

collect export — Export table from a collection

[Description](#)

[Remarks and examples](#)

[Quick start](#)

[Stored results](#)

[Menu](#)

[Also see](#)

[Syntax](#)

[Options](#)

Description

`collect export` exports a table from a collection to a specified document type.

Quick start

Export a table from the current collection to `myfile.xlsx`

```
collect export myfile.xlsx
```

Export a table from the collection `c2` to `myfile2.xlsx`

```
collect export myfile2.xlsx, name(c2)
```

As above, but export the table to sheet `Table1`, instead of the default `Sheet1`

```
collect export myfile2.xlsx, name(c2) sheet(Table1)
```

Export a table from the current collection to `table1.docx`, replacing the file if it exists

```
collect export table1.docx, replace
```

As above, and save the `putdocx` commands used to export the table in `table1.do`

```
collect export table1.docx, replace dofile(table1)
```

Export a table from the current collection to `table1.tex`

```
collect export table1.tex
```

As above, but export the table only, instead of creating a complete \LaTeX document

```
collect export table1.tex, tableonly
```

Menu

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

Syntax

```
collect export filename.suffix [ , export_options document_options ]
```

<i>export_options</i>	Description
<code>name(<i>cname</i>)</code>	export collection <i>cname</i>
<code>as(<i>fileformat</i>)</code>	specify document type
<code>replace</code>	overwrite existing file

By default, `collect export` will try to determine the document type from *suffix* using the following table:

<i>suffix</i>	Implied option	File type
<code>.docx</code>	<code>as(docx)</code>	Microsoft Word
<code>.html</code>	<code>as(html)</code>	HTML 5 with CSS
<code>.pdf</code>	<code>as(pdf)</code>	PDF (Portable Document Format)
<code>.xlsx</code>	<code>as(xlsx)</code>	Microsoft Excel 2007/2010 or newer
<code>.xls</code>	<code>as(xls)</code>	Microsoft Excel 1997/2003
<code>.tex</code>	<code>as(latex)</code>	L ^A T _E X
<code>.smcl</code>	<code>as(smcl)</code>	SMCL (Stata Markup and Control Language)
<code>.txt</code>	<code>as(txt)</code>	plain text
<code>.markdown</code>	<code>as(markdown)</code>	Markdown
<code>.md</code>	<code>as(markdown)</code>	Markdown

<i>document_options</i>	Description
<i>docx_options</i>	when exporting to <code>.docx</code> files
<i>html_options</i>	when exporting to <code>.html</code> files
<i>pdf_options</i>	when exporting to <code>.pdf</code> files
<i>excel_options</i>	when exporting to <code>.xls</code> and <code>.xlsx</code> files
<i>tex_option</i>	when exporting to <code>.tex</code> files

<i>docx_options</i>	Description
<code>noisily</code>	show the <code>putdocx</code> commands used to export to the <code>.docx</code> file
<code>dofile(<i>filename</i> [, <code>replace</code>])</code>	save the <code>putdocx</code> commands used for exporting to the named do-file

<i>html_options</i>	Description
<code>prefix(<i>prefix</i>)</code>	use <i>prefix</i> to identify style classes
<code>cssfile(<i>cssfile</i>)</code>	define the styles in <i>cssfile</i> instead of <i>filename</i>
<code>tableonly</code>	export only the table to the specified file

<i>pdf_options</i>	Description
<code>noisily</code>	show the <code>putpdf</code> commands used to export to the PDF file
<code>dofile(filename [, replace])</code>	save the <code>putpdf</code> commands used for exporting to the named do-file
<i>excel_options</i>	Description
<code>noisily</code>	show the <code>putexcel</code> commands used to export to the Excel file
<code>dofile(filename [, replace])</code>	save the <code>putexcel</code> commands used for exporting to the named do-file
<code>cell(cell)</code>	specify the Excel upper-left cell as the starting position to export the table; the default is <code>cell(A1)</code>
<code>open</code>	open Excel file in memory
<code>modify</code>	modify Excel file
<code>sheet(sheetname [, replace])</code>	specify the worksheet to use; the default sheet name is <code>Sheet1</code>
<i>tex_option</i>	Description
<code>tableonly</code>	export only the table to the specified file

Options

Options are presented under the following headings:

[export_options](#)
[docx_options](#)
[html_options](#)
[pdf_options](#)
[excel_options](#)
[tex_option](#)

export_options

`name(cname)` specifies a collection to export instead of the current collection.

`as(fileformat)` specifies the file format to which the collection is to be exported. This option is rarely specified because, by default, `collect export` determines the format from the suffix of the file being created.

`replace` permits `collect export` to overwrite an existing file.

docx_options

`noisily` specifies that `collect export` show the `putdocx` commands used to export to the `.docx` file.

