

## A (basic) command-redefinition package for SILE

The **redefine** package can be used to easily redefine a command under a new name.

Sometimes one may want to redefine a command (e.g. a font switching hook for some other command, etc.), but would also want to restore the initial command definition afterwards at some point, or to invoke the original definition from the newly redefined one.

This package is just some sort of quick "hack" in order to do it in an easy way. It's far from perfect, it likely has implications if users start saving and restoring commands in a disordered way, but it can do the magic in some fairly reasonable symmetric cases.

The first syntax below allows one to change the definition of command  $\langle name \rangle$  to a new  $\langle content \rangle$ , but saving the previous definition to  $\langle saved-name \rangle$ :

```
\redefine[command= $\langle name \rangle$ , as= $\langle saved-name \rangle$ ]{ $\langle content \rangle$ }
```

From now on, invoking  $\langle name \rangle$  will result in the new definition to be applied, while  $\langle saved-name \rangle$  will invoke the previous definition, whatever it was.

Of course, be sure to use a unique save name – otherwise, if overwriting an existing command, you will get a warning, at your own risks...

If invoked without  $\langle content \rangle$ , the redefinition will just define an alias to the current command:

```
\redefine[command= $\langle name \rangle$ , as= $\langle saved-name \rangle$ ]
```

The following syntax allows one to restore command  $\langle name \rangle$  to whatever was saved in  $\langle saved-name \rangle$ , and to clear the latter:

```
\redefine[command= $\langle name \rangle$ , from= $\langle saved-name \rangle$ ]
```

So now on  $\langle name \rangle$  is restored to whatever was saved, and  $\langle saved-name \rangle$  is no longer defined. Again, if the saved name corresponded to some existing command in a broader scope, things may break.