font 15, wd 5pt, ht 4.48pt, dp 0.11pt
      -WHATSIT (pdf_colorstack) data '1 0 0 rg 1 0 0 RG'
      -GLYPH (glyph) 'r', font 15, wd 3.92pt, ht 4.42pt
      -GLYPH (glyph) 'e', font 15, wd 4.44pt, ht 4.48pt, dp 0.11pt
      -WHATSIT (pdf_colorstack) data ''
      -GLYPH (glyph) 'm', font 15, wd 8.33pt, ht 4.42pt
      -GLYPH (glyph) '.', font 15, wd 2.78pt, ht 1.06pt
      -PENALTY (linepenalty) 10000
      -GLUE (parfillskip) plus +ifil
      -GLUE (rightskip)
    -----

```

8.12 Type dir(10)

```

\textdir TRT nur {\textdir TLT run \textdir TRT NUR} nur

```

```

before callback post_linebreak_filter
-----
-GLUE (baselineskip) wd 5.17pt
-HLIST (line) wd 345pt, dp 0.22pt, ht 6.83pt
  └─head
    └─LOCAL_PAR
      -DIR (+TRT) level 2
      -HLIST (indent) wd 15pt
      -GLYPH (glyph) 'n', font 15, wd 5.56pt, ht 4.42pt
      -GLYPH (glyph) 'u', font 15, wd 5.56pt, ht 4.42pt, dp 0.11pt
      props {'injections'} = {'rightkern'} = -18350.08}
      -GLYPH (glyph) 'r', font 15, wd 3.92pt, ht 4.42pt
      -GLUE (spaceskip) wd 3.33pt, plus 1.66pt, minus 1.11pt
      -DIR (+TLT) level 2
      -GLYPH (glyph) 'r', font 15, wd 3.92pt, ht 4.42pt
      -GLYPH (glyph) 'u', font 15, wd 5.56pt, ht 4.42pt, dp 0.11pt
      -GLYPH (glyph) 'n', font 15, wd 5.56pt, ht 4.42pt
      -GLUE (spaceskip) wd 3.33pt, plus 1.66pt, minus 1.11pt
      -DIR (-TLT) level 2
      -DIR (+TRT) level 2
      -GLYPH (glyph) 'N', font 15, wd 7.5pt, ht 6.83pt
      -GLYPH (glyph) 'U', font 15, wd 7.5pt, ht 6.83pt, dp 0.22pt
      -GLYPH (glyph) 'R', font 15, wd 7.36pt, ht 6.83pt, dp 0.22pt
      -DIR (-TRT) level 1
      -GLUE (spaceskip) wd 3.33pt, plus 1.66pt, minus 1.11pt
      -GLYPH (glyph) 'n', font 15, wd 5.56pt, ht 4.42pt
      -GLYPH (glyph) 'u', font 15, wd 5.56pt, ht 4.42pt, dp 0.11pt
    -----

```

```

      props {'injections' = {'rightkern' = -18350.08}}
      GLYPH (glyph) 'r', font 15, wd 3.92pt, ht 4.42pt
      PENALTY (linepenalty) 10000
      DIR (-TRT) level 1
      GLUE (parfillskip) plus +1fil
      GLUE (rightskip)
-----

```

8.13 Type glue(12), subtype baselineskip(2)

\baselineskip=5cm Lorem Lorem

