```
(RPAQQ LOADMENUITEMSCOMS
      [

;;; Defines a new FILEPKGCOM called LOADMENUITEMS that can be used to add entries onto the background menu for easy loading of utility files

        (FNS AddLoadMenuItem LoadUtilityPackage PickLoadUtilityItem LOADMENUITEMSCOMS)
        (FILEPKGCOMS LOADMENUITEMS)
        ;; Make the Filemanager happy

        (DECLARE%: DONTCOPY (PROP MAKEFILE-ENVIRONMENT LOADMENUITEMS))
        (DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVARS (ADDVARS (NLAMA)
                                                                             (NLAML)
                                                                             (LAMA PickLoadUtilityItem)])


;;; Defines a new FILEPKGCOM called LOADMENUITEMS that can be used to add entries onto the background menu for easy loading of utility files

(DEFINEQ

(AddLoadMenuItem
  [LAMBDA (group fileDescr startUpForm)                          ; Edited  2-Dec-87 11:00 by smL

;;; Add a menu item to the background menu that will load the file{s} --- group is the submenu name for this file, default is Misc --- fileDescr is a list that
;;; can be passed to DOFILESLOAD to load the file{s} --- startUpForm is an optional form that will be evaluated after the LOAD, default will print a nice
;;; msg in the prompt window

    (DECLARE (GLOBALVARS BackgroundMenu BackgroundMenuCommands))
    (LET* [(group (OR group "Misc"))
           (fileDescr (MKLIST fileDescr))
           [fileName (for x in fileDescr thereis (NOT (LISTP x]
           (startUpFormProvided (NOT (NULL startUpForm)))
           [startUpForm (OR startUpForm '(PROMPTPRINT ,fileName " loaded"]
           [loadForm ''(LISPXEVAL '(LoadUtilityPackage ',fileDescr ',startUpForm)
                             "<mouse event>_"]
           (topLevelItem (OR (for item in BackgroundMenuCommands thereis (STRING-EQUAL (CAR item)
                                                                                   "Load utility"))
                             (LET [(topLevelItem (LIST "Load utility" NIL "Load a utility file"
                                                       (LIST 'SUBITEMS]
                                  (/push BackgroundMenuCommands topLevelItem)
                                  topLevelItem)))
           (groupItem (OR (for item in (CDAR (LAST topLevelItem)) thereis (STRING-EQUAL group (CAR item)))
                          (LET [(groupList (LIST group NIL "Load a utility file" (LIST 'SUBITEMS]
                               (/push (CDAR (LAST topLevelItem))
                                      groupList)
                               [SORT (CDAR (LAST topLevelItem))
                                     (FUNCTION (LAMBDA (x y)
                                                (ALPHORDER (CAR x)
                                                           (CAR y]
                               groupList)))
           (newItem (LIST fileName loadForm (CONCAT "Load " fileName]
      (COND
        ((for item in (CAR (LAST groupItem)) thereis (EQUAL item newItem))
                                                                ; Don't bother to add the item if it's already there
          NIL)
        ([AND (NOT startUpFormProvided)
              (for file in fileDescr when (NOT (LISTP file))
                 always (NOT (NULL (GETPROP (ROOTFILENAME (U-CASE file)
                                                   T)
                                            'FILEDATES]        ; If there is no start-up form and all the files are already loaded,
                                                                ; don't bother adding the item
          NIL)
        (T                                                      ; Go ahead and add the item
          (/NCONC1 (CAR (LAST groupItem))
                   newItem)
          [SORT (CDAR (LAST groupItem))
                (FUNCTION (LAMBDA (x y)
                           (ALPHORDER (CAR x)
                                      (CAR y]
          (SETQ BackgroundMenu NIL)
          (UNDOSAVE '(SETQ ,BackgroundMenu NIL])
```

₍**LoadUtilityPackage**
  [LAMBDA (fileDescr startUpForm)   ; Edited 20-Apr-88 18:04 by smL

