<AFFIRM>PROFILE..26 17-Jun-81 16:36:31

```
(FILECREATED "31-Mar-81 20:38:15" <AFFIRM>PROFILE..26 74800

     previous date: "28-Mar-81 16:58:45" <AFFIRM>PROFILE..25)


(PRETTYCOMPRINT PROFILECOMS)

(RPAQQ PROFILECOMS [(VARS AutoCompile AutoFix AutoInfix AutoSave (AutoSufficient NIL)
                          CommandCategories
                          (LastProgramUnits NIL)
                          (ReferencedInterfaces NIL)
                          (ShortenedReferencedInterfaces NIL)
                          (InAutoMechanism NIL))
        (* Profile Mechanism)
        (FNS APE APG APGgetName AtomList canUserChangeProfileEntry? DefaultUserProfile
             definedEntryName? displayProfileEntry formChangePairs getNewValue
             getProfileAssociatedVariables getProfileConditions getProfileDefaultValue
             getProfileEntryDef getProfilePossibleValues gnvKeyList gnvQuestion InitializeUserProfile
             IsInKeyList? performAssociatedUpdates profile ReadUserProfileFile
             ReInitializeUserProfile reportProfileDBError reportUndefinedEntryNames saveUserProfile?
             SpecialValueOK? UserProfile UserProfileDefault UserProfileEnquiry UserProfilePut
             UserProfileSet validateValue WriteUserProfileFile)
        (* Profile Data Base)
        (PROP ALL UserProfileEntryDefinitions)
        (RECORDS ProfileChangePair UserProfileGroupInformation)
        [P (KnownNames:ProfileEntries← (SORT (for pe on (GETPROPLIST (QUOTE
                                                                     UserProfileEntryDefinitions))
                                                  by pe::2 unless pe:1 MEMB SYSPROPS collect
                                                  pe:2:ProfileEntryName)
                                             (QUOTE UPPER]
        (VARS (EnquiryAborted NIL)
              (HelpWanted NIL)
              [OnOffKeyList (QUOTE ((False NIL RETURN (QUOTE Off))
                                    (No NIL RETURN (QUOTE Off))
                                    (Off NIL)
                                    (On NIL)
                                    (True NIL RETURN (QUOTE On))
                                    (Yes NIL RETURN (QUOTE On]
              (OffValues (QUOTE (False NIL No Off)))
              (OnValues (QUOTE (On T True Yes)))
              (ProfileAlreadyLoaded NIL)
              [SameOldKeyList (QUOTE ((" " "(old value)" NOECHOFLG T EXPLAINSTRING
                                           "(space: old value)"
                                           RETURN NIL)
                                      ("
" "(old value)" NOECHOFLG T EXPLAINSTRING "(carriage return: old value)" RETURN NIL)
                                      ("Exit!" NIL RETURN (PROGN (SETQ EnquiryAborted T)
                                                                 NIL))
                                      ("Help!" NIL RETURN (PROGN (SETQ HelpWanted T)
                                                                 NIL]
              UserProfileGroups)
        (* BLOCK Specifications)
        (FNS * AUTOFNS)
        (BLOCKS (PROFILEBLOCK APE APG APGgetName AtomList canUserChangeProfileEntry?
                              DefaultUserProfile definedEntryName? displayProfileEntry
                              formChangePairs getNewValue getProfileAssociatedVariables
                              getProfileConditions getProfileEntryDef getProfilePossibleValues
                              gnvKeyList gnvQuestion InitializeUserProfile IsInKeyList?
                              performAssociatedUpdates profile ReadUserProfileFile
                              ReInitializeUserProfile reportProfileDBError reportUndefinedEntryNames
                              saveUserProfile? SpecialValueOK? UserProfile UserProfileDefault
                              UserProfileEnquiry UserProfilePut UserProfileSet validateValue
                              WriteUserProfileFile
                              (ENTRIES APE APG DefaultUserProfile InitializeUserProfile UserProfile
                                       UserProfileDefault UserProfilePut UserProfileSet profile)
                              (GLOBALVARS EnquiryAborted FONTCHANGEFLG GOODGUY HelpWanted KnownNames
                                          OffValues OnOffKeyList OnValues ProfileAlreadyLoaded
                                          ProfileReadTable SameOldKeyList SYSPROPS UserProfileGroups
                                          USERNAME)
                              (NOLINKFNS . T)
                              (SPECVARS casedName])

(RPAQQ AutoCompile NIL)

(RPAQQ AutoFix NIL)

(RPAQQ AutoInfix NIL)

(RPAQQ AutoSave NIL)

(RPAQ AutoSufficient NIL)
```

```
(RPAQQ CommandCategories ((; abort batch compile e exec fix forget freeze gripe help ImplicitE lisp
                             load monitor needs note ok print profile quit read redo renumber review
                             save stop storage thaw transcript undo)
        (adopt axiom declare define discard edit end infix interface lemma rulelemma schema
              sufficient? type)
        (@ annotate apply arc assume augment cases choose clear complete denote down employ eval
           invoke let name next normalize normint put replace resume retry search split suppose swap
           theorem try up use)
        (affirmed? genvcs readp)
        (annotate clear eval name theorem)))

(RPAQ LastProgramUnits NIL)

(RPAQ ReferencedInterfaces NIL)

(RPAQ ShortenedReferencedInterfaces NIL)

(RPAQ InAutoMechanism NIL)
[DECLARE: DONTEVAL@LOAD DONTCOPY
```

```
(* Profile Mechanism) ]

(DEFINEQ

                                                                                    1

(APE
  [LAMBDA (casedName)                                    (* D.Thompson " 1-Oct-79 16:27")
    (PROG (fields newVal keyList question done groupName PEname record)
          (DECLARE (SPECVARS fields))
          (PEname←(U-CASE casedName))
          (for field in '(("Entry Name" casedName)
                          ("Display String" (QUOTE NormalDisplay))
                          ("PossibleValues" (QUOTE OnOffKeyList))
                          ["DefaultValue" (COND
                                            ((EQ (CADDR fields)
                                                 (QUOTE OnOffKeyList))
                                              (QUOTE Off))
                                            (T (QUOTE NoDefault]
                          ("SubKeys" (QUOTE NoSubKeys))
                          ("Conditions" (QUOTE NoConditions))
                          ("Associated Variables" (QUOTE NoAssociatedVariables)))
             do (printout T .TAB0 0 field:1 ":" .)
                (fields← < !! fields (if (READ)
                                         else (EVAL field:2))
                         >))
          (record←(create UserProfileEntryInformation
                          ProfileEntryName ← fields:1
                          DisplayString ← fields:2
                          PossibleValues ← fields:3
                          DefaultValue ← fields:4
                          SubKeys ← fields:5
                          Conditions ← fields:6
                          AssociatedVariables ← fields:7))
          (PUTPROP 'UserProfileEntryDefinitions PEname record)
          (newVal←(GETPROP 'UserProfileEntryDefinitions PEname))
          (printout T T "New Entry" , PEname ":" T 5 "Profile Entry Name =" 28 .PPV
                    newVal:ProfileEntryName T 5 "Display String =" 28 .PPV newVal:DisplayString T 5
                    "Possible Values ="
                    28 .PPV newVal:PossibleValues T 5 "Default Value =" 28 .PPV newVal:DefaultValue T
                    5 "SubKeys =" 28 .PPV newVal:SubKeys T 5 "Conditions =" 28 .PPV newVal:Conditions
                    T 5 "Associated Variables =" 28 .PPV newVal:AssociatedVariables T)
          (UserProfileSet PEname newVal:DefaultValue)
          (keyList← <<'/" (a new group)" > !(for g in UserProfileGroups collect <g:GroupName NIL>)
                    >)
          (question←(CONCAT "What Profile Group does " newVal:ProfileEntryName " belong to? "))
          (until done do (groupName←(AFFIRMUSER NIL NIL question keyList T T))
                         (if groupName='/
                             then (if groupName←(APG)
                                      then done←T
                                    else (printout T .TAB0 0
                                                   "... that still leaves us with the question:"
                                                   T))
                           else done←T))
          (for g in UserProfileGroups
             do (if g:GroupName=groupName
                    then (g:EntryNames←(MERGEINSERT newVal:ProfileEntryName g:EntryNames T))
                  elseif newVal:ProfileEntryName MEMB g:EntryNames
                    then g:EntryNames←(DREMOVE newVal:ProfileEntryName g:EntryNames)
                         (printout T .TAB0 0 "(Removed" , newVal:ProfileEntryName , "from the" ,
                                   g:GroupName , "group)" T)))
          (DefinedEntryNames←(MERGEINSERT PEname DefinedEntryNames T))
          (RETURN newVal:ProfileEntryName])

                                                                                    2

(APG
  [LAMBDA (groupName)                                    (* R.Bates "23-Jan-80 12:58")
    (PROG (anchor blank c cr done groupNames groupNameKeyList groupString keyList newGroup position
                  question response tail)
          (blank←(FCHARACTER 32))
          (cr←(FCHARACTER 31))
          (groupNames←(for g in UserProfileGroups collect g:GroupName))
          (groupNameKeyList←(for g in groupNames collect <g NIL>))
          (until groupName←(APGgetName groupName groupNames) do (printout T .TAB0 0
                                                                          "What's the new Profile Group's name? ")
                                                                 (groupName←(RATOM)))
          (until done do (printout T .TAB0 0 "What's the display name for" , groupName "?" .)
                         (repeatwhile c=blank or c=cr do c←(READC))
```

```
                    (if c='%"
                        then groupString←(readString)
                             done←T
                      else (printout T .TAB0 0 "??")))
        [if groupName ~MEMB groupNames
            then question←(CONCAT "Where should the " groupName
                                  " group be positioned in the list of groups? ")
                 keyList←
                 <<'F "ront of list" > <'R "ear of list" > <'A "fter (group name) " 'EXPLAINSTRING
                                                          "After "
                                                          'KEYLST groupNameKeyList>
                   <'B "efore (group name) " 'EXPLAINSTRING "Before " 'KEYLST groupNameKeyList>>
                 response←(AFFIRMUSER NIL NIL question keyList T T)
                 position←(NTHCHAR response 1)
                 newGroup←(create UserProfileGroupInformation
                                  GroupName ← groupName
                                  DisplayName ← groupString
                                  EntryNames ← NIL
                                  HiddenEntryNames ← NIL)
                 (SELECTQ position
                          (A anchor← (PACK (UNPACK response)::1)
                             tail←
                             ((FASSOC anchor UserProfileGroups)
                              MEMB UserProfileGroups)
                             tail::1← <newGroup ! tail::1>)
                          (B anchor← (PACK (UNPACK response)::1)
                             tail←
                             (NLEFT UserProfileGroups 1 ((FASSOC anchor UserProfileGroups)
                                     MEMB UserProfileGroups))
                             (if tail
                                 then (tail::1← <newGroup ! tail::1>)
                               else UserProfileGroups← <newGroup ! UserProfileGroups>))
                          (F UserProfileGroups← <newGroup ! UserProfileGroups>)
                          (R UserProfileGroups← < !! UserProfileGroups newGroup>)
                          (PROG NIL
                                (printout T T "*** unexpected response! (" position ")" T)
                                (RETURN NIL]
        (RETURN groupName])
```

