

The nodetree package

Josef Friedrich

josef@friedrich.rocks

github.com/Josef-Friedrich/nodetree

v2.2 from 2020/10/23

```
Callback: post_linebreak_filter
```

```
-----  
└─GLUE subtype: baselineskip, width: 5.06pt  
└─HLIST subtype: line, width: 345pt, depth: 0.11pt, height: 6.94pt  
  └─head:  
    └─LOCAL_PAR  
      └─HLIST subtype: indent, width: 15pt  
        └─GLYPH subtype: 256, char: n, width: 5.56pt, height: 4.42pt  
          └─GLYPH subtype: 256, char: o, width: 5pt, height: 4.48pt, depth: 0.11pt  
            └─KERN kern: 0.28pt  
              └─GLYPH subtype: 256, char: d, width: 5.56pt, height: 6.94pt, depth: 0.11pt  
                properties: {['injections'] = {['leftkern'] = 18350.08}}  
              └─GLYPH subtype: 256, char: e, width: 4.44pt, height: 4.48pt, depth: 0.11pt  
                └─DISC subtype: regular, penalty: 50  
                  └─pre:  
                    └─GLYPH subtype: 256, char: -, width: 3.33pt, height: 2.45pt  
                      └─GLYPH subtype: 256, char: t, width: 3.89pt, height: 6.15pt, depth: 0.11pt  
                        └─GLYPH subtype: 256, char: r, width: 3.92pt, height: 4.42pt  
                          └─GLYPH subtype: 256, char: e, width: 4.44pt, height: 4.48pt, depth: 0.11pt  
                            └─GLYPH subtype: 256, char: e, width: 4.44pt, height: 4.48pt, depth: 0.11pt  
                              └─PENALTY subtype: linepenalty, penalty: 10000  
                                └─GLUE subtype: parfillskip, stretch: +1fil  
                                  └─GLUE subtype: 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 _L _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>	14
5.3.1	Example: <code>verbosity=1</code>	14
5.3.2	Example: <code>verbosity=2</code>	15
5.4	Option <code>color</code>	15
5.5	Option <code>unit</code>	15
5.5.1	Example: <code>unit=pt</code>	16
5.5.2	Example: <code>unit=sp</code>	16
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>	17
5.6.3	Example: <code>decimalplaces=5</code>	17
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>	18
5.7.4	Example: <code>theme=monokaisoda thememode=light</code>	18
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>	19
5.9.1	Example:	19
5.9.2	Example:	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>	23
7.4	Node types	23
7.4.1	Type: <code>hlist(0)</code> Subtype: <code>line(1)</code>	23
7.4.2	Type: <code>hlist(0)</code> Subtype: <code>box(2)</code>	24
7.4.3	Type: <code>hlist(0)</code> Subtype: <code>indent(3)</code>	24
7.4.4	Type: <code>vlist(1)</code>	25
7.4.5	Type: <code>rule(2)</code>	25
7.4.6	Type: <code>mark(4)</code>	26
7.4.7	Type: <code>disc(7)</code> Subtype: <code>discretionary(0)</code>	27
7.4.8	Type: <code>disc(7)</code> Subtype: <code>regular(3)</code>	27
7.4.9	Type: <code>whatsit(8)</code> Subtype: <code>pdfaction(22)</code>	28
7.4.10	Type: <code>whatsit(8)</code> Subtype: <code>pdfcolorstack(28)</code>	29
7.4.11	Type: <code>glue(12)</code> Subtype: <code>baselineskip(2)</code>	29
7.4.12	Type: <code>glue(12)</code> Subtype: <code>parskip(3)</code>	30
7.4.13	Type: <code>glue(12)</code> Subtype: <code>spaceskip(13)</code>	31
7.4.14	Type: <code>glue(12)</code> Subtype: <code>leaders(100)</code>	31
7.4.15	Type: <code>glue(12)</code> Subtype: <code>cleaders(101)</code>	31
7.4.16	Type: <code>glue(12)</code> Subtype: <code>xleaders(102)</code>	32
7.4.17	Type: <code>glue(12)</code> Subtype: <code>gleaders(102)</code>	32
7.4.18	Type: <code>kern(13)</code> Subtype: <code>userkern(0)</code>	33
7.4.19	Type: <code>kern(13)</code> Subtype: <code>fontkern(1)</code>	33
7.4.20	Type: <code>kern(13)</code> Subtype: <code>accentkern(2)</code>	34
7.4.21	Type: <code>kern(13)</code> Subtype: <code>italiccorrection(3)</code>	34
7.4.22	Type: <code>penalty(14)</code>	34
7.4.23	Type: <code>glyph(29)</code>	35
7.4.24	Type: <code>attribute(38)</code>	35
7.4.25	Type: <code>attributelist(40)</code>	35

8	Implementation	37
8.1	The file <code>nodetree.tex</code>	37
8.2	The file <code>nodetree.sty</code>	38
8.3	The file <code>nodetree.lua</code>	42

1 Abstract

`nodetree` is a development package that visualizes the structure of node lists. `nodetree` shows its debug informations in the consoles' output when you compile a Lua \TeX file. It uses a similar visual representation for node lists as the UNIX `tree` command does for a folder tree.

Node lists are the main building blocks of each document generated by the \TeX engine *Lua \TeX* . The package `nodetree` doesn't change the rendered document. The tree view can only be seen when using a terminal to generate the document.

`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 Lua \TeX , a Lua \LaTeX package or as package to embed `nodetree` views in a Lua \LaTeX 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>channel</code>	Page 14, Section 5.2
<code>verbosity</code>	Page 14, Section 5.3
<code>color</code>	Page 15, Section 5.4
<code>unit</code>	Page 15, Section 5.5
<code>decimalplaces</code>	Page 17, Section 5.6

2.2 As a Lua^AT_EX package

Or run `lualatex lualatex-test.tex` to show a node tree using Lua^AT_EX. In Lua^AT_EX you can omit `\NodetreeRegisterCallback{postline}`. `\usepackage{nodetree}` registers automatically the `post_linebreak_filter`. If you don't want debug the `post_linebreak_filter` use `\NodetreeUnregisterCallback{postline}`.

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

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

2.2.1 Available macros

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

2.2.2 Available options

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

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' })
```

This are the default options:

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

The following code snippet demonstrates the usage in Lua_T_E_X. `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_A_T_E_X.

```
\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 the single intermediate steps:

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]{1}
\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 embed in the environment `NodetreeEmbed`

```
\par{}\par{}Callback: \textcolor{NTEred}{post\_linebreak\_filter}\par{}
-----\par{}
\mbox{ \textcolor{NTEmagentabright}{GLUE\hspace{0.5em}}\textcolor{NTEyellow}{subtype:}
↪ baselineskip, \textcolor{NTEyellow}{width:}
↪ 5.06\textcolor{NTEwhite}{pt}}\par{}
...
```

4. Finally the result:

```
Callback: post_linebreak_filter
-----
├─GLUE subtype: baselineskip, width: 5.06pt
├─HLIST subtype: line, width: 345pt, depth: 0.11pt, height: 6.94pt
└─head:
  ├─LOCAL_PAR
  ├─HLIST subtype: indent, width: 15pt
  ├─GLYPH subtype: 256, char: n, width: 5.56pt, height: 4.42pt
  ├─GLYPH subtype: 256, char: o, width: 5pt, height: 4.48pt, depth: 0.11pt
  └─KERN kern: 0.28pt
```

```

├─GLYPH subtype: 256, char: d, width: 5.56pt, height: 6.94pt, depth: 0.11pt
│  properties: {[ 'injections' ] = {[ 'leftkern' ] = 18350.08}}
├─GLYPH subtype: 256, char: e, width: 4.44pt, height: 4.48pt, depth: 0.11pt
├─DISC subtype: regular, penalty: 50
├─┬pre:
│  └─GLYPH subtype: 256, char: -, width: 3.33pt, height: 2.45pt
├─GLYPH subtype: 256, char: t, width: 3.89pt, height: 6.15pt, depth: 0.11pt
├─GLYPH subtype: 256, char: r, width: 3.92pt, height: 4.42pt
├─GLYPH subtype: 256, char: e, width: 4.44pt, height: 4.48pt, depth: 0.11pt
├─GLYPH subtype: 256, char: e, width: 4.44pt, height: 4.48pt, depth: 0.11pt
├─PENALTY subtype: linepenalty, penalty: 10000
├─GLUE subtype: parfillskip, stretch: +1fil
└─GLUE subtype: 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>channel</code>	Page 14, Section 5.2
<code>verbosity</code>	Page 14, Section 5.3
<code>color</code>	Page 15, Section 5.4
<code>unit</code>	Page 15, 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 19, Section 5.9

3 Macros

3.1 `\NodetreeRegisterCallback`

`\NodetreeRegisterCallback` `\NodetreeRegisterCallback{<callbacks>}`: The argument `{<callbacks>}` takes a comma separated list of callback aliases as described in (→ 5.1).