`dofile(filename [, replace])` specifies that `collect export` save to `filename` the `putdocx` commands used to export to the `.docx` file. If `filename` already exists, it can be overwritten by specifying `replace`. If `filename` is specified without an extension, `.do` is assumed.

html_options

`prefix(prefix)` specifies that `collect export` use *prefix* to identify style classes for the exported HTML table.

`cssfile(cssfile)` specifies that `collect export` define the styles in *cssfile* instead of *filename*. If this option is not specified, a CSS filename is constructed from *filename*, with the extension replaced with `.css`.

`tableonly` specifies that only the table be exported to the specified file. With this option, the produced file may be included in other HTML documents. By default, `collect export` produces a complete HTML document.

If option `cssfile()` is not specified, a CSS filename is constructed from *filename*, with the extension replaced with `.css`.

pdf_options

`noisily` specifies that `collect export` show the `putpdf` commands used to export to the PDF file.

`dofile(filename[, replace])` specifies that `collect export` save to *filename* the `putpdf` commands used to export to the PDF file. If *filename* already exists, it can be overwritten by specifying `replace`. If *filename* is specified without an extension, `.do` is assumed.

excel_options

`noisily` specifies that `collect export` show the `putexcel` commands used to export to the `.xls` or `.xlsx` file.

`dofile(filename[, replace])` specifies that `collect export` save to *filename* the `putexcel` commands used to export to the `.xls` or `.xlsx` file. If *filename* already exists, it can be overwritten by specifying `replace`. If *filename* is specified without an extension, `.do` is assumed.

`cell(cell)` specifies an Excel upper-left cell as the starting position to publish the table. The default is `cell(A1)`.

`open` permits `putexcel set` to open the Excel file in memory for modification. For more information about this option, see [RPT] [putexcel](#). This option could improve the speed of the export if many cells or style edits are in the collection.

`modify` permits `putexcel set` to modify an Excel file. For more information about this option, see [RPT] [putexcel](#).

`sheet(sheetname [, replace])` saves to the worksheet named *sheetname*. For more information about this option, see [RPT] [putexcel](#).

tex_option

`tableonly` specifies that only the table be exported to the specified file. With this option, the produced file may be included in other L^AT_EX documents via the `\input` or `\include` macro.

Remarks and examples

Remarks are presented under the following headings:

[Introduction](#)

[Styles for different documents](#)

[Creating more extensive documents](#)

Introduction

One goal of creating a customized table may be to present your findings to others. With `collect export`, you can export a collection to a variety of file types. For example, after creating a table from a collection of results and making styling edits to obtain the look you want, you can export a table from the current collection to an Excel file by typing the following:

```
. collect export myfile.xlsx
```

By specifying the `.xlsx` suffix, we have indicated that we want to export our work to a Microsoft Excel file. Equivalently, we could have instead specified the document type as follows:

```
. collect export myfile, as(xlsx)
```

Either way, we would have exported a table from the current collection to the file `myfile.xlsx`.

Styles for different documents

The `collect` suite of commands has many formatting features that can be applied to any collection of results, regardless of the document you may be exporting your table to. For example, you may specify the numeric formatting for your results or modify the labels for the dimensions. But there are also some style specifications that are specific to the type of document you will be exporting to. For example, you can use `collect style html` to specify whether adjacent cell borders should be collapsed in the resulting HTML file. If you will be exporting a table to a `.docx` or `.pdf` file, you can see `collect style putdocx` and `collect style putpdf` for some style specifications specific to those types of documents.

Creating more extensive documents

With `collect export`, you can export a customized table to the file format of your choice, but you may want to create documents with more than just this customized table.

For example, when you export a table to an HTML or a \LaTeX file, `collect export` creates a complete document with the table from the current collection. If you want to incorporate this table in a more extensive document, you can use the `tableonly` option to export just the table to the specified file.

Additionally, suppose that you wish to export your table to a `.docx` or `.pdf` file but you want that table to be part of a report that also includes graphs, text, and other results from Stata. By using the `putdocx` and `putpdf` suites, you can incorporate the customized table in the active document with `putdocx collect` and `putpdf collect`.

Similarly, if you want to create an Excel file with the table from the current collection and other Stata results or graphs, you can incorporate the table in the active Excel file by using the `collect` output type with `putexcel`.

Stored results

`collect export` stores the following in `s()`:

Macros

<code>s(filename)</code>	name of the file
<code>s(collection)</code>	name of collection
<code>s(noisily)</code>	1 if option <code>noisily</code> specified, 0 otherwise
<code>s(dofile)</code>	name of the new do-file
<code>s(cssfile)</code>	name of the new css-file

Also see

[TABLES] **collect style html** — Collection styles for HTML files

[TABLES] **collect style putdocx** — Collection styles for `putdocx`

[TABLES] **collect style putpdf** — Collection styles for `putpdf`

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

[RPT] **putexcel** — Export results to an Excel file

[RPT] **putpdf collect** — Add a table from a collection to a PDF file