

Description

`collect style cell` specifies the cell appearance styles in the collection. This includes numeric formats, borders, bolding, italics, font, text color, cell color, margins, justification, and more. These styles can be applied to all cells in the collection, to cells of a particular dimension, or to specific cells of a particular dimension. Certain appearance edits can be rendered only on certain export formats.

Quick start

Use a comma as the delimiter for the upper and lower bounds of confidence intervals

```
collect style cell result[_r_ci], cidelimiter(,)
```

Set the cell margin for all cells in the collection to 10 points on the left and right

```
collect style cell, margin(left right, width(10))
```

Format standard errors and coefficients with two decimal places

```
collect style cell result[_r_se _r_b], nformat(%8.2f)
```

Menu

Statistics > Summaries, tables, and tests > Tables and collections > Build and style table

Syntax

```
collect style cell [ taglist ] [ , options ]
```

taglist is a list of tags that identify table cells to which styles are to be applied. Within the *taglist*, if tags are joined by #, cells identified by all of these specified tags are selected; if tags are separated by a space, cells identified by any of these tags are selected. If no *taglist* is specified, styles are applied to all cells.

taglist contains

tagspec

tagspec taglist

tagspec contains

tag

tag#tag[#tag[...]]

tag contains

dimension

dimension [*levels*]

dimension is a dimension in the collection.

levels are levels of the corresponding dimension.

Distinguish between `[]`, which are to be typed, and `[]`, which indicate optional arguments.

<i>options</i>	Description
Main	
<code>name(<i>cname</i>)</code>	apply cell appearance styles to collection <i>cname</i>
<code>basestyle</code>	update base style
<code>[no]warn</code>	display or suppress notes about tags that are not recognized; default is to display
Borders	
<code>border(<i>bspec</i>)</code>	set pattern, color, and width for cell border; option may be repeated
Diagonals	
<code>dborder(<i>dbspec</i>)</code>	set direction, pattern, and color for cell diagonal border
Fonts	
<code>font([<i>fontfamily</i>][<i>, font_opts</i>])</code>	set font style for cell text
<code>smcl(<i>smcl</i>)</code>	specify formatting for SMCL files
<code>latex(<i>latex</i>)</code>	specify L ^A T _E X macro
Shading	
<code>shading(<i>sspec</i>)</code>	set background color, foreground color, and fill pattern for cells
Margins	
<code>margin(<i>mspec</i>)</code>	set margins for cells; option may be repeated
Alignments	
<code>halign(<i>hvalue</i>)</code>	set horizontal alignment for cells
<code>valign(<i>vvalue</i>)</code>	set vertical alignment for cells
Formats	
<code>nformat(<i>%fmt</i>)</code>	specify numeric format for cell text
<code>sformat(<i>sfmt</i>)</code>	specify string format for cell text
<code>minimum([#][<i>, label(string)</i>])</code>	specify minimum value to report
<code>maximum([#][<i>, label(string)</i>])</code>	specify maximum value to report
<code>empty(<i>string</i>)</code>	specify text to place in empty cells
<code>cidelimiter(<i>char</i>)</code>	use character as delimiter for confidence interval limits
<code>cridelimiter(<i>char</i>)</code>	use character as delimiter for credible interval limits
<i>font_opts</i>	Description
<code>size([<i>unit</i>])</code>	specify font size
<code>color(<i>color</i>)</code>	specify font color
<code>variant(<i>variant</i>)</code>	specify font variant and capitalization
<code>[no]bold</code>	specify whether to format text as bold
<code>[no]italic</code>	specify whether to format text as italic
<code>[no]strikeout</code>	specify whether to strike out text
<code>underline(<i>upattern</i>)</code>	specify whether to underline text

bspec is

```
[ borders ] [ , width(bwidth) pattern(bpattern) color(bcolor) ]
```

borders specifies one or more border locations and identifies where to apply the border style edits.

bwidth is defined as # [*unit*] and specifies the border line width. If # is specified without the optional *unit*, points is assumed.

bpattern is a keyword specifying the look of the border. The default is `single`. For a complete list of border patterns, see [Border patterns](#) of [TABLES] [Appendix](#). To remove an existing border, specify `nil` as the *bpattern*.

bcolor specifies the border color.

dbspec is

```
direction [ , pattern(dbpattern) color(dbcOLOR) ]
```

direction specifies the diagonal border direction and may be one of `down`, `up`, or `both`.

dbpattern is a keyword specifying the look of the diagonal border. The default is `thin`. For a complete list of diagonal border patterns, see [Diagonal border patterns](#) of [TABLES] [Appendix](#).

dbcOLOR specifies the diagonal border line color.

sspec is

```
[ background(bgcolor) foreground(fgcolor) pattern(fpattern) ]
```

bgcolor specifies the background color.

fgcolor specifies the foreground color.

fpattern specifies the fill pattern. A complete list of fill patterns is shown in [Shading patterns](#) of [TABLES] [Appendix](#).

fontfamily specifies a valid font family.

unit may be `in` (inch), `pt` (point), or `cm` (centimeter). An inch is equivalent to 72 points and 2.54 centimeters. The default is `pt`.

variant may be `allcaps`, `smallcaps`, or `normal`.