```

before callback post_linebreak_filter
-----
GLUE (baselineskip) wd 4.76cm
HLIST (line) wd 12.13cm, dp 0cm, ht 0.24cm
  LL-head
    LOCAL_PAR
    HLIST (indent) wd 0.53cm
    GLYPH (glyph) 'L', font 15, wd 0.22cm, ht 0.24cm
    GLYPH (glyph) 'o', font 15, wd 0.18cm, ht 0.16cm, dp 0cm
    GLYPH (glyph) 'r', font 15, wd 0.14cm, ht 0.16cm
    GLYPH (glyph) 'e', font 15, wd 0.16cm, ht 0.16cm, dp 0cm
    GLYPH (glyph) 'm', font 15, wd 0.29cm, ht 0.16cm
    PENALTY (linepenalty) 10000
    GLUE (parfillskip) plus +1fil
    GLUE (rightskip)
-----
before callback post_linebreak_filter
-----
GLUE (baselineskip) wd 4.76cm
HLIST (line) wd 12.13cm, dp 0cm, ht 0.24cm
  LL-head
    LOCAL_PAR
    HLIST (indent) wd 0.53cm
    GLYPH (glyph) 'L', font 15, wd 0.22cm, ht 0.24cm
    GLYPH (glyph) 'o', font 15, wd 0.18cm, ht 0.16cm, dp 0cm
    GLYPH (glyph) 'r', font 15, wd 0.14cm, ht 0.16cm
    GLYPH (glyph) 'e', font 15, wd 0.16cm, ht 0.16cm, dp 0cm
    GLYPH (glyph) 'm', font 15, wd 0.29cm, ht 0.16cm
    PENALTY (linepenalty) 10000
    GLUE (parfillskip) plus +1fil
    GLUE (rightskip)
-----

```

8.14 Type glue(12), subtype parskip(3)

\parskip=5cm Lorem Lorem

```

before callback pre_output_filter
- groupcode: output
- direction: TLT
- maxdepth: 327680
- packtype: exactly
- size: 36044800
-----
WHATSIIT (write) stream 129, data ''
GLUE (topskip) wd 3.17pt
HLIST (line) wd 345pt, dp 0.11pt, ht 6.83pt
  LL-head
    LOCAL_PAR
    HLIST (indent) wd 15pt
    GLYPH (glyph) 'L', font 15, wd 6.25pt, ht 6.83pt
    GLYPH (glyph) 'o', font 15, wd 5pt, ht 4.48pt, dp 0.11pt
    GLYPH (glyph) 'r', font 15, wd 3.92pt, ht 4.42pt

```

```

-GLYPH (glyph) 'e', font 15, wd 4.44pt, ht 4.48pt, dp 0.11pt
-GLYPH (glyph) 'm', font 15, wd 8.33pt, ht 4.42pt
-PENALTY (linepenalty) 10000
-GLUE (parfillskip) plus +1fil
-GLUE (rightskip)
-GLUE (parskip) wd 142.26pt
-GLUE (parskip)
-GLUE (baselineskip) wd 5.06pt
-HLIST (line) wd 345pt, dp 0.11pt, ht 6.83pt
  └─head
    └─LOCAL_PAR
      -HLIST (indent) wd 15pt
      -GLYPH (glyph) 'l', font 15, wd 6.25pt, ht 6.83pt
      -GLYPH (glyph) 'o', font 15, wd 5pt, ht 4.48pt, dp 0.11pt
      -GLYPH (glyph) 'r', font 15, wd 3.92pt, ht 4.42pt
      -GLYPH (glyph) 'e', font 15, wd 4.44pt, ht 4.48pt, dp 0.11pt
      -GLYPH (glyph) 'm', font 15, wd 8.33pt, ht 4.42pt
      -PENALTY (linepenalty) 10000
      -GLUE (parfillskip) plus +1fil
      -GLUE (rightskip)
-GLUE (userskip) wd -0.11pt
-GLUE (userskip) plus +1fil
-----
<no registered function for 'pre_output_filter' callback>
before callback pre_output_filter
- groupcode: output
- direction: TLT
- maxdepth: 327680
- packtype: exactly
- size: 36044800
-----
- WHATSIT (write) stream 129, data ''
- GLUE (topskip) wd 10pt
- VLIST (unknown)
-----
<no registered function for 'pre_output_filter' callback>
before callback pre_output_filter
- groupcode: output
- direction: TLT
- maxdepth: 327680
- packtype: exactly
- size: 36044800
-----
- WHATSIT (write) stream 129, data ''
- GLUE (topskip) wd 10pt
- HLIST (unknown) wd 345pt
- GLUE (userskip) plus +1fill
-----
<no registered function for 'pre_output_filter' callback>

```

8.15 Type glue(12), subtype spaceskip(13)

\spaceskip =5cm a a

