fralias — Alias variables from linked frames

Description

fralias add defines variable aliases, names that reference variables in a linked frame. An alias defined by fralias add is a variable that behaves like a copy of a variable from a linked frame, which you could obtain from frget. Unlike a copy, however, an alias uses very little memory, and you cannot modify its observations. Almost all of Stata’s statistical and data-management commands allow you to specify an alias just as you would specify the name of a variable in the current frame.

fralias describe produces a summary of the alias variables in the current frame.

See [D] frames intro if you do not know what a frame is.

Quick start

Define aliases for variables v1, v2, and v3 from another frame linked by lnk
fralias add v1 v2 v3, from(lnk)

Define aliases newv4 and newv5 for variables v4 and v5 linked via lnk
fralias add newv4=v4 newv5=v5, from(lnk)

Define aliases for all variables in linkage lnk, prefixing them with l_
fralias add *, from(lnk) prefix(l_)

Define aliases for all variables via linkage lnk, excluding those matching pattern ind*
fralias add *, from(lnk) exclude(ind*)

Report on all the alias variables in the current frame
fralias describe

Report on the alias variables starting with l_
fralias describe l_*
**Syntax**

Add alias variables

```plaintext
fralias add varlist, from(linkname) [ rename_options ]
```

(1)

```plaintext
fralias add newalias1 = varname1
    [newalias2 = varname2 [...]] , from(linkname)
```

(2)

Describe alias variables

```plaintext
fralias describe [ varlist ]
```

*linkname* is the name of a *linkvar* in the current frame that was created by *frlink*; see [D] *frlink*.

<table>
<thead>
<tr>
<th>rename_options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>prefix(string)</td>
<td>prefix new alias names with <em>string</em></td>
</tr>
<tr>
<td>suffix(string)</td>
<td>suffix new alias names with <em>string</em></td>
</tr>
<tr>
<td>exclude(varlist)</td>
<td>exclude specified variables</td>
</tr>
</tbody>
</table>

*collect* is allowed; see [U] 11.1.10 Prefix commands.

Syntax 1 defines aliases for the variable names specified by *varlist* from the frame linked by *linkname*.

Syntax 2 defines alias *newalias1* in the current frame to be a reference to *varname1* from the frame linked by *linkname*. Similarly, alias *newalias2* is a reference to *varname2* and so on.

**Options**

*from(linkname)* specifies the identity of the linked frame from which variables are aliased. Linkages to frames are created by the *frlink* command. Linkages are usually named for the frame to which they link. Linkage counties links to frame counties, and so you specify *from(counties)*. If linkage c links to frame counties, you specify *from(c)*. *from()* is required.

*prefix(string)* specifies a string to be prefixed to the names of the new aliases created in the current frame. Say that you type

```
. fralias add inc*, from(counties)
```

to define aliases for variables income and income_family. If variable income already exists in the current frame, the command would issue an error message to that effect and alias neither variable. To alias the two variables, you could type

```
. fralias add inc*, from(counties) prefix(c_)
```

Then the new aliases would be named c_income and c_income_family.

*suffix(string)* works like *prefix(string)*, the difference being that the string is suffixed rather than prefixed to the alias names. Both options may be specified if you wish.

*exclude(varlist)* specifies variables that are not to be aliased. An example of the option is

```
. fralias add *, from(counties) exclude(emp*)
```

All variables except variables starting with *emp* would get an alias. More correctly, all variables except *emp*, _*, and the *match variables* would be aliased because *fralias add* always omits the underscore and match variables. See the explanation below.
Remarks and examples

Remarks are presented under the following headings:

- Overview
- Everything you need to know about fralias add
- Where are alias variables not allowed
- Breaking alias variables
  - Rename or drop the linked variable
  - Rename or drop the linkage variable
  - Rename or drop a matching variable
  - Rename or drop the linked frame
  - Change sort order in the linked frame

Overview

You have data on people and data on counties. You loaded the datasets and created a linkage named uscounties by typing

```
. use people
. frame create uscounties
. frame uscounties: use uscounties
. frlink m:1 countyid, frame(uscounties)
```

See example 1 in [D] frlink for details.

Among the variables in uscounties.dta is median_income. Instead of copying the variable into the current frame, you could define an alias for the variable by typing either of the following:

```
. fralias add median_income, from(uscounties)
. fralias add medinc = median_income, from(uscounties)
```

The first command defines an alias named median_income in the current frame. The second names it medinc.