3

```
(APGgetName
  [LAMBDA (groupName groupNames)                    (* D Thompson "11-Jun-79 10:43")
    (PROG NIL
          (if groupName=NIL
              then (RETURN NIL)
            else (if (ATOM groupName)
                     then (if groupName MEMB groupNames
                              then (printout T .TAB0 0 groupName . "already exists!" .)
                                   (if (AFFIRMUSER NIL NIL
                                                   "Do you want to use it, rather than a new one? "
                                                    NIL T T)='Y
                                       then (RETURN groupName)
                                     else (RETURN NIL))
                            else (RETURN groupName))
                   else (printout T .TAB0 0 "What??" .. groupName "??  Please try again:" .)
                        (RETURN NIL])
```

4

```
(AtomList
  [LAMBDA (l)                                       (* D.Thompson " 5-Jun-79 10:33")
    (if (LISTP l)
        then (for a in l join (AtomList a))
      elseif l
        then <l>])
```

5

```
(canUserChangeProfileEntry?
  [LAMBDA (casedEntryName newValue dontNotifyUser)  (* D.Thompson "15-Jun-79 17:16")
                                                    (* The default, if nothing is explicitly said about a
                                                    profile entry, is to assume it is OK to change it)

    (PROG ((entryName (U-CASE casedEntryName)))
          (if GOODGUY
              then (RETURN T)
            else (RETURN (SELECTQ entryName
                                  (nilEntry NIL)
```

```
T])
```

6

```
(DefaultUserProfile
  [LAMBDA NIL                                          (* R.Bates "14-Dec-79 14:42")
    (PROG NIL

          (* * First, clear the current profile)

          (SETPROPLIST 'UserProfile (for pe on (GETPROPLIST 'UserProfile) by pe::2 join <pe:1 pe:2>
                                       when pe:1 MEMB SYSPROPS))

          (* * Next, individually set each profile entry to its default value, if any)

          (for pe on (GETPROPLIST 'UserProfileEntryDefinitions) by pe::2
             unless pe:1 MEMB SYSPROPS or pe:2:DefaultValue='NoDefault
             do (UserProfileSet pe:1 pe:2:DefaultValue T))

          (* * Finally, set anything else: any special things should be inserted here)

          (RootProfileName←(UserProfile 'UserProfileFileName))
          (ProfileAlreadyLoaded←NIL])
```

7

```
(definedEntryName?
  [LAMBDA (entryName)                                  (* D.Thompson " 5-Feb-80 12:41")

          (* * This routine acts as the spelling corrector for profile entry names.)

    (PROG (newName)
          (RETURN (if newName←(SuperSpell entryName KnownNames:ProfileEntries NIL T T NIL NIL)
                      then (getProfileEntryDef newName)
                    else NIL])
```

8

```
(displayProfileEntry
  [LAMBDA (entryName value queryMode)                  (* D.Thompson "14-Jun-79 11:36")
    (PROG (entryDef displayMethod)
          (entryDef←(getProfileEntryDef entryName))
          (displayMethod←entryDef:DisplayString)
          (if displayMethod='NormalDisplay or ((LISTP displayMethod)
                                               and (EVALA displayMethod
                                                          < <'newValue ! value> >))
              then (printout NIL .TAB (if queryMode='See
                                          then 5
                                        else 0)
                                 entryDef:ProfileEntryName . "is" . .PPV
                                 (if value=NIL or value='NoDefault
                                     then (PACK* "!!" 'undefined "!!")
                                   else value)
                                 T])
```

9

```
(formChangePairs
  [LAMBDA (paramList)                                  (* D.Thompson " 8-Jun-79 15:21")
    (PROG (action changePairs done p pendingPairs state)
          (p←paramList)
          (state←'A)
          (until done
             do (until p=NIL or p:1 ~MEMB '(. and) do p←p::1)
                (SELECTQ state
                         (A (if p
                                then (if (LITATOM p:1) and p:1~='=
                                         then action←1
                                              state←'B
                                       else action←0
                                            state←'A)
                              else action←3
                                   done←T))
```

```
                    (B (if p
                           then (if p:1='=
                                    then action←3
                                         state←'C
                                  elseif p:1 MEMB '(? . and)
                                    then action←4
                                         state←'A
                                  else action←2
                                       state←'A)
                                else action←4
                                     done←T))
                    (C (if p
                           then (if p:1='=
                                    then action←0
                                         state←'C
                                  elseif p:1 MEMB '(? . and)
                                    then action←4
                                         state←'A
                                  else action←5
                                       state←'D)
                                else action←4
                                     done←T))
                    (D (if p
                           then (if ~(LITATOM p:1) or p:1='?
                                    then action←0
                                         state←'D
                                  elseif p:1 MEMB '(. and)
                                    then action←8
                                         state←'A
                                  elseif p:1='=
                                    then action←7
                                         state←'C
                                  else action←6
                                       state←'B)
                                else action←8
                                     done←T))
                    (PROG NIL
                          (printout T T "**Internal error[1], form change pairs: param =" T .PPV
                                    paramList T "state =" . .PPV state T "action =" . .PPV
                                    action T "list ptr =" . .PPV p T)))
            (SELECTQ action
                    (1 pendingPairs← < (create ProfileChangePair
                                               ProfileEntryName ← p:1)
                       >)
                    (2 pendingPairs:1:NewValue←p:1 pendingPairs:1:Wish←'Modify changePairs← < !!
                       changePairs ! pendingPairs> pendingPairs←NIL)
                    (3 NIL)
                    (4 (for c in pendingPairs do (c:Wish←'Query))
                       changePairs← < !! changePairs ! pendingPairs> pendingPairs←NIL)
                    (5 pendingPairs:-1:NewValue←p:1 pendingPairs:-1:Wish←'Modify)
                    (6 (for c in pendingPairs do (c:Wish←'Modify)
                                                 (c:NewValue←pendingPairs:-1:NewValue))
                       changePairs← < !! changePairs ! pendingPairs> pendingPairs← <
                       (create ProfileChangePair
                               ProfileEntryName ← p:1)
                       >)
                    (7 pendingPairs← < !! pendingPairs (create ProfileChangePair
                                                               ProfileEntryName ←
                                                               pendingPairs:-1:NewValue)
                       >)
                    (8 (for c in pendingPairs do (c:Wish←'Modify)
                                                 (c:NewValue←pendingPairs:-1:NewValue))
                       changePairs← < !! changePairs ! pendingPairs> pendingPairs←NIL)
                    (0 (printout NIL . "%"" p:1 "%"?" .))
                    (PROG NIL
                          (printout T T "**Internal error [2], form change pairs: param =" T
                                    .PPV paramList T "action =" . .PPV action T "state =" . .PPV
                                    state T "list ptr =" . .PPV p T)))
            (p←p::1))
        (RETURN changePairs])
```

10

(getNewValue

```
  [LAMBDA (entryName currentValue proposedValue)          (* D.Thompson "26-Feb-80 12:13")
```

(* This routine obtains a value from the user and validates it. It also will attempt to quietly validate a proposed value, in which case it avoids the user interaction.)*

```
    (PROG (category done entryDef keyList newValue optionList possibleValues question)
          (entryDef←(getProfileEntryDef entryName))
          (possibleValues←(getProfilePossibleValues entryDef))
          (category←possibleValues:1)
          (possibleValues←(AtomList possibleValues::1))
          (if proposedValue←(validateValue entryDef:ProfileEntryName proposedValue currentValue
                                           category possibleValues NIL)
              then (RETURN proposedValue))
          (question←(gnvQuestion category possibleValues entryName entryDef))
          (keyList←(gnvKeyList category possibleValues entryName entryDef))
          (optionList← NIL)
          (until done do (newValue←(AFFIRMUSER NIL T question keyList T T optionList))
                         (if newValue
                             then (if newValue←(validateValue entryDef:ProfileEntryName newValue
                                                              currentValue category possibleValues
                                                              NIL)
                                      then done←T)
                           elseif HelpWanted
                             then HelpWanted←NIL
                                  (AFFIRMHelp 'Profile entryName)
                           else newValue←currentValue
                                done←T))
          (RETURN newValue])
```

11

```
(getProfileAssociatedVariables
  [LAMBDA (entryDef)                                    (* D Thompson " 6-Aug-79 10:25")
    (PROG (associatedVariables)
          (associatedVariables←entryDef:AssociatedVariables)
          (if associatedVariables=NIL or (ATOM associatedVariables)
                and (U-CASE associatedVariables)='NOASSOCIATEDVARIABLES
              then (RETURN NIL)
            elseif (LISTP associatedVariables)
              then (RETURN associatedVariables)
            else (reportProfileDBError NIL entryDef)
                 (RETURN NIL])
```