```

before callback post_linebreak_filter
-----
-GLUE (baselineskip) wd 7.52pt
-HLIST (line) wd 345pt, dp 0.11pt, ht 4.48pt
  └─head
    └─LOCAL_PAR
      -HLIST (indent) wd 15pt
      -GLYPH (glyph) 'a', font 15, wd 5pt, ht 4.48pt, dp 0.11pt
      -GLUE (spaceskip) wd 142.26pt
      -GLYPH (glyph) 'a', font 15, wd 5pt, ht 4.48pt, dp 0.11pt
      -PENALTY (linepenalty) 10000
      -GLUE (parfillskip) plus +1fil

```

```
└GLUE (rightskip)
```

8.16 Type glue(12), subtype leaders(100)

```
a \leavevmode \leaders \hbox { . }\hfill \kern 0pt a
```

```
before callback post_linebreak_filter
-----
└GLUE (baselineskip) wd 7.52pt
└HLIST (line) wd 345pt, dp 0.11pt, ht 4.48pt
  └head
    └LOCAL_PAR
      └HLIST (indent) wd 15pt
        └GLYPH (glyph) 'a', font 15, wd 5pt, ht 4.48pt, dp 0.11pt
          └GLUE (spaceskip) wd 3.33pt, plus 1.66pt, minus 1.11pt
            └GLUE (leaders) plus +1fill
              └leader
                └HLIST (box) wd 10.55pt, ht 1.06pt
                  └head
                    └GLUE (spaceskip) wd 3.33pt, plus 1.66pt, minus 1.11pt
                      └GLYPH (glyph) '.', font 15, wd 2.78pt, ht 1.06pt
                        └GLUE (spaceskip) wd 4.44pt, plus 4.99pt, minus 0.37pt
          └KERN (userkern)
          └GLYPH (glyph) 'a', font 15, wd 5pt, ht 4.48pt, dp 0.11pt
          └PENALTY (linepenalty) 10000
          └GLUE (parfillskip) plus +1fil
          └GLUE (rightskip)
```

8.17 Type glue(12), subtype cleaders(101)

```
a \leavevmode \cleaders \hbox { . }\hfill \kern 0pt a
```

```
before callback post_linebreak_filter
-----
└GLUE (baselineskip) wd 7.52pt
└HLIST (line) wd 345pt, dp 0.11pt, ht 4.48pt
  └head
    └LOCAL_PAR
      └HLIST (indent) wd 15pt
        └GLYPH (glyph) 'a', font 15, wd 5pt, ht 4.48pt, dp 0.11pt
          └GLUE (spaceskip) wd 3.33pt, plus 1.66pt, minus 1.11pt
            └GLUE (cleaders) plus +1fill
              └leader
                └HLIST (box) wd 10.55pt, ht 1.06pt
                  └head
                    └GLUE (spaceskip) wd 3.33pt, plus 1.66pt, minus 1.11pt
                      └GLYPH (glyph) '.', font 15, wd 2.78pt, ht 1.06pt
                        └GLUE (spaceskip) wd 4.44pt, plus 4.99pt, minus 0.37pt
          └KERN (userkern)
          └GLYPH (glyph) 'a', font 15, wd 5pt, ht 4.48pt, dp 0.11pt
          └PENALTY (linepenalty) 10000
          └GLUE (parfillskip) plus +1fil
          └GLUE (rightskip)
```

8.18 Type glue(12), subtype xleaders(102)

```
a \leavevmode \xleaders \hbox { . }\hfill \kern 0pt a
```

```

before callback post_linebreak_filter
-----
├─GLUE (baselineskip) wd 7.52pt
├─HLIST (line) wd 345pt, dp 0.11pt, ht 4.48pt
│  └─head
│     └─LOCAL_PAR
│        └─HLIST (indent) wd 15pt
│           └─GLYPH (glyph) 'a', font 15, wd 5pt, ht 4.48pt, dp 0.11pt
│              └─GLUE (spaceskip) wd 3.33pt, plus 1.66pt, minus 1.11pt
│                 └─GLUE (xleaders) plus +1fill
│                    └─leader
│                       └─HLIST (box) wd 10.55pt, ht 1.06pt
│                          └─head
│                             └─GLUE (spaceskip) wd 3.33pt, plus 1.66pt, minus 1.11pt
│                                └─GLYPH (glyph) '.', font 15, wd 2.78pt, ht 1.06pt
│                                   └─GLUE (spaceskip) wd 4.44pt, plus 4.99pt, minus 0.37pt
├─KERN (userkern)
├─GLYPH (glyph) 'a', font 15, wd 5pt, ht 4.48pt, dp 0.11pt
├─PENALTY (linepenalty) 10000
├─GLUE (parfillskip) plus +1fil
└─GLUE (rightskip)
-----

```

8.19 Type glue(12), subtype gleaders(102)

a \leavevmode \gleaders \hbox { . }\hfill \kern 0pt a

