

quarto-index empty-level probe

An empty level in the middle of an entry

! separates sub-entry levels, so a value that opens, closes, or consists of one puts a level in the entry that prints no text at all. The LaTeX index tool rejects an entry whose first or middle level is empty — "Illegal null field" — drops it, and still exits cleanly, so the entry disappears from a build that looks fine. An empty level therefore prints nothing and is dropped, in every format, and the entry indexes at the levels that remain.

Only two shapes can be written. A run of ! yields at most one separator, because !! is a literal exclamation mark, so two separators can never sit side by side and a level between two others can never be empty.

A leading empty level, the shape that destroys the entry: cats.

A trailing empty level, which the index tool tolerates by swallowing it, so the two shapes fail in opposite ways and neither is evidence about the other: dogs.

Two empty levels around one that survives, so a mark carrying more than one can be seen to draw a single report naming both of them rather than one report per level, byte-identical to its twin: sub.

An empty level next to the rest of the syntax

A cross-reference whose target is the entry's own surviving level. The target already dropped its empty levels; now the entry does too, so the two spellings meet and the self-reference is reported in every format: owls.

A sort key written across both levels. The key at the empty level goes with it, so the surviving level keeps the key written for it and files under **aardvark** rather than under **mmm**: zebra.

A sort key reaching past an entry whose own depth the empty-level drop reduces, so three depths differ at once: the value is written with two levels, the sort key with three, and the entry indexes at one. The report about the extra sort levels must therefore state which count

it is measuring against rather than leave the number to be read as the depth the entry indexes at. The term is one no other mark here indexes, so its sort key cannot attach to a probe this shape is not about: moles.

The same overreaching sort key with no **entry=** at all. The mark indexes at the one level its visible text makes, a level the author wrote no **entry=** for, so the report must not tell them what their index entry was written with: ferns.

An entry that is nothing but empty levels

Nothing survives the drop, so there is no entry to index and the mark says so rather than claiming no **entry=** was written.

With visible text to fall back on, the mark indexes under that text. Its sort key was written for levels that are all gone, so none of it lands on the level it falls back to, and it is told so: Ferrets.

With no visible text and nothing to fall back on, the mark indexes nothing and is removed: .

A control with no empty level

Two ordinary levels, so a check that the empty ones are gone is told apart from a check that lost every level: wrens.

An entry with no empty level whose sort key reaches past its own depth, marked twice. Restating a level's own text on the way to a deeper level declares nothing for that level, so the first mark's R is positional filler and the second mark's yyy is the key both marks file under — one entry, one key, no rival reported. This shape has no empty level anywhere in it, which is the point: dropping empty levels must not change what a sort key means for an entry that has none: quail and rook.

Index

ferns, [2](#)

Moles, [2](#)

Zebra, [1](#)

Birds

Wrens, [2](#)

Cats, [1](#)

Dogs, [1](#)

Ferrets, [2](#)

Owls, [1](#)

Q

R, [2](#)

Sub, [1](#)