

Re: Woohoo! My macros are now (semi) hygienic

Source: <http://coding.derkeiler.com/Archive/Lisp/comp.lang.lisp/2008-02/msg01469.html>

- *From:* Marco Antoniotti <marcoxa@xxxxxxxx>
 - *Date:* Wed, 27 Feb 2008 01:34:24 -0800 (PST)
-

On Feb 27, 10:54 am, r...@xxxxxxxx (Rob Warnock) wrote:

Kaz Kylheku <kkylh...@xxxxxxxx> wrote:

+-----

| Maciej Katafiasz <mathr...@xxxxxxxx> wrote:

| > And that is different from (use-package :foo) how?

|

| It's different in that USE-PACKAGE is required to detect conflicts,
| which you have to explicitly deal with using SHADOW (resolve in favor
| of local package) or UNINTERN (resolve in favor of library being used).

+-----

A third option [much less blunt than UNINTERN] is SHADOWING-IMPORT.

+-----

| But USE-PACKAGE (and the :USE interface of DEFPACKAGE) are
| nevertheless blunt instruments.

+-----

True, but often a handful of prior SHADOW or SHADOWING-IMPORT
operations can make USE-PACKAGE bearable.

And of course both :SHADOW and :SHADOWING-IMPORT clauses exist
in DEFPACKAGE, and their sequencing with the :USE clause is both
well-defined and useful:

http://alu.org/HyperSpec/Body/mac_defpackage.html

Macro DEFPACKAGE

...

The order in which the options appear in a DEFPACKAGE form is
irrelevant. The order in which they are executed is as follows:

1. :shadow and :shadowing-import-from.
2. :use.
3. :import-from and :intern.
4. :export.

SHADOWs are established first, since they might be necessary to
block spurious name conflicts when the :USE option is processed.

Re: Woohoo! My macros are now (semi) hygienic

The :USE option is executed next so that :INTERN and :EXPORT options can refer to normally inherited symbols. The :EXPORT option is executed last so that it can refer to symbols created by any of the other options; in particular, shadowing symbols and imported symbols can be made external.

I have found that careful use of the :SHADOW and :SHADOWING-IMPORT clauses can significantly improve the usability of the :USE clause.

```
+-----  
| If you care about precisely what is accessible where, you have to  
| maintain lists, no matter what namespace system you are using.  
|  
| In the case of packages, you'd be maintaining import lists rather  
| than export lists. I.e. lists of symbols which are arguments to  
| :IMPORT-FROM or :SHADOWING-IMPORT-FROM in a DEFPACKAGE.  
+-----
```

Exactly. But depending upon the packages you want to inherit from, the lists you will need of :SHADOW plus :SHADOWING-IMPORT-FROM symbols plus packages :USE'd might (or might not) be a smaller maintenance burden than the lists you's need for only :IMPORT-FROM or :SHADOWING-IMPORT-FROM without any :USE.

-Rob

```
-----  
Rob Warnock          <r...@xxxxxxx>  
627 26th Avenue     <URL:http://rpw3.org/>  
San Mateo, CA 94403 (650)572-2607
```

I also remember TFB's "conduit packages". They looked pretty neat to me.

Cheers

Marco

.