3.2 `\NodetreeUnregisterCallback`

`\NodetreeUnregisterCallback` `\NodetreeUnregisterCallback{<callbacks>}`: The argument `{<callbacks>}` takes a comma separated list of callback aliases as described in (→ 5.1).

3.3 `\NodetreeSetOption`

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

3.4 `\NodetreeResetOption`

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

3.5 `\NodetreeSet`

`\NodetreeSet` `\NodetreeSet{<kv-options>}`: This macro sets multiple options a once. It only can 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`: This macro resets multiple options to its 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 the `xparse +v` option to grab the verbatim content. Only available in the package `nodetree-embed`.

3.8 `\NodetreeEmbedInput`

`\NodetreeEmbedInput` `\NodetreeEmbedInput[<kv-options>]{<nttex-file>}`: The path or filename of `*.nttex` file without the extension. Only available in the package `nodetree-embed`.

4 Environments

4.1 NodetreeEmbedEnv

`NodetreeEmbedEnv` `\begin{NodetreeEmbedEnv}[\langle kv-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 ([→ 3.7](#)). Uses the `\detokenize` command to grab the verbatim content. Only available in the package `nodetree-embed`.

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.

This macros process callback options: `\NodetreeRegisterCallback{<callbacks>}`, `\NodetreeUnregisterCallback{<callbacks>}`, `\NodetreeSet{<callback=<callbacks>}` and `\usepackage[<callback=<callbacks>]{<nodetree>}`.

Use commas to specify multiple callbacks. Avoid using whitespaces:

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

Wrap your callback aliases in curly braces for the macro `\NodetreeSet`:

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

The same applies for the macro `\usepackage`:

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

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 `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 options is 1. At the moment only verbosity level 2 is implemented.

5.3.1 Example: verbosity=1

```
Callback: pre_linebreak_filter
-----
|LOCAL_PAR
|HLIST subtype: indent, width: 15pt
|GLYPH subtype: 256, char: ., width: 2.78pt, height: 1.06pt
|PENALTY subtype: linepenalty, penalty: 10000
|GLUE subtype: parfillskip, stretch: +1fil
```

The callbacks are listed in the same order as in the LuaTeX reference manual.

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

```

-----
5.3.2 Example: verbosity=2

Callback: pre_linebreak_filter
-----
|LOCAL_PAR[9] no: 464
|GLYPH[29] no: 398, subtype: indent[3], width: 15pt
|GLYPH[29] no: 398, subtype: 256, char: . [46], font: 25, width: 2.78pt, height: 1.06pt
|PENALTY[14] no: 329, subtype: linepenalty[2], penalty: 10000
|GLUE[12] no: 470, subtype: parfillskip[15], stretch: +1fil
-----

```

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 figure 2 and 3 for possible values.

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

5.5.1 Example: unit=pt

```

Callback: pre_linebreak_filter
-----
|LOCAL_PAR
|HLIST subtype: indent, width: 15pt
|GLYPH subtype: 256, char: L, width: 6.25pt, height: 6.83pt
|GLYPH subtype: 256, char: o, width: 5pt, height: 4.48pt, depth: 0.11pt
|GLYPH subtype: 256, char: r, width: 3.92pt, height: 4.42pt
|GLYPH subtype: 256, char: e, width: 4.44pt, height: 4.48pt, depth: 0.11pt
|GLYPH subtype: 256, char: m, width: 8.33pt, height: 4.42pt
|GLYPH subtype: 256, char: ., width: 2.78pt, height: 1.06pt
|PENALTY subtype: linepenalty, penalty: 10000
|GLUE subtype: parfillskip, stretch: +1fil
-----

```

5.5.2 Example: unit=sp

```

Callback: pre_linebreak_filter
-----
|LOCAL_PAR
|HLIST subtype: indent, width: 983040sp
|GLYPH subtype: 256, char: L, width: 409600sp, height: 447611sp
|GLYPH subtype: 256, char: o, width: 327680sp, height: 293601sp, depth: 7209sp
|GLYPH subtype: 256, char: r, width: 256901sp, height: 289669sp
|GLYPH subtype: 256, char: e, width: 290980sp, height: 293601sp, depth: 7209sp
|GLYPH subtype: 256, char: m, width: 545915sp, height: 289669sp
|GLYPH subtype: 256, char: ., width: 182190sp, height: 69468sp
|PENALTY subtype: linepenalty, penalty: 10000
|GLUE subtype: parfillskip, stretch: +1fil
-----

```

5.5.3 Example: unit=cm

```
Callback: pre_linebreak_filter
-----
|LOCAL_PAR
|HLIST subtype: indent, width: 0.53cm
|GLYPH subtype: 256, char: L, width: 0.22cm, height: 0.24cm
|GLYPH subtype: 256, char: o, width: 0.18cm, height: 0.16cm, depth: 0cm
|GLYPH subtype: 256, char: r, width: 0.14cm, height: 0.16cm
|GLYPH subtype: 256, char: e, width: 0.16cm, height: 0.16cm, depth: 0cm
|GLYPH subtype: 256, char: m, width: 0.29cm, height: 0.16cm
|GLYPH subtype: 256, char: ., width: 0.1cm, height: 0.04cm
|PENALTY subtype: linepenalty, penalty: 10000
|GLUE subtype: parfillskip, stretch: +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

```
Callback: pre_linebreak_filter
-----
|LOCAL_PAR
|HLIST subtype: indent, width: 1cc
|GLYPH subtype: 256, char: L, width: 0cc, height: 1cc
|GLYPH subtype: 256, char: o, width: 0cc, height: 0cc, depth: 0cc
|GLYPH subtype: 256, char: r, width: 0cc, height: 0cc
|GLYPH subtype: 256, char: e, width: 0cc, height: 0cc, depth: 0cc
|GLYPH subtype: 256, char: m, width: 1cc, height: 0cc
|GLYPH subtype: 256, char: ., width: 0cc, height: 0cc
|PENALTY subtype: linepenalty, penalty: 10000
|GLUE subtype: parfillskip, stretch: +1fil
-----
```

5.6.2 Example: decimalplaces=2

```
Callback: pre_linebreak_filter
-----
|LOCAL_PAR
|HLIST subtype: indent, width: 1.17cc
|GLYPH subtype: 256, char: L, width: 0.49cc, height: 0.53cc
|GLYPH subtype: 256, char: o, width: 0.39cc, height: 0.35cc, depth: 0.01cc
|GLYPH subtype: 256, char: r, width: 0.31cc, height: 0.34cc
|GLYPH subtype: 256, char: e, width: 0.35cc, height: 0.35cc, depth: 0.01cc
|GLYPH subtype: 256, char: m, width: 0.65cc, height: 0.34cc
|GLYPH subtype: 256, char: ., width: 0.22cc, height: 0.08cc
|PENALTY subtype: linepenalty, penalty: 10000
|GLUE subtype: parfillskip, stretch: +1fil
-----
```

5.6.3 Example: decimalplaces=5

```

Callback: pre_linebreak_filter
-----
└LOCAL_PAR
└HLIST subtype: indent, width: 1.16821cc
└GLYPH subtype: 256, char: L, width: 0.48676cc, height: 0.53193cc
└GLYPH subtype: 256, char: o, width: 0.3894cc, height: 0.34891cc, depth: 0.00857cc
└GLYPH subtype: 256, char: r, width: 0.30529cc, height: 0.34423cc
└GLYPH subtype: 256, char: e, width: 0.34579cc, height: 0.34891cc, depth: 0.00857cc
└GLYPH subtype: 256, char: m, width: 0.64875cc, height: 0.34423cc
└GLYPH subtype: 256, char: ., width: 0.21651cc, height: 0.08255cc
└PENALTY subtype: linepenalty, penalty: 10000
└GLUE subtype: parfillskip, stretch: +1fil
-----

```

5.7 Option theme and thememode

5.7.1 Example: theme=bwdark thememode=dark

```

Callback: pre_linebreak_filter
-----
└LOCAL_PAR
└HLIST subtype: indent, width: 15pt
└GLYPH subtype: 256, char: ., width: 2.78pt, height: 1.06pt
└PENALTY subtype: linepenalty, penalty: 10000
└GLUE subtype: parfillskip, stretch: +1fil
-----

```

5.7.2 Example: theme=bwlight thememode=light

```

Callback: pre_linebreak_filter
-----
└LOCAL_PAR
└HLIST subtype: indent, width: 15
└GLYPH subtype: 256, char: ., width: 2.78 , height: 1.06
└PENALTY subtype: linepenalty, penalty: 10000
└GLUE subtype: parfillskip, stretch: +1
-----

```

5.7.3 Example: theme=monokaisoda thememode=dark

```

Callback: pre_linebreak_filter
-----
└LOCAL_PAR
└HLIST subtype: indent, width: 15pt
└GLYPH subtype: 256, char: ., width: 2.78pt, height: 1.06pt
└PENALTY subtype: linepenalty, penalty: 10000
└GLUE subtype: parfillskip, stretch: +1fil
-----

```

5.7.4 Example: theme=monokaisoda thememode=light

```

Callback: pre_linebreak_filter
-----

```

```

├LOCAL_PAR
├HLIST subtype: indent, width: 15pt
├GLYPH subtype: 256, char: ., width: 2.78pt, height: 1.06pt
├PENALTY subtype: linepenalty, penalty: 10000
└GLUE subtype: parfillskip, stretch: +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 a monospaced and have some box drawing glyphs (See table [UNICODE glyphs 4](#)).