Everything you need to know about fralias add

Here is everything you need to know in outline form:

1. What it means to alias a linked variable
2. fralias add can define aliases one at a time
3. fralias add allows variable names to be abbreviated
4. fralias add can define groups of aliases
5. fralias add works with all the variables specified, or none of them
6. fralias add ignores repeated variables
7. How to define aliases for all the variables 1: fralias add *
8. How to define aliases for all the variables 2: fralias add *, prefix()

We make two assumptions in what follows:

A1. The current frame contains data on people. A frame named uscounties contains data on counties. That is, we assume

```
. use people
. frame create uscounties
. frame uscounties: use uscounties
```
A2. The frames are linked on the match variable countyid, which appears in both datasets. The linkage between the frames is named uscounties, the same name as the frame being linked. That is, we assume

```
.frlink m:1 countyid, frame(uscounties)
```

1. What it means to alias a linked variable

When you type

```
.fralias add median_income, from(uscounties)
```

`fralias` add defines an alias named `median_income` in the current frame that references variable `median_income` from frame `uscounties`. This allows you to use `median_income` as if it were a variable in the current frame. It is like a copy of the original variable, but it uses much less memory, and you cannot modify its observations.

2. `fralias` add can define aliases one at a time

To define alias `median_income` of variable `median_income` from linked frame `uscounties`, type

```
.fralias add median_income, from(uscounties)
```

To instead define alias `medinc` of variable `median_income` from the same linked frame, type

```
.fralias add medinc=median_income, from(uscounties)
```

3. `fralias` add allows variable names to be abbreviated

`fralias` add allows abbreviations if you have not `set varabbrev off`. If `median_income` is the only variable beginning with `median` in the linked frame, you can type

```
.fralias add median, from(uscounties)
```

The new alias will be named `median_income`.

When using `fralias` add's `newvar=varname` syntax, you can abbreviate the variable being copied that appears to the right of the equals sign:

```
.fralias add medinc=median, from(uscounties)
```

4. `fralias` add can define groups of aliases

`fralias` add allows you to specify a `varlist`. Even though you type `fralias` add in the current frame, the `varlist` is interpreted in the linked frame. You can type

```
.fralias add emp*, from(uscounties)
```

When you specify a `varlist`, `fralias` add automatically omits the match variable or variables and any variables starting with an underscore ( `_`). First, we will tell you why, and then, we will tell you a work-around.

We start with a match variable. The match variable in our example is match variable `countyid`. The variable has the same name in both frames. Pretend for a moment that `fralias` add did not exclude match variables. Then, if you tried to alias `countyid`, that would be an error because `fralias` add will not overwrite an existing variable with a new alias. That seems reasonable until you realize that it would also mean that `fralias` add would issue an error if you typed
fralias — Alias variables from linked frames

or even if you typed

```
.fralias add *, from(uscounties)
```

fralias add would issue errors because c* and * would include countyid, which, being the match variable, already exists in the current frame. fralias add automatically omits match variables so that you can type fralias add c* and fralias add * and get aliases for all the other variables.

fralias add omits _* variables because they tend to be Stata system variables that are valid only in the dataset in which they appear. You do not want them.

What if you need to get one of these variables? Use the newvar=varname syntax. Type, for instance,

```
.fralias add _myvar=_myvar, frame(uscounties)
```

Automatic omission is not applied to this syntax.

5. fralias add works with all the variables specified, or none of them

fralias add will not replace existing variables with aliases. If just one variable in the specified list already exists in the current frame, fralias add issues an error.

```
.fralias add emp* m*, from(uscounties)
```

variable mvalues already exists

r(110);

If you want all the m* variables except mvalues, use the exclude() option:

```
.fralias add emp* m*, from(uscounties) exclude(mvalues)
```

If you also want mvalues to have alias mvals in the current frame, type

```
.fralias add mvals=mvalues, from(uscounties)
```

6. fralias add ignores repeated variables

It is not an error to type

```
.fralias add employment employment, from(uscounties)
```

We specified employment twice, but fralias add ignores that and defines the alias once. This is convenient because variables can be inadvertently repeated, as in

```
.fralias add m* employment-larea, from(uscounties)
```

Although you cannot see it, variable mds is repeated in the example. m* contains mds, and so does employment-larea because mds is among the variables stored between them.

