**collect combine** — Combine collections

**Description**

*collect combine* combines separate collections into a single new collection.

**Quick start**

Create new collection *newc* by combining existing collections *c1*, *c2*, and *c3*

```
collect combine newc = c1 c2 c3
```

As above, but use the layout defined in the rightmost collection, *c3*

```
collect combine newc = c1 c2 c3, layout(right)
```

As above, but use the style defined in the rightmost collection, *c3*

```
collect combine newc = c1 c2 c3, layout(right) style(right)
```

**Menu**

Statistics > Summaries, tables, and tests > Tables and collections > Combine collections
2 collect combine — Combine collections

Syntax

```bash
collect combine newcname = cnamelist [, options ]
```

where `newcname` is the name of the new collection and `cnamelist` is a list of names of existing collections.

<table>
<thead>
<tr>
<th>options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main</td>
<td></td>
</tr>
<tr>
<td>replace</td>
<td>overwrite <code>newcname</code> if it exists</td>
</tr>
<tr>
<td>Options</td>
<td></td>
</tr>
<tr>
<td>layout(left</td>
<td>right)</td>
</tr>
<tr>
<td>style(left</td>
<td>right)</td>
</tr>
<tr>
<td>label(left</td>
<td>right)</td>
</tr>
<tr>
<td>[no]warn</td>
<td>display notes when encountering unrecognized tags</td>
</tr>
</tbody>
</table>

Options

replace permits `collect combine` to overwrite `newcname` if it already exists. This option is required if the new collection already exists and is not empty.

layout(left|right) specifies the collection whose layout will be used in the new collection.

layout(left) is the default; it applies the layout from the leftmost collection in `cnamelist` to the new collection.

layout(right) applies the layout from the rightmost collection in `cnamelist` to the new collection.

style(left|right) specifies the collection whose style will be used in the new collection.

style(left) is the default; it applies the style definitions from the leftmost collection in `cnamelist` to the new collection.

style(right) applies the style definitions from the rightmost collection in `cnamelist` to the new collection.

label(left|right) specifies the collection whose labels will be used in the new collection.

label(left) is the default; it applies the labels from the leftmost collection in `cnamelist` to the new collection.

label(right) applies the labels from the rightmost collection in `cnamelist` to the new collection.

warn and nowarn control the display of notes when `collect` encounters a tag it does not recognize.

warn, the default, specifies that `collect` show the notes.

nowarn specifies that `collect` not show the notes.

These options override the `collect_warn` setting; see [TABLES] `set collect_warn`.
**Remarks and examples**

`collect combine` combines existing collections into a new collection. The new collection becomes the current collection.

The `label()`, `layout()`, and `style()` options specify whether `collect` should apply the labels, layout, and style from the leftmost or rightmost collection specified. The default is to apply the style, layout, and labels from the leftmost collection in `cnamelist` to the new collection. This is equivalent to specifying `label(left)`, `layout(left)`, and `style(left)`. If any of these attributes is not defined in the leftmost collection, `collect` will search for that attribute in the collections listed in `cnamelist`, from left to right. However, if the rightmost collection is specified with any of these options, and that attribute is not defined in the rightmost collection, `collect` will search for that attribute in the collections listed, from right to left.

For example, we create a collection called `new` by combining the collections `c1`, `c2`, `c3`, and `c4`.

\[. collect combine new = c1 c2 c3 c4\]

If collection `c1` has an empty style, `collect` will apply the style from `c2` to the new collection. If `c2` also has an empty style, `collect` will apply the style from `c3`.

Suppose that we instead type the following:

\[. collect combine new = c1 c2 c3 c4, style(right)\]

`collect` will apply the style from collection `c4` to the collection `new`. If collection `c4` has an empty style, `collect` will apply the style from `c3` to the new collection.

**Stored results**

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

Macros

- `s(current)` name of new collection
- `s(collections)` list of combined collections

**Also see**

[TABLES] `collect use` — Use a collection from disk

[TABLES] `collect save` — Save a collection to disk