Description

`frunalias` changes the storage type of alias variables identified in *varlist* to that of the variable they reference in another frame. If *varlist* is not specified, then all alias variables are changed.

`frunalias` ignores variables that do not have storage type `alias`.

Quick start

Recast alias variables v1, v2, and v3 to be copies of the variables they reference in another frame

```
frunalias v1 v2 v3
```

Recast all alias variables in the current dataset

```
frunalias
```

Syntax

```
frunalias [varlist]
```

Remarks and examples

If x is an alias variable, linked to a *type* variable in another frame, then

```
frunalias x
```

will recast x to be a *type* variable. This effectively makes x a copy of the variable from the linked frame.

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);
```

As this message indicates, we could now type

```
frunalias var1 var2
```

to make a copy of var1 and var2 in the current frame. Then we can proceed with the xpose command.

Also see

[D] **fralias** — Alias variables from linked frames

[D] **frlink** — Link frames

[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