5.8.1 Example: `font={Liberation Mono}`

```

Callback: post_linebreak_filter
-----
├GLUE subtype: baselineskip, width: 10.94pt
├HLIST subtype: line, width: 345pt, height: 1.06pt
└head:
  ├──LOCAL_PAR
  ├──HLIST subtype: indent, width: 15pt
  ├──GLYPH subtype: 256, char: ., width: 2.78pt, height: 1.06pt
  ├──PENALTY subtype: linepenalty, penalty: 10000
  ├──GLUE subtype: parfillskip, stretch: +1fil
  └GLUE subtype: rightskip
-----

```

5.8.2 Example: `font={Ubuntu Mono}`

```

Callback: post_linebreak_filter
-----
├GLUE subtype: baselineskip, width: 10.94pt
├HLIST subtype: line, width: 345pt, height: 1.06pt
└head:
  ├──LOCAL_PAR
  ├──HLIST subtype: indent, width: 15pt
  ├──GLYPH subtype: 256, char: ., width: 2.78pt, height: 1.06pt
  ├──PENALTY subtype: linepenalty, penalty: 10000
  ├──GLUE subtype: parfillskip, stretch: +1fil
  └GLUE subtype: rightskip
-----

```

5.9 Option fontsize

5.9.1 Example: `\small`

```

Callback: pre_linebreak_filter
-----
├LOCAL_PAR
├HLIST subtype: indent, width: 15pt

```

```
└GLYPH subtype: 256, char: ., width: 2.78pt, height: 1.06pt
└PENALTY subtype: linepenalty, penalty: 10000
└GLUE subtype: parfillskip, stretch: +1fil
-----
```

5.9.2 Example: \tiny

```
Callback: pre_linebreak_filter
-----
└LOCAL_PAR
└LIST subtype: indent, width: 15pt
└GLYPH subtype: 256, char: ., width: 2.78pt, height: 1.06pt
└PENALTY subtype: linepenalty, penalty: 10000
└GLUE subtype: parfillskip, stretch: +1fil
-----
```

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 the `list` and `field` connections.

- `list`: Nodes, which are double connected by `next` and `previous` fields.
- `field`: Connections to nodes by other fields than `next` and `previous` fields, e. g. `head`, `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

In this section lists some examples of the `nodetree` output.

7.1 The node list of the package name

`nodetree`

```
Callback: post_linebreak_filter
-----
├─GLUE subtype: baselineskip, width: 5.06pt
├─HLIST subtype: line, width: 345pt, depth: 0.11pt, height: 6.94pt
├─head:
│   └─LOCAL_PAR
│       └─HLIST subtype: indent, width: 15pt
│           └─GLYPH subtype: 256, char: n, width: 5.56pt, height: 4.42pt
│               └─GLYPH subtype: 256, char: o, width: 5pt, height: 4.48pt, depth: 0.11pt
│                   └─KERN kern: 0.28pt
│                       └─GLYPH subtype: 256, char: d, width: 5.56pt, height: 6.94pt, depth: 0.11pt
│                           properties: {'injections': {'leftkern': 18350.08}}
│                               └─GLYPH subtype: 256, char: e, width: 4.44pt, height: 4.48pt, depth: 0.11pt
│                                   └─DISC subtype: regular, penalty: 50
│                                       └─pre:
│                                           └─GLYPH subtype: 256, char: -, width: 3.33pt, height: 2.45pt
│                                               └─GLYPH subtype: 256, char: t, width: 3.89pt, height: 6.15pt, depth: 0.11pt
│                                                   └─GLYPH subtype: 256, char: r, width: 3.92pt, height: 4.42pt
│                                                       └─GLYPH subtype: 256, char: e, width: 4.44pt, height: 4.48pt, depth: 0.11pt
│                                                           └─GLYPH subtype: 256, char: e, width: 4.44pt, height: 4.48pt, depth: 0.11pt
│                                                               └─PENALTY subtype: linepenalty, penalty: 10000
│                                                                   └─GLUE subtype: parfillskip, stretch: +1fil
│                                                                       └─GLUE subtype: rightskip
└─-----
```

7.2 The node list of a mathematical formula

`$1+2$`

```
Callback: post_linebreak_filter
-----
├─GLUE subtype: baselineskip, width: 5.56pt
├─HLIST subtype: line, width: 345pt, depth: 0.83pt, height: 6.44pt
├─head:
│   └─LOCAL_PAR
│       └─HLIST subtype: indent, width: 15pt
│           └─MATH
│               └─GLYPH subtype: 256, char: 1, width: 5pt, height: 6.44pt
│                   └─GLUE subtype: medmuskip, width: 2.22pt, stretch: 1.11pt, shrink: 2.22pt
│                       └─GLYPH subtype: 256, char: +, width: 7.78pt, height: 5.83pt, depth: 0.83pt
│                           └─PENALTY subtype: noadpenalty, penalty: 700
│                               └─GLUE subtype: medmuskip, width: 2.22pt, stretch: 1.11pt, shrink: 2.22pt
│                                   └─GLYPH subtype: 256, char: 2, width: 5pt, height: 6.44pt
│                                       └─MATH subtype: endmath
└─-----
```

```

└PENALTY subtype: linepenalty, penalty: 10000
└GLUE subtype: parfillskip, stretch: +1fil
└GLUE subtype: rightskip
-----

```

7.3 The node list of the word *Office*

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

```

Callback: post_linebreak_filter
-----
└GLUE subtype: baselineskip, width: 4.95pt
└HLIST subtype: line, width: 345pt, depth: 0.22pt, height: 7.05pt
  └head:
    └LOCAL_PAR
      └HLIST subtype: indent, width: 15pt
        └GLYPH subtype: 256, char: 0, width: 7.78pt, height: 7.05pt, depth: 0.22pt
          └DISC subtype: regular, penalty: 50
            └replace:
              └GLYPH subtype: 258, char: [], width: 8.33pt, height: 7.05pt
                └components:
                  └GLYPH subtype: ghost, char: [], width: 5.83pt, height: 7.05pt
                    └components:
                      └GLYPH char: f, width: 3.06pt, height: 7.05pt
                        └GLYPH char: f, width: 3.06pt, height: 7.05pt
                      └GLYPH char: i, width: 2.78pt, height: 6.57pt
                    └pre:
                      └GLYPH subtype: 256, char: f, width: 3.06pt, height: 7.05pt
                        └GLYPH subtype: 256, char: -, width: 3.33pt, height: 2.45pt
                      └post:
                        └GLYPH subtype: 258, char: fi, width: 5.56pt, height: 7.05pt
                          └components:
                            └GLYPH char: f, width: 3.06pt, height: 7.05pt
                              └GLYPH char: i, width: 2.78pt, height: 6.57pt
                            └GLYPH subtype: 256, char: c, width: 4.44pt, height: 4.48pt, depth: 0.11pt
                              └GLYPH subtype: 256, char: e, width: 4.44pt, height: 4.48pt, depth: 0.11pt
                          └PENALTY subtype: linepenalty, penalty: 10000
                          └GLUE subtype: parfillskip, stretch: +1fil
                          └GLUE subtype: rightskip
                        -----

```

7.4 Node types

This chapter shows some node types in a `nodetree` view.