```

before callback post_linebreak_filter
-----
├─GLUE (baselineskip) wd 7.52pt
├─HLIST (line) wd 345pt, dp 0.11pt, ht 4.48pt
│  └─head
│     └─LOCAL_PAR
│        └─HLIST (indent) wd 15pt
│           └─GLYPH (glyph) 'a', font 15, wd 5pt, ht 4.48pt, dp 0.11pt
│              └─GLUE (spaceskip) wd 3.33pt, plus 1.66pt, minus 1.11pt
│                 └─GLUE (gleaders) plus +1fill
│                    └─leader
│                       └─HLIST (box) wd 10.55pt, ht 1.06pt
│                          └─head
│                             └─GLUE (spaceskip) wd 3.33pt, plus 1.66pt, minus 1.11pt
│                                └─GLYPH (glyph) '.', font 15, wd 2.78pt, ht 1.06pt
│                                   └─GLUE (spaceskip) wd 4.44pt, plus 4.99pt, minus 0.37pt
├─KERN (userkern)
├─GLYPH (glyph) 'a', font 15, wd 5pt, ht 4.48pt, dp 0.11pt
├─PENALTY (linepenalty) 10000
├─GLUE (parfillskip) plus +1fil
└─GLUE (rightskip)
-----

```

8.20 Type kern(13), subtype userkern(0)

a \kern 2pt

```

before callback post_linebreak_filter
-----
├─GLUE (baselineskip) wd 7.52pt
├─HLIST (line) wd 345pt, dp 0.11pt, ht 4.48pt
│  └─head
│     └─LOCAL_PAR
│        └─HLIST (indent) wd 15pt
│           └─GLYPH (glyph) 'a', font 15, wd 5pt, ht 4.48pt, dp 0.11pt
│              └─KERN (userkern) 2pt
├─PENALTY (linepenalty) 10000
└─GLUE (parfillskip) plus +1fil
-----

```



```
└─GLUE (rightskip)
```

8.21 Type kern(13), subtype fontkern(1)

Ve

```
before callback post_linebreak_filter
-----
└─GLUE (baselineskip) wd 5.17pt
└─HLIST (line) wd 345pt, dp 0.22pt, ht 6.83pt
  └─head
    └─LOCAL_PAR
      └─HLIST (indent) wd 15pt
        └─GLYPH (glyph) 'V', font 15, wd 7.5pt, ht 6.83pt, dp 0.22pt
          └─KERN (fontkern) -0.83pt
            └─GLYPH (glyph) 'e', font 15, wd 4.44pt, ht 4.48pt, dp 0.11pt
              props {'injections'} = {'leftkern'} = -54394.88}
            └─PENALTY (linepenalty) 10000
          └─GLUE (parfillskip) plus +1fil
        └─GLUE (rightskip)
```

8.22 Type kern(13), subtype accentkern(2)

\accent 96 a

```
before callback post_linebreak_filter
-----
└─GLUE (baselineskip) wd 4.85pt
└─HLIST (line) wd 345pt, dp 0.11pt, ht 7.15pt
  └─head
    └─LOCAL_PAR
      └─HLIST (indent) wd 15pt
        └─KERN (accentkern)
          └─HLIST (unknown) wd 5pt, ht 6.98pt, shift -0.17pt
            └─head
              └─GLYPH (glyph) '`, font 15, wd 5pt, ht 6.98pt
                └─KERN (accentkern) -5pt
                  └─GLYPH (glyph) 'a', font 15, wd 5pt, ht 4.48pt, dp 0.11pt
                    └─PENALTY (linepenalty) 10000
                  └─GLUE (parfillskip) plus +1fil
                └─GLUE (rightskip)
```

8.23 Type kern(13), subtype italiccorrection(3)

\textit {L}\OL

