

**Intro 4** — Overview of commands[Description](#)[Remarks and examples](#)[Also see](#)

## Description

We give an overview of all commands in the `collect` suite, organized by their intended use.

## Remarks and examples

stata.com

Remarks are presented under the following headings:

*[Introduction](#)**[Prepare to collect results](#)**[Collect results](#)**[Combine collections](#)**[Explore the collection](#)**[Modify the collection](#)**[Lay out rows and columns of the table](#)**[Preview the table](#)**[Modify labels in row and column headers](#)**[Control display of zero coefficients in regression results](#)**[Change styles—formats, bolding, colors, and more](#)**[Query collection style properties](#)**[Export the table](#)**[Save styles and labels](#)**[Save the collection](#)**[Manage collections](#)*

## Introduction

In [TABLES] **Intro 3**, we introduced the basic workflow for creating a table using `collect`. Here we provide a more detailed overview of all the commands in the `collect` suite and information on how each one may be useful in the process of creating a table.

## Prepare to collect results

Before collecting results for a new table, you will want to start with an empty collection. There are two ways to do this. You can create a new empty collection or clear all collections from memory.

`collect create`                      create a new collection

`collect clear`                      remove all collections from memory

If you have not collected any results since you opened Stata, you can skip this step—the collected results will be placed in the empty `default` collection.

### Collect results

The next step in creating a table is to collect results from one or more Stata commands.

<code>collect prefix</code>	collect results from the prefixed command
<code>collect get</code>	collect results from a previously run command

Alternatively, you can use `table` to create an initial table and place the results in a collection in one step.

### Combine collections

You can work with multiple collections at once by iteratively using `collect create` followed by `collect get` or the `collect prefix`. If you want to create a single table with results from multiple collections, you can first combine the collections.

<code>collect combine</code>	collect results from existing collections
------------------------------	---

### Explore the collection

Values in the collection are organized according to their associated `tags` (comprising dimensions and levels within the dimensions). Before creating and modifying a table, you will need to know about the tags, dimensions, and levels of those dimensions in your collection. These will be used in subsequent `collect` commands.

<code>collect dims</code>	list the dimensions in the collection
<code>collect levelsof</code>	list the levels of a dimension
<code>collect label list</code>	list the levels of a dimension along with their labels

### Modify the collection

After collecting results, you may want to modify the tags that are associated with the values in your collection. This allows you to customize the way values will later align in row and column headers when you lay out the table.

<code>collect addtags</code>	add tags to items
<code>collect recode</code>	recode levels of a dimension
<code>collect remap</code>	remap tags (modifying tags within or across dimensions)

## Lay out rows and columns of the table

With results stored in a collection, you can construct a table by identifying what belongs on the rows and columns (and possibly even separate tables).

- `collect layout`                      arrange values in the collection into a table
- `collect style autolevels` specify statistics to be automatically added to the table

## Preview the table

At the time you lay out your table, you will see a preview of the table. As you make changes to the table using the commands described in the following sections, you will likely want to see a preview of the table after each change.

- `collect preview`                      preview the current table

## Modify labels in row and column headers

Once an initial table is created, you may want to modify what appears in the row, column, and table headers by default. You can select whether labels, titles, or nothing appears for each dimension and for each level of a dimension. You can also modify the default labels.

- `collect label dim`                      add or modify the label for a dimension
- `collect label levels`                      add or modify labels for levels within a dimension
- `collect label use`                      apply labels from an external label file
- `collect label drop`                      drop dimension and level labels
- `collect style header`                      specify whether titles, labels, or nothing is shown for a dimension or for levels of a dimension
- `collect style row`                      change arrangement of row headers, how factor variables are displayed, how duplicates are reported, and how long labels wrap
- `collect style column`                      change arrangement of column headers, how factor variables are displayed, how duplicates are reported, and the width and spacing of columns
- `collect style table`                      change display of factor variables in table headers

## Control display of zero coefficients in regression results

When regression results are included in a table, coefficients with values of 0 are reported for covariates that are dropped because of collinearity, base levels of factor variables, and empty cells in factor-variable interactions. You can choose to show or hide these 0-valued coefficients.

- `collect style showomit` show or hide omitted covariates
- `collect style showbase` show or hide base levels of factor variables
- `collect style showempty` show or hide empty cells of factor-variable interactions

## Change styles—formats, bolding, colors, and more

To complete a table, you may want to modify the look of cells in the body of the table or in the row and column headers.

- `collect style cell` modify formats, bolding, italics, colors, and more
- `collect stars` add stars representing statistical significance
- `collect style html` change appearance of cell borders and header cells for tables exported to HTML
- `collect style putdocx` change titles, width, indentation, and spacing of tables to be included in a report created by `putdocx`
- `collect style putpdf` change titles, width, indentation, and spacing of tables to be included in a report created by `putpdf`
- `collect style use` apply styles from an external style file
- `collect style clear` clear all styles

## Query collection style properties

After applying styles from an external file and making several edits, you may want to check the current settings. You can query style properties for row headers, base levels, the position of the intercept, and more.

- `collect query` query collection style properties

## Export the table

After customizing the table, you can export it to Microsoft Word, HTML, PDF, Microsoft Excel, L<sup>A</sup>T<sub>E</sub>X, Markdown, SMCL, or plain text. You can also incorporate the table into a report created with `putdocx`, `putpdf`, or `putexcel`.

<code>collect export</code>	export a table
<code>putdocx collect</code>	add a table to a report created by <code>putdocx</code>
<code>putpdf collect</code>	add a table to a report created by <code>putpdf</code>
<code>putexcel ul_cell = collect</code>	add a table to a report created by <code>putexcel</code> with the top left cell of the table in <code>ul_cell</code>

## Save styles and labels

If you have built a table with styles or labels you would like to apply to other tables, you can save these to a file.

<code>collect label save</code>	save labels to a file
<code>collect style save</code>	save styles to a file

## Save the collection

If you would like to use the collection you created in the future to build a new table or further modify the existing table, you can save the collection and use it later.

<code>collect save</code>	save a collection to a file
<code>collect use</code>	use a collection from a file

## Manage collections

You can work with one or more collections in memory. With multiple collections, you can set the active collection. You can also list, copy, rename, and drop collections.

<code>collect dir</code>	list collections in memory
<code>collect set</code>	set the current collection
<code>collect copy</code>	copy a collection
<code>collect rename</code>	rename a collection
<code>collect drop</code>	drop collections from memory

## Also see

[TABLES] **Intro 3** — Workflow outline

[TABLES] **Intro 5** — Other tabulation commands