7.4.1 Type: `hlist(0)` Subtype: `line(1)`

Lorem

```

Callback: post_linebreak_filter
-----

```

```

├─GLUE subtype: baselineskip, width: 5.17pt
├─HLIST subtype: line, width: 345pt, depth: 0.11pt, height: 6.83pt
├─head:
├─LOCAL_PAR
├─HLIST subtype: indent, width: 15pt
├─GLYPH subtype: 256, char: L, width: 6.25pt, height: 6.83pt
├─GLYPH subtype: 256, char: o, width: 5pt, height: 4.48pt, depth: 0.11pt
├─GLYPH subtype: 256, char: r, width: 3.92pt, height: 4.42pt
├─GLYPH subtype: 256, char: e, width: 4.44pt, height: 4.48pt, depth: 0.11pt
├─GLYPH subtype: 256, char: m, width: 8.33pt, height: 4.42pt
├─PENALTY subtype: linepenalty, penalty: 10000
├─GLUE subtype: parfillskip, stretch: +1fil
├─GLUE subtype: rightskip
-----

```

7.4.2 Type: hlist(0) Subtype: box(2)

L\hbox to 40pt{ore}m

```

Callback: post_linebreak_filter
-----
├─GLUE subtype: baselineskip, width: 5.17pt
├─HLIST subtype: line, width: 345pt, depth: 0.11pt, height: 6.83pt
├─head:
├─LOCAL_PAR
├─HLIST subtype: indent, width: 15pt
├─GLYPH subtype: 256, char: L, width: 6.25pt, height: 6.83pt
├─HLIST subtype: box, width: 40pt, depth: 0.11pt, height: 4.48pt
├─head:
├─GLYPH subtype: 256, char: o, width: 5pt, height: 4.48pt, depth: 0.11pt
├─GLYPH subtype: 256, char: r, width: 3.92pt, height: 4.42pt
├─GLYPH subtype: 256, char: e, width: 4.44pt, height: 4.48pt, depth: 0.11pt
├─GLYPH subtype: 256, char: m, width: 8.33pt, height: 4.42pt
├─PENALTY subtype: linepenalty, penalty: 10000
├─GLUE subtype: parfillskip, stretch: +1fil
├─GLUE subtype: rightskip
-----

```

7.4.3 Type: hlist(0) Subtype: indent(3)

\setlength {\parindent }{5cm} I

```

Callback: post_linebreak_filter
-----
├─GLUE subtype: baselineskip, width: 0.18cm
├─HLIST subtype: line, width: 12.13cm, height: 0.24cm
├─head:
├─LOCAL_PAR
├─HLIST subtype: indent, width: 5cm
├─GLYPH subtype: 256, char: I, width: 0.13cm, height: 0.24cm
├─PENALTY subtype: linepenalty, penalty: 10000
├─GLUE subtype: parfillskip, stretch: +1fil
├─GLUE subtype: rightskip
-----

```



```

├─HLIST subtype: box, width: 10mm, depth: 2mm, height: 2mm
├─┬head:
├─└─RULE width: 10mm, depth: 2mm, height: 2mm
├─PENALTY subtype: linepenalty, penalty: 10000
├─GLUE subtype: parfillskip, stretch: +1fil
├─GLUE subtype: rightskip
-----

```

7.4.6 Type: mark(4)

\mark {Lorem}.

```

Callback: pre_output_filter
- size: 36044800
- maxdepth: 327680
- groupcode: output
- packtype: exactly
- direction: TLT
-----
├─WHATSIT subtype: write, stream: 129, data:
├─MARK mark: table: 0x6dcae20
├─GLUE subtype: topskip, width: 3.14mm
├─HLIST subtype: line, width: 121.25mm, height: 0.37mm
├─┬head:
├─└─LOCAL_PAR
├─└─HLIST subtype: indent, width: 5.27mm
├─└─GLYPH subtype: 256, char: ., width: 0.98mm, height: 0.37mm
├─└─PENALTY subtype: linepenalty, penalty: 10000
├─└─GLUE subtype: parfillskip, stretch: +1fil
├─└─GLUE subtype: rightskip
├─GLUE stretch: +1fil
-----
Callback: pre_output_filter
- size: 36044800
- maxdepth: 327680
- groupcode: output
- packtype: exactly
- direction: TLT
-----
├─WHATSIT subtype: write, stream: 129, data:
├─GLUE subtype: topskip, width: 3.51mm
├─VLIST
-----
Callback: pre_output_filter
- size: 36044800
- maxdepth: 327680
- groupcode: output
- packtype: exactly
- direction: TLT
-----
├─WHATSIT subtype: write, stream: 129, data:
├─GLUE subtype: topskip, width: 3.51mm
├─HLIST width: 121.25mm

```

```
└─GLUE stretch: +1fill
```

7.4.7 Type: disc(7) Subtype: discretionary(0)

L\~0\L

```
Callback: post_linebreak_filter
-----
└─GLUE subtype: baselineskip, width: 4.95pt
└─HLIST subtype: line, width: 345pt, depth: 0.22pt, height: 7.05pt
  └─head:
    └─LOCAL_PAR
      └─HLIST subtype: indent, width: 15pt
        └─GLYPH subtype: 256, char: L, width: 6.25pt, height: 6.83pt
          └─DISC subtype: explicit, penalty: 50
            └─pre:
              └─GLYPH subtype: 256, char: -, width: 3.33pt, height: 2.45pt
                └─GLYPH subtype: 256, char: 0, width: 7.78pt, height: 7.05pt, depth: 0.22pt
                  └─DISC subtype: explicit, penalty: 50
                    └─pre:
                      └─GLYPH subtype: 256, char: -, width: 3.33pt, height: 2.45pt
                        └─GLYPH subtype: 256, char: L, width: 6.25pt, height: 6.83pt
                          └─PENALTY subtype: linepenalty, penalty: 10000
                            └─GLUE subtype: parfillskip, stretch: +1fil
                              └─GLUE subtype: rightskip
```

7.4.8 Type: disc(7) Subtype: regular(3)

Office

```
Callback: post_linebreak_filter
-----
└─GLUE subtype: baselineskip, width: 4.95pt
└─HLIST subtype: line, width: 345pt, depth: 0.22pt, height: 7.05pt
  └─head:
    └─LOCAL_PAR
      └─HLIST subtype: indent, width: 15pt
        └─GLYPH subtype: 256, char: 0, width: 7.78pt, height: 7.05pt, depth: 0.22pt
          └─DISC subtype: regular, penalty: 50
            └─post:
              └─GLYPH subtype: 258, char: fi, width: 5.56pt, height: 7.05pt
                └─components:
                  └─GLYPH char: f, width: 3.06pt, height: 7.05pt
                    └─GLYPH char: i, width: 2.78pt, height: 6.57pt
            └─replace:
              └─GLYPH subtype: 258, char: [], width: 8.33pt, height: 7.05pt
                └─components:
                  └─GLYPH subtype: ghost, char: [], width: 5.83pt, height: 7.05pt
                    └─components:
                      └─GLYPH char: f, width: 3.06pt, height: 7.05pt
                        └─GLYPH char: f, width: 3.06pt, height: 7.05pt
```

```

| ||      └─GLYPH char: i, width: 2.78pt, height: 6.57pt
|   └─pre:
|       └─GLYPH subtype: 256, char: f, width: 3.06pt, height: 7.05pt
|           └─GLYPH subtype: 256, char: -, width: 3.33pt, height: 2.45pt
|       └─GLYPH subtype: 256, char: c, width: 4.44pt, height: 4.48pt, depth: 0.11pt
|       └─GLYPH subtype: 256, char: e, width: 4.44pt, height: 4.48pt, depth: 0.11pt
|       └─PENALTY subtype: linepenalty, penalty: 10000
|       └─GLUE subtype: parfillskip, stretch: +1fil
|       └─GLUE subtype: rightskip
|-----

```

7.4.9 Type: whatsit(8) Subtype: pdfaction(22)

```

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

```

```

Callback: post_linebreak_filter
|-----
| └─GLUE subtype: baselineskip, width: 5.06pt
|   └─HLIST subtype: line, width: 345pt, depth: 2.29pt, height: 6.94pt
|     └─head:
|         └─LOCAL_PAR
|             └─HLIST subtype: indent, width: 15pt
|                 └─WHATSIT subtype: pdf_start_link, width: -16384pt, depth: -16384pt, height: -16384pt, objnum: 4
|                     └─action:
|                         └─WHATSIT subtype: pdf_action, action_type: 3, file: , data: /Subtype/Link/A<</Type/Action/S
| └─MATH
|   └─GLYPH subtype: 256, char: h, width: 5.25pt, height: 6.11pt
|   └─GLYPH subtype: 256, char: t, width: 5.25pt, height: 5.54pt, depth: 0.06pt
|   └─GLYPH subtype: 256, char: t, width: 5.25pt, height: 5.54pt, depth: 0.06pt
|   └─GLYPH subtype: 256, char: p, width: 5.25pt, height: 4.37pt, depth: 2.22pt
|   └─GLUE subtype: thickmuskip
|   └─GLYPH subtype: 256, char: :, width: 5.25pt, height: 4.31pt
|   └─PENALTY subtype: noadpenalty, penalty: 500
|   └─GLUE subtype: thickmuskip
|   └─GLYPH subtype: 256, char: /, width: 5.25pt, height: 6.94pt, depth: 0.83pt
|   └─GLYPH subtype: 256, char: l, width: 5.25pt, height: 6.11pt
|   └─GLYPH subtype: 256, char: u, width: 5.25pt, height: 4.31pt, depth: 0.06pt
|   └─GLYPH subtype: 256, char: a, width: 5.25pt, height: 4.4pt, depth: 0.06pt
|   └─GLYPH subtype: 256, char: t, width: 5.25pt, height: 5.54pt, depth: 0.06pt
|   └─GLYPH subtype: 256, char: e, width: 5.25pt, height: 4.4pt, depth: 0.06pt
|   └─GLYPH subtype: 256, char: x, width: 5.25pt, height: 4.31pt
|   └─GLUE subtype: medmuskip
|   └─GLYPH subtype: 256, char: ., width: 5.25pt, height: 1.25pt
|   └─PENALTY subtype: noadpenalty, penalty: 700
|   └─GLUE subtype: medmuskip
|   └─GLYPH subtype: 256, char: o, width: 5.25pt, height: 4.4pt, depth: 0.06pt
|   └─GLYPH subtype: 256, char: r, width: 5.25pt, height: 4.37pt
|   └─GLYPH subtype: 256, char: g, width: 5.25pt, height: 4.42pt, depth: 2.29pt
|   └─MATH subtype: endmath