;;; Load the package. See the function AddLoadMenuItem.

```
    (ALLOW.BUTTON.EVENTS)
    (LET* ((myWindow (WFROMDS (TTYDISPLAYSTREAM)))
        (windowOpen? (OPENWP myWindow)))                        ; Let the use know what is going on
      (printout NIL T "Loading " (for x in fileDescr thereis (NOT (LISTP x)))
            T)                                                  ; Do the load
      [DOFILESLOAD (for x in fileDescr collect (if (LISTP x)
                                                    then x
                                                  else (MKATOM (U-CASE x]
      (EVAL startUpForm)                                        ; Now remove the load menu item, if you can find it
      (LET [(topLevelItem (for item in BackgroundMenuCommands thereis (STRING-EQUAL (CAR item)
                                                                                    "Load utility"]
          (for groupItem in (CDAR (LAST topLevelItem)) bind thisItem
            eachtime [SETQ thisItem (for item in (CDAR (LAST groupItem))
                                       thereis (EQ fileDescr (CADR (CADR (CADR (CADR (CADR (CADR item]
            when thisItem do (DREMOVE thisItem (CAR (LAST groupItem)))
                             [if (NULL (CDAR (LAST groupItem)))
                                 then (DREMOVE groupItem (CDAR (LAST topLevelItem]
                             (change BackgroundMenu NIL))) ; Close any TTY window that we may have opened
      (if (NOT windowOpen?)
          then (CLOSEW myWindow])
```

₍**PickLoadUtilityItem**
  [CL:LAMBDA (XCL-USER::UTILITY-NAME &OPTIONAL XCL-USER::GROUP-NAME XCL-USER::NO-ERRORS-P)
                                        ; Edited 3-Dec-87 11:29 by smL

;;; Many thanks to Andy Cameron for this function.