12

```
(getProfileConditions
  [LAMBDA (entryDef)                                    (* D.Thompson " 4-Oct-79 09:08")
    (PROG (conditions)
          (conditions←entryDef:Conditions)
          (if conditions=NIL or (ATOM conditions) and (U-CASE conditions)='NOCONDITIONS
              then (RETURN NIL)
            elseif (LISTP conditions) and ((FGETD conditions:1) or conditions:1 MEMB CLISPIFWORDSPLST
                                           or conditions:1 MEMB CLISPFORWORDSPLST)
              then (RETURN conditions)
            else (reportProfileDBError NIL entryDef)
                 (RETURN NIL])
```

13

```
(getProfileDefaultValue
  [LAMBDA (entryDef)                                    (* D.Thompson " 2-Sep-80 10:39")
    entryDef:DefaultValue])
```

14

```
(getProfileEntryDef
  [LAMBDA (casedEntryName)                              (* D.Thompson " 5-Jun-79 10:38")
    (PROG (entryDef (entryName (U-CASE casedEntryName)))
          (if (entryDef←(GETPROP 'UserProfileEntryDefinitions entryName))
              then (RETURN entryDef)
            else (reportProfileDBError casedEntryName NIL)
                 (RETURN (create UserProfileEntryInformation ProfileEntryName←casedEntryName
                                 DisplayString←'NormalDisplay
                                 PossibleValues←'OnOffKeyList
                                 DefaultValue←NIL
                                 SubKeys←'NoSubKeys
                                 Conditions←'NoConditions
                                 AssociatedVariables←'NoAssociatedVariables])
```

15

```
(getProfilePossibleValues
  [LAMBDA (entryDef)                                    (* D.Thompson " 4-Oct-79 09:05")
    (PROG (possibleValues)
          (possibleValues←entryDef:PossibleValues)
          (if (ATOM possibleValues)
              then (possibleValues←(U-CASE possibleValues))
                   [SELECTQ possibleValues
                            ((FILENAME KEYLIST ONOFFKEYLIST SPECIAL STRING)
                              (RETURN <possibleValues>))
                            (NUMBER (RETURN '(NUMBER NoLow NoHigh)))
                            (PROG NIL
                                  (reportProfileDBError NIL entryDef)
                                  (RETURN '(ONOFFKEYLIST]
            elseif (LISTP possibleValues)
              then (RETURN <(U-CASE possibleValues:1) ! possibleValues::1>)
            else (reportProfileDBError NIL entryDef)
                 (RETURN '(ONOFFKEYLIST])
```

16

```
(gnvKeyList
  [LAMBDA (category possibleValues entryName entryDef)      (* D.Thompson " 5-Feb-80 15:28")

          (* * This routine builds the key list for the profile interactive mode.)


    < !(if category~='NUMBER
           then (for v in possibleValues collect <v NIL>))
     !(SELECTQ category
               (FILENAME (if possibleValues::1
                             else <<'↑[ NIL 'EXPLAINSTRING "(a new file name)" >>))
               (KEYLIST NIL)
               (NUMBER <<'↑[ NIL 'EXPLAINSTRING "(a new integer)" >>)
               (ONOFFKEYLIST OnOffKeyList)
               (SPECIAL (if possibleValues::1
                            else <<'↑[ NIL 'EXPLAINSTRING "(a new value)" >>))
               [STRING (if possibleValues::1
                           else '((%" NIL EXPLAINSTRING "(a new string)" RETURN (readString]
               (PROGN (reportProfileDBError entryName entryDef)
                      NIL))
     ! SameOldKeyList>])
```

17

```
(gnvQuestion
  [LAMBDA (category possibleValues entryName entryDef)      (* D.Thompson " 5-Feb-80 15:28")

          (* * This routine builds the question for the profile package interactive mode.)


    (SELECTQ category
             (FILENAME "New file name: ")
             ((KEYLIST ONOFFKEYLIST SPECIAL)
               "New value: ")
             (NUMBER                                        (* low value = possibleValues:1;
                                                               high value = possibleValues:2)
                     (CONCAT "New integer value" (if (FIXP possibleValues:1)
                                                     then (if (FIXP possibleValues:2)
                                                              then (CONCAT " between "
                                                                           possibleValues:1 " and "
                                                                           possibleValues:2)
                                                              else (CONCAT " greater than or equal to "
                                                                           possibleValues:1))
                                                     else (if (FIXP possibleValues:2)
                                                              then (CONCAT " less than or equal to "
                                                                           possibleValues:2)
                                                              else ""))
                                 ": "))
             (STRING "New string value: ")
             (PROGN (reportProfileDBError entryName entryDef)
                    "New value: "])
```

18

```
(InitializeUserProfile
  [LAMBDA NIL                                           (* D.Thompson "26-Jan-80 12:29")

          (* * This routine attempts to load the user's profile file, if necessary, when a session first begins, and when a
```

frozen system is restarted. If there is no profile file, one with all default values is established.)

```
(PROG (currentFileName done fileNames question)
      [if ProfileAlreadyLoaded
          then                                          (* restarting a frozen system: a profile is present.
                                                        Should the user's profile file be loaded?)
               (ReInitializeUserProfile)
               (if (LITATOM RootProfileName) and fileNames←(INFILEP (PACKFILENAME 'DIRECTORY
                                                                                  USERNAME 'BODY
                                                                                RootProfileName))
                   then (AffirmNews fileNames))
               (if (UserProfile 'EnquireAfterFreeze T)
                   then (UserProfileEnquiry))
        else                                            (* A new session is just beginning.
                                                        Find and process the user's profile file)
             ProfileAlreadyLoaded←T
             currentFileName←(UserProfile 'UserProfileFileName)
             fileNames← <currentFileName>
             (if (ReadUserProfileFile currentFileName T T)
                 then                                   (* The user had a profile file, and it read in OK.
                                                        Now process any file chainings.)
                      (eachtime currentFileName←(UserProfile 'UserProfileFileName)
                         until done or ~(UserProfile 'ReadAnotherProfileFile T)
                                 or currentFileName MEMB fileNames
                         do (fileNames← <currentFileName ! fileNames>)
                            (if (ReadUserProfileFile currentFileName T NIL)
                              else (printout NIL .TAB0 0 "Profile file" , currentFileName ,
                                             "isn't usable."
                                             T)
                                   done←T))
                      (if (UserProfile 'EnquireInitially T)
                          then (UserProfileEnquiry))
               else                                     (* there is no profile file: establish one full of
                                                        defaults)
                    (PrintNews 0)
                    (printout NIL .TAB0 0 "... Establishing default user profile, file" , #
                              (WriteUserProfileFile currentFileName]
      (reportUndefinedEntryNames])
```

19

```
(IsInKeyList?
  [LAMBDA (casedValue keyList)                          (* D.Thompson "  5-Jun-79 09:57")
    (PROG ((value (U-CASE casedValue))
           foundValue)
          (for key in keyList until foundValue do (if value=(U-CASE key)
                                                      then foundValue←key))
          (RETURN foundValue])
```

20

```
(performAssociatedUpdates
  [LAMBDA (casedEntryName value)                        (* D.Thompson "  6-Aug-79 10:42")
    (PROG (associatedVariables entryDef (entryName (U-CASE casedEntryName)))
          (entryDef←(getProfileEntryDef entryName))
          (if associatedVariables←(getProfileAssociatedVariables entryDef)
              then (for av in associatedVariables do (if av and (LITATOM av)
                                                         then (SET av value)
                                                       elseif (LISTP av)
                                                         then (EVALA av <<'newValue ! value>>)
                                                       else (reportProfileDBError entryName entryDef])
```

21

```
(profile
  [LAMBDA (param)                                       (* D.Thompson "  1-Nov-79 16:36")

          (* * This routine oversees the user interaction with the user profile package. -
          An AFFIRM command function.)


    (PROG (currentValue entryDef entryName)
          (if param
              then (for p in (formChangePairs param)
                      do (if entryDef←(definedEntryName? p:ProfileEntryName)
                             then entryName←entryDef:ProfileEntryName
                                  currentValue←(UserProfile entryName)
```

```
                    (if p:Wish='Modify
                        then (if (canUserChangeProfileEntry? entryName currentValue T)
                                 then (UserProfilePut entryName
                                                      (getNewValue entryName
                                                                   currentValue
                                                                   p:NewValue))
                                      (printout NIL .TAB0 5 entryName ":" , .PPV
                                                (UserProfile entryName)
                                                T)
                                 else (printout NIL .TAB0 2 "**" , entryName ,
                                                "can't be changed!"
                                                T))
                             else (printout NIL .TAB0 5 entryName , "=" , currentValue T))
                    else (printout NIL .TAB0 2 "??" , p:ProfileEntryName , "??" T)))
    else (UserProfileEnquiry])
```

22

(ReadUserProfileFile

```
  [LAMBDA (fileName dontReportUndefinedYet RootProfile)      (* R.Bates "19-Aug-80 15:55")
    (PROG ((Open T))
          (if RootProfile
              then fileName←(PACKFILENAME 'DIRECTORY USERNAME 'BODY fileName))
          (if (LITATOM fileName) and fileName←(INFILEP fileName)
              then (if (UserProfile 'UserProfileTrace T)
                       then (printout NIL .TAB0 0 "(... reading User Profile File" , fileName ")"))
                   (PROG (HELPFLAG)
                         (OR (if RootProfile
                                 then (ERSETQ fileName←(OPENFILE fileName 'BOTH)))
                             (ERSETQ (PROGN Open←'ERROR
                                            fileName←(OPENFILE fileName 'INPUT)))
                             fileName←NIL))
                   (if fileName=NIL
                       then (RETURN 'ABORTED))
                   (for pe in (READ fileName) do (UserProfileSet pe:1 pe:2))
                   (if RootProfile
                       then (AffirmNews fileName Open))
                   (CLOSEF fileName)
                   (if (UserProfile 'UserProfileTrace T)
                       then (printout NIL .TAB0 0 "(... finished with file" , fileName ")"))
                   (if dontReportUndefinedYet
                     else (reportUndefinedEntryNames))
                   (RETURN fileName)
          else (RETURN NIL])
```