```

```

├─WHATSIT subtype: pdf_end_link
├─PENALTY subtype: linepenalty, penalty: 10000
├─GLUE subtype: parfillskip, stretch: +1fil
└─GLUE subtype: rightskip
-----

```

7.4.10 Type: whatsit(8) Subtype: pdfcolorstack(28)

```

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

```

```

Callback: post_linebreak_filter
-----
├─GLUE subtype: baselineskip, width: 5.17pt
└─HLIST subtype: line, width: 345pt, depth: 0.11pt, height: 6.83pt
  └─head:
    └─LOCAL_PAR
      └─HLIST subtype: indent, width: 15pt
        └─GLYPH subtype: 256, char: L, width: 6.25pt, height: 6.83pt
          └─GLYPH subtype: 256, char: o, width: 5pt, height: 4.48pt, depth: 0.11pt
            └─WHATSIT subtype: pdf_colorstack, data: 1 0 0 rg 1 0 0 RG
              └─GLYPH subtype: 256, char: r, width: 3.92pt, height: 4.42pt
                └─GLYPH subtype: 256, char: e, width: 4.44pt, height: 4.48pt, depth: 0.11pt
                  └─WHATSIT subtype: pdf_colorstack, data:
                    └─GLYPH subtype: 256, char: m, width: 8.33pt, height: 4.42pt
                      └─GLYPH subtype: 256, char: ., width: 2.78pt, height: 1.06pt
                        └─PENALTY subtype: linepenalty, penalty: 10000
                          └─GLUE subtype: parfillskip, stretch: +1fil
                            └─GLUE subtype: rightskip
                              -----

```

7.4.11 Type: glue(12) Subtype: baselineskip(2)

```

\baselineskip=5cm Lorem Lorem

```

```

Callback: post_linebreak_filter
-----
├─GLUE subtype: baselineskip, width: 4.76cm
└─HLIST subtype: line, width: 12.13cm, depth: 0cm, height: 0.24cm
  └─head:
    └─LOCAL_PAR
      └─HLIST subtype: indent, width: 0.53cm
        └─GLYPH subtype: 256, char: L, width: 0.22cm, height: 0.24cm
          └─GLYPH subtype: 256, char: o, width: 0.18cm, height: 0.16cm, depth: 0cm
            └─GLYPH subtype: 256, char: r, width: 0.14cm, height: 0.16cm
              └─GLYPH subtype: 256, char: e, width: 0.16cm, height: 0.16cm, depth: 0cm
                └─GLYPH subtype: 256, char: m, width: 0.29cm, height: 0.16cm
                  └─PENALTY subtype: linepenalty, penalty: 10000
                    └─GLUE subtype: parfillskip, stretch: +1fil
                      └─GLUE subtype: rightskip
                        -----

```

```

-----
Callback: post_linebreak_filter
-----
├─GLUE subtype: baselineskip, width: 4.76cm
└─HLIST subtype: line, width: 12.13cm, depth: 0cm, height: 0.24cm
  └─head:
    ├─LOCAL_PAR
    │ └─HLIST subtype: indent, width: 0.53cm
    │   └─GLYPH subtype: 256, char: L, width: 0.22cm, height: 0.24cm
    │     └─GLYPH subtype: 256, char: o, width: 0.18cm, height: 0.16cm, depth: 0cm
    │       └─GLYPH subtype: 256, char: r, width: 0.14cm, height: 0.16cm
    │         └─GLYPH subtype: 256, char: e, width: 0.16cm, height: 0.16cm, depth: 0cm
    │           └─GLYPH subtype: 256, char: m, width: 0.29cm, height: 0.16cm
    │             └─PENALTY subtype: linepenalty, penalty: 10000
    │               └─GLUE subtype: parfillskip, stretch: +1fil
    │                 └─GLUE subtype: rightskip
    
```

7.4.12 Type: glue(12) Subtype: parskip(3)

\parskip=5cm Lorem Lorem

```

-----
Callback: post_linebreak_filter
-----
├─GLUE subtype: baselineskip, width: 5.17pt
└─HLIST subtype: line, width: 345pt, depth: 0.11pt, height: 6.83pt
  └─head:
    ├─LOCAL_PAR
    │ └─HLIST subtype: indent, width: 15pt
    │   └─GLYPH subtype: 256, char: L, width: 6.25pt, height: 6.83pt
    │     └─GLYPH subtype: 256, char: o, width: 5pt, height: 4.48pt, depth: 0.11pt
    │       └─GLYPH subtype: 256, char: r, width: 3.92pt, height: 4.42pt
    │         └─GLYPH subtype: 256, char: e, width: 4.44pt, height: 4.48pt, depth: 0.11pt
    │           └─GLYPH subtype: 256, char: m, width: 8.33pt, height: 4.42pt
    │             └─PENALTY subtype: linepenalty, penalty: 10000
    │               └─GLUE subtype: parfillskip, stretch: +1fil
    │                 └─GLUE subtype: rightskip
    
```

```

-----
Callback: post_linebreak_filter
-----
├─GLUE subtype: baselineskip, width: 5.06pt
└─HLIST subtype: line, width: 345pt, depth: 0.11pt, height: 6.83pt
  └─head:
    ├─LOCAL_PAR
    │ └─HLIST subtype: indent, width: 15pt
    │   └─GLYPH subtype: 256, char: L, width: 6.25pt, height: 6.83pt
    │     └─GLYPH subtype: 256, char: o, width: 5pt, height: 4.48pt, depth: 0.11pt
    │       └─GLYPH subtype: 256, char: r, width: 3.92pt, height: 4.42pt
    │         └─GLYPH subtype: 256, char: e, width: 4.44pt, height: 4.48pt, depth: 0.11pt
    │           └─GLYPH subtype: 256, char: m, width: 8.33pt, height: 4.42pt
    │             └─PENALTY subtype: linepenalty, penalty: 10000
    │               └─GLUE subtype: parfillskip, stretch: +1fil
    │                 └─GLUE subtype: rightskip
    
```

7.4.13 Type: glue(12) Subtype: spaceskip(13)

`\spaceskip =5cm a a`

```
Callback: post_linebreak_filter
-----
├─GLUE subtype: baselineskip, width: 7.52pt
├─HLIST subtype: line, width: 345pt, depth: 0.11pt, height: 4.48pt
├─head:
│   └─LOCAL_PAR
│       └─HLIST subtype: indent, width: 15pt
│           └─GLYPH subtype: 256, char: a, width: 5pt, height: 4.48pt, depth: 0.11pt
│               └─GLUE subtype: spaceskip, width: 142.26pt
│                   └─GLYPH subtype: 256, char: a, width: 5pt, height: 4.48pt, depth: 0.11pt
│                       └─PENALTY subtype: linepenalty, penalty: 10000
│                           └─GLUE subtype: parfillskip, stretch: +1fil
│                               └─GLUE subtype: rightskip
└─
```

7.4.14 Type: glue(12) Subtype: leaders(100)

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

```
Callback: post_linebreak_filter
-----
├─GLUE subtype: baselineskip, width: 7.52pt
├─HLIST subtype: line, width: 345pt, depth: 0.11pt, height: 4.48pt
├─head:
│   └─LOCAL_PAR
│       └─HLIST subtype: indent, width: 15pt
│           └─GLYPH subtype: 256, char: a, width: 5pt, height: 4.48pt, depth: 0.11pt
│               └─GLUE subtype: spaceskip, width: 3.33pt, stretch: 1.66pt, shrink: 1.11pt
│                   └─GLUE subtype: leaders, stretch: +1fill
│                       └─leader:
│                           └─HLIST subtype: box, width: 10.55pt, height: 1.06pt
│                               └─head:
│                                   └─GLUE subtype: spaceskip, width: 3.33pt, stretch: 1.66pt, shrink: 1.11pt
│                                       └─GLYPH subtype: 256, char: ., width: 2.78pt, height: 1.06pt
│                                           └─GLUE subtype: spaceskip, width: 4.44pt, stretch: 4.99pt, shrink: 0.37pt
├─KERN subtype: userkern
├─GLYPH subtype: 256, char: a, width: 5pt, height: 4.48pt, depth: 0.11pt
├─PENALTY subtype: linepenalty, penalty: 10000
├─GLUE subtype: parfillskip, stretch: +1fil
└─GLUE subtype: rightskip
└─
```

