`quartable` test document

colspan · rowspan · midrule · cline · vline · align

Eric Bavu

2026-05-09

Table of contents

1 Features

`quartable` extends Markdown pipe tables with six constructs that Pandoc does not support natively. This document is **self-documenting**: each example shows the markdown source above its rendered output, so the rendered HTML / PDF / Reveal is itself a usage reference.

Markup	Effect
`[text]{cs=N}`	merge `N` consecutive columns
`[text]{rs=N}` + `^`	merge `N` consecutive rows (`^` marks the continuations)
`===`	full-width horizontal separator (booktabs `\midrule`)
`===N-M`, `===N`, `===1,3-5,7`	partial horizontal separator (booktabs `\cmidrule`)
`[text]{cs=N .vl .vr rspan=K}`	vertical line(s) on a colspan boundary (LaTeX only)
`[text]{align=l|c|r}`	per-cell alignment override (overrides Markdown col-spec)

By design `quartable` does **not** draw an automatic head/body separator — write `===` (or `===N-M`) explicitly as the first body row when you want one. This matches booktabs' philosophy of leaving that decision to the author, especially when the header contains column spans or row spans.

2 Midrule

A row whose first cell contains exactly `===` is replaced by a full-width horizontal separator. The very first `===` (sitting just under the header separator `|---|`) acts as the head/body divider; later `===` rows split the body into visually distinct groups.

Code:

```
| Group | Value | Note   |
|-------|------:|--------|
| ===   |       |        |
| A     |    12 | first  |
| A     |     8 | second |
| ===   |       |        |
| B     |    15 | third  |
```

```
| B     |     7 | fourth |
| ===   |       |        |
| C     |    22 | fifth  |
```

Rendered:

Group	Value	Note
A	12	first
A	8	second
B	15	third
B	7	fourth
C	22	fifth

3 Cline (partial horizontal line)

`===N-M` draws a partial line spanning columns `N..M` instead of a full midrule. A bare number `N` is shorthand for `N-N`. Multiple ranges and singletons can be combined with commas: `===1,3-5,7`.

In LaTeX each range emits a booktabs `\cmidrule(l|r|lr){N-M}`; the trim option is chosen automatically based on adjacency to other ranges on the same line, so isolated segments keep the booktabs default `(lr)` look while adjacent segments drop the trim on the adjacent side. In HTML and Reveal, the `border-bottom` of the previous row's cells is set on exactly the targeted columns.

3.1 Single range

Code:

```
| A | B  | C  | D  |
|---|----|----|----|
| === |    |    |    |
| 1 | 2  | 3  | 4  |
| 5 | 6  | 7  | 8  |
| ===2-3 |  |  |    |
| 9 | 10 | 11 | 12 |
```

Rendered:

A	B	C	D
1	2	3	4
5	6	7	8
9	10	11	12

3.2 Multiple ranges

`===1-2,4-5` produces two `\cmidrule` calls (cols 1–2 and 4–5), each at the same horizontal position, with a gap on the unmarked column.

Code:

```
| A | B  | C  | D | E  |
|---|----|----|---|----|
| === |  |    |   |    |
| 1 | 2  | 3  | 4 | 5  |
| ===1-2,4-5 | | | |   |
| 6 | 7  | 8  | 9 | 10 |
```

Rendered:

A	B	C	D	E
1	2	3	4	5
6	7	8	9	10

A	B	C	D	E

3.3 Mixed ranges and single columns

`===1,3-4` mixes a single-column cline with a range. Adaptive trim: the single col on the left gets `(r)` only (right side touches the neighbour), so its segment keeps full width on the left.

Code:

```
| A    | B  | C  | D  |
|------|----|----|----|
| ===  |    |    |    |
| 1    | 2  | 3  | 4  |
| ===1,3-4 | | |    |
| 5    | 6  | 7  | 8  |
```

Rendered:

A	B	C	D
1	2	3	4
5	6	7	8

4 Colspan

4.1 Colspan in the header

`[Measurements]{cs=3}` merges its cell with the next two empty placeholder cells. The `===2-4` row demonstrates a partial cline that underlines exactly the merged span — useful to visually anchor the spanned label above its sub-columns.

