

THE TAGDIRTREE PACKAGE

VO.1.0 2026-09-14

Tagged Directory-trees

Matthias WERNER¹

<https://github.com/werner-matthias/osglecture>

Unlike existing directory-tree packages (`dirtree`, `dirtreex`), `TAGDIRTREE` does not draw the tree as a self-contained graphic that subsequently needs an alt-text description – it sets it as a genuine nested list from the start: every entry is ordinary, readable text, the nesting itself carries the parent-child hierarchy, and each entry additionally records whether it is a file, a directory, or an archive. When `\DocumentMetadata{tagging=on}` is set, this is further exposed to assistive technology via a per-entry alt description (kind, name, note); without tagging it remains an ordinary nested list.

Table of Contents

1	Purpose and usage	1	2	Implementation	2
1.1	Icons	2		Index	3

1 Purpose and usage

The package has no package options. The `tagdirtree` environment frames a tree; `\dtdir`, `\dtfile`, and `\dtarchive` enter directories, files, and archives respectively. `\dtdir` takes the directory's entries as its third argument.

```
\begin{tagdirtree}
  \dtdir{project}{root directory}{
    \dtdir{src}{source code}{
      \dtfile{main.py}{entry point}
      \dtfile{util.py}{}
    }
  }
```

1. matthias.werner@informatik.tu-chemnitz.de

```

\dtfile{README.md}{documentation}
\dtarchive{legacy.zip}{old sources}
}
\end{tagdirtree}

```

The second brace argument (also the second for `\dt{dir}`) is the entry’s note and may be empty. It appears in the text in parentheses and, when set, as part of the alt description.

Directories are set bold, archives italic, files in the body font; entries form an ordinary (via `\enumitem` tightly set) nested list, not a custom drawing – appearance can therefore be adjusted with the usual `\enumitem` means on the `tagdirtree` environment itself.

1.1 Icons

Every entry sets a short, language-independent icon before its name and note – `\tagdirtreefileicon`, `\tagdirtreedirectoryicon`, or `\tagdirtreearchiveicon`, depending on kind. The default is a plain placeholder ([D], [F], [A]) and meant to be overridden; while tagging is active, the icon is marked as a purely decorative artifact, since the entry’s kind already lives entirely in its alt text (see below) – a screen reader should not read it out a second time.

```

\RenewDocumentCommand \tagdirtreedirectoryicon { } { \faFolder }
\RenewDocumentCommand \tagdirtreefileicon { } { \faFile }
\RenewDocumentCommand \tagdirtreearchiveicon { } { \faFileArchive }

```

The presentation of names and notes can likewise be changed without affecting the text used for tagging. The corresponding commands are `\tagdirtreefilename`, `\tagdirtreedirectoryname`, `\tagdirtreearchivename`, and `\tagdirtreeannotation`; each receives the respective text as its only argument.

With `osgstyler`, icons can instead be set in a theme color rather than a fixed one, via the public `\OsgColorName`. (`osgstyler`’s own affix decoration was checked and does *not* work as an attachment point: it is bound to `osgstyler-block`’s own application sites, has no public “apply a single affix standalone” command, and its `symbol=` key has no resolving backend anyway – `content=` would be the only usable option left, with no real benefit over `\OsgColorName` directly.)

```

\RenewDocumentCommand \tagdirtreedirectoryicon { }
{ \textcolor{\OsgColorName{accent}}{[D]} }

```

2 Implementation

To view the implementation description, please uncomment `\OnlyDescription` in `tagdirtree.dtx`.

Index

D

\DocumentMetadata.....	1
\dtarchive.....	1
\dtdir.....	1 f.
\dtfile.....	1

O

\OnlyDescription.....	2
\OsgColorName.....	2

T

tagdirtree (environment).....	1 f.
\tagdirtreeannotation.....	2
\tagdirtreearchiveicon.....	2
\tagdirtreearchivename.....	2
\tagdirtreedirectoryicon.....	2
\tagdirtreedirectoryname.....	2
\tagdirtreefileicon.....	2
\tagdirtreefilename.....	2