7.4.15 Type: glue(12) Subtype: cleaders(101)

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

```

Callback: post_linebreak_filter
-----
├─GLUE subtype: baselineskip, width: 7.52pt
├─HLIST subtype: line, width: 345pt, depth: 0.11pt, height: 4.48pt
├─head:
│   ├─LOCAL_PAR
│   │   └─HLIST subtype: indent, width: 15pt
│   │       └─GLYPH subtype: 256, char: a, width: 5pt, height: 4.48pt, depth: 0.11pt
│   │           └─GLUE subtype: spaceskip, width: 3.33pt, stretch: 1.66pt, shrink: 1.11pt
│   │               └─GLUE subtype: cleaders, stretch: +1fill
│   │                   └─leader:
│   │                       └─HLIST subtype: box, width: 10.55pt, height: 1.06pt
│   │                           └─head:
│   │                               └─GLUE subtype: spaceskip, width: 3.33pt, stretch: 1.66pt, shrink: 1.11pt
│   │                                   └─GLYPH subtype: 256, char: ., width: 2.78pt, height: 1.06pt
│   │                                       └─GLUE subtype: spaceskip, width: 4.44pt, stretch: 4.99pt, shrink: 0.37pt
│   └─KERN subtype: userkern
├─GLYPH subtype: 256, char: a, width: 5pt, height: 4.48pt, depth: 0.11pt
├─PENALTY subtype: linepenalty, penalty: 10000
├─GLUE subtype: parfillskip, stretch: +1fil
└─GLUE subtype: rightskip
-----

```

7.4.16 Type: glue(12) Subtype: xleaders(102)

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

```

Callback: post_linebreak_filter
-----
├─GLUE subtype: baselineskip, width: 7.52pt
├─HLIST subtype: line, width: 345pt, depth: 0.11pt, height: 4.48pt
├─head:
│   ├─LOCAL_PAR
│   │   └─HLIST subtype: indent, width: 15pt
│   │       └─GLYPH subtype: 256, char: a, width: 5pt, height: 4.48pt, depth: 0.11pt
│   │           └─GLUE subtype: spaceskip, width: 3.33pt, stretch: 1.66pt, shrink: 1.11pt
│   │               └─GLUE subtype: xleaders, stretch: +1fill
│   │                   └─leader:
│   │                       └─HLIST subtype: box, width: 10.55pt, height: 1.06pt
│   │                           └─head:
│   │                               └─GLUE subtype: spaceskip, width: 3.33pt, stretch: 1.66pt, shrink: 1.11pt
│   │                                   └─GLYPH subtype: 256, char: ., width: 2.78pt, height: 1.06pt
│   │                                       └─GLUE subtype: spaceskip, width: 4.44pt, stretch: 4.99pt, shrink: 0.37pt
│   └─KERN subtype: userkern
├─GLYPH subtype: 256, char: a, width: 5pt, height: 4.48pt, depth: 0.11pt
├─PENALTY subtype: linepenalty, penalty: 10000
├─GLUE subtype: parfillskip, stretch: +1fil
└─GLUE subtype: rightskip
-----

```

7.4.17 Type: glue(12) Subtype: gleaders(102)

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

```

Callback: post_linebreak_filter
-----
├─GLUE subtype: baselineskip, width: 7.52pt
├─HLIST subtype: line, width: 345pt, depth: 0.11pt, height: 4.48pt
├─head:
│   └─LOCAL_PAR
│       └─HLIST subtype: indent, width: 15pt
│           └─GLYPH subtype: 256, char: a, width: 5pt, height: 4.48pt, depth: 0.11pt
│               └─GLUE subtype: spaceskip, width: 3.33pt, stretch: 1.66pt, shrink: 1.11pt
│                   └─GLUE subtype: gleaders, stretch: +1fill
│                       └─leader:
│                           └─HLIST subtype: box, width: 10.55pt, height: 1.06pt
│                               └─head:
│                                   └─GLUE subtype: spaceskip, width: 3.33pt, stretch: 1.66pt, shrink: 1.11pt
│                                       └─GLYPH subtype: 256, char: ., width: 2.78pt, height: 1.06pt
│                                           └─GLUE subtype: spaceskip, width: 4.44pt, stretch: 4.99pt, shrink: 0.37pt
│                                               └─KERN subtype: userkern
│                                                   └─GLYPH subtype: 256, char: a, width: 5pt, height: 4.48pt, depth: 0.11pt
│                                                       └─PENALTY subtype: linepenalty, penalty: 10000
│                                                           └─GLUE subtype: parfillskip, stretch: +1fil
│                                                               └─GLUE subtype: rightskip
└─GLUE subtype: baselineskip, width: 7.52pt

```

7.4.18 Type: kern(13) Subtype: userkern(0)

a\kern 2pt

```

Callback: post_linebreak_filter
-----
├─GLUE subtype: baselineskip, width: 7.52pt
├─HLIST subtype: line, width: 345pt, depth: 0.11pt, height: 4.48pt
├─head:
│   └─LOCAL_PAR
│       └─HLIST subtype: indent, width: 15pt
│           └─GLYPH subtype: 256, char: a, width: 5pt, height: 4.48pt, depth: 0.11pt
│               └─KERN subtype: userkern, kern: 2pt
│                   └─PENALTY subtype: linepenalty, penalty: 10000
│                       └─GLUE subtype: parfillskip, stretch: +1fil
│                           └─GLUE subtype: rightskip
└─GLUE subtype: baselineskip, width: 7.52pt

```

7.4.19 Type: kern(13) Subtype: fontkern(1)

Ve

```

Callback: post_linebreak_filter
-----
├─GLUE subtype: baselineskip, width: 5.17pt
├─HLIST subtype: line, width: 345pt, depth: 0.22pt, height: 6.83pt
├─head:
│   └─LOCAL_PAR
│       └─HLIST subtype: indent, width: 15pt
│           └─GLYPH subtype: 256, char: V, width: 7.5pt, height: 6.83pt, depth: 0.22pt
└─GLUE subtype: baselineskip, width: 5.17pt

```

```

├KERN kern: -0.83pt
├GLYPH subtype: 256, char: e, width: 4.44pt, height: 4.48pt, depth: 0.11pt
  └properties: [['injections'] = [['leftkern'] = -54394.88]]
├PENALTY subtype: linepenalty, penalty: 10000
├GLUE subtype: parfillskip, stretch: +1fil
└GLUE subtype: rightskip
-----

```

7.4.20 Type: kern(13) Subtype: accentkern(2)

\`{a}

```

Callback: post_linebreak_filter
-----
├GLUE subtype: baselineskip, width: 5.02pt
├HLIST subtype: line, width: 345pt, depth: 0.11pt, height: 6.98pt
  └head:
    └LOCAL_PAR
      └HLIST subtype: indent, width: 15pt
        └GLYPH subtype: 256, char: à, width: 5pt, height: 6.98pt, depth: 0.11pt
          └PENALTY subtype: linepenalty, penalty: 10000
            └GLUE subtype: parfillskip, stretch: +1fil
              └GLUE subtype: rightskip
-----

```

7.4.21 Type: kern(13) Subtype: italiccorrection(3)

\textit {L}\OL

```

Callback: post_linebreak_filter
-----
├GLUE subtype: baselineskip, width: 4.95pt
├HLIST subtype: line, width: 345pt, depth: 0.22pt, height: 7.05pt
  └head:
    └LOCAL_PAR
      └HLIST subtype: indent, width: 15pt
        └GLYPH subtype: 256, char: L, width: 6.27pt, height: 6.83pt
          └KERN subtype: italiccorrection, kern: 0.17pt
            └GLYPH subtype: 256, char: O, width: 7.78pt, height: 7.05pt, depth: 0.22pt
              └GLYPH subtype: 256, char: L, width: 6.25pt, height: 6.83pt
                └PENALTY subtype: linepenalty, penalty: 10000
                  └GLUE subtype: parfillskip, stretch: +1fil
                    └GLUE subtype: rightskip
-----

```

7.4.22 Type: penalty(14)

L \penalty 23 OL

```

Callback: post_linebreak_filter
-----
├GLUE subtype: baselineskip, width: 4.95pt
├HLIST subtype: line, width: 345pt, depth: 0.22pt, height: 7.05pt

