

COMPILEBANG

This provides an interface to the compiler that avoids the interview for the common cases of in-core compilation. It contains a single function `COMPILE!`, and the `Lisp` and `edit` macros `C`:

`(COMPILE! X NOSAVE NOREDEFINE PRINTLAP)` [Function]

Calls the compiler to compile `X`. If `X` is a `litatom`, its definition is compiled and stored in the function cell unless `NOREDEFINE`, and the old definition if any is saved on the property list unless `NOSAVE`. No printing of `lap` or machine code is done unless `PRINTLAP`.

Thus, to simply compile the function `BAR`, do `COMPILE!(BAR)`.

`X` may also be a list form. In this case, `COMPILE!` assumes that the user is interested just in seeing how that form would compile. The form is embedded in a `Lambda` expression and compiled. Of course, there is no function-cell to be stored into or saved.

`C` [Lisp Macro]

The `LISPMACRO C` calls `COMPILE!`, with `PRINTLAP` on, on the next element of the input line. Thus, `C BAR` will compile, redefine, and save the old definition for `BAR`.

`C (CONS)` will show how a call to `CONS` would compile.

The `editmacro C` calls `COMPILE!` on the current expression if it is a list, or on the form of which the current expression is an element.