```
before callback post_linebreak_filter
-----
└─GLUE (baselineskip) wd 4.95pt
└─HLIST (line) wd 345pt, dp 0.22pt, ht 7.05pt
  └─head
    └─LOCAL_PAR
      └─HLIST (indent) wd 15pt
        └─GLYPH (glyph) 'L', font 16, wd 6.27pt, ht 6.83pt
          └─KERN (italiccorrection) 0.17pt
            └─GLYPH (glyph) 'O', font 15, wd 7.78pt, ht 7.05pt, dp 0.22pt
              └─GLYPH (glyph) 'L', font 15, wd 6.25pt, ht 6.83pt
                └─PENALTY (linepenalty) 10000
              └─GLUE (parfillskip) plus +1fil
            └─GLUE (rightskip)
```

8.24 Type penalty(14)

L \penalty 23 0L

```
before callback post_linebreak_filter
-----
├─GLUE (baselineskip) wd 4.95pt
├─HLIST (line) wd 345pt, dp 0.22pt, ht 7.05pt
└─head
  └─LOCAL_PAR
    ├─HLIST (indent) wd 15pt
    ├─GLYPH (glyph) 'L', font 15, wd 6.25pt, ht 6.83pt
    ├─GLUE (spaceskip) wd 3.33pt, plus 1.66pt, minus 1.11pt
    ├─PENALTY (userpenalty) 23
    ├─GLYPH (glyph) 'O', font 15, wd 7.78pt, ht 7.05pt, dp 0.22pt
    ├─GLYPH (glyph) 'L', font 15, wd 6.25pt, ht 6.83pt
    ├─PENALTY (linepenalty) 10000
    ├─GLUE (parfillskip) plus +1fil
    └─GLUE (rightskip)
```

8.25 Type glyph(29)

abc

```
before callback post_linebreak_filter
-----
├─GLUE (baselineskip) wd 5.06pt
├─HLIST (line) wd 345pt, dp 0.11pt, ht 6.94pt
└─head
  └─LOCAL_PAR
    ├─HLIST (indent) wd 15pt
    ├─GLYPH (glyph) 'a', font 15, wd 5pt, ht 4.48pt, dp 0.11pt
    ├─GLYPH (glyph) 'b', font 15, wd 5.56pt, ht 6.94pt, dp 0.11pt
    ├─KERN (fontkern) 0.28pt
    ├─GLYPH (glyph) 'c', font 15, wd 4.44pt, ht 4.48pt, dp 0.11pt
    │   props {'injections'} = [{'leftkern'} = 18350.08]}
    ├─PENALTY (linepenalty) 10000
    ├─GLUE (parfillskip) plus +1fil
    └─GLUE (rightskip)
```

8.26 Type attributelist(40)

{\attribute 0=1 A}

```
before callback hpack_filter
- size: 0
- groupcode: hbox
- direction: TLT
- packtype: additional
- attributelist: <node nil < 83 > 149 : attribute_list 0>
-----
└─RULE (normal) dp 3.6pt, ht 8.4pt
-----
before callback hpack_filter
- size: 0
- groupcode: hbox
- direction: TLT
- packtype: additional
- attributelist: <node nil < 145 > 147 : attribute_list 0>
-----
└─RULE (normal) dp 3.6pt, ht 8.4pt
-----
before callback hpack_filter
```