When variables are repeated using the newvar=varname syntax, fralias add does not ignore repetition. It defines an alias for each variable that you specify:

```
.fralias add medinc=income inc=income, from(uscounties)
```
7. How to define aliases for all the variables 1: `fralias add *`

To define an alias for all the variables, try typing

```
.fralias add *, from(uscounties)
```

This sometimes works. Other times it does not because some of the variables in `uscounties` already exist in the current frame. When it does not work, `fralias add` lists the variable names that exist in both frames and, even better, stores them in `r(dups)`. Thus, if you are willing to exclude those variables, you can type

```
.fralias add *, from(uscounties) exclude('r(dups)')
```

8. How to define aliases for all the variables 2: `fralias add *, prefix()`

Another way to define aliases for all the variables in a linked frame is to type

```
.fralias add *, from(uscounties) prefix(c_)
```

This defines aliases for all the variables in the linked frame, using their original names but prefixed with `c_`. The variable `mvalues` in the linked frame, for instance, is aliased to `c_mvalues`.

Another advantage of this approach is how easily you can drop the aliases from the data should you desire to do so. Type

```
.drop c_*
```

You can choose your own prefix. If you prefer suffixing them, type

```
.fralias add *, from(uscounties) suffix(_c)
```

This names the aliases `mvalues1_c`, `mvalues2_c`, etc. These names are more like the originals, at least if you use tab completion for typing them. Type the first characters of the original name, and press `Tab`. And if you wish, you can later drop the suffixed variables just as easily as prefixed ones. Type

```
.drop *_c
```

**Where are alias variables not allowed**

The following commands change the values in variables they operate on, so by their very nature, they cannot work with alias variables: `cross`, `dyngen`, `fillin`, the `icd` suite of commands, `recode`, `reshape`, `stack`, `xpose`, the `mi` suite of commands, and `snapspan`.

The error message they produce, when they detect alias variables, will mention using `frunalias` to work around this restriction.

```
.xpose, clear
 alias variables not allowed
    Alias variables detected: `var1` and `var2`.
    You could use command `frunalias` to recast these variables to avoid this
    error message.
```

`r(109);`
Breaking alias variables

We can break the linkages that alias variables depend on. In the following, we cover the various ways this can happen.

We use the datasets and linkage described in Example 1: A typical m:1 linkage of [D] frlink for our setup. Recall that persons.dta contains data on people and txcounty.dta contains data on Texas counties, and we link the two using variable countyid.

```
. use https://www.stata-press.com/data/r18/persons
. frame create txcounty
. frame txcounty: webuse txcounty
(Median income in Texas counties)
. frlink m:1 countyid, frame(txcounty)
(all observations in frame default matched)
```

Let’s create an alias for each variable in the linked frame.

```
. fralias add *, from(txcounty)
(variable not aliased from linked frame: countyid)
(1 variable aliased from linked frame)
fralias add informed us that it added 1 alias variable.
```

For alias variables, describe will try to report the storage type of the linked variable. If the link is broken, then describe will report unknown for the storage type. In either case, describe will note when it detects alias variables. The note indicates that alias variables have a clickable type.

```
. describe
Contains data from https://www.stata-press.com/data/r18/persons.dta
Observations: 20
Variables: 5
       16 Apr 2022 13:36
(_dta has notes)
```

```
<table>
<thead>
<tr>
<th>Variable name</th>
<th>Storage type</th>
<th>Display format</th>
<th>Value label</th>
</tr>
</thead>
<tbody>
<tr>
<td>personid</td>
<td>byte</td>
<td>%9.0g</td>
<td>Person ID</td>
</tr>
<tr>
<td>countyid</td>
<td>byte</td>
<td>%9.0g</td>
<td>County ID</td>
</tr>
<tr>
<td>income</td>
<td>float</td>
<td>%9.0g</td>
<td>Household income</td>
</tr>
<tr>
<td>txcounty</td>
<td>byte</td>
<td>%10.0g</td>
<td></td>
</tr>
<tr>
<td>median_income</td>
<td>float</td>
<td>%9.0g</td>
<td>Household median income</td>
</tr>
</tbody>
</table>
```

Sorted by:
Note: Alias variables have clickable types.
Note: Dataset has changed since last saved.

Clicking on the storage type link (float) in Stata will run the fralias describe command on the associated variable.

```
. fralias describe median_income
```

```
<table>
<thead>
<tr>
<th>Alias</th>
<th>Type</th>
<th>Target</th>
<th>Link</th>
<th>Frame</th>
</tr>
</thead>
<tbody>
<tr>
<td>median_income</td>
<td>float</td>
<td>median_income</td>
<td>txcounty</td>
<td>txcounty</td>
</tr>
</tbody>
</table>
```
Rename or drop the linked variable

Let's break the link in our alias variable by renaming the linked variable `median_income` to `medinc`. `describe` now reports `unknown` for the storage type of our alias variable.

```stata
.frame txcounty: rename median_income medinc
.describe
```

Clicking on the link (unknown) shows the same information as before, except the target type is (unknown).

```stata
.fralias describe median_income
```