23

(ReInitializeUserProfile

```
  [LAMBDA NIL                                                (* D.Thompson " 6-Aug-79 10:15")
    (for p on (GETPROPLIST 'UserProfileEntryDefinitions) by p::2 unless p:1 MEMB SYSPROPS
                                                                 or (
getProfileAssociatedVariables p:2)=NIL
       do (UserProfileSet p:1 (if (UserProfile p:1)
                                  else p:2:DefaultValue)
                          NIL])
```

24

(reportProfileDBError

```
  [LAMBDA (entryName entryDef)                               (* D.Thompson " 6-Feb-80 13:07")
    (if entryName = NIL
        then (if entryDef:ProfileEntryName = NIL
                 then (printout NIL .TAB0 0 "*** A profile entry is quite screwed up!!" T
                                "entryName is NIL ..."
                                T "entry definition =" T .PPV entryDef T)
             else (entryName←entryDef:ProfileEntryName)))
    (Unexpected 'ProfileEntryName])
```

25

(reportUndefinedEntryNames

```
  [LAMBDA NIL                                                (* D.Thompson " 5-Jun-79 10:38")
    (PROG (undefinedEntryNames)
          (undefinedEntryNames←(LDIFFERENCE (LDIFFERENCE (PROPNAMES 'UserProfile)
                                                         (PROPNAMES 'UserProfileEntryDefinitions))
                                            SYSPROPS))
          (if undefinedEntryNames
```

```
        then (printout NIL .TAB0 0 "The following User Profile entry names" T
                       "are not defined in the current version of AFFIRM:")
             (for name in undefinedEntryNames do (printout NIL .TAB0 5 name ":" , .PPV
                                                           (UserProfile name)
                                                           T))

             (RETURN T)
        else (RETURN NIL])
```

26

(saveUserProfile?

```
  [LAMBDA NIL                                        (* D.Thompson " 5-Sep-80 20:16")
```

(* This routine asks the user (at the end of an interactive profile dialogue) wether the current profile should be saved. and then if necessary calls the profile save routine.)*

```
    (PROG (keyList optionList question response fileName)
          (keyList←NIL)
          (optionList←NIL)
          (question←"Do you wish to save this profile? ")
          (printout NIL T)
          (if response←(AFFIRMUSER NIL 'N question keyList T NIL optionList)='Y
              then (if fileName←(getFileName NIL (UserProfile 'UserProfileFileName)
                                             "Profile file: " 'OUTPUT)
                       then (printout NIL .TAB0 0 "(Using" , fileName ")" T)
                            (WriteUserProfileFile fileName)
                     else (printout NIL .TAB0 0 "No default file!!" T])
```

27

(SpecialValueOK?

```
  [LAMBDA (entryName currentValue proposedValue)     (* D.Thompson " 3-Mar-80 15:27")
    proposedValue])
```

28

(UserProfile

```
  [LAMBDA (casedEntryName makeItBoolean)             (* R.Bates " 2-Oct-79 13:16")
                                                     (* There is compiler macro for this function)
    (PROG (value (entryName (U-CASE casedEntryName)))
          (if value←(GETPROP 'UserProfile entryName)
              then (if makeItBoolean
                       then (RETURN (value ~MEMB OffValues))
                     else (RETURN value))
            else (RETURN NIL])
```

29

(UserProfileDefault

```
  [LAMBDA (entryName)                                (* D.Thompson " 2-Sep-80 10:37")
    (getProfileDefaultValue (getProfileEntryDef entryName])
```

30

(UserProfileEnquiry

```
  [LAMBDA NIL                                        (* D.Thompson " 5-Sep-80 20:18")
```

(* This routine drives the interactive profile dialogue. Each family of profile entries can be displayed and updated. -*
with due apologies to XED mode file dialogue.)

```
    (PROG (done fileName keyList mainKeyList optionList queryKeyListHead queryKeyListTail question
                response value)
          (queryKeyListTail←'((No NIL)
             (Yes " (Modify)" RETURN 'Modify)
             (Exit NIL)))
          (queryKeyListHead←'((Modify NIL)
             (See NIL)))
          (mainKeyList← < ! queryKeyListHead ! queryKeyListTail>)
          (EnquiryAborted←NIL)
          (question←"Do you want to Modify, See, or Read your profile? ")
          (keyList← < ! queryKeyListHead '(Read NIL) ! queryKeyListTail>)
          (optionList←'(EXPLAINDELIMITER ", " AUTOCOMPLETEFLG T))
          (while response←(AFFIRMUSER NIL 'See question keyList T T optionList)='Read
```

```
        do (done←NIL)
           (until done do (if fileName←(getFileName NIL (UserProfile 'UserProfileFileName)
                                                        "Profile file: ")=NIL
                              then (printout NIL .TAB0 0 "No default file!!" T)
                            else (printout NIL .TAB0 0 "(Reading User Profile from" . fileName
                                                       ")"
                                                       T)
                                 done←T))
           (ReadUserProfileFile fileName))
     (if response='Exit
         then (RETURN)
       elseif response='See
         then (for group in UserProfileGroups
                 do (printout NIL T T group:DisplayName ":" T)
                    (for entry in (if (UserProfile 'DisplayHiddenEntries T)
                                      then < ! group:EntryNames ! group:HiddenEntryNames>
                                    else group:EntryNames)
                       do (displayProfileEntry entry (UserProfile entry)
                                               'See)))
              (printout NIL T)
              question←"NOW do you want to modify the current profile? " response←(AFFIRMUSER
                NIL 'No question queryKeyListTail T T optionList)
              (printout NIL T)
              (if response='Exit
                  then (RETURN)))
     [if response='Modify
         then done←NIL
              (for group in UserProfileGroups until done or EnquiryAborted
                 do (question←(CONCAT "Do you wish to Modify or See " group:DisplayName "? "))
                    (if response←(AFFIRMUSER NIL 'See question mainKeyList T T optionList)='Exit
                        then done←T
                      elseif response='See
                        then (for entry in (if (UserProfile 'DisplayHiddenEntries T)
                                               then < ! group:EntryNames
                                                      ! group:HiddenEntryNames>
                                             else group:EntryNames)
                                do (displayProfileEntry entry (UserProfile entry)
                                                        'See))
                             question←(CONCAT "NOW do you want to modify " group:DisplayName
                                              "? ")
                             response←(AFFIRMUSER NIL 'No question queryKeyListTail T T
                                                  optionList)
                             (printout NIL T)
                             (if response='Exit
                                 then (RETURN)))
                    (if response='Modify
                        then [for entry in (if (UserProfile 'DisplayHiddenEntries T)
                                               then < ! group:EntryNames
                                                      ! group:HiddenEntryNames>
                                             else group:EntryNames)
                                until EnquiryAborted
                                do (value←(UserProfile entry))
                                   (if (canUserChangeProfileEntry? entry value T)
                                       then (displayProfileEntry entry value 'Modify)
                                            (UserProfilePut entry (getNewValue entry value NIL]
                             (printout NIL T]
     (saveUserProfile?])
```

31

(UserProfilePut

```
  [LAMBDA (casedEntryName value)                    (* R.Erickson "12-May-80 13:32")

          (* * For internal use only. assumes values have been checked.)


     (PROG ((entryName (U-CASE casedEntryName)))
           (UNDOSAVE <'UserProfilePut entryName (GETPROP 'UserProfile entryName)
                     >)                             (* Make undoable. To get old value quickly, we cheat
                                                       instead of calling UserProfile)
           (PUTPROP 'UserProfile entryName value)
           (performAssociatedUpdates entryName value)
           (RETURN value])
```

32

(UserProfileSet

```
  [LAMBDA (casedEntryName value dontNotifyUser)     (* D Thompson "16-Jun-79 10:59"
    (PROG (category currentValue possibleValues (entryName (U-CASE casedEntryName)))
```

```
        (currentValue←(UserProfile entryName))
        (possibleValues←(getProfilePossibleValues (getProfileEntryDef entryName)))
        (category←possibleValues:1)
        (possibleValues←(AtomList possibleValues::1))
        (if (canUserChangeProfileEntry? entryName value dontNotifyUser)
              and (validateValue casedEntryName value currentValue category possibleValues
                                 dontNotifyUser)
            then (UserProfilePut entryName value)
                 (RETURN value)
          else (RETURN NIL])
```

33

(validateValue

```
  [LAMBDA (entryName proposedValue currentValue category possibleValues dontNotifyUser)
                                                          (* D Thompson "12-Jul-79 17:04")
    (PROG (newValue)
          (if proposedValue
              then (SELECTQ category
                            (FILENAME newValue←proposedValue)
                            (KEYLIST (if newValue←(IsInKeyList? proposedValue possibleValues)
                                       elseif dontNotifyUser
                                       else (printout NIL .TAB0 5 entryName . "can't be set to" .
                                                      proposedValue "." T)))
                            [NUMBER (if (FIXP proposedValue)
                                        then (if (FIXP possibleValues:1)
                                                   and possibleValues:1 gt proposedValue
                                                 then newValue←NIL
                                                      (if dontNotifyUser
                                                        else (printout NIL .TAB0 5 entryName .
                                                                   "must be greater than or equal to"
                                                                   . possibleValues:1 "." T))
                                               elseif (FIXP possibleValues:2)
                                                   and possibleValues:2 lt proposedValue
                                                 then newValue←NIL
                                                      (if dontNotifyUser
                                                        else (printout NIL .TAB0 5 entryName .
                                                                   "must be less than or equal to"
                                                                   . possibleValues:2 "." T))
                                               else newValue←proposedValue)
                                      else newValue←NIL
                                           (if dontNotifyUser
                                             else (printout NIL .TAB0 5 entryName .
                                                            "must be an integer."
                                                            T]
                            (ONOFFKEYLIST (if newValue←(IsInKeyList? proposedValue
                                                                     < ! possibleValues
                                                                       !(for v in OnOffKeyList
                                                                           collect v:1)
                                                                       >)
                                            elseif dontNotifyUser
                                            else (printout NIL .TAB0 5 entryName . "can't be set to"
                                                           . proposedValue "." T)))
                            (SPECIAL newValue← (SpecialValueOK? entryName currentValue proposedValue))
                            [STRING (if (STRINGP proposedValue)
                                        then newValue←proposedValue
                                      else newValue←NIL
                                           (if dontNotifyUser
                                             else (printout NIL .TAB0 5 entryName .
                                                            "must be a string."
                                                            T]
                            (PROGN newValue←NIL))
            else newValue←NIL)
          (RETURN newValue])
```