`variant(allcaps)` changes the text to all uppercase letters; applicable when publishing items from a collection to Microsoft Word, PDF, \LaTeX , and HTML files.

`variant(smallcaps)` changes the text to use large capitals for uppercase letters and smaller capitals for lowercase letters; applicable when publishing items from a collection to Microsoft Word, \LaTeX , and HTML files.

`variant(normal)` changes the font variant back to normal and leaves the capitalization unchanged from the original text; applicable when publishing items from a collection to Microsoft Word, PDF, \LaTeX , and HTML files.

bcolor, *dbcOLOR*, *bgcolor*, *fgcolor*, and *color* may be one of the colors listed in [Colors](#) of [TABLES] [Appendix](#); a valid RGB value in the form `### ## #`, for example, `171 248 103`; or a valid RRGGBB hex value in the form `#####`, for example, `ABF867`.

Options

Main

`name(cname)` specifies a collection *cname* to which appearance styles are applied.

`basestyle` indicates that the appearance styles be applied to the base style, instead of overriding the current style for the specified cells.

Each cell begins with baseline style properties. (You can view your table with these baseline style properties by first clearing out the collection styles with `collect style clear`.) The appearance of the cells is then updated with any changes specified in the default style used by `collect` and `table`. Any `collect style cell` command you issue will override the current style for the specified cells. If you specify the `basestyle` option, the style changes will instead apply to the baseline style and they will not override any current style edits targeted to specific tags.

For example, suppose you have created a table with coefficients, standard errors, *p*-values, confidence intervals, and R^2 values. You then format your coefficients to display only two digits after the decimal. If you then decide to format all other statistics to display only three digits, you can type `collect style cell result, basestyle nformat(%9.3f)` to apply this change while retaining the formatting you applied to the coefficients.

`warn` and `nowarn` control the display of notes when `collect` encounters a tag it does not recognize.

The notes are displayed by default unless you used `set collect_warn off` to suppress them. `warn` specifies that `collect` show the notes. `nowarn` specifies that `collect` not show the notes. These options override the `collect_warn` setting; see [TABLES] [set collect_warn](#).

Borders

`border([borders] [, width(bwidth) pattern(bpattern) color(bcolor)])` specifies line styles for cell borders. *borders* specifies one or more border locations and identifies where to apply the border style edits. The border locations are `left`, `right`, `top`, `bottom`, or `all`. If *borders* is not specified, `all` is assumed. You may change the width, pattern, and color for the border by specifying *bwidth*, *bpattern*, and *bcolor*.

This option may be specified multiple times in a single command to accommodate different border settings. If multiple `border()` options are specified, they are applied in the order specified from left to right. Additionally, these border style properties are applicable when publishing items from a collection to all file types, except Markdown.

Diagonals

`dborder(direction [, pattern(dbpattern) color(dbcolor)])` specifies line styles for diagonal cell borders. The direction of the diagonal border is specified by *direction*, which may be `down`, `up`, or `both`. Optionally, you may change the pattern and color for the border by specifying *dbpattern* and *dbcOLOR*.

These diagonal border style properties are applicable when publishing items from a collection to a Microsoft Excel file.

Fonts

`font([fontfamily] [, size(#[unit]) color(color) variant(variant) [no]bold [no]italic [no]strikeout [no]underline underline(upattern)])` specifies the font style for the cell text.

These font style properties are applicable when publishing items from a collection to Microsoft Word, Microsoft Excel, PDF, L^AT_EX, and HTML files, unless otherwise specified.

fontfamily specifies a valid font family. This font style property is applicable when publishing items from a collection to Microsoft Word, Microsoft Excel, PDF, and HTML files.

size(# [*unit*]) specifies the font size as a number optionally followed by units. If # is specified without the optional *unit*, points is assumed. This font style property is applicable when publishing items from a collection to Microsoft Word, Microsoft Excel, PDF, and HTML files.

variant(*variant*) specifies the font variant and capitalization.

bold and *nobold* specify the font weight. *bold* changes the font weight to bold; *nobold* changes the font weight back to normal.

italic and *noitalic* specify the font style. *italic* changes the font style to italic; *noitalic* changes the font style back to normal.

strikeout and *nostrikeout* specify whether to add a strikeout mark to the text. *strikeout* adds a strikeout mark to the text; *nostrikeout* changes the text back to normal.

Only one of *strikeout* or *underline* is allowed when publishing to HTML files.

underline(*upattern*), *underline*, and *nounderline* specify how to underline the text.

underline(*upattern*) adds an underline to the text using a specified pattern. *upattern* may be any of the patterns listed in [Underline patterns](#) of [\[TABLES\] Appendix](#). For example, *underline*(*none*) removes the underline from the text, and *underline*(*single*) underlines the text. All other *upatterns* are available only when publishing items from a collection to Microsoft Word.