Code:

```
| Item      | [Measurements]{cs=3} |        |        |
|-----------|:--------------------:|:------:|-------:|
| ===       |                      |        |        |
|           | T1                   | T2     |     T3 |
| ===2-4    |                      |        |        |
| Sample A  | 42.1                 | 41.8   |  43.0  |
| Sample B  | 38.5                 | 39.1   |  37.9  |
| Sample C  | 51.2                 | 50.8   |  52.1  |
```

Rendered:

Item	Measurements		
	T1	T2	T3
Sample A	42.1	41.8	43.0
Sample B	38.5	39.1	37.9
Sample C	51.2	50.8	52.1

4.2 Colspan in the body

Colspan also works on body rows. Here each group label spans the first two columns; a `===1-2` cline under the label then underlines just the spanned columns, leaving the `Value` column intact.

Code:

```
| Category         | Detail   | Value |
|------------------|:--------:|------:|
| ===              |          |       |
| [Group 1]{cs=2}  |          |  18.3 |
| ===1-2           |          |       |
| Sub-cat. X       | $\alpha$ |  10.1 |
| Sub-cat. Y       | $\beta$  |   8.2 |
| ===              |          |       |
| [Group 2]{cs=2}  |          |  12.7 |
| ===1-2           |          |       |
| Sub-cat. X       | $\alpha$ |   7.4 |
| Sub-cat. Y       | $\beta$  |   5.3 |
```

Rendered:

Category	Detail	Value
Group 1		18.3
Sub-cat. X	α	10.1
Sub-cat. Y	β	8.2
Group 2		12.7
Sub-cat. X	α	7.4
Sub-cat. Y	β	5.3

5 Rowspan

5.1 Rowspan in the body

`[Group A]{rs=3}` merges the cell with the next two rows in the same column. Continuation rows must put a single `^` (or an empty cell) at that column position so the column count of each row stays consistent.

Code:

```
| Group            | Item | Value |
|------------------|:----:|------:|
| ===              |      |       |
| [Group A]{rs=3}  | x1   |   5.2 |
| ^                | x2   |   4.8 |
| ^                | x3   |   6.1 |
| ===              |      |       |
| [Group B]{rs=2}  | y1   |  20.0 |
| ^                | y2   |  19.4 |
```

Rendered:

Group	Item	Value
Group A	x1	5.2
	x2	4.8
	x3	6.1
Group B	y1	20.0
	y2	19.4

5.2 Rowspan combined with colspan

A single cell can span both rows and columns at the same time (`{rs=2 cs=2}`). The continuation row uses `^` placeholders for all spanned columns.

Code:

```
| [Block A]{rs=2 cs=2} |       | Value 1 |
|----------------------|-------|---------|
| ^                    | ^     | Value 2 |
| ===                  |       |         |
| Col 1                | Col 2 | Col 3   |
```

Rendered:

Block A		Value 1 Value 2
Col 1	Col 2	Col 3

6 Midrule + rowspan

`===` separates row groups, each containing its own rowspan. This is the typical academic-table pattern.

Code:

```
| Condition        | Item | Score 1 | Score 2 |
|------------------|:----:|--------:|--------:|
| ===              |      |         |         |
| [Baseline]{rs=3} | x1   |    0.71 |     2.1 |
| ^                | x2   |    0.68 |     2.0 |
| ^                | x3   |    0.74 |     2.3 |
| ===              |      |         |         |
| [Improved]{rs=3} | x1   |    0.89 |     3.4 |
| ^                | x2   |    0.87 |     3.3 |
| ^                | x3   |    0.91 |     3.6 |
```

Rendered:

Condition	Item	Score 1	Score 2
Baseline	x1	0.71	2.1
	x2	0.68	2.0
	x3	0.74	2.3
Improved	x1	0.89	3.4
	x2	0.87	3.3
	x3	0.91	3.6

7 Full table — colspan + rowspan + cline + midrule

A typical multi-feature table. The header pairs `[Method]{rs=2}` with two `cs=2` group labels (`Set A`, `Set B`), and the cline `===2-3,4-5` underlines each group's two sub-columns in one shot. Row groups in the body are separated by `===` midrules.