34

(WriteUserProfileFile

```
  [LAMBDA (fileName)                                      (* D.Thompson "20-Sep-79 12:16")
    (PROG (currentValues def saveOnlyChangedEntries FONTCHANGEFLG)
          (saveOnlyChangedEntries←(UserProfile 'SaveOnlyChangedEntries T))
          (for pe on (GETPROPLIST 'UserProfile) by pe::2 unless pe:1 MEMB SYSPROPS
             do (def←(getProfileEntryDef pe:1))
                (if saveOnlyChangedEntries and (EQUAL pe:2 def:DefaultValue)
                  else currentValues← < ! currentValues <def:ProfileEntryName pe:2>>))
          (if (OKtoUseCOMS? fileName)
              then fileName←(OPENFILE (makeFileName fileName)
                                      'OUTPUT)
                   (printout fileName .PPV currentValues T ('STOP)
```

```
                    T)
                (if (NUMBERP (EVALV 'NewsDate))
                    then (printout fileName NewsDate T))
                (CLOSEF fileName)
                (reportUndefinedEntryNames)
                (markFileNameAsUsed fileName 'UserProfile)
                (printout NIL .TAB0 0 fileName T)
            else (printout NIL .TAB0 0 "...not saving User Profile on" , fileName])
)
[DECLARE: DONTEVAL@LOAD DONTCOPY
```