```

- size: 0
- groupcode: hbox
- direction: TLT
- packtype: additional
- attributelist: <node 12 < 513 > 594 : attribute_list 0>
-----
└─RULE (normal) dp 3.6pt, ht 8.4pt
-----
before callback hpack_filter
- size: 0
- groupcode: adjusted_hbox
- direction: TLT
- packtype: additional
- attributelist: <node nil < 511 > 605 : attribute_list 0>
-----
└─HLIST (box) wd 345pt
-----
before callback hpack_filter
- size: 22609920
- groupcode: hbox
- direction: TLT
- packtype: exactly
- attributelist: <node nil < 684 > 691 : attribute_list 0>
-----
└─GLUE (userskip) plus +1fil
└─GLYPH (glyph) '1', font 15, wd 5pt, ht 6.66pt
└─GLUE (userskip) plus +1fil
-----
before callback hpack_filter
- size: 0
- groupcode: adjusted_hbox
- direction: TLT
- packtype: additional
- attributelist: <node nil < 684 > 691 : attribute_list 0>
-----
└─HLIST (box) wd 345pt, ht 6.66pt
  └─head
    └─GLUE (userskip) plus +1fil
    └─GLYPH (glyph) '1', font 15, wd 5pt, ht 6.66pt
    └─GLUE (userskip) plus +1fil
-----

```

9 Implementation

9.1 The file nodetree.tex

```
23 \directlua{
24   nodetree = require('nodetree')
25 }
```

\NodetreeSetOption

```
26 \def\NodetreeSetOption[#1]#2{
27   \directlua{
28     nodetree.set_option('#1', '#2')
29   }
30 }
31 \let\nodetreeoption\NodetreeSetOption
```

\NodetreeResetOption

```
32 \def\NodetreeResetOption#1{
33   \NodetreeSetOption[#1]{%
34     \directlua{
35       tex.print(nodetree.get_default_option('#1'))
36     }%
37   }%
38 }
```

\NodetreeReset

```
39 \def\NodetreeReset{
40   \NodetreeResetOption{callback}
41   \NodetreeResetOption{channel}
42   \NodetreeResetOption{color}
43   \NodetreeResetOption{decimalplaces}
44   \NodetreeResetOption{unit}
45   \NodetreeResetOption{verbosity}
46 }
47 \let\nodetreereset\NodetreeReset
```

\NodetreeRegisterCallback

```
48 \def\NodetreeRegisterCallback#1{
49   \directlua{
50     nodetree.set_option('callback', '#1')
51     nodetree.register_callbacks()
52   }
53 }
54 \let\nodetreeregister\NodetreeRegisterCallback
```

\NodetreeUnregisterCallback

```
55 \def\NodetreeUnregisterCallback#1{
56   \directlua{
57     nodetree.set_option('callback', '#1')
```

```

58     nodetree.unregister_callbacks()
59   }
60 }
61 \let\nodetreeunregister\NodetreeUnregisterCallback

```

9.2 The file nodetree.sty

```

26 \input{nodetree}

27 \RequirePackage{kvoptions}

28 \SetupKeyvalOptions{
29   family=NT,
30   prefix=NTK@
31 }

32 \DeclareStringOption[term]{channel}
33 \define@key{NT}{channel}[]{\NodetreeSetOption[channel]{#1}}

34 \DeclareStringOption[postlinebreak]{callback}
35 \define@key{NT}{callback}[]{\NodetreeSetOption[callback]{#1}}

36 \DeclareStringOption[0]{verbosity}
37 \define@key{NT}{verbosity}[]{\NodetreeSetOption[verbosity]{#1}}

38 \DeclareStringOption[colored]{color}
39 \define@key{NT}{color}[]{\NodetreeSetOption[color]{#1}}

40 \DeclareStringOption[1]{unit}
41 \define@key{NT}{unit}[]{\NodetreeSetOption[unit]{#1}}

42 \DeclareStringOption[1]{decimalplaces}
43 \define@key{NT}{decimalplaces}[]{\NodetreeSetOption[decimalplaces]{#1}}

44 \ProcessKeyvalOptions{NT}
45 \directlua{
46   nodetree.register_callbacks()
47 }

```

`\NodetreeSet`

```

48 \newcommand{\NodetreeSet}[1]{%
49   \setkeys{NT}{#1}%
50 }
51 \let\nodetreeset\NodetreeSet

```

9.3 The file nodetree-embed.sty

```

23 \NeedsTeXFormat{LaTeX2e}[1994/06/01]
24 \ProvidesPackage{nodetree-embed}
25   [2023/09/10 v2.3.0 Embed node trees into a LaTeX document]

26 \RequirePackage{xcolor,mdframed,expl3,xparse,fontspec}

```

```

27 \input{nodetree}

28 \RequirePackage{kvoptions}
29 \SetupKeyvalOptions{
30   family=NTE,
31   prefix=NTEK@
32 }

