

File created: 10-May-88 15:41:17 {DSK}<LISPFILS>LOOPS>LYRIC>RULES>LOOPSRULESTTY.;3

changes to: (FNS RS.INTTY RS.INTTYL RS.WRITE RS.CHECKREPLY)
(VARS LOOPSRULESTTYCOMS)

previous date: 10-May-88 13:23:02 {DSK}<LISPFILS>LOOPS>LYRIC>RULES>LOOPSRULESTTY.;1

Read Table: XCL

Package: INTERLISP

Format: XCCS

; Copyright (c) 1988 by XEROX Corporation. All rights reserved.

(RPAQQ **LOOPSRULESTTYCOMS**

```
((* * |Simplified| TTY |handling| |fns.| |Written| |by| |Mark| |Stefik.|)
(FNS RS.INTTY RS.INTTYL RS.WRITE)
(* * |Internal| |fns| |not| |called| |directly| |by| |User.|)
(FNS RS.CHECKREPLY)
(VARS (|tty| T))
(DECLARE\ : DONTVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILEVARS (ADDVARS (NLAMA)
(NLAML)
(LAMA RS.WRITE))))
```

```
(* * |Simplified| TTY |handling| |fns.| |Written| |