Code:

```
| [Method]{rs=2}        | [Set A]{cs=2} |           | [Set B]{cs=2} |           |
|-----------------------|:-------------:|:---------:|:-------------:|----------:|
| ===2-3,4-5            |               |           |               |           |
|                       | Score X       | Score Y   | Score X       |   Score Y |
| ===                   |               |           |               |           |
| Baseline              | 0.71          | 2.1       | 0.68          |       1.9 |
| Wiener                | 0.82          | 2.8       | 0.79          |       2.6 |
| ===                   |               |           |               |           |
| GAN-A                 | 0.87          | 3.1       | 0.85          |       2.9 |
| GAN-B                 | 0.91          | 3.5       | 0.89          |       3.3 |
| ===                   |               |           |               |           |
| [**Proposed**]{rs=2}  | **0.93**      | **3.7**   | **0.92**      |   **3.6** |
| ^                     | *(large)*     | *(large)* | *(large)*     | *(large)* |
```

Rendered:

Method	Set A		Set B	
	Score X	Score Y	Score X	Score Y
Baseline	0.71	2.1	0.68	1.9
Wiener	0.82	2.8	0.79	2.6
GAN-A	0.87	3.1	0.85	2.9
GAN-B	0.91	3.5	0.89	3.3
Proposed	**0.93** *(large)*	**3.7** *(large)*	**0.92** *(large)*	**3.6** *(large)*

8 Vlines (vertical lines, LaTeX only)

Adding `.vl` and/or `.vr` to a cell draws a vertical line on that cell's left and/or right edge (or, for a colspan cell, on the boundary of the merged span). By default the line spans the full table height; the optional `rspan=K` attribute limits it to the declaring row plus `K-1` rows below.

> Note: `.vl`/`.vr` are LaTeX-only in v0.1. HTML and Reveal silently ignore them.

8.1 Full-height vline (`.vr`)

Code:

```
| Item | [Measurements]{cs=2 .vr} |   | Note |
|------|:-------------------------:|:-:|------|
| ===  |                           |   |      |
| A    | 1                         | 2 | foo  |
| B    | 3                         | 4 | bar  |
| C    | 5                         | 6 | baz  |
```

Rendered:

Item	Measurements		Note
A	1	2	foo
B	3	4	bar
C	5	6	baz

8.2 Full-height vlines on both sides (`.vl .vr`)

Code:

```
| Item | [Measurements]{cs=2 .vl .vr} |   | Note |
|------|:----------------------------:|:-:|------|
| ===  |                              |   |      |
| A    | 1                            | 2 | foo  |
| B    | 3                            | 4 | bar  |
```

Rendered:

Item	Measurements		Note
A	1	2	foo
B	3	4	bar

8.3 Vline limited with `rspan=K`

`.vl rspan=2` on the cell `3` draws a vline on its left edge for two rows. Rows above and below do not receive the vline.

Code:

```
| Item | [Measurements]{cs=2} |   | Note |
|------|----------------------|---|------|
| ===  |                      |   |      |
| A    | 1                    | 2 | foo  |
| B    | [3]{.vl rspan=2}     | 4 | bar  |
| C    | 5                    | 6 | baz  |
| D    | 7                    | 8 | qux  |
```

Rendered:

Item	Measurements		Note
A	1	2	foo
B	3	4	bar
C	5	6	baz
D	7	8	qux

8.4 Two limited vlines at different positions

Code:

```
| A | B | C                | D                |
|---|---|------------------|------------------|
| F | G | [H]{.vl rspan=2} | I                |
| I | J | K                | [L]{.vl rspan=2} |
| M | N | O                | P                |
```

Rendered:

A	B	C	D
F	G	H	I
I	J	K	L
M	N	O	P

9 Per-cell alignment override (`align=l|c|r`)

The `align=` attribute forces an alignment on a single cell, overriding the column's Markdown alignment. Both short (`l`, `c`, `r`) and long (`left`, `center`, `right`) values are accepted. Without `align=`, regular cells inherit the column alignment as usual; colspan cells default to centered.

9.1 Override on regular (non-colspan) cells

The `Value` column uses a right-aligned col-spec (`---:`). Body cells without `align=` follow that default; cells with explicit `align=` switch sides for that one cell only. To make the alignment visible, each row mixes a short value (`42`) with a longer phrase (`a longer phrase`) so the column is forced wider than any single cell content.

