

Description

`collect query` displays the settings of various collection parameters.

Quick start

Display the current layout specification

```
collect query layout
```

Display the current row header style properties, such as the binder used to separate factor variables from their levels and whether row header elements are stacked or placed in separate columns

```
collect query row
```

Display the current putdocx settings, such as the table width and cell spacing

```
collect query putdocx
```

Display the levels of dimension `dim1` that will be automatically displayed when `dim1` is included in a table

```
collect query autolevels dim1
```

Syntax

Query the current layout specification

```
collect query layout [ , name(cname) ]
```

Query automatic levels for a dimension

```
collect query autolevels dim [ , name(cname) ]
```

Query row header style properties

```
collect query row [ , name(cname) ]
```

Query column header style properties

```
collect query column [ , name(cname) ]
```

Query table header style properties

```
collect query table [ , name(cname) ]
```

Query dimension header style properties

```
collect query header [ dim [ level ] ] [ , name(cname) ]
```

Query collection styles for HTML files

```
collect query html [ , name(cname) ]
```

Query collection styles for Microsoft Word files

```
collect query putdocx [ , name(cname) ]
```

Query collection styles for PDF files

```
collect query putpdf [ , name(cname) ]
```

Query collection styles for L^AT_EX files

```
collect query tex [ , name(cname) ]
```

Query collection styles for displaying base levels

```
collect query showbase [ , name(cname) ]
```

Query collection styles for displaying empty cells

```
collect query showempty [ , name(cname) ]
```

Query collection styles for displaying omitted coefficients

```
collect query showomit [ , name(cname) ]
```

Query cell appearance styles

```
collect query cell [ tag ] [ , name(cname) ]
```

Query collection style for intercept position

```
collect query _cons [ , name(cname) ]
```

Query collection styles for identifying significant results

```
collect query stars [ , name(cname) ]
```

Query collection styles for table titles

```
collect query title [ , name(cname) ]
```

Query collection styles for table notes

```
collect query notes [ , name(cname) ]
```

Query collection composite results

```
collect query composite [ , name(cname) ]
```

Query collection composite result definition

```
collect query composite clevel [ , name(cname) ]
```

where *cname* is a collection name, *dim* is a dimension in the specified collection, *level* is a level of the specified dimension, *clevel* is a composite result, and *tag* identifies cells in the table. *tag* is one or more *dim* [*level*] joined by #.

Option

`name(cname)` specifies the collection for which settings should be displayed. By default, settings are displayed for the current collection.

Remarks and examples

`collect query` provides information on collection style properties. This can be useful when you want information on style properties without having to preview the table in a collection or lay out a table. For example, perhaps you have adopted a colleague's layout and style with `collect style use`. You can type

```
. collect query layout
```

to see the layout specification, or type

```
. collect query autolevels result
```

to see what levels of dimension `result` will be automatically displayed.

Or perhaps you are building a table, and you want to check whether the column header style properties are to your liking. You can type

```
. collect query column
```

and see which characters will be used to delimit interaction terms, whether extra spaces will be inserted between columns, and more.

Stored results

`collect query layout` stores the following in `s()`:

Macros

<code>s(collection)</code>	name of collection
<code>s(rows)</code>	rows specification
<code>s(columns)</code>	columns specification
<code>s(tables)</code>	tables specification
<code>s(k_tables)</code>	number of tables
<code>s(table#)</code>	layout for the #th table

collect query autolevels stores the following in s():

Macros

s(dimname)	name of dimension
s(levels)	levels of dimension to be automatically displayed
s(collection)	name of collection

collect query row after collect style row split stores the following in s():

Macros

s(span)	true or false
s(position)	left or right
s(dups)	repeat, first, or center
s(spacer)	on or off; whether a blank line is added between stacked row dimensions
s(binder)	binder used to separate factor variables from their levels
s(nobinder)	on or off; whether to bind factor variables and their levels
s(bardelimiter)	delimiter for interaction terms containing the symbol
s(atdelimiter)	delimiter for interaction terms containing the @ symbol
s(delimiter)	delimiter for interaction terms composed in a single cell
s(nodelimiter)	on or off
s(arrangement)	split
s(collection)	name of collection

collect query row after collect style row stack stores the following in s():

