collect style putdocx — Collection styles for putdocx

Description

collect style putdocx specifies styles to be used by putdocx when exporting a table from a collection with putdocx collect.

collect style putdocx, typed without any options, will clear the current styles for the current collection.

Quick start

Specify that tables exported with putdocx collect have a width 80% of the default and be right-aligned

    collect style putdocx, width(80%) halign(right)

Same as above, but for the collection c2

    collect style putdocx, width(80%) halign(right) name(c2)

Clear the current styles for putdocx collect

    collect style putdocx

Menu

Statistics > Summaries, tables, and tests > Tables and collections > Collect styles > Styles for putdocx

Syntax

Specify styles to be used when exporting a collection with `putdocx collect`

>   `collect style putdocx` [`,` *options*]

Clear the current styles for `putdocx collect`

>   `collect style putdocx` [`,` `name(`*cname*`)`]

cname is the name of an existing collection.

options	Description
Main	
`name(`*cname*`)`	apply styles to collection *cname*
`halign(`*hvalue*`)`	set table horizontal alignment
`width(#`[*unit* | `%`] `|`*matname*`)`	set table width; option may be repeated
`layout(`*layouttype*`)`	adjust column width
`indent(#`[*unit*]`)`	set table indentation
`cellspacing(#`[*unit*]`)`	set spacing between adjacent cells and the edges of the table
`headerrow(#)`	set number of the top rows that constitute the table header
Cell margins	
`cellmargin(`*cmarg*`, #`[*unit*]`)`	set margins for each table cell; option may be repeated

unit may be `in` (inch), `pt` (point), `cm` (centimeter), or `twip` (20th of a point). An inch is equivalent to 72 points, 2.54 centimeters, or 1440 twips. The default is `in`.

Options

    [Main]

`name(`*cname*`)` specifies that styles for `putdocx collect` be applied to collection *cname*.

    When `name(`*cname*`)` is specified without any other options, styles for `putdocx collect` are cleared from collection *cname*.

    The default in both cases is to apply the style changes to the current collection.

`halign(`*hvalue*`)` sets the horizontal alignment of the table within the page. *hvalue* may be `left`, `right`, or `center`. The default is `halign(left)`.

`width(#`[*unit* \| `%`]`)` and `width(`*matname*`)` set the table width. Any two of the types of width specifications can be combined.

    `width(#`[*unit* \| `%`]`)` sets the width based on a specified value. *#* may be an absolute width or a percentage of the default table width, which is determined by the page width of the document. When specifying the table width as a percentage, it cannot be greater than 100%. For example, `width(50%)` sets the table width to 50% of the default table width. The default is `width(100%)`.

`width(`*matname*`)` sets the table width based on the dimensions specified in the Stata matrix *matname*, which has contents in the form of $(\#_1, \#_2, \ldots, \#_n)$ to denote the percent of the default table width for each column. n is the number of columns of the table, and the sum of $\#_1$ to $\#_n$ must be equal to 100.

`width()` may be specified multiple times in a single command to control the table width and individual column widths simultaneously.

`layout(`*layouttype*`)` adjusts the column width of the table. *layouttype* may be `fixed`, `autofitwindow`, or `autofitcontents`. `fixed` means the width is the same for all columns in the table. When `autofitwindow` is specified, the column width automatically resizes to fit the window. When `autofitcontents` is specified, the table width is determined by the overall table layout algorithm, which automatically resizes the column width to fit the contents. The default is `layout(autofitwindow)`.

`indent(`#[*unit*]`)` specifies the table indentation from the left margin of the current document.

`cellspacing(`#[*unit*]`)` sets the spacing between adjacent cells and the edges of the table.

`headerrow(`#`)` sets the top # rows to be repeated as header rows at the top of each page on which the table is displayed. This setting has a visible effect only when the table crosses multiple pages.

⎡ Cell margins ⎤

`cellmargin(`*cmarg*, #[*unit*]`)` sets the cell margins for table cells. *cmarg* may be `top`, `bottom`, `left`, or `right`. This option may be specified multiple times in a single command to accommodate different margin settings.

Remarks and examples

After finalizing your table of results, you can export it to a Word document (`.docx` file) in two ways. One is to simply use `collect export`—this command will create a document with a table from a collection. The other method is to incorporate the table into a larger report created by `putdocx`. In this case, you create an active `.docx` file, to which you can add a table from a collection along with formatted text, graphs, and other results created in Stata.

The second method utilizing `putdocx` allows you to take advantage of additional formatting features for the table you are exporting to the `.docx` file. Before you export your table, you use `collect style putdocx` to apply your desired styles to your collection. For example, you can specify the option `layout(autofitcontents)` so that the width of the columns of the table will automatically be resized to fit the contents. Then, as you are creating your report with `putdocx`, you can export the customized table to your document with `putdocx collect`.

Stored results

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

Macros
    `s(collection)`   name of collection

Reference

Huber, C. 2021. 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/.

Also see

[TABLES] **collect export** — Export table from a collection

[TABLES] **collect query** — Query collection style properties

[RPT] **putdocx collect** — Add a table from a collection to an Office Open XML (.docx) file