(* Profile Data Base)]

```
(PUTPROPS UserProfileEntryDefinitions ACTIVATEINTERLISPENVIRONMENT (ActivateInterLispEnvironment
                                                                    NormalDisplay OnOffKeyList Off
                                                                    NoSubKeys NoConditions
                                                                    NoAssociatedVariables)
                                      ANNOTATINGTRANSCRIPT (AnnotatingTranscript NormalDisplay
                                                                                 OnOffKeyList Off
                                                                                 NoSubKeys
                                                                                 NoConditions
                                                                                 ((Annotating)))
                                      AUTOCASES (AutoCases NormalDisplay (OnOffKeyList (Ask Tell))
                                                           Off NoSubKeys NoConditions
                                                           NoAssociatedVariables)
                                      AUTORESUME (AutoResume NormalDisplay OnOffKeyList Off
                                                             NoSubKeys NoConditions
                                                             NoAssociatedVariables)
                                      AUTOTRANSCRIPT (AutoTranscript NormalDisplay OnOffKeyList On
                                                                     (AutoLoggingOfTranscripts
                                                                       TranscriptFileName)
                                                                     NoConditions
                                                                     NoAssociatedVariables)
                                      AXIOMGROUPING (AxiomGrouping NormalDisplay OnOffKeyList On
                                                                   NoSubkeys NoConditions
                                                                   NoAssociatedVariables)
                                      BREAKACCESS (BreakAccess NormalDisplay OnOffKeyList Off
                                                               NoSubKeys NoConditions
                                                               NoAssociatedVariables)
                                      CAUTIOUSCOMPLETION [CautiousCompletion
                                                          NormalDisplay OnOffKeyList Off NoSubKeys
                                                          NoConditions
                                                          ((SETQ CautiousCompletion
                                                                 (UserProfile (QUOTE
                                                                                CautiousCompletion)
                                                                              T]
                                      CAUTIOUSSUBEXPSPEC (CautiousSubExpSpec NormalDisplay
                                                                             OnOffKeyList Off
                                                                             NoSubKeys NoConditions
                                                                           NoAssociatedVariables)
                                      DEFINEGROUPING (DefineGrouping NormalDisplay OnOffKeyList On
                                                                     NoSubKeys NoConditions
                                                                     NoAssociatedVariables)
                                      DONTASKJUSTTAKE (DontAskJustTake NormalDisplay (Number 0 100)
                                                                       40 NoSubKeys NoConditions
                                                                       (DontAskJustTake))
                                      ENQUIREAFTERFREEZE (EnquireAfterFreeze NormalDisplay
                                                                             OnOffKeyList Off
                                                                             NoSubKeys NoConditions
                                                                           NoAssociatedVariables)
                                      ENQUIREINITIALLY (EnquireInitially NormalDisplay OnOffKeyList
                                                                         Off NoSubKeys NoConditions
                                                                         NoAssociatedVariables)
                                      EXPERTUSER (ExpertUser NormalDisplay OnOffKeyList Off
                                                             NoSubKeys NoConditions
                                                             NoAssociatedVariables)
                                      FULLYPARENTHESIZE (FullyParenthesize NormalDisplay
                                                                           OnOffKeyList Off
                                                                           NoSubKeys NoConditions
                                                                           (FullyParenthesize))
                                      GARBAGECOLLECTIONMESSAGE (GarbageCollectionMessage
                                                                NormalDisplay
                                                                (OnOffKeyList (Normal None Compact)
                                                                              )
                                                                Off
                                                                (GarbageCollectionPages)
                                                                NoConditions
                                                                ((setGCmessage newValue)))
                                      GARBAGECOLLECTIONPAGES (GarbageCollectionPages
                                                              NormalDisplay
                                                              (Number 0 NoHigh)
                                                              40 NoSubKeys
                                                              (if (UserProfile (QUOTE
                                                                                 GarbageCollectionMessage))
                                                                  ~=
                                                                  (QUOTE None))
                                                              ((setGCpages newValue)))
                                      HISTORYWINDOWSIZE [HistoryWindowSize
                                                         NormalDisplay
                                                         (Number 3 NoHigh)
                                                         30 NoSubKeys NoConditions
                                                         ((COND ((FIXP newValue)
```

```
                                    (PROG (largestEventNumber)
                                          (SETQ largestEventNumber
                                                (CADDDR LISPXHISTORY))
                                          (CHANGESLICE newValue
                                                       LISPXHISTORY)
                                          (FRPLACA (CDDDR
                                                       LISPXHISTORY)
                                                   largestEventNumber]
IMPLICITECOMMAND [ImplicitECommand
                  NormalDisplay OnOffKeyList Off NoSubKeys
                  NoConditions
                  ((SETQ ImplicitE
                         (UserProfile (QUOTE
                                          ImplicitECommand)
                                      T]
INTERFACEGROUPING (InterfaceGrouping NormalDisplay
                                     OnOffKeyList On
                                     NoSubKeys NoConditions
                                     NoAssociatedVariables)
LEMMAGROUPING (LemmaGrouping NormalDisplay OnOffKeyList On
                             NoSubKeys NoConditions
                             NoAssociatedVariables)
LESSOUTPUTDESIRED [LessOutputDesired
                   NormalDisplay OnOffKeyList On NoSubKeys
                   NoConditions
                   ((SETQ LessOutputDesired
                          (UserProfile (QUOTE
                                           LessOutputDesired)
                                       T]
LISPACCESS (LispAccess NormalDisplay OnOffKeyList On
                       NoSubKeys NoConditions
                       NoAssociatedVariables)
LISPBREAKACCESS (LISPBreakAccess NormalDisplay OnOffKeyList
                                 Off NoSubKeys NoConditions
                                 NoAssociatedVariables)
LISPGARBAGECOLLECTIONMESSAGE (LISPGarbageCollectionMessage
                              NormalDisplay
                              (KeyList (Normal None Compact))
                              Normal NoSubKeys NoConditions
                              NoAssociatedVariables)
LISPGARBAGECOLLECTIONPAGES (LISPGarbageCollectionPages
                            NormalDisplay
                            (Number 0 NoHigh)
                            40 NoSubKeys NoConditions
                            NoAssociatedVariables)
LISPHISTORYWINDOWSIZE (LISPHistoryWindowSize NormalDisplay
                                             (Number 3 NoHigh)
                                             30 NoSubKeys
                                             NoConditions
                                      NoAssociatedVariables)
LISPTERMINALLINEWIDTH (LISPTerminalLineWidth NormalDisplay
                                             (Number 20 132)
                                             79 NoSubKeys
                                             NoConditions
                                    NoAssociatedVariables)
LISTAPPLIEDEXPRS (ListAppliedExprs NormalDisplay OnOffKeyList
                                   Off NoSubKeys NoConditions
                                   NoAssociatedVariables)
LOADMETHOD (LoadMethod NormalDisplay (KeyList (Slow Fast))
                       Fast NoSubKeys NoConditions
                       NoAssociatedVariables)
PAGESIZE (PageSize NormalDisplay (Number NoLow NoHigh)
                   24 NoSubKeys
                   (COND ((UserProfile (QUOTE TerminalPaging)
                                       T)))
                   NoAssociatedVariables)
READANOTHERPROFILEFILE (ReadAnotherProfileFile NormalDisplay
                                               OnOffKeyList On
                                                 NoSubKeys
                                               NoConditions
                                      NoAssociatedVariables)
REPORTFLAG [ReportFlag NormalDisplay OnOffKeyList Off
                       NoSubKeys NoConditions
                       ((SETQ REPORTFLAG
                              (UserProfile (QUOTE REPORTFLAG)
                                           T]
RULELHSPERCENTAGE (RuleLHSPercentage NormalDisplay
                                     (Number 0 100)
                                     49 NoSubKeys
                                     NoConditions
                                     (RuleLHSPercentage))
```

```
SAVEMETHOD (SaveMethod NormalDisplay (KeyList (Slow Fast))
                       Fast NoSubKeys NoConditions
                       NoAssociatedVariables)
SAVEONLYCHANGEDENTRIES (SaveOnlyChangedEntries NormalDisplay
                                               OnOffKeyList On
                                                 NoSubKeys
                                                NoConditions
                                            NoAssociatedVariables)
SAVETYPEMETHOD (SaveTypeMethod NormalDisplay
                               (KeyList (Slow Fast))
                               Fast NoSubKeys NoConditions
                               NoAssociatedVariables)
SCHEMAGROUPING (SchemaGrouping NormalDisplay OnOffKeyList On
                               NoSubKeys NoConditions
                               NoAssociatedVariables)
SHOWRULESIMPLIFICATION (ShowRuleSimplification NormalDisplay
                                               OnOffKeyList On
                                                 NoSubKeys
                                                NoConditions
                                            NoAssociatedVariables)
SUBEXPSPECCOMPLETION (SubExpSpecCompletion
                      NormalDisplay
                      (KeyList (Normal Verbose))
                      Normal NoSubKeys NoConditions
                      NoAssociatedVariables)
TERMINALLINEWIDTH (TerminalLineWidth NormalDisplay
                                     (Number 20 132)
                                     79 NoSubKeys
                                     NoConditions
                                     ((LINELENGTH newValue)))
TERMINALPAGEEJECTSEQUENCE (TerminalPageEjectSequence
                                                NormalDisplay
                                                       String " "
                                                   NoSubKeys
                                                  NoConditions
                                            NoAssociatedVariables)
TERMINALPAGING (TerminalPaging NormalDisplay OnOffKeyList Off
                               (PageSize)
                               NoConditions
                               NoAssociatedVariables)
TERMINALTYPE (TerminalType NormalDisplay
                           (KeyList (Paper Screen))
                           Screen NoSubKeys NoConditions
                           NoAssociatedVariables)
TEXTEDITOR (TextEditor NormalDisplay
                       (KeyList (SOS TECO XED EMACS RMODE TED
                                 POET))
                       XED NoSubKeys NoConditions
                       NoAssociatedVariables)
TIMER [Timer NormalDisplay OnOffKeyList Off NoSubKeys
             NoConditions ((SETQ Timer
                                 (UserProfile (QUOTE Timer)
                                              T]
TRANSCRIPTFILENAME (TranscriptFileName
                    NormalDisplay FileName AFFIRMTRANSCRIPT
                    NoSubKeys
                    (if (UserProfile (QUOTE AutoTranscript)
                                     T))
                    NoAssociatedVariables)
TYPESININTERFACES (TypesInInterfaces NormalDisplay
                                     (KeyList (Types
                                                Variables))
                                     Variables NoSubKeys
                                     NoConditions
                                     NoAssociatedVariables)
USERPROFILEFILENAME (UserProfileFileName NormalDisplay
                                         FileName
                                        --AFFIRMUserProfile--
                                         NoSubKeys
                                         NoConditions
                                     NoAssociatedVariables)
USERPROFILETRACE (UserProfileTrace NormalDisplay OnOffKeyList
                                   Off NoSubKeys NoConditions
                                   NoAssociatedVariables)
COMMANDLINEBUFFERING (CommandLineBuffering NormalDisplay
                                             OnOffKeyList On
                                             NoSubKeys
                                             NoConditions
                                         NoAssociatedVariables)
USERINITIALIZATIONFILENAME (UserInitializationFileName
                                              NormalDisplay
```

```
                                                       FileName
                                           --Initialization-AFFIRM--
                                                     NoSubKeys
                                                   NoConditions
                                             NoAssociatedVariables)
FREEZEFILENAME (FreezeFileName NormalDisplay FileName
                               Frozen-AFFIRM NoSubKeys
                               NoConditions (FreezeFileName))
KBAXIOMORDER (KBAxiomOrder NormalDisplay
                           (KeyList Chronological Inverse)
                           Inverse NoSubKeys NoConditions
                           NoAssociatedVariables)
LOADNEEDEDTYPES (LoadNeededTypes NormalDisplay
                                 (OnOffKeyList Ask)
                                 On NoSubKeys NoConditions
                                 NoAssociatedVariables)
NEEDSGROUPING (NeedsGrouping NormalDisplay OnOffKeyList On
                             NoSubKeys NoConditions
                             NoAssociatedVariables)
ERRORTOKENSOUTPUT (ErrorTokensOutput NormalDisplay
                                     (Number 1 NoHigh)
                                     15 NoSubKeys
                                     NoConditions
                                    NoAssociatedVariables)
USEORINPROPS (UseORinProps NormalDisplay OnOffKeyList On
                           NoSubKeys NoConditions
                           NoAssociatedVariables)
DUMMYVARNAME (DummyVarName NormalDisplay Special dummy
                           NoSubKeys NoConditions
                           NoAssociatedVariables)
COMPILEOPTION (CompileOption NormalDisplay
                             (KeyList (Redefine FileOnly))
                             Redefine NoSubKeys NoConditions
                             NoAssociatedVariables)
DISPLAYHIDDENENTRIES (DisplayHiddenEntries NormalDisplay
                                           OnOffKeyList Off
                                           NoSubKeys
                                           NoConditions
                                    NoAssociatedVariables)
NEWPP [NewPP NormalDisplay OnOffKeyList On NoSubKeys
             NoConditions ((SETQ PFFlag
                                 (UserProfile (QUOTE NewPP)
                                              T]
AVERAGENAMELENGTH [AverageNameLength
                    NormalDisplay
                    (Number 1 30)
                    5 NoSubKeys NoConditions
                    ((SETQ MagicCount
                           (UserProfile (QUOTE
                                         AverageNameLength]
USINGTED [UsingTed NormalDisplay OnOffKeyList Off NoSubKeys
                   NoConditions ((SETQ
                      UsingTed
                      (UserProfile (QUOTE UsingTed)
                                   T]
AUTOUNDO [AutoUndo NormalDisplay (OnOffKeyList (Ask Tell))
                   Off NoSubKeys NoConditions
                   ((AutoNotYet (QUOTE AutoUndo]
AUTOANNOTATE (AutoAnnotate NormalDisplay
                           (OnOffKeyList (Ask Tell))
                           Off NoSubKeys NoConditions
                           NoAssociatedVariables)
AUTOCOMPILE [AutoCompile NormalDisplay
                         (OnOffKeyList (Ask Tell))
                         Off NoSubKeys NoConditions
                         ((AutoNotYet (QUOTE AutoCompile]
AUTOFIX [AutoFix NormalDisplay (OnOffKeyList (Ask Tell))
                 Off NoSubKeys NoConditions
                 ((AutoNotYet (QUOTE AutoFix]
AUTOFREEZE (AutoFreeze NormalDisplay (OnOffKeyList
                         (Ask Tell))
                       Off NoSubKeys NoConditions
                       NoAssociatedVariables)
AUTOGENVCS (AutoGenvcs NormalDisplay (OnOffKeyList
                         (Ask Tell))
                       Off NoSubKeys NoConditions
                       NoAssociatedVariables)
AUTOINFIX [AutoInfix NormalDisplay (OnOffKeyList (Ask Tell))
                     Off NoSubKeys NoConditions
                     ((AutoNotYet (QUOTE AutoInfix]
AUTOINVOKEIH (AutoInvokeIH NormalDisplay
```

```
                                        (OnOffKeyList (Ask Tell))
                                        Off NoSubKeys NoConditions
                                        NoAssociatedVariables)
                    AUTONEXT (AutoNext NormalDisplay (OnOffKeyList (Ask Tell))
                                       Off NoSubKeys NoConditions
                                       NoAssociatedVariables)
                    AUTONORMINT [AutoNormint
                                 NormalDisplay
                                 (OnOffKeyList (Ask Tell))
                                 Off NoSubKeys NoConditions
                                 ((SETQ
                                    IntegerInterfaces
                                    (SORT
                                      (for i in (getInterfaces (QUOTE Integer))
                                           collect
                                           (MakeExtension
                                             (Shorten (fetch Operator of
                                                             (fetch LHS of i)))
                                             (QUOTE Integer)))
                                      (QUOTE UPPER]
                    AUTOPRINTPROOF (AutoPrintProof NormalDisplay
                                                   (OnOffKeyList (Ask Tell))
                                                   Off NoSubKeys NoConditions
                                                   NoAssociatedVariables)
                    AUTOPRINTPROOFTHEOREMS (AutoPrintProofTheorems
                                            NormalDisplay
                                            (OnOffKeyList (Ask Tell))
                                            Off NoSubKeys NoConditions
                                            NoAssociatedVariables)
                    AUTOREPLACE (AutoReplace NormalDisplay
                                             (OnOffKeyList (Ask Tell))
                                             Off NoSubKeys NoConditions
                                             NoAssociatedVariables)
                    AUTOSAVE [AutoSave NormalDisplay (OnOffKeyList (Ask Tell))
                                       Off NoSubKeys NoConditions
                                       ((AutoNotYet (QUOTE AutoSave]
                    AUTOSEARCH (AutoSearch NormalDisplay (OnOffKeyList
                                                          (Ask Tell))
                                           Off NoSubKeys NoConditions
                                           NoAssociatedVariables)
                    AUTOSUFFICIENT (AutoSufficient NormalDisplay
                                                   (OnOffKeyList (Ask Tell))
                                                   Off NoSubKeys NoConditions
                                                   NoAssociatedVariables)
                    SHOWRULES [ShowRules NormalDisplay OnOffKeyList Off NoSubKeys
                                         NoConditions ((SETQ
                                                        REPORTFLAG
                                                        (UserProfile (QUOTE ShowRules)
                                                                     T]
                    SHOWNORMINT [ShowNormint NormalDisplay OnOffKeyList Off
                                             NoSubKeys NoConditions
                                             ((SETQ ShowIntegerSimplification
                                                    (UserProfile (QUOTE
                                                                  ShowNormint)
                                                                 T])

[DECLARE: EVAL@COMPILE

(RECORD ProfileChangePair (ProfileEntryName NewValue Wish))

(RECORD UserProfileGroupInformation (GroupName DisplayName EntryNames HiddenEntryNames))
]
(KnownNames:ProfileEntries← (SORT (for pe on (GETPROPLIST (QUOTE UserProfileEntryDefinitions))
                                     by pe::2 unless pe:1 MEMB SYSPROPS collect
                                     pe:2:ProfileEntryName)
                                  (QUOTE UPPER)))

(RPAQ EnquiryAborted NIL)

(RPAQ HelpWanted NIL)

(RPAQQ OnOffKeyList ((False NIL RETURN (QUOTE Off))
                     (No NIL RETURN (QUOTE Off))
                     (Off NIL)
                     (On NIL)
                     (True NIL RETURN (QUOTE On))
                     (Yes NIL RETURN (QUOTE On))))

(RPAQQ OffValues (False NIL No Off))

(RPAQQ OnValues (On T True Yes))
```

```
(RPAQ ProfileAlreadyLoaded NIL)

(RPAQQ SameOldKeyList ((" " "(old value)" NOECHOFLG T EXPLAINSTRING "(space: old value)" RETURN NIL)
                       ("
" "(old value)" NOECHOFLG T EXPLAINSTRING "(carriage return: old value)" RETURN NIL)
                       ("Exit!" NIL RETURN (PROGN (SETQ EnquiryAborted T)
                                                  NIL))
                       ("Help!" NIL RETURN (PROGN (SETQ HelpWanted T)
                                                  NIL))))

(RPAQQ UserProfileGroups ((TerminalCharacteristics "your terminal characteristics" (TerminalLineWidth)
                                                    (LISPTerminalLineWidth PageSize
                                                                           TerminalPageEjectSequence
                                                                             TerminalPaging TerminalType)
                                                    )
        (PrintoutModes "the printout modes"
                       (LessOutputDesired AverageNameLength AxiomGrouping LemmaGrouping
                                          InterfaceGrouping DefineGrouping NeedsGrouping NewPP
                                          SchemaGrouping UseORinProps DummyVarName ErrorTokensOutput
                                          ReportFlag RuleLHSPercentage ShowNormint
                                          ShowRuleSimplification ShowRules TypesInInterfaces)
                       (FullyParenthesize))
        (History "the event history flags" (HistoryWindowSize)
                 (LISPHistoryWindowSize))
        (ExecInfo "the executive flags" (TextEditor Timer CompileOption UsingTed)
                  (CommandLineBuffering ExpertUser))
        (FileInformation "the information about files" (FreezeFileName LoadNeededTypes)
                         (LoadMethod SaveMethod))
        (SpellingCorrector "the spelling correction flags" (DontAskJustTake)
                           NIL)
        (TheoremProver "the theorem prover flags" (AutoResume ListAppliedExprs AutoCases)
                       (CautiousSubExpSpec SubExpSpecCompletion))
        (InterLisp "the Interlisp mode flags" (GarbageCollectionMessage GarbageCollectionPages
                                                                        BreakAccess)
                   (ActivateInterLispEnvironment ImplicitECommand LISPBreakAccess
                                                 LISPGarbageCollectionMessage
                                                 LISPGarbageCollectionPages
                                                 UserInitializationFileName))
        (TranscriptInfo "the transcript information" (AnnotatingTranscript TranscriptFileName)
                        (AutoTranscript))
        (ProfileInfo "the profile information" (SaveOnlyChangedEntries UserProfileFileName
                                                                       EnquireAfterFreeze
                                                                       EnquireInitially
                                                                       ReadAnotherProfileFile)
                     (DisplayHiddenEntries UserProfileTrace))
        (KnuthBendix "the Knuth-Bendix flags" (CautiousCompletion KBAxiomOrder)
                     NIL)
        (AutoMechanism "the automatically-applied event flags"
                       (AutoAnnotate AutoCompile AutoFix AutoFreeze AutoGenvcs AutoInfix AutoInvokeIH
                                     AutoNext AutoNormint AutoPrintProof AutoPrintProofTheorems
                                     AutoReplace AutoSave AutoSearch AutoSufficient AutoUndo)
                       NIL)))
[DECLARE: DONTEVAL@LOAD DONTCOPY
```

```
(* BLOCK Specifications) ]

(RPAQQ AUTOFNS (Auto2Abort AutoApplicable AutoCommand AutoMechExec1 AutoMechSpec1 AutoMechTP1
                       AutoMechTP2 AutoMechTP3 AutoMechTP4 AutoMechanism AutoNotYet AutoPerform?))
(DEFINEQ
```