If we try to use this broken alias variable in a calculation, Stata will exit with an informative error message.

```stata
.summarize median_income
variable median_income not found in frame txcounty
You created alias variable median_income using the fralias command. When you did that, you specified median_income as the target variable and txcounty as the link variable for frame txcounty. The target variable median_income no longer exists in frame txcounty. Without it, the alias variable median_income is broken. If you renamed the target variable in frame txcounty, rename it back to median_income.
r(111);
```

We did rename `median_income`, so let's rename back to the original and try `summarize` again.

```stata
.frame txcounty: rename medinc median_income
.summarize median_income
```

```
<table>
<thead>
<tr>
<th>Variable</th>
<th>Obs</th>
<th>Mean</th>
<th>Std. dev.</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>median_inc~e</td>
<td>20</td>
<td>56182.1</td>
<td>12207.6</td>
<td>43788</td>
<td>72785</td>
</tr>
</tbody>
</table>
```
Rename or drop the linkage variable

Renaming or dropping a linkage variable will break all the alias variables that depend on it. A linkage variable is the variable created by \texttt{frlink}. In our example, this is the variable named \texttt{txcounty}. If we rename \texttt{txcounty} to \texttt{txcnty}, \texttt{describe} reports \textit{unknown} for the storage type of our alias variable.

```
. rename txcounty txcnty
. describe
Contains data from https://www.stata-press.com/data/r18/persons.dta
Observations: 20
Variables: 5

<table>
<thead>
<tr>
<th>Variable name</th>
<th>Storage type</th>
<th>Display format</th>
<th>Value label</th>
</tr>
</thead>
<tbody>
<tr>
<td>personid</td>
<td>byte</td>
<td>%9.0g</td>
<td>Person ID</td>
</tr>
<tr>
<td>countyid</td>
<td>byte</td>
<td>%9.0g</td>
<td>County ID</td>
</tr>
<tr>
<td>income</td>
<td>float</td>
<td>%9.0g</td>
<td>Household income</td>
</tr>
<tr>
<td>txcnty</td>
<td>byte</td>
<td>%10.0g</td>
<td></td>
</tr>
<tr>
<td>median_income</td>
<td>unknown</td>
<td>%9.0g</td>
<td>Household median income</td>
</tr>
</tbody>
</table>
```

Sorted by:
Note: Alias variables have clickable types.
Note: Dataset has changed since last saved.

Now, if we try to use this broken alias variable in a calculation, Stata will exit with another informative error message.

```
. summarize median_income
variable \texttt{txcounty} not found
You created alias variable \texttt{median_income} using the \texttt{fralias} command. When you did that, you specified \texttt{txcounty} as the link variable. The link variable \texttt{txcounty} no longer exists. Without it, the alias variable \texttt{median_income} is broken. If you renamed the link variable, rename it back to \texttt{txcounty}.
```

```
r(111);
. rename txcnty txcounty
```

Here, we simply renamed the linkage variable back to the original.

Rename or drop a matching variable

Renaming or dropping the variables used to link the frames will break alias variables that depend on that link. In our example, variable \texttt{countyid} is used to link our frames. After we rename \texttt{countyid} to \texttt{cnty} in frame \texttt{txcounty}, \texttt{describe} reports \textit{unknown} for the storage type of our alias variable.

```
. rename countyid cnty
. describe
```
. frame txcounty: rename countyid cnty
.

describe
Contains data from https://www.stata-press.com/data/r18/persons.dta
Observations: 20
Variables: 5 16 Apr 2022 13:36 (_dta has notes)

<table>
<thead>
<tr>
<th>Variable name</th>
<th>Storage type</th>
<th>Display format</th>
<th>Value label</th>
<th>Variable label</th>
</tr>
</thead>
<tbody>
<tr>
<td>personid</td>
<td>byte</td>
<td>%9.0g</td>
<td></td>
<td>Person ID</td>
</tr>
<tr>
<td>countyid</td>
<td>byte</td>
<td>%9.0g</td>
<td></td>
<td>County ID</td>
</tr>
<tr>
<td>income</td>
<td>float</td>
<td>%9.0g</td>
<td></td>
<td>Household income</td>
</tr>
<tr>
<td>txcounty</td>
<td>byte</td>
<td>%10.0g</td>
<td>median_income</td>
<td>Household median income</td>
</tr>
<tr>
<td>median_income</td>
<td><strong>unknown</strong></td>
<td>%9.0g</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Sorted by:

Note: Alias variables have clickable types.
Note: Dataset has changed since last saved.

Now, if we try to use this broken alias variable in a calculation, Stata will exit with a different informative error message.