Macros

s(abbreviate)	on or off
s(truncate)	tail, middle, or head
s(wrap)	# of lines allowed for long headers, 0 if not specified
s(wrapon)	word or length
s(length)	maximum display length for stacked headers
s(indent)	on or off; whether stacked headers are to be indented
s(spacer)	on or off; whether a blank line is added between stacked row dimensions
s(binder)	binder used to separate factor variables from their levels
s(nobinder)	on or off; whether to bind factor variables and their levels
s(bardelimiter)	delimiter for interaction terms containing the symbol
s(atdelimiter)	delimiter for interaction terms containing the @ symbol
s(delimiter)	delimiter for interaction terms composed in a single cell
s(nodelimiter)	on or off
s(arrangement)	stack
s(collection)	name of collection

collect query column stores the following in `s()`:

Macros

<code>s(width)</code>	asis or equal
<code>s(extraspace)</code>	# of spaces used to pad columns
<code>s(position)</code>	top or bottom
<code>s(dups)</code>	repeat, first, or center
<code>s(binder)</code>	binder used to separate factor variables from their levels
<code>s(bardelimiter)</code>	delimiter for interaction terms containing the symbol
<code>s(atdelimiter)</code>	delimiter for interaction terms containing the @ symbol
<code>s(delimiter)</code>	delimiter for interaction terms composed in a single cell
<code>s(nodelimiter)</code>	on or off
<code>s(collection)</code>	name of collection

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

Macros

<code>s(dimbinder)</code>	delimiter used to separate dimensions from their levels
<code>s(dimdelimiter)</code>	delimiter for table headers
<code>s(binder)</code>	binder used to separate factor variables from their levels
<code>s(bardelimiter)</code>	delimiter for interaction terms containing the symbol
<code>s(atdelimiter)</code>	delimiter for interaction terms containing the @ symbol
<code>s(delimiter)</code>	delimiter for interaction terms composed in a single cell
<code>s(nodelimiter)</code>	on or off
<code>s(collection)</code>	name of collection

collect query header stores the following in `s()`:

Macros

<code>s(dimname)</code>	name of dimension, if specified
<code>s(dimlevel)</code>	level of dimension, if specified
<code>s(title)</code>	label, name, or hide
<code>s(level)</code>	label, value, or hide
<code>s(collection)</code>	name of collection

collect query html stores the following in `s()`:

Macros

<code>s(useth)</code>	true or false
<code>s(bcollapse)</code>	true or false
<code>s(collection)</code>	name of collection

collect query putdocx stores the following in `s()`:

Macros

<code>s(title)</code>	table title
<code>s(width)</code>	table width
<code>s(indent)</code>	table indentation from left margin
<code>s(cellspacing)</code>	spacing between adjacent cells and edges of the table
<code>s(halign)</code>	left, right, or center
<code>s(layout)</code>	fixed, autofitwindow, or autofitcontents
<code>s(cellmargin_top)</code>	margin between text and top cell border
<code>s(cellmargin_bottom)</code>	margin between text and bottom cell border
<code>s(cellmargin_left)</code>	margin between text and left cell border
<code>s(cellmargin_right)</code>	margin between text and right cell border
<code>s(note#)</code>	note #
<code>s(collection)</code>	name of collection

collect query putpdf stores the following in s():

Macros

s(title)	table title
s(width)	table width
s(indent)	table indentation from left margin
s(halign)	left, right, or center
s(spacing_before)	spacing before table
s(spacing_after)	spacing after table
s(note#)	note #
s(collection)	name of collection

collect query tex stores the following in s():

Macros

s(centering)	true or false
s(begintable)	true or false
s(collection)	name of collection

collect query showbase stores the following in s():

