

[Description](#)[Remarks and examples](#)[Quick start](#)[Stored results](#)[Menu](#)[Reference](#)[Syntax](#)[Also see](#)[Options](#)

Description

`collect style table` specifies table header style properties. When `collect layout` specifies that multiple tables be created, `collect style table` determines how the headers for each of the tables are displayed.

Quick start

Use an `x` to delimit interaction terms

```
collect style table, delimiter(" x ")
```

Use a colon to delimit table headers with multiple dimensions

```
collect style table, dimdelimiter(" : ")
```

Menu

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

Syntax

```
collect style table [ , options ]
```

<i>options</i>	Description
<code>name(<i>cname</i>)</code>	specify table header styles for collection <i>cname</i>
<code>nodelimiter</code>	place factor-variable and interaction elements in separate cells without a delimiter
<code>delimiter(<i>delim</i>)</code>	use <i>delim</i> to delimit interaction terms composed in a single cell
<code>atdelimiter(<i>atdelim</i>)</code>	use <i>atdelim</i> to delimit interaction terms containing the @ symbol
<code>bardelimiter(<i>bardelim</i>)</code>	use <i>bardelim</i> to delimit interaction terms containing the symbol
<code>binder(<i>binder</i>)</code>	use <i>binder</i> to separate factor variables from their levels
<code>dimdelimiter(<i>dimdelim</i>)</code>	use <i>dimdelim</i> to delimit table headers
<code>dimbinder(<i>dimbinder</i>)</code>	use <i>dimbinder</i> to separate dimensions from their levels

Options

`name` (*cname*) specifies the collection to which table header style properties are to be applied. By default, properties are applied to the current collection.

`nodelimiter`, `delimiter()`, `atdelimiter()`, and `bardelimiter()` control how to compose factor-variable and interaction terms in headers.

`nodelimiter`, the default, specifies that factor-variable and interaction term elements (matrix stripe elements) be split into separate cells.

`delimiter(delim)` specifies that factor-variable and interaction term elements (matrix stripe elements) be composed in a single cell.

The variables in an interaction term are composed in a single cell using *delim* as the delimiter.

Factor-variable terms serve as their own dimension nested within the stripe dimensions `coleq`, `colname`, `roweq`, and `rowname`. Option `binder()` controls how levels of factor variables are composed within a single cell.

`atdelimiter(atdelim)` specifies that *atdelim* be used to delimit interaction terms containing the @ symbol. This option is applicable when, for example, working with results from `contrast`, `mean`, `proportion`, `ratio`, and `total`.

`bardelimiter(bardelim)` specifies that *bardelim* be used to delimit interaction terms containing the | symbol. This option is applicable when, for example, working with results from `anova` and `manova`.

`binder(binder)` specifies how to compose levels of factor variables within a single cell.

The binder will be applied as long as the factor variable and its levels are not hidden. Note that the default style used by `collect`, which is `style-default.stjson`, will hide the dimension from the headers. You can use `collect style header` to specify whether to display the label or name for a dimension and whether to display the label or value for the level of a dimension.

`dimdelimiter(dimdelim)` specifies how to delimit table headers composed from multiple dimensions. The default is to use a comma as the delimiter.

`dimbinder(dimbinder)` specifies how dimension-level pairs are composed within the table headers.

Remarks and examples

`collect style table` specifies how the headers of individual tables are to be composed when `collect` layout specifies that multiple tables are to be created. When creating multiple tables, you may have one or more dimensions defining the tables. With a single dimension, you may have the title of the dimension and the label for the level of the dimension. For these tables, you can specify the delimiter used to separate the dimension from its level. With two dimensions, you may have multiple titles and labels, in which case you may also want to specify the delimiter for the dimensions.

▷ Example 1 Delimiters for dimensions

We use data from the Second National Health and Nutrition Examination Survey (NHANES II) (McDowell et al. 1981), and we model the occurrence of a heart attack as a function of systolic blood pressure, age, and body mass index (`bmi`). We fit two different models and collect the coefficients (`_r_b`) and standard errors (`_r_se`) for each.

```
. use https://www.stata-press.com/data/r19/nhanes2
(Second National Health and Nutrition Examination Survey)
. quietly: collect _r_b _r_se: logit heartatk bpsystol bmi
. quietly: collect _r_b _r_se : logit heartatk bpsystol age
```

We would like to create tables that focus on the coefficient and standard error for `bpsystol`. We can include just those variables in which we are interested by specifying the levels of `colname` when arranging the items in our collection with `collect layout`. The dimension `cmdset` identifies the commands from which we have collected results. We place the levels of this dimension on the columns and leave the row specification empty. We also create separate tables for the results of the covariate `bpsystol`:

```
. collect layout () (cmdset) (colname[bpsystol]#result)
Collection: default
Columns: cmdset
Tables: colname[bpsystol]#result
Table 1: 1 x 2
Table 2: 1 x 2
Systolic blood pressure, Coefficient
-----
      1      2
-----
.0155575 -.0021038
-----
Systolic blood pressure, Std. error
-----
      1      2
-----
.0018229 .0020584
-----
```

The first label we see corresponds to the level `bpsystol` of the dimension `colname`. The second corresponds to the level of the dimension `result`. By default, a comma is used as the delimiter for the dimensions. We would instead like to use a colon, with spaces on each side. We specify that below with the `dimdelimiter()` option. We also label the levels of `cmdset` to indicate coefficients are adjusted for another covariate, center the results, and add extra space between columns. Then, we preview our table:

```
. collect style table, dimdelimiter(" : ")
. collect label levels cmdset 1 "Model 1 (BMI adjusted)"
> 2 "Model 2 (age adjusted)"
. collect style cell, halign(center)
. collect style column, extraspace(1)
. collect preview
Systolic blood pressure : Coefficient
-----
Model 1 (BMI adjusted)  Model 2 (age adjusted)
-----
      .0155575          -.0021038
-----
Systolic blood pressure : Std. error
-----
Model 1 (BMI adjusted)  Model 2 (age adjusted)
-----
      .0018229          .0020584
-----
```

While it is rare that we would want to report coefficients and standard errors in separate tables, we can see that `collect style table` is useful for controlling the look of table headers when `collect` creates multiple tables at once.



Stored results

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

Macros

```
s(collection)  name of collection
```

Reference

McDowell, A., A. Engel, J. T. Massey, and K. Maurer. 1981. “Plan and operation of the Second National Health and Nutrition Examination Survey, 1976–1980”. In *Vital and Health Statistics*, ser. 1, no. 15. Hyattsville, MD: National Center for Health Statistics.

Also see

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

[TABLES] [collect style header](#) — Collection styles for hiding and showing header components

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](#).