Code:

```
| Description                 |                         Value |
|:----------------------------|------------------------------:|
| Default (inherits col)      |                            42 |
| Default (inherits col)      |              a longer phrase |
| Override `align=l` (short)  | [42]{align=l}                 |
| Override `align=l` (long)   | [two words]{align=l}  |
| Override `align=c` (short)  | [42]{align=c}                 |
| Override `align=c` (long)   | [two words]{align=c}  |
| Override `align=r` (no-op)  | [42]{align=r}                 |
```

Rendered:

Description	Value
Default (inherits col)	42
Default (inherits col)	a longer phrase
Override `align=l` (short)	42
Override `align=l` (long)	two words
Override `align=c` (short)	42
Override `align=c` (long)	two words
Override `align=r` (no-op)	42

9.2 Override on colspan cells

`cs > 1` cells default to centered. `align=` overrides that. The single-word colspan content (`grouped data`) is the same in all three rows — only the alignment changes — so the difference is purely positional within the merged span.

Code:

```
| Step | A   | B   | Total |
|:-----|:---:|:---:|------:|
| ===  |     |     |       |
| 1    | [grouped data]{cs=3}            |   |   |
| 2    | [grouped data]{cs=3 align=l}    |   |   |
| 3    | [grouped data]{cs=3 align=r}    |   |   |
| ===2-4 |   |     |       |
| 4    | 12  | 34  |   100 |
| 5    | 78  | 90  |   200 |
| 6    | 999 | 888 |  1234 |
```

Rendered:

Step	A	B	Total
1	grouped data		
2	grouped data		
3	grouped data		
4	12	34	100
5	78	90	200

Step	A	B	Total
6	999	888	1234

9.3 Override on body cells of a numeric column

A common real-world case: a value column is right-aligned for numbers, but a few rows carry a non-numeric note (e.g. `N/A`, `pending`) that reads better left- or center-aligned. `align=` lets you opt those specific cells out of the column default without affecting the rest.

Code:

```
| Sample | Result                          |
|:-------|--------------------------------:|
| ===    |                                 |
| s001   |                          1234.5 |
| s002   |                            98.7 |
| s003   | [N/A]{align=l}                  |
| s004   |                           542.0 |
| s005   | [pending review]{align=c}       |
| s006   |                          7890.1 |
| s007   | [aborted]{align=l} |
```

Rendered:

Sample	Result
s001	1234.5
s002	98.7
s003	N/A
s004	542.0
s005	pending review
s006	7890.1
s007	aborted

9.4 Combined: `align=` + vline on a colspan

`align=` and the vline classes are independent and can be combined. Here the colspan cell is left-aligned AND carries a right-side vline that extends through both rows of the body.

Code:

```
| Item | [Forced left]{cs=2 .vr align=l} |    | Note |
|------|---------------------------------|---|------|
| ===  |                                 |   |      |
| A    | 1                               | 2 | foo  |
| B    | 3                               | 4 | bar  |
```

Rendered:

Item	Forced left		Note
A	1	2	foo
B	3	4	bar

10 Edge cases

10.1 Explicit empty cell after a colspan

A reminder that colspan needs as many empty placeholder cells as there are extra columns spanned — Pandoc requires every row to have the same physical cell count.

Code:

```
| [Long title]{cs=2} |   | Col C |
|--------------------|---|-------|
| ===                |   |       |
| a                  | b | c     |
```

Rendered:

Long title		Col C
a	b	c

10.2 `cs=1` (no merge, normal cell)

`cs=1` is a no-op: the cell takes its single natural column. Useful to assert that the attribute parses cleanly even when it doesn't trigger any merge.

Code:

```
| [Normal]{cs=1} | Other |
|----------------|-------|
| val A          | val B |
```

Rendered:

Normal	Other
val A	val B

10.3 Expected warning: rowspan crossing a midrule

A rowspan that extends across a `===` separator is flagged at parse time (warning printed to stderr; the filter does not abort). HTML rendering remains correct; LaTeX rendering of this specific case is imperfect because `\multirow` cannot cross `\midrule` cleanly.

Code:

```
| Group            | Value |
|------------------|------:|
| ===              |       |
| [CONFLICT]{rs=3} |     1 |
| ^                |     2 |
| ===              |       |
| ^                |     3 |
```

Rendered:

Group	Value
CONFLICT	1
	2
3	