Macros

s(showbase)	off, factor, or all
s(collection)	name of collection

collect query showempty stores the following in s():

Macros

s(showempty)	on or off
s(collection)	name of collection

collect query showomit stores the following in s():

Macros

s(showomit)	on or off
s(collection)	name of collection

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

Macros

<code>s(cridelimiter)</code>	delimiter for credible intervals
<code>s(cidelimiter)</code>	delimiter for confidence intervals
<code>s(sformat)</code>	string format
<code>s(nformat)</code>	numeric format
<code>s(valign)</code>	vertical alignment
<code>s(halign)</code>	horizontal alignment
<code>s(shading_pattern)</code>	fill pattern
<code>s(shading_foreground)</code>	foreground color
<code>s(shading_background)</code>	background color
<code>s(latex)</code>	L ^A T _E X macro
<code>s(smcl)</code>	SMCL directive
<code>s(font_underline)</code>	underline pattern
<code>s(font_strikeout)</code>	normal or <code>strikeout</code>
<code>s(font_variant)</code>	<code>allcaps</code> , <code>smallcaps</code> , or <code>normal</code>
<code>s(font_style)</code>	whether text is formatted as italic
<code>s(font_weight)</code>	whether text is formatted as bold
<code>s(font_color)</code>	font color
<code>s(font_size)</code>	font size
<code>s(font_family)</code>	font family
<code>s(dborder_color)</code>	color for diagonal cell borders
<code>s(dborder_pattern)</code>	pattern for diagonal cell borders
<code>s(dborder_direction)</code>	<code>down</code> , <code>up</code> , or <code>both</code>
<code>s(margin_left_width)</code>	margin between text and left cell border
<code>s(margin_right_width)</code>	margin between text and right cell border
<code>s(margin_top_width)</code>	margin between text and top cell border
<code>s(margin_bottom_width)</code>	margin between text and bottom cell border
<code>s(border_left_color)</code>	color for left cell borders
<code>s(border_left_pattern)</code>	pattern for left cell borders
<code>s(border_left_width)</code>	width for left cell borders
<code>s(border_right_color)</code>	color for left cell borders
<code>s(border_right_pattern)</code>	pattern for left cell borders
<code>s(border_right_width)</code>	width for left cell borders
<code>s(border_top_color)</code>	color for left cell borders
<code>s(border_top_pattern)</code>	pattern for left cell borders
<code>s(border_top_width)</code>	width for left cell borders
<code>s(border_bottom_color)</code>	color for left cell borders
<code>s(border_bottom_pattern)</code>	pattern for left cell borders
<code>s(border_bottom_width)</code>	width for left cell borders
<code>s(tag)</code>	tags identifying table cells to which styles were applied
<code>s(collection)</code>	name of collection

collect query `_cons` stores the following in `s()`:

Macros

<code>s(position)</code>	<code>first</code> or <code>last</code>
<code>s(collection)</code>	name of collection

collect query stars stores the following in `s()`:

Macros

<code>s(type)</code>	result or dimension
<code>s(results)</code>	<i>resultID</i>
<code>s(attach)</code>	<i>attachres</i>
<code>s(fortags)</code>	<i>taglist</i>
<code>s(value1)</code>	star cutoff value 1
<code>s(label1)</code>	star label 1
<code>s(value2)</code>	star cutoff value 2
<code>s(label2)</code>	star label 2
<code>s(value3)</code>	star cutoff value 3
<code>s(label3)</code>	star label 3
<code>s(value4)</code>	star cutoff value 4
<code>s(label4)</code>	star label 4
<code>s(value5)</code>	star cutoff value 5
<code>s(label5)</code>	star label 5
<code>s(shownote)</code>	on or off; whether to show the stars note after the table
<code>s(pvname)</code>	note <i>p</i> -value name
<code>s(nformat)</code>	note numeric format
<code>s(delimiter)</code>	note delimiter between labels
<code>s(prefix)</code>	note prefix, if specified
<code>s(suffix)</code>	note suffix, if specified
<code>s(note)</code>	the stars note
<code>s(collection)</code>	name of collection

collect query title stores the following in `s()`:

Macros

<code>s(shading_pattern)</code>	fill pattern
<code>s(shading_foreground)</code>	foreground color
<code>s(shading_background)</code>	background color
<code>s(latex)</code>	L ^A T _E X macro
<code>s(smcl)</code>	SMCL directive
<code>s(font_underline)</code>	underline pattern
<code>s(font_strikeout)</code>	normal or strikeout
<code>s(font_variant)</code>	allcaps, smallcaps, or normal
<code>s(font_style)</code>	whether text is formatted as italic
<code>s(font_weight)</code>	whether text is formatted as bold
<code>s(font_color)</code>	font color
<code>s(font_size)</code>	font size
<code>s(font_family)</code>	font family
<code>s(title)</code>	custom title
<code>s(collection)</code>	name of collection

collect query notes stores the following in `s()`:

Macros

<code>s(shading_pattern)</code>	fill pattern
<code>s(shading_foreground)</code>	foreground color
<code>s(shading_background)</code>	background color
<code>s(latex)</code>	L ^A T _E X macro
<code>s(smcl)</code>	SMCL directive
<code>s(font_underline)</code>	underline pattern
<code>s(font_strikeout)</code>	normal or strikeout
<code>s(font_variant)</code>	allcaps, smallcaps, or normal
<code>s(font_style)</code>	whether text is formatted as italic
<code>s(font_weight)</code>	whether text is formatted as bold
<code>s(font_color)</code>	font color
<code>s(font_size)</code>	font size

<code>s(font_family)</code>	font family
<code>s(note1)</code>	first note
<code>s(note2)</code>	second note
<code>s(note3)</code>	third note
<code>s(note#)</code>	#th note
<code>s(k_notes)</code>	number of notes
<code>s(tag1_note1)</code>	first note for the first tag
<code>s(tag1_note2)</code>	second note for the first tag
<code>s(tag1_note#)</code>	#th note for the first tag
<code>s(k_tag1_notes)</code>	number of notes for the first tag
<code>s(tag2_note1)</code>	first note for the second tag
<code>s(tag2_note2)</code>	second note for the second tag
<code>s(tag2_note#)</code>	#th note for the second tag
<code>s(k_tag2_notes)</code>	number of notes for the second tag
<code>s(tag#_{<i>t</i>}_note#_{<i>n</i>})</code>	# _{<i>n</i>} th note for the # _{<i>t</i>} th tag
<code>s(k_tag#_notes)</code>	number of notes for the #th tag
<code>s(k_notes_tags)</code>	number of notes tags
<code>s(notes_tags)</code>	tags for notes
<code>s(tag1_notes_tag)</code>	first tag
<code>s(tag2_notes_tag)</code>	second tag
<code>s(tag#_{<i>t</i>}_notes_tag)</code>	# _{<i>t</i>} th tag
<code>s(collection)</code>	name of collection

collect query composite stores the following in `s()`:

Macros

<code>s(levels)</code>	composite result levels
<code>s(collection)</code>	name of collection

collect query composite *level* stores the following in `s()`:

Macros

<code>s(override)</code>	on or off; whether to override trim properties of elements
<code>s(trim)</code>	on or off; whether to remove extra spaces from numeric format
<code>s(delimiter)</code>	delimiter placed between elements
<code>s(elements)</code>	composite result elements
<code>s(composite)</code>	composite result level
<code>s(collection)</code>	name of collection

Also see

[TABLES] [collect layout](#) — Specify table layout for the current collection

[TABLES] [collect style autolevels](#) — Collection styles for automatic dimension levels

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

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

Stata, Stata Press, Mata, NetCourse, and NetCourseNow are registered trademarks of StataCorp LLC. Stata and Stata Press are registered trademarks with the World Intellectual Property Organization of the United Nations. StataNow is a trademark 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](#).