33 \define@key{NTE}{callback}[]{\NodetreeSetOption[callback]{#1}}

34 \DeclareStringOption[0]{verbosity}
35 \define@key{NTE}{verbosity}[]{\NodetreeSetOption[verbosity]{#1}}

36 \DeclareStringOption[colored]{color}
37 \define@key{NTE}{color}[]{\NodetreeSetOption[color]{#1}}

38 \DeclareStringOption[1]{unit}
39 \define@key{NTE}{unit}[]{\NodetreeSetOption[unit]{#1}}

40 \DeclareStringOption[1]{decimalplaces}
41 \define@key{NTE}{decimalplaces}[]{\NodetreeSetOption[decimalplaces]{#1}}

42 \DeclareStringOption[monokaisoda]{theme}

43 \DeclareStringOption[dark]{thememode}

44 \DeclareStringOption[Ubuntu Mono]{font}

45 \DeclareStringOption[\footnotesize]{fontsize}

46 \DeclareStringOption[1]{firstline}
47 \define@key{NTE}{firstline}[]{\NodetreeSetOption[firstline]{#1}}

48 \DeclareStringOption[-1]{lastline}
49 \define@key{NTE}{lastline}[]{\NodetreeSetOption[lastline]{#1}}

50 \DeclareBoolOption{showmarkup}

51 \ProcessKeyvalOptions{NTE}

```

\NTE@colors

```

52 \ExplSyntaxOn
53 \def\NTE@colors{
54   \str_case_e:nn{\NTEK@theme}{
55     {bwdark}{
56       \definecolor{NTEblack}{gray}{0}
57       \definecolor{NTEred}{gray}{1}
58       \definecolor{NTEgreen}{gray}{1}
59       \definecolor{NTEyellow}{gray}{1}
60       \definecolor{NTEblue}{gray}{1}
61       \definecolor{NTEmagenta}{gray}{1}
62       \definecolor{NTEcyan}{gray}{1}

```

```

63     \definecolor{NTEwhite}{gray}{1}
64     \definecolor{NTEblackbright}{gray}{0}
65     \definecolor{NTEredbright}{gray}{1}
66     \definecolor{NTEgreenbright}{gray}{1}
67     \definecolor{NTEyellowbright}{gray}{1}
68     \definecolor{NTEbluebright}{gray}{1}
69     \definecolor{NTEmagentabright}{gray}{1}
70     \definecolor{NTEcyanbright}{gray}{1}
71     \definecolor{NTEwhitebright}{gray}{1}
72 }
73 {bilight}{
74     \definecolor{NTEblack}{gray}{0}
75     \definecolor{NTEred}{gray}{0}
76     \definecolor{NTEgreen}{gray}{0}
77     \definecolor{NTEyellow}{gray}{0}
78     \definecolor{NTEblue}{gray}{0}
79     \definecolor{NTEmagenta}{gray}{0}
80     \definecolor{NTEcyan}{gray}{0}
81     \definecolor{NTEwhite}{gray}{1}
82     \definecolor{NTEblackbright}{gray}{0}
83     \definecolor{NTEredbright}{gray}{0}
84     \definecolor{NTEgreenbright}{gray}{0}
85     \definecolor{NTEyellowbright}{gray}{0}
86     \definecolor{NTEbluebright}{gray}{0}
87     \definecolor{NTEmagentabright}{gray}{0}
88     \definecolor{NTEcyanbright}{gray}{0}
89     \definecolor{NTEwhitebright}{gray}{1}
90 }
91 {monokaisoda}{
92     \definecolor{NTEblack}{HTML}{1a1a1a}
93     \definecolor{NTEred}{HTML}{f4005f}
94     \definecolor{NTEgreen}{HTML}{98e024}
95     \definecolor{NTEyellow}{HTML}{fa8419}
96     \definecolor{NTEblue}{HTML}{9d65ff}
97     \definecolor{NTEmagenta}{HTML}{f4005f}
98     \definecolor{NTEcyan}{HTML}{58d1eb}
99     \definecolor{NTEwhite}{HTML}{c4c5b5}
100    \definecolor{NTEblackbright}{HTML}{625e4c}
101    \definecolor{NTEredbright}{HTML}{f4005f}
102    \definecolor{NTEgreenbright}{HTML}{98e024}
103    \definecolor{NTEyellowbright}{HTML}{e0d561}
104    \definecolor{NTEbluebright}{HTML}{9d65ff}
105    \definecolor{NTEmagentabright}{HTML}{f4005f}
106    \definecolor{NTEcyanbright}{HTML}{58d1eb}
107    \definecolor{NTEwhitebright}{HTML}{f6f6ef}
108 }
109 }
110 \str_case_e:nn{\NTEK@thememode}{
111     {dark}{
112         \definecolor{NTEbackground}{named}{NTEblack}
113         \definecolor{NTEfont}{named}{NTEwhitebright}
114     }
115     {light}{
116         \definecolor{NTEbackground}{named}{NTEwhitebright}

```