```

```

└─head:
  └─LOCAL_PAR
  └─HLIST subtype: indent, width: 15pt
  └─GLYPH subtype: 256, char: L, width: 6.25pt, height: 6.83pt
  └─GLUE subtype: spaceskip, width: 3.33pt, stretch: 1.66pt, shrink: 1.11pt
  └─PENALTY penalty: 23
  └─GLYPH subtype: 256, char: 0, width: 7.78pt, height: 7.05pt, depth: 0.22pt
  └─GLYPH subtype: 256, char: L, width: 6.25pt, height: 6.83pt
  └─PENALTY subtype: linepenalty, penalty: 10000
  └─GLUE subtype: parfillskip, stretch: +1fil
  └─GLUE subtype: rightskip
-----

```

7.4.23 Type: glyph(29)

abc

```

Callback: post_linebreak_filter
-----
└─GLUE subtype: baselineskip, width: 5.06pt
└─HLIST subtype: line, width: 345pt, depth: 0.11pt, height: 6.94pt
  └─head:
    └─LOCAL_PAR
    └─HLIST subtype: indent, width: 15pt
    └─GLYPH subtype: 256, char: a, width: 5pt, height: 4.48pt, depth: 0.11pt
    └─GLYPH subtype: 256, char: b, width: 5.56pt, height: 6.94pt, depth: 0.11pt
    └─KERN kern: 0.28pt
    └─GLYPH subtype: 256, char: c, width: 4.44pt, height: 4.48pt, depth: 0.11pt
    └─properties: {'injections' = {'leftkern' = 18350.08}}
    └─PENALTY subtype: linepenalty, penalty: 10000
    └─GLUE subtype: parfillskip, stretch: +1fil
    └─GLUE subtype: rightskip
-----

```

7.4.24 Type: attribute(38)

{\attribute 0=1 A}

```

Callback: post_linebreak_filter
-----
└─GLUE subtype: baselineskip, width: 4.84pt
└─HLIST subtype: line, width: 345pt, height: 7.16pt
  └─head:
    └─LOCAL_PAR attr: 0=1
    └─HLIST subtype: indent, width: 15pt, attr: 0=1
    └─GLYPH subtype: 256, char: A, width: 7.5pt, height: 7.16pt, attr: 0=1
    └─PENALTY subtype: linepenalty, penalty: 10000
    └─GLUE subtype: parfillskip, stretch: +1fil
    └─GLUE subtype: rightskip
-----

```

7.4.25 Type: attributelist(40)

{\attribute 0=1 A}

```
Callback: post_linebreak_filter
-----
├─GLUE subtype: baselineskip, width: 4.84pt
├─HLIST subtype: line, width: 345pt, height: 7.16pt
├─head:
│   ├──LOCAL_PAR attr: 0=1
│   ├──HLIST subtype: indent, width: 15pt, attr: 0=1
│   ├──GLYPH subtype: 256, char: A, width: 7.5pt, height: 7.16pt, attr: 0=1
│   ├──PENALTY subtype: linepenalty, penalty: 10000
│   ├──GLUE subtype: parfillskip, stretch: +1fil
│   └─GLUE subtype: rightskip
-----
```

8 Implementation

8.1 The file nodetree.tex

```
26 \directlua{
27   nodetree = require('nodetree')
28 }
```

`\NodetreeSetOption`

```
29 \def\NodetreeSetOption[#1]#2{
30   \directlua{
31     nodetree.set_option('#1', '#2')
32   }
33 }
34 \let\nodetreeroption\NodetreeSetOption
```

`\NodetreeResetOption`

```
35 \def\NodetreeResetOption#1{
36   \NodetreeSetOption[#1]{%
37     \directlua{
38       tex.print(nodetree.get_default_option('#1'))
39     }%
40   }%
41 }
```

`\NodetreeReset`

```
42 \def\NodetreeReset{
43   \NodetreeResetOption{callback}
44   \NodetreeResetOption{channel}
45   \NodetreeResetOption{color}
46   \NodetreeResetOption{decimalplaces}
47   \NodetreeResetOption{unit}
48   \NodetreeResetOption{verbosity}
49 }
50 \let\nodetreereset\NodetreeReset
```

`\NodetreeRegisterCallback`

```
51 \def\NodetreeRegisterCallback#1{
52   \directlua{
53     nodetree.set_option('callback', '#1')
54     nodetree.register_callbacks()
55   }
56 }
57 \let\nodetreeregister\NodetreeRegisterCallback
```

`\NodetreeUnregisterCallback`

```
58 \def\NodetreeUnregisterCallback#1{
59   \directlua{
60     nodetree.set_option('callback', '#1')
61     nodetree.unregister_callbacks()
62   }
63 }
64 \let\nodetreeunregister\NodetreeUnregisterCallback
```

8.2 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[1]{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
```

```

52 \NeedsTeXFormat{LaTeX2e}[1994/06/01]
53 \ProvidesPackage{nodetree-embed}
54 [2020/10/23 v2.2 Embed node trees into a LaTeX document]

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

56 \input{nodetree}

57 \RequirePackage{kvoptions}
58 \SetupKeyvalOptions{
59 family=NTE,
60 prefix=NTEK@
61 }

62 \directlua{
63 nodetree = require('nodetree')
64 nodetree.check_shell_escape()
65 }

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

67 \DeclareStringOption[1]{verbosity}
68 \define@key{NTE}{verbosity}[]{\NodetreeSetOption[verbosity]{#1}}

69 \DeclareStringOption[colored]{color}
70 \define@key{NTE}{color}[]{\NodetreeSetOption[color]{#1}}

71 \DeclareStringOption[1]{unit}
72 \define@key{NTE}{unit}[]{\NodetreeSetOption[unit]{#1}}

73 \DeclareStringOption[1]{decimalplaces}
74 \define@key{NTE}{decimalplaces}[]{\NodetreeSetOption[decimalplaces]{#1}}

75 \DeclareStringOption[monokaisoda]{theme}

76 \DeclareStringOption[dark]{thememode}

77 \DeclareStringOption[Ubuntu Mono]{font}

78 \DeclareStringOption[\footnotesize]{fontsize}

79 \DeclareBoolOption{showmarkup}

80 \ProcessKeyvalOptions{NTE}

\NTE@colors

81 \ExplSyntaxOn
82 \def\NTE@colors{

```

```

83 \str_case_e:nn{\NTEK@theme}{
84   {bwdark}{
85     \definecolor{NTEblack}{gray}{0}
86     \definecolor{NTEred}{gray}{1}
87     \definecolor{NTEgreen}{gray}{1}
88     \definecolor{NTEyellow}{gray}{1}
89     \definecolor{NTEblue}{gray}{1}
90     \definecolor{NTEmagenta}{gray}{1}
91     \definecolor{NTEcyan}{gray}{1}
92     \definecolor{NTEwhite}{gray}{1}
93     \definecolor{NTEblackbright}{gray}{0}
94     \definecolor{NTEredbright}{gray}{1}
95     \definecolor{NTEgreenbright}{gray}{1}
96     \definecolor{NTEyellowbright}{gray}{1}
97     \definecolor{NTEbluebright}{gray}{1}
98     \definecolor{NTEmagentabright}{gray}{1}
99     \definecolor{NTEcyanbright}{gray}{1}
100    \definecolor{NTEwhitebright}{gray}{1}
101   }
102   {bwlight}{
103     \definecolor{NTEblack}{gray}{0}
104     \definecolor{NTEred}{gray}{0}
105     \definecolor{NTEgreen}{gray}{0}
106     \definecolor{NTEyellow}{gray}{0}
107     \definecolor{NTEblue}{gray}{0}
108     \definecolor{NTEmagenta}{gray}{0}
109     \definecolor{NTEcyan}{gray}{0}
110     \definecolor{NTEwhite}{gray}{1}
111     \definecolor{NTEblackbright}{gray}{0}
112     \definecolor{NTEredbright}{gray}{0}
113     \definecolor{NTEgreenbright}{gray}{0}
114     \definecolor{NTEyellowbright}{gray}{0}
115     \definecolor{NTEbluebright}{gray}{0}
116     \definecolor{NTEmagentabright}{gray}{0}
117     \definecolor{NTEcyanbright}{gray}{0}
118     \definecolor{NTEwhitebright}{gray}{1}
119   }
120   {monokaisoda}{
121     \definecolor{NTEblack}{HTML}{1a1a1a}
122     \definecolor{NTEred}{HTML}{f4005f}
123     \definecolor{NTEgreen}{HTML}{98e024}
124     \definecolor{NTEyellow}{HTML}{fa8419}
125     \definecolor{NTEblue}{HTML}{9d65ff}
126     \definecolor{NTEmagenta}{HTML}{f4005f}
127     \definecolor{NTEcyan}{HTML}{58d1eb}
128     \definecolor{NTEwhite}{HTML}{c4c5b5}
129     \definecolor{NTEblackbright}{HTML}{625e4c}
130     \definecolor{NTEredbright}{HTML}{f4005f}
131     \definecolor{NTEgreenbright}{HTML}{98e024}
132     \definecolor{NTEyellowbright}{HTML}{e0d561}

```