35

```
(Auto2Abort
  [LAMBDA (event)                                          (* D.Thompson " 7-Aug-80 17:16")
    (printout NIL .TAB0 0 "... Aborting auto mechanism ..." T])
```

36

```
(AutoApplicable
  [LAMBDA (profileEntryName aux)                           (* D.Thompson "28-Oct-80 13:12")
    (SELECTQ profileEntryName
             (AutoAnnotate                                 (* proof of theorem is complete)
                       (Proved? aux))
             (AutoCases                                    (* current proposition contains embedded if-expressions)
                        (AND (type? Qexpression CurrentPropn)
                             (AnyInnerIfs CurrentPropn:expr NIL)))
             (AutoCompile                                  (* quitting this session and one or more types have been
                                                           newly defined or modified)
                          (AND CurrentCommand='quit AutoCompile))
             (AutoFix                                      (* always applicable. Danger of looping controlled by
                                                           abort processor)
                      T)
             (AutoFreeze                                   (* quitting this session)
                         CurrentCommand='quit)
             (AutoGenvcs                                   (* current command is READP and there are some Pascal
                                                           program units)
                         (AND CurrentCommand='readp LastProgramUnits))
             (AutoInfix                                    (* current command is INTERFACE and some of the declared
                                                           interfaces were typed in infix notation)
                        (AND CurrentCommand='interface AutoInfix))
             (AutoInvokeIH                                 (* current proposition contains IH)
                           (IH2OP MEMB ReferencedInterfaces))
             (AutoNext                                     (* proof of this branch is complete)
                       CurrentPropn=TRUE)
             (AutoNormint                                  (* current proposition contains arithmetic expressions.
                          (INTERSECTION ReferencedInterfaces IntegerInterfaces))
             [AutoPrintProof                               (* proof of theorem name provided as extra parameter is
                                                           complete)
                             (AND aux (Proved? (GetTheorem aux]
             (AutoPrintProofTheorems                       (* quitting this session)
                                     CurrentCommand='quit)
             (AutoReplace                                  (* current proposition contains equalities)
                          (EQOP MEMB ShortenedReferencedInterfaces))
             (AutoSave                                     (* quiting this session and one or more types have been
                                                           newly defined or modified)
                       (AND CurrentCommand='quit AutoSave))
             (AutoSearch                                   (* current proposition contains existential quantifiers)
                         (AND (type? Qexpression CurrentPropn)
                              CurrentPropn:find~=NIL))
             (AutoSufficient                               (* current command is END and current type is known)
                             (AND CurrentCommand='end CurrentType (CurrentType MEMB KnownNames:Types))
                             )
             (Unexpected 'AutoProfileEntryName])
```

37

```
(AutoCommand
  [LAMBDA (profileEntryName explanatoryString command parameters aux)
                                                           (* D.Thompson "19-Sep-80 16:50")
    (if (AutoPerform? profileEntryName explanatoryString aux)
        then (SELECTQ profileEntryName
                      (AutoFix (PROG (input)
                                     (input←(MKLIST (UserEdit <command parameters> PascalReadTable)))
                                     (CurrentCommand←(AFFIRMSpellingCorrect (L-CASE input:1)
                                                                            KnownNames:Commands))
                                     (ProcessCommand CurrentCommand input::1 Terminal NIL)))
                      (AutoFreeze (ProcessCommand 'freeze
                                                  (if parameters
                                                    else (getFileName NIL
                                                                      <'BODY (UserProfile
                                                                                 'FreezeFileName)
```

```
                                                    'EXTENSION
                                                    (if SYSTEMTYPE='TOPS20
                                                        then 'EXE
                                                      else 'SAV)
                                                    >
                                                  "Freeze file: " 'OUTPUT))
                                        Terminal 'internal))
                          (ProcessCommand command parameters Terminal NIL))
    else NIL])
```

38

```
(AutoMechExec1
  [LAMBDA (freezeFileName)                                    (* R.Erickson "29-Jan-81 16:28")
    (AutoCommand 'AutoFreeze "save this AFFIRM session" 'freeze freezeFileName)
    (if ComingBackFromSYSOUT
      else (AutoCommand 'AutoPrintProofTheorems "summarize the proof attempts" 'print '(proof
                                                                                         theorems))
           (AutoCommand 'AutoSave "save new or modified type specifications" 'save
                        <'types !(Separate AutoSave)
                          >)
           (AutoCommand 'AutoCompile "compile new or modified type specifications" 'compile
                        <'types !(Separate AutoCompile)
                          >])
```

39

```
(AutoMechSpec1
  [LAMBDA NIL                                                 (* R.Erickson "29-Jan-81 16:30")

        (* * This routine handles the auto-applied commands generated by specification commands. -
        EXCEPTION: AutoSufficient is performed directly by the END command processor.)


    (AutoCommand 'AutoInfix "declare infix operators" 'infix (Separate AutoInfix])
```