Only one of *strikeout* or *underline* is allowed when publishing to HTML files; underline patterns are not allowed when publishing to HTML files.

smcl(*smcl*) specifies how to render cell text for SMCL output. The supported SMCL directives are *input*, *error*, *result*, and *text*. This style property is applicable only when publishing items from a collection to a SMCL file.

latex(*latex*) specifies the name of a L^AT_EX macro to render cell text for L^AT_EX output. This style property is applicable only when publishing items from a collection to a L^AT_EX file.

Example L^AT_EX macro names are *textbf*, *textsf*, *textrm*, and *texttt*. Custom L^AT_EX macros are also allowed. If *value* is the value for a given cell, then *latex* is translated to the following when exporting to L^AT_EX:

```
\latex {value}
```

Shading

shading([*background*(*bgcolor*) *foreground*(*fgcolor*) *pattern*(*fpattern*)]) sets the background color, foreground color, and fill pattern for cells. The background color is applicable when exporting the table to Microsoft Word, Microsoft Excel, PDF, HTML, and L^AT_EX files. The foreground color and fill pattern are applicable when exporting the table to Microsoft Word and Microsoft Excel.

Margins

margin([*margins*] [, *width*(# [*unit*])]) specifies margins inside the cell.

These margin style properties are applicable when publishing items from a collection to PDF and HTML files.

margins specifies one or more margin locations and identifies where to apply the margin style edits.

The margin locations are left, right, top, bottom, and all. If *margins* is not specified, all is assumed.

`width(# [unit])` specifies the margin width as a number optionally followed by units.

Alignments

`halign(hvalue)` specifies the horizontal alignment for the cell text. *hvalue* may be left, center, and right.

These alignment style properties are applicable when publishing items from a collection to all file types, except Markdown.

`valign(vvalue)` specifies the vertical alignment for the cell text. *vvalue* may be top, bottom, or center.

These alignment style properties are applicable when publishing items from a collection to all file types, except Markdown.

Formats

`nformat(%fmt)` applies the Stata numeric format *%fmt* to cell text constructed from numeric items.

`sformat(sfmt)` applies a string format to cell text. You can, for instance, add symbols or text to the values reported in the collection by modifying the string format.

sfmt may contain a mix of text and %s. Here %s refers to the numeric value that is formatted as specified using `nformat()`. The text will be placed around the numeric values in the collection as it is placed around %s in this option. For instance, to place parentheses around results, you can specify `sformat("(%s)")`.

Two text characters must be specified using a special character sequence if you want them to be displayed in your collection. To include %, type %%. To include \, type \\. For instance, to place a percent sign after results, you can specify `sformat("%s%%")`.

`minimum([#][, label(string)])` specifies that numeric items less than # be displayed as “<#”, where # is formatted according to `nformat()`.

If suboption `label(string)` is specified, then “string” is used instead of “<#”. If *string* contains %s, then %s is replaced by # formatted according to `nformat()`.

If suboption `label()` is not specified, it effectively defaults to `label("<%s")`.

`maximum([#][, label(string)])` specifies that numeric items greater than # be displayed as “>#”, where # is formatted according to `nformat()`.

If suboption `label(string)` is specified, then “string” is used instead of “>#”. If *string* contains %s, then %s is replaced by # formatted according to `nformat()`.

If suboption `label()` is not specified, it effectively defaults to `label(">%s")`.

`empty(string)` specifies text to place in empty cells.

`cidelimiter(char)` changes the delimiter between confidence interval limits. The default is `cidelimiter(" ")`.

`cridelimiter(char)` changes the delimiter between credible interval limits. The default is `cridelimiter(" ")`.

Remarks and examples

`collect style cell` allows you to specify the cell appearance styles for tables built from the collection. These styles include the numeric format for results, borders around cells, font, and much more. If you do not specify a tag, your appearance style will be applied to all cells in the table, including those in the body of the table and the headers.

Stored results

`collect style cell` stores the following in `s()`:

Macros

`s(collection)` name of collection

References

Huber, C. 2021a. Customizable tables in Stata 17, part 2: The new `collect` command. *The Stata Blog: Not Elsewhere Classified*. <https://blog.stata.com/2021/06/07/customizable-tables-in-stata-17-part-2-the-new-collect-command/>.

———. 2021b. Customizable tables in Stata 17, part 3: The classic table 1. *The Stata Blog: Not Elsewhere Classified*. <https://blog.stata.com/2021/06/24/customizable-tables-in-stata-17-part-3-the-classic-table-1/>.

Also see

[TABLES] [collect query](#) — Query collection style properties

[TABLES] [collect style column](#) — Collection styles for column headers

[TABLES] [collect style row](#) — Collection styles for row headers

Stata, Stata Press, and Mata are registered trademarks of StataCorp LLC. Stata and Stata Press are registered trademarks with the World Intellectual Property Organization of the United Nations. StataNow and NetCourseNow are trademarks of StataCorp LLC. Other brand and product names are registered trademarks or trademarks of their respective companies. Copyright © 1985–2025 StataCorp LLC, College Station, TX, USA. All rights reserved.

For suggested citations, see the FAQ on [citing Stata documentation](#).

