

## LOADMENUITEMS

By: sML (Lanning.pa@Xerox.com)

### INTRODUCTION

Some utility files are so useful that users will always want them in their system: these files are typically loaded from the users INIT file. A (rather large) number of other utilities are only sometimes useful. Users are faced with the choice of either loading these files from their INIT files (slowing down the initialization process and consuming space, whether the utility is needed or not) or having to remember how to load and initialize these files.

LOADMENUITEMS addresses this problem: it defines a new filepackage command that can be used to add entries onto the background menu for easy loading of utility files.

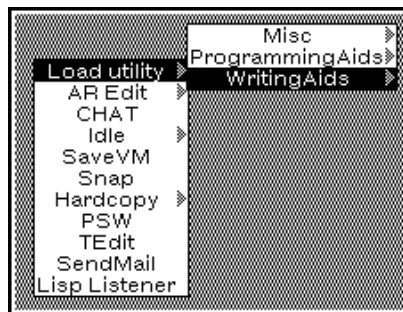
[NOTE: All (advertised) symbols in this utility are in the INTERLISP package.]

### EXAMPLE

The filepackage command

```
(COMS
  ;; Make it easy to load some oft-used utilities
  (FILES LoadMenuItems)
  (LOADMENUITEMS WritingAids Sketch VirtualKeyboards ProofReader)
  (LOADMENUITEMS ProgrammingAids (Spy (SPY.BUTTON)))
  (LOADMENUITEMS NIL VStats Calendar))
```

will add an entry "Load utility" to the background menu. "Load utility" will have three subitems: Misc, ProgrammingAids, and WritingAids:



WritingAids will in turn have three subitems: ProofReader, Sketch, and VirtualKeyboards; Misc will have the subitems Calendar and VStats; ProgrammingAids will have the single subitem Spy.



Selecting any of these final menu items will load the corresponding file. In addition, the `spy` menu item will evaluate the form `(SPY.BUTTON)` after loading the file `Spy`.

## INTERFACE

(LOADMENUITEMS *group utilDescr1 utilDescr2 ...*) [FilePackageCommand]

Dumps out to the file a form that will add items to the background menu for loading *utilDescr1*, *utilDescr2*, ... Each item will be added to the *group* subitem of the "Load utility" item on the background menu; if *group* is NIL it defaults to "Misc".

In the simplest case, *utilDescr* is a LITATOM. This is used when you want to load a file without any extra initialization, and the file is on one of the directories in `DIRECTORIES`. Selecting the resulting item will evaluate `(DOFILESLOAD 'utilDescr)` and print an informative message in the prompt window when the `DOFILESLOAD` is finished. The added item will have the label *utilDescr*.

In the general case, *utilDescr* is a list. This is used when you want to specify an initialization form to be evaluated when the utility is loaded, or when the file description is not a LITATOM. In this case, selecting the menu item will evaluate `(DOFILESLOAD (CAR 'utilDescr))`. If *utilDescr* is a list of two elements, the `CADR` of *utilDescr* will be evaluated after the utility is loaded; otherwise an informative message will be printed in the prompt window. The added item will have as a label the first LITATOM in the `CAR` of *utilDescr*; this is the first file that will be loaded when the item is selected.

In each of the above cases, the item is removed from the background menu after the utility is loaded and initialized.

When a utility is loaded from the "Load utility" menu, the event is added to the history list. This way you can UNDO loading a utility.

Some illustrative examples:

```
;; This adds the item "VStats" to the "Misc" subitem
(LOADMENUITEMS NIL VStats)

;; Selecting the "Spy" item will load SPY and call
;; SPY.BUTTON to bring up the spy button icon
(LOADMENUITEMS ProgrammingAids (Spy (SPY.BUTTON)))

;; This will add the item "GO" to the "Games" group
(LOADMENUITEMS Games (((SYSLOAD FROM {PHYLUM}<Foster>Lisp>) GO)))

;; These items are useful for Lafite users, but aren't always
;; needed
(LOADMENUITEMS MailTools LafiteFind Undigestify MailScavenge)
```

## FUNCTIONS

(AddLoadMenuItem *group fileDescr startUpForm*)

[Function]

Add a menu item to the background menu that will load the files. The item will be added under the top level item "Load utility". *group* is the submenu name for this file; the default is `Misc`. *fileDescr* is a list that can be passed to `DOFILESLOAD` to load the files. *startUpForm* is an optional form that will be evaluated after the `DOFILESLOAD`; the default will print a nice message in the prompt window. The `LOADMENUITEMS` filepackage command described above expands to calls to `AddLoadMenuItem`.

`AddLoadMenuItem` is UNDOable.

(PickLoadUtilityItem *utility-name* &OPTIONAL *group-name no-errors-p*)

[Function]

This is the programatic equivalent of selecting the item named *utility-name* from the "Load utility" item on the background menu. If *group-name* is given, only that group under the "Load utility" item is searched for the utility; otherwise the entire menu item is searched. If multiple matching items are found, a continuable error is signaled. Proceeding from this error will let you pick one of the items to execute. If no matching items are found, a continuable error is signaled. The *no-errors-p* flag controls whether or not these errors are actually signaled: if *no-errors-p* is true, `PickLoadUtilityItem` ignores the errors. `PickLoadUtilityItem` return T if the utility was loaded, NIL otherwise.