```

117     \definecolor{NTEfont}{named}{NTEblack}
118   }
119 }
120 }
121 \ExplSyntaxOff

```

`\NTE@fonts`

```

122 \def\NTE@fonts{
123   \bfseries%
124   \NTEK@fontsize%
125   \setmonofont{\NTEK@font}%
126   \ttfamily%
127   \setlength{\parindent}{0pt}%
128   \setlength{\parskip}{-0.9pt}%
129 }

```

`\NodetreeSet` Same definition as in `nodetree.sty`. Only implement this command if not already registered.

```

130 \providecommand{\NodetreeSet}[1]{%
131   \setkeys{NTE}{#1}%
132 }

133 \newenvironment{NodetreeEmbedView}[1][]{
134   \directlua{nodetree.push_options()}
135   \setkeys{NTE}{#1}
136   \NTE@colors
137   \begin{mdframed}[
138     linecolor=black,
139     backgroundcolor=NTEbackground,
140     fontcolor=NTEfont,
141   ]%
142   \NTE@fonts
143 }{
144   \end{mdframed}%
145   \directlua{nodetree.pop_options()}%
146 }

```

`NodetreeEmbedEnv` (*env.*)

```

147 \NewDocumentEnvironment { NodetreeEmbedEnv } { 0{} +b } {
148   \directlua{
149     nodetree.check_shell_escape('NodetreeEmbedEnv', false)
150     nodetree.push_options()
151   }
152   \setkeys{NTE}{#1}
153   \ifNTEK@showmarkup
154     \noindent
155     \texttt{\detokenize{#2}}
156   \fi
157   \NTE@colors
158   \begin{NodetreeEmbedView}
159     \directlua{

```



```

160     nodetree.compile_include('\luaescapestring{\unexpanded{#2}}')
161   }
162   \end{NodetreeEmbedView}
163   \directlua{nodetree.pop_options()}
164 }{}

```

`\NodetreeEmbedCmd`

```

165 \NewDocumentCommand { \NodetreeEmbedCmd } { 0{} +v } {
166   \directlua{
167     nodetree.check_shell_escape('\string\NodetreeEmbedCmd', true)
168     nodetree.push_options()
169   }
170   \setkeys{NTE}{#1}
171   \ifNTEK@showmarkup
172     \noindent
173     \texttt{#2}
174   \fi
175   \NTE@colors
176   \begin{NodetreeEmbedView}
177     \directlua{
178       nodetree.compile_include('\luaescapestring{\unexpanded{#2}}')
179     }
180   \end{NodetreeEmbedView}
181   \directlua{nodetree.pop_options()}
182 }

```

`\NodetreeEmbedInput`

```

183 \newcommand{\NodetreeEmbedInput}[2] []{
184   \directlua{nodetree.push_options()}
185   \setkeys{NTE}{#1}
186   \begin{NodetreeEmbedView}
187     \directlua{nodetree.input('#2.nttex')}
188   \end{NodetreeEmbedView}
189   \directlua{nodetree.pop_options()}
190 }
191 \let\nodetreeterminalemulator\NodetreeEmbedInput

```

9.4 The file `nodetree.lua`

There is a source code documentation of the file `nodetree.lua` compiled with Ldoc on Github: <http://josef-friedrich.github.io/nodetree/>