40

```
(AutoMechTP1
  [LAMBDA (theorem status)                                    (* D.Thompson " 9-Sep-80 14:41")
    (PROG (name)
          (name←(TheoremId theorem))
          (RETURN (AutoCommand 'AutoAnnotate (CONCAT "provide a time stamp for " name)
                               'annotate
                               (LIST name Comma status 'by (L-CASE USERNAME T)
                                     'using 'AFFIRM AFFIRMVERSION 'on (SUBSTRING (DATE)
                                                                                 1 9)
                                     'in 'transcript (MKSTRING (DRIBBLEFILE)))
                               theorem])
```

41

```
(AutoMechTP2
  [LAMBDA NIL                                                 (* D.Thompson "16-Oct-80 15:08")

        (* * This routine is invoked after a theorem proving command. It sees what can apply, and performs the "extra"
        commands if they're armed (set to ON or TELL). The routine cycles until no more changes occur.
        It then checks the result. If TRUE, get another and start over (if AutoNext is armed).)


    (bind (newActions profileEntry toBePerformed) first toBePerformed←
                                                    <'AutoCases 'AutoNormint 'AutoInvokeIH
                                                      'AutoReplace 'AutoSearch >
      while CurrentPropn~=TRUE and toBePerformed
      do (profileEntry←toBePerformed:1)
         (toBePerformed←toBePerformed::1)
         (if newActions←(SELECTQ profileEntry
                                 (AutoCases (if (AutoCommand 'AutoCases
                                                             "raise embedded if-expressions"
                                                             'cases)
                                                then <'AutoNormint 'AutoReplace 'AutoSearch >))
                                 (AutoInvokeIH (if (AutoCommand 'AutoInvokeIH
                                                                "invoke induction hypothesis IH"
                                                                'invoke '(IH %| all %|))
                                                   then <'AutoCases 'AutoNormint 'AutoInvokeIH
                                                          'AutoReplace 'AutoSearch >))
                                 (AutoNormint (if (AutoCommand 'AutoNormint
```

```
                                          "apply integer simplifier"
                                          'normint)
                                then <'AutoReplace 'AutoSearch >))
                        (AutoReplace (if (AutoCommand 'AutoReplace "replace all equalities"
                                          'replace)
                                then <'AutoCases 'AutoNormint 'AutoSearch >))
                        (AutoSearch (AutoCommand 'AutoSearch "search for instantiation"
                                          'search)
                                NIL)
                        (Unexpected 'AutoTheoremProverProfileEntry))
        then toBePerformed← < !! toBePerformed !(LDIFFERENCE newActions toBePerformed)
                         >)
    finally (AutoMechTP4])
```

42

(AutoMechTP3

```
  [LAMBDA NIL                                         (* D.Thompson "16-Oct-80 15:16")
```

(* This routine is applied after a lemma is used in theorem proving. Perform AutoSearch. if necessary and it's armed; then check for TRUE for AutoNext. etc.)*

```
    (AutoCommand 'AutoSearch "search for instantiation" 'search)
    (AutoMechTP4])
```

43

(AutoMechTP4

```
  [LAMBDA NIL                                         (* D.Thompson "16-Oct-80 15:17")
```

(* This routine is called after the auto mechanism has exhausted its possibilities on simplifying the current proposition any further. First. print out the proofs of any theorems that have just become TRUE. Then. if the current proposition is itself TRUE. go get something else to prove (if AutoNext is armed and there IS something else to prove). and hand it over to the auto mechanism simplifier (which will recursively call this routine).)*

```
    (for theorem in NewlyProved finally NewlyProved←NIL do (AutoCommand 'AutoPrintProof
                                                             (CONCAT "print the proof of "
                                                                     theorem)
                                                             'print <'proof theorem
                                                             theorem))
    (if CurrentPropn=TRUE
        then (if (AutoCommand 'AutoNext "go to the next proposition to prove" 'next)
                 then (AutoMechTP2])
```

44

(AutoMechanism

```
  [LAMBDA (event aux)                                 (* R.Erickson "29-Jan-81 16:26")
```

(* This routine is the general overseer for the auto mechanism. -
Usual invocation is after a command (i,e.. an event). The particular command is used to decide the event category.
Some events can occur several times within one command processing (such as "Theorem proved". or "proposition reduced to TRUE"). -
At each event. this routine is called.)*

```
    (PROG (HELPFLAG (InAutoMechanism T))
          (if (UNDONLSETQ (SELECTQ event
                                   (CommandCompletion NIL)
                                   (CompletedCommandAbort (AutoCommand 'AutoFix
                                                            "edit the aborted command"
                                                            'fix AutoFix))
                                   (ExecutiveCommandCompletion NIL)
                                   (IncompleteCommandAbort (AutoCommand 'AutoUndo
                                                         "make sure effects are undone"
                                                            'undo)
                                                           (AutoCommand 'AutoFix
                                                            "edit the aborted command"
                                                            'fix AutoFix))
                                   (LemmaApplication (AutoMechTP3))
                                   [MiscellaneousCommandCompletion (AutoCommand 'AutoGenvcs
                                                   "generate verification conditions"
                                                                  'genvcs
                                                                  (Separate (for unit
                                                                                 in
```

```
                                                                    LastProgramUnits
                                                                            collect
                                                                             unit:1]
                                  (ProgramVerified NewlyProved← < !! NewlyProved aux>
                                               (AutoMechTP1 (GetTheorem aux)
                                                        'verified))
                                  (ProofCycleStep (AutoMechTP2))
                                  (ProofMaintenanceCommand NIL)
                                  (SessionCompletion (AutoMechExec1 aux))
                                  (SpecificationCommandCompletion (AutoMechSpec1))
                                  (TheoremAssumed (AutoMechTP4))
                                  (TheoremProved NewlyProved← < !! NewlyProved aux> aux←
                                              (GetTheorem aux)
                                              (AutoMechTP1 aux (Status aux)))
                                  (TypeClosed (AutoCommand 'AutoSufficient
                                                      "test for sufficient-completeness"
                                                      'sufficient? CurrentType))
                                  (Unexpected 'AutoMechanismCategory)))
                 else (Auto2Abort event])
```

45

(AutoNotYet

```
  [LAMBDA (entry)                                        (* D Thompson " 6-Aug-80 19:20")
    (if (UserProfile entry T)
        then (UserProfileSet entry 'Off)
             (printout NIL .TAB0 0 "The" , entry , "profile entry isn't yet functional." T])
```

46

(AutoPerform?

```
  [LAMBDA (profileEntryName explanatoryString aux)       (* D.Thompson "15-Oct-80 12:50")
    (SELECTQ (UserProfile profileEntryName)
             ((False NIL No Off)
               NIL)
             ((On T True Yes)
               (AutoApplicable profileEntryName aux))
             (Tell (if (AutoApplicable profileEntryName aux)
                       then (printout NIL .TAB0 0 LeftParenthesis "will" , explanatoryString
                                      RightParenthesis T)
                            T))
             (Ask (if (AutoApplicable profileEntryName aux)
                      then (AFFIRMUSER NIL (if (UserProfileDefault profileEntryName) MEMB
                                              <'Tell ! OnValues>
                                              then 'Y
                                            else 'N)
                                       (CONCAT (U-CASE (SUBSTRING explanatoryString 1 1))
                                               (SUBSTRING explanatoryString 2)
                                               QuestionMark Blank))='Y))
             (Unexpected 'AutoProfileEntryValue])
)
[DECLARE: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY
(BLOCK: PROFILEBLOCK APE APG APGgetName AtomList canUserChangeProfileEntry? DefaultUserProfile
        definedEntryName? displayProfileEntry formChangePairs getNewValue
        getProfileAssociatedVariables getProfileConditions getProfileEntryDef
        getProfilePossibleValues gnvKeyList gnvQuestion InitializeUserProfile IsInKeyList?
        performAssociatedUpdates profile ReadUserProfileFile ReInitializeUserProfile
        reportProfileDBError reportUndefinedEntryNames saveUserProfile? SpecialValueOK? UserProfile
        UserProfileDefault UserProfileEnquiry UserProfilePut UserProfileSet validateValue
        WriteUserProfileFile (ENTRIES APE APG DefaultUserProfile InitializeUserProfile UserProfile
                                      UserProfileDefault UserProfilePut UserProfileSet profile)
        (GLOBALVARS EnquiryAborted FONTCHANGEFLG GOODGUY HelpWanted KnownNames OffValues OnOffKeyList
                    OnValues ProfileAlreadyLoaded ProfileReadTable SameOldKeyList SYSPROPS
                    UserProfileGroups USERNAME)
        (NOLINKFNS . T)
        (SPECVARS casedName))
]
(DECLARE: DONTCOPY
  (FILEMAP (NIL (4411 40034 (APE 4423 . 7271) (APG 7275 . 9717) (APGgetName 9721 . 10415) (AtomList
10419 . 10648) (canUserChangeProfileEntry? 10652 . 11147) (DefaultUserProfile 11151 . 12034) (
definedEntryName? 12038 . 12448) (displayProfileEntry 12452 . 13150) (formChangePairs 13154 . 16232) (
getNewValue 16236 . 17733) (getProfileAssociatedVariables 17737 . 18298) (getProfileConditions 18302 .
 18901) (getProfileDefaultValue 18905 . 19059) (getProfileEntryDef 19063 . 19701) (
getProfilePossibleValues 19705 . 20491) (gnvKeyList 20495 . 21388) (gnvQuestion 21392 . 22488) (
InitializeUserProfile 22492 . 24831) (IsInKeyList? 24835 . 25172) (performAssociatedUpdates 25176 .
25807) (profile 25811 . 27062) (ReadUserProfileFile 27066 . 28296) (ReInitializeUserProfile 28300 .
28715) (reportProfileDBError 28719 . 29178) (reportUndefinedEntryNames 29182 . 29875) (
saveUserProfile? 29879 . 30804) (SpecialValueOK? 30808 . 30947) (UserProfile 30951 . 31440) (
```