



— Shareware Five Dollars US\$5 —

This application has been carefully handcrafted by native California computer craftsmen using only pure organic bits and bytes for your health and safety.

If you use this program send \$5 (Cash, Check, Money Order, Negotiable Securities) to...

Josh & Nik Franco

1117 Kagawa Street

Pacific Palisades, CA 90272

- What is MacLibs?

Funny you should ask! MacLibs is partly a game - a computerized version of an old party favorite, and partly a literary tool. On one level, you may use MacLibs simply as an amusing diversion, or at a higher level, you may use it as a powerful literary tool which can enable anyone, no matter how untalented, to produce great imaginative literary masterpieces. With MacLibs you will be able to produce impressive and deeply moving literature for all occasions: homework, doctoral dissertations, political debates and fudge recipes.

- Wow! Sounds great!!! How do I use it?

There are two things you can do with MacLibs: (1) Play a game and (2) invent new games.

## To play a game:

Click Play under the Run Menu. You'll see a standard Mac Dialog Box with a list of files. Just pick the file you want and click OK. Next you'll see a small dialog box that will say something like "Enter an adjective" or "Enter a kind of rodent" or something like that. Just do what it says, and when you've done it, click OK. (If you decide to quit without finishing, click on Cancel.) After a while, the program will stop begging for new parts of speech and will show you your masterpiece. You can save it as a file for later viewing, or you can print it out on your printer.

## To invent new games (or change old ones):

There are two basic kinds of **MacLibs** files, **coded files**, for playing games, and **text files** for printing the results of a game. You can create **coded files** by clicking on New under the File Menu. You'll get a new window and some menus that look sort of like a feeble-minded word processor. You use this to write a story. What you do is write your story, and imbed little codes in it so that **MacLibs** will read the codes and ask the user to substitute a word in the story, sort of fill in the blanks. You make one of these "blank" codes by using two backslashes — `\\`. So to get the program to ask for a noun, for example, you write `\noun\`. You can do the same thing by clicking on noun under the Insert Menu, just to save a little time. The insert menu gives a few of the most common kinds of substitutions you may use, but you can use anything you want as a prompt. You could write something like: "...and so my brother fell madly in love with `\name of a girl in brother's class\` and kissed her `\number\` times and..." and when **MacLibs** read this it would ask the user to "Enter a name of a girl in brother's class" and then "Enter a number."

**The important thing to remember is this: You have to have two backslashes: `\\` Otherwise **MacLibs** will get confused.**

When you are all done editing your story, then save it, and you can run it like any other **MacLibs** game.

To change an already existing game, just click on Open, under the File Menu, and open an old game. Be sure to save it using Save As... under a different name if you want to keep the old story the same. (**Secret Trick:** If you hold down the option key when you select Open, you'll get to choose not only **MacLibs** files but also any TEXT files made by any other program (TEXT files are files that word processors can save without all the styles and stuff).

## About Special Codes:

**You must include special codes in brackets, and you can not have any blank spaces before or after! They must go first, before the word prompt!**

**[art]**: Use this to have the program include "a" or "an" in front of the word it is substituting. (There are rules for this sort of thing, ya know.)

For example: `\[art]noun\` will request a noun, then when it puts it into the story it will put 'a' or 'an' before it.

**[Cap]**: Use this to have the program capitalize the word it is substituting.

For example: `\[Cap]noun\` will request a name, then capitalize it when it puts it into the story.

**[noE]**: Use this if you need to strip the last 'e' off a word for some reason. If there is no 'e', nothing happens.

For example: You write "she had a nicely `\[noE]verb\ed` face"; someone enters "rearrange"; the result is "she had a nicely rearranged face".

**[plu]**: Use this to make a word into a plural. All it does is add an 's' at the end, unless the word ends in 'y' in which case it changes it to end in 'ies' if the preceding letter is a consonant in which case it changes the ending to 'ies' unless it ends in s, z, x, ch, or sh in which case it adds 'es' but only if your mother is an English teacher.

For example: You write "it rained `\[plu]animal\` and `\[plu]animal\`"; someone enters "frog" and "fly"; the result is "it rained frogs and flies". If someone says "Kathy" for the kind of animal you'll get "Kathies" which is gramatically wrong but, hey, what'ya want fer nuttin.

**[#1]...#1**: You put a number (1 to 100) with a #-sign in brackets, like this: `\[#62]noun\`, then the word entered is coded with the number. Later you can write `\#62\` (note: with the #-sign but without the brackets, and instead of asking for a new word, the program will insert the same word that was entered before.

For example: You write "...he ate a whole bag of `\[#1][plu]noun\`. He picked up `\[art]#1\` and gobbled it up. `\[Cap][plu]#1\` sure taste good...", the machine asks: "Enter a noun" and if you enter "earmuff" the story will say: "...he ate a whole bag of earmuffs. He picked up an earmuff and gobbled it up. Earmuffs sure taste good..."

That's all there is to it. After the story is done, pass it off as homework. Your parents will be amazed at what happens to your school grades!