set collect_label — Label settings for collections

Description Syntax Option Remarks and examples Also see

Description

`set collect_label` controls the default labels used in tables created by `collect`. The default setting is `default`, which means that `collect` uses the labels defined in the file `label-default.stjson`. This file can be found in the ado-path.

Syntax

Use the system default labels in tables

```
set collect_label default [ , permanently ]
```

Specify a label set to be used as default labels in tables

```
set collect_label label [ , permanently ]
```

Option

`permanently` specifies that, in addition to making the change right now, the setting be remembered and become the default setting when you invoke Stata.

Remarks and examples

Remarks are presented under the following headings:

Overview

`set collect_label` controls the default labels used in tables created by `collect`. The default setting is `default`, which means that `collect` uses the labels defined in the file `label-default.stjson`. This file can be found in the ado-path.

However, if you have a set of labels that you plan to use with many of the tables that you will be creating, you can save those labels to a file with `collect label save`. For example, you can save your labels under the filename `mylabels.stjson` by typing the following:

```
. collect label save mylabels.stjson
```

Then, to use these labels by default when creating tables with `collect`, you would type

```
. set collect_label mylabels
```

`set collect_label` will then search for `label-mylabels.stjson` in the ado-path. If `label-mylabels.stjson` is not found, it will search the ado-path for `mylabels.stjson`.

To see the current setting, type

```
. display c(collect_label)
```

In the following sections, we outline the logic that `collect` uses to determine the labels to be used when there is not a label for the result that was collected.

Labels for e-class results

When collecting an e-class result, which we will call `e(`*res*`)`, `collect` will use the label from the collection that corresponds to that result. If that label does not exist, then a label is determined using the following steps:

1. If macro `e(`*res*`__CL)` exists, the label is pulled from this macro.
2. Search the system labels for a command-specific label attached to result *res*. If results are collected using the `collect` prefix, the prefixed command name is used. If results are collected using the `collect get` command, the command name is taken from macro `e(cmd)`.
3. Search the system labels for a global (command-agnostic) label attached to result *res*.

Labels for r-class results

When collecting an r-class result, which we will call `r(`*res*`)`, `collect` will use the label from the collection that corresponds to that result. If that label does not exist, then a label is determined using the following steps:

1. If macro `r(`*res*`__CL)` exists, the label is pulled from this macro.
2. Search the system labels for a command-specific label attached to result *res*. If results are collected using the `collect` prefix, the prefixed command name is used. If results are collected using the `collect get` command, the command name is taken from macro `r(cmd)`.
3. Search the system labels for a global (command-agnostic) label attached to result *res*.

Labels for other results

When collecting results from other commands, `collect` will use the label from the collection if one exists. If there is no label for this result, then a label is determined by searching the system labels for a global (command-agnostic) label attached to that result.

Also see

[TABLES] **collect label** — Manage custom labels in a collection