```

133     \definecolor{NTEbluebright}{HTML}{9d65ff}
134     \definecolor{NTEmagentabright}{HTML}{f4005f}
135     \definecolor{NTEcyanbright}{HTML}{58d1eb}
136     \definecolor{NTEwhitebright}{HTML}{f6f6ef}
137   }
138 }
139 \str_case_e:nn{\NTEK@thememode}{
140   {dark}{
141     \definecolor{NTEbackground}{named}{NTEblack}
142     \definecolor{NTEfont}{named}{NTEwhitebright}
143   }
144   {light}{
145     \definecolor{NTEbackground}{named}{NTEwhitebright}
146     \definecolor{NTEfont}{named}{NTEblack}
147   }
148 }
149 }
150 \ExplSyntaxOff

```

`\NTE@fonts`

```

151 \def\NTE@fonts{
152   \bfseries%
153   \NTE@fontsize%
154   \setmonofont{\NTEK@font}%
155   \ttfamily%
156   \setlength{\parindent}{0pt}%
157   \setlength{\parskip}{-0.9pt}%
158 }

```

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

```

159 \providecommand{\NodetreeSet}[1]{%
160   \setkeys{NTE}{#1}%
161 }

162 \newenvironment{NodetreeEmbedView}[1][]{
163   \setkeys{NTE}{#1}
164   \NTE@colors
165   \begin{mdframed}[
166     linecolor=black,
167     backgroundcolor=NTEbackground,
168     fontcolor=NTEfont,
169   ]%
170   \NTE@fonts
171 }{
172   \end{mdframed}%
173 }

```

NodetreeEmbedEnv

```
174 \NewDocumentEnvironment { NodetreeEmbedEnv } { 0{} +b } {
175   \setkeys{NTE}{#1}
176   \ifNTEK@showmarkup
177     \noindent
178     \texttt{\detokenize{#2}}
179   \else
180   \fi
181   \NTE@colors
182   \begin{NodetreeEmbedView}
183     \directlua{
184       nodetree.compile_include('\luaescapestring{\unexpanded{#2}}')
185     }
186   \end{NodetreeEmbedView}
187 }{}
```

\NodetreeEmbedCmd

```
188 \NewDocumentCommand { \NodetreeEmbedCmd } { 0{} +v } {
189   \setkeys{NTE}{#1}
190   \ifNTEK@showmarkup
191     \noindent
192     \texttt{#2}
193   \else
194   \fi
195   \NTE@colors
196   \begin{NodetreeEmbedView}
197     \directlua{
198       nodetree.compile_include('\luaescapestring{\unexpanded{#2}}')
199     }
200   \end{NodetreeEmbedView}
201 }
```

\NodetreeEmbedInput

```
202 \newcommand{\NodetreeEmbedInput}[2] []{
203   \setkeys{NTE}{#1}
204   \begin{NodetreeEmbedView}
205     \input{#2.nttex}
206   \end{NodetreeEmbedView}
207 }
208 \let\nodetreeterminalemulator\NodetreeEmbedInput
```

8.3 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/>

Change History

v0.1	General: Converted to DTX file . . .	36	
v1.0	General: Inital release	36	nodetree-embed.sty for embedding nodetree views into a L ^A T _E X document. * Add support for new node subtype names. * Add support for a new LuaT _E X node callback. * Add support for node properties. * Less verbose representation of node attributes. * Minor tree output adjustments.
v1.1	General: Fix the registration of same callbacks	36	
v1.2	General: Fix difference between README.md in the upload and that from nodetree.dtx . .	36	
v2.0	General: * Switch from lowercase macro names to PascalCase names for better readability. * The Lua code is no longer developed inside the DTX file, instead in a separate file named nodetree.lua. * Add a sub package named		v2.1
			General: * Make the package compatible with the Harfbuzz mode of the luaotfload fontloader. * Print node properties of copied nodes. . . .
			v2.2
			General: * Fix unavailable library error (utf8 not in Lua5.1)

Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.

B		L		<code>\NTE@colors</code>
<code>\begin</code> 165, 182, 196, 204	<code>\let</code> 34, 50, 51, 57, 64, 208	<code>\NTE@fonts</code> . . . <u>151</u> , 170	<code>\NTEK@font</code> 154	<code>\NTEK@font</code> 154
<code>\bfseries</code> 152	<code>\luaescapestring</code> 184, 198	<code>\NTEK@fontsize</code> 153	<code>\NTEK@theme</code> 83	<code>\NTEK@thememode</code> . . . 139
D		N		P
<code>\DeclareBoolOption</code> . 79	<code>\NeedsTeXFormat</code> 52	<code>\NewDocumentCommand</code> 188	<code>\NewDocumentEnvironment</code> 174	<code>\parindent</code> 156
<code>\DeclareStringOption</code> 32, 34, 36, 38, 40, 42, 67, 69, 71, 73, 75–78	<code>\NodetreeEmbedCmd</code> . <u>188</u>	<code>\NewDocumentEnvironment</code> 174	<code>\NodetreeEmbedEnv</code> (en- vironment) .. <u>174</u>	<code>\parskip</code> 157
<code>\define@key</code> . 33, 35, 37, 39, 41, 43, 66, 68, 70, 72, 74	<code>\NodetreeEmbedInput</code> <u>202</u>	<code>\NodetreeEmbedInput</code> <u>202</u>	<code>\nodetreereoption</code> 34	<code>\ProcessKeyvalOptions</code> 44, 80
<code>\definecolor</code> 85–100, 103–118, 121–136, 141, 142, 145, 146	<code>\nodetreeregister</code> .. 57	<code>\nodetreeregister</code> .. 57	<code>\nodetreeregistercallback</code> <u>51</u>	<code>\providecommand</code> . . . 159
<code>\detokenize</code> 178	<code>\NodetreeReset</code> <u>42</u>	<code>\NodetreeRegisterCallback</code> <u>51</u>	<code>\NodetreeReset</code> <u>42</u>	<code>\ProvidesPackage</code> . . . 53
E		<code>\NodetreeResetOption</code> <u>35</u> , 43–48	<code>\NodetreeResetOption</code> <u>35</u> , 43–48	R
<code>\else</code> 179, 193	<code>\NodetreeResetOption</code> <u>35</u> , 43–48	<code>\NodetreeResetOption</code> <u>35</u> , 43–48	<code>\NodetreeResetOption</code> <u>35</u> , 43–48	<code>\RequirePackage</code> 27, 55, 57
<code>\end</code> . 172, 186, 200, 206	<code>\NodetreeSet</code> .. <u>48</u> , <u>159</u>	<code>\NodetreeSet</code> .. <u>48</u> , <u>159</u>	<code>\NodetreeSet</code> .. <u>48</u> , <u>159</u>	S
environments:	<code>\nodetreereset</code> 50	<code>\nodetreereset</code> 50	<code>\nodetreereset</code> 50	<code>\setkeys</code> . . . 49, 160, 163, 175, 189, 203
<code>NodetreeEmbedEnv</code> <u>174</u>	<code>\NodetreeSet</code> .. <u>48</u> , <u>159</u>	<code>\NodetreeSet</code> .. <u>48</u> , <u>159</u>	<code>\NodetreeSet</code> .. <u>48</u> , <u>159</u>	<code>\setlength</code> . . . 156, 157
<code>\ExplSyntaxOff</code> 150	<code>\NodetreeSetOption</code> . . <u>29</u> , 33, 35–37, 39, 41, 43, 66, 68, 70, 72, 74	<code>\NodetreeSetOption</code> . . <u>29</u> , 33, 35–37, 39, 41, 43, 66, 68, 70, 72, 74	<code>\NodetreeSetOption</code> . . <u>29</u> , 33, 35–37, 39, 41, 43, 66, 68, 70, 72, 74	<code>\setmonofont</code> 154
<code>\ExplSyntaxOn</code> 81	<code>\nodetreeterminalemulator</code> 208	<code>\nodetreeterminalemulator</code> 208	<code>\nodetreeterminalemulator</code> 208	<code>\SetupKeyvalOptions</code> 28, 58
F		<code>\nodetreeunregister</code> 64	<code>\nodetreeunregister</code> 64	<code>\str</code> 83, 139
<code>\fi</code> 180, 194	<code>\NodetreeUnregisterCallback</code> <u>58</u>	<code>\NodetreeUnregisterCallback</code> <u>58</u>	<code>\NodetreeUnregisterCallback</code> <u>58</u>	T
<code>\footnotesize</code> 78	<code>\noindent</code> 177, 191	<code>\noindent</code> 177, 191	<code>\noindent</code> 177, 191	<code>\texttt</code> 178, 192
I		U		<code>\ttfamily</code> 155
<code>\ifNTEK@showmarkup</code> 176, 190	<code>\unexpanded</code> .. 184, 198	<code>\unexpanded</code> .. 184, 198	<code>\unexpanded</code> .. 184, 198	
<code>\input</code> 26, 56, 205				