;;; Provides a way to programatically chose items from the LoadUtility Menu item

;;; Eg: (PickLoadUtilityItem "FileWatch")

```
        (CL:LABELS ((XCL-USER::ITEM-LABEL (XCL-USER::MENU-ITEM)
                      (CL:FIRST XCL-USER::MENU-ITEM))
                    (XCL-USER::ITEM-BODY (XCL-USER::MENU-ITEM)
                      (CL:SECOND XCL-USER::MENU-ITEM))
                    (XCL-USER::ITEM-SUB-ITEMS (XCL-USER::MENU-ITEM)
                      (CL:REST (CL:FOURTH XCL-USER::MENU-ITEM)))
                    [XCL-USER::DO-ITEM (XCL-USER::MENU-ITEM)
                      (CL:EVAL (CL:EVAL (XCL-USER::ITEM-BODY XCL-USER::MENU-ITEM]
                    (XCL-USER::LOOKUP (XCL-USER::X XCL-USER::ALIST)
                      (CL:ASSOC XCL-USER::X XCL-USER::ALIST :TEST #'STRING-EQUAL))
                    (XCL-USER::MAKE-MENU (&KEY XCL-USER::ITEMS XCL-USER::TITLE)
                      (create MENU
                              TITLE _ XCL-USER::TITLE
                              ITEMS _ XCL-USER::ITEMS
                              CENTERFLG _ T)))
            ;;
            ;; Find all the matching utilities
            (LET [(XCL-USER::GROUP-ITEMS (CL:REMOVE-IF-NOT #'[CL:LAMBDA (XCL-USER::GROUP-ITEM)
                                                               (AND (XCL-USER::LOOKUP
                                                                      XCL-USER::UTILITY-NAME
                                                                      (XCL-USER::ITEM-SUB-ITEMS
                                                                        XCL-USER::GROUP-ITEM))
                                                                    (OR (NULL XCL-USER::GROUP-NAME)
                                                                        (STRING-EQUAL
                                                                          XCL-USER::GROUP-NAME
                                                                          (XCL-USER::ITEM-LABEL
                                                                            XCL-USER::GROUP-ITEM]
                                           (XCL-USER::ITEM-SUB-ITEMS (XCL-USER::LOOKUP "Load utility"
                                                                       BackgroundMenuCommands]
                (COND
                  ((= 1 (CL:LENGTH XCL-USER::GROUP-ITEMS))
                   ;; Exactly one applicable item, so use it
                   [XCL-USER::DO-ITEM (XCL-USER::LOOKUP XCL-USER::UTILITY-NAME (XCL-USER::ITEM-SUB-ITEMS
                                                                                 (CL:FIRST
                                                                                   XCL-USER::GROUP-ITEMS
                                                                                   ]
                   T)
                  (XCL-USER::NO-ERRORS-P
                      ;; Forget it.
                      NIL)
                  ((NULL XCL-USER::GROUP-ITEMS)
                   ;; Not found anywhere.
                   (CL:CERROR "Forget loading the utility." "~%%Utility ~A not found under ~A on the
                         LoadUtilty menu." XCL-USER::UTILITY-NAME (OR XCL-USER::GROUP-NAME "any group"))
                   NIL)
```

```
                              (T  ;; Found in more that one place.
                                  (CL:CERROR "Pick one of them to load." "~%%Utility ~A found under multiple groups on
                                           the LoadUtilty menu.~%%~{ ~A~}" XCL-USER::UTILITY-NAME (CL:MAPCAR
                                                                                                      #'
                                                                                                   XCL-USER::ITEM-LABEL

                                                                                                XCL-USER::GROUP-ITEMS
                                                                                                     ))
                                  (LET [(XCL-USER::GROUP-ITEM (MENU (XCL-USER::MAKE-MENU
                                                                        :TITLE "Load utility from which group?" :ITEMS
                                                                      (CL:MAPCAR #'(CL:LAMBDA (XCL-USER::GROUP-ITEM)
                                                                                      (LIST (XCL-USER::ITEM-LABEL
                                                                                             XCL-USER::GROUP-ITEM
                                                                                             XCL-USER::GROUP-ITEM))
                                                                                   XCL-USER::GROUP-ITEMS]
                                       (CL:WHEN XCL-USER::GROUP-ITEM
                                           (XCL-USER::DO-ITEM (XCL-USER::LOOKUP XCL-USER::UTILITY-NAME (
                                                                                               XCL-USER::ITEM-SUB-ITEMS

                                                                                               XCL-USER::GROUP-ITEM
                                                                                                    )))
                                       T)])
```

(**LOADMENUITEMSCOMS**
```
  [LAMBDA (GROUP X)                                                 (* smL "16-Sep-86 18:37")

              (* * Expand the LOADMENUITEMS filepackage command)

     (LIST (CONS 'P (for packageDescr in X collect (if (LITATOM packageDescr)
                                                       then `(AddLoadMenuItem ',GROUP ',packageDescr)
                                                    elseif (EQLENGTH packageDescr 1)
                                                       then `[AddLoadMenuItem ',GROUP ',(CAR packageDescr]
                                                    elseif (EQLENGTH packageDescr 2)
                                                       then `[AddLoadMenuItem ',GROUP ',(CAR packageDescr)
                                                                 ',(CADR packageDescr]
                                                    else (ERROR "Utility description has too many arguments"
                                                                packageDescr])
)

[PUTDEF 'LOADMENUITEMS 'FILEPKGCOMS '((COM MACRO [(GROUP . X)
                                                  (COMS * (LOADMENUITEMSCOMS 'GROUP 'X]
                                        CONTENTS NILL]
```

;; Make the Filemanager happy

(DECLARE%: DONTCOPY

(PUTPROPS **LOADMENUITEMS MAKEFILE-ENVIRONMENT** (:PACKAGE "IL" :READTABLE "INTERLISP" :BASE 10))
)

(DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVARS

(ADDTOVAR **NLAMA**)

(ADDTOVAR **NLAML**)

(ADDTOVAR **LAMA** PickLoadUtilityItem)
)

FUNCTION INDEX

PROPERTY INDEX