. summarize median_income
variable countyid not found in frame txcounty
You created the link variable txcounty using the frlink command. When you did that, you specified variable countyid as the link variable, or as one of them. That variable no longer exists in frame txcounty. Without it, the frames can no longer be linked. If you renamed the variable in the frame, rename it back to countyid.
r(111);
.
.frame txcounty: rename cnty countyid

Renaming cnty back to countyid in frame txcounty resolves this problem.

### Rename or drop the linked frame

Renaming or dropping a linked frame will break alias variables linked to that frame. Let’s rename frame txcounty to county. As before, describe now reports unknown for the storage type of our alias variable.

. frame rename txcounty county
.
describe
Contains data from https://www.stata-press.com/data/r18/persons.dta
Observations: 20
Variables: 5 16 Apr 2022 13:36 (_dta has notes)

<table>
<thead>
<tr>
<th>Variable name</th>
<th>Storage type</th>
<th>Display format</th>
<th>Value label</th>
<th>Variable label</th>
</tr>
</thead>
<tbody>
<tr>
<td>personid</td>
<td>byte</td>
<td>%9.0g</td>
<td></td>
<td>Person ID</td>
</tr>
<tr>
<td>countyid</td>
<td>byte</td>
<td>%9.0g</td>
<td></td>
<td>County ID</td>
</tr>
<tr>
<td>income</td>
<td>float</td>
<td>%9.0g</td>
<td></td>
<td>Household income</td>
</tr>
<tr>
<td>txcounty</td>
<td>byte</td>
<td>%10.0g</td>
<td>median_income</td>
<td>Household median income</td>
</tr>
<tr>
<td>median_income</td>
<td><strong>unknown</strong></td>
<td>%9.0g</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Sorted by:

Note: Alias variables have clickable types.
Note: Dataset has changed since last saved.
Now, if we try to use this broken alias variable in a calculation, Stata will exit with another informative error message.

```
. summarize median_income
  frame txcounty not found
  You created the link variable txcounty using the frlink command with
  txcounty specified in option frame(). That frame no longer exists.
  Without it, the frames can no longer be linked. If you renamed the frame,
  rename it back to txcounty.
  r(111);
. frame rename county txcounty
```

Renaming the frame back to txcounty again resolves this issue.

### Change sort order in the linked frame

Changing the sort order in the linked frame will break alias variables linked to that frame. Let’s sort frame txcounty on median_income. As evidence that the link is broken, describe reports unknown for the storage type of our alias variable.

```
. frame txcounty: sort median_income
. describe
  Contains data from https://www.stata-press.com/data/r18/persons.dta
  Observations: 20
  Variables: 5
  16 Apr 2022 13:36
    (_dta has notes)

  Variable Storage Display Value              Variable label
    name    type format label                          
  personid  byte %9.0g   Person ID
  countyid  byte %9.0g   County ID
  income    float %9.0g  Household income
  txcounty  byte %10.0g
  median_income unknown %9.0g Household median income

  Sorted by:
    Note: Alias variables have clickable types.
    Note: Dataset has changed since last saved.
```

Now, if we try to use this broken alias variable in a calculation, Stata will exit with another informative error message.

```
. summarize median_income
  data in frame txcounty not sorted
    Type frlink describe txcounty. frlink describe will sort the data in the
    frame, thus correcting the problem, and it will verify that the link
    variable is otherwise still valid. If it is not, frlink describe will
    tell you how to fix the problem.
  r(5);
. quietly frlink describe txcounty
```

Using frlink describe restores the original sort order.
Stored results

`fralias add` stores the following in `r()`:

Scalars

- `r(k)`: number of aliases created

Macros

- `r(newlist)`: new aliases in the current frame
- `r(srclist)`: variables aliased from linked frame
- `r(excluded)`: variables not aliased from linked frame
- `r(dups)`: variables already present in the current frame
- `r(notfound)`: variables not found in the linked frame

`r(dups)` is present only if `fralias add` exits with an error message because a prospective new alias name already exists in the current frame.

`r(notfound)` is present only for syntax 2 when `fralias add` exits with an error message because a `varname` is not found in the linked frame.

`fralias describe` stores the following in `r()`:

Macros

- `r(varlist)`: alias variables in the current frame

Also see

[D] `frlink` — Link frames
[D] `frget` — Copy variables from linked frame
[D] `frunalias` — Change storage type of alias variables
[D] `frames intro` — Introduction to frames
[D] `merge` — Merge datasets
[M-5] `st_addalias()` — Add alias variable to current Stata dataset
[M-5] `st_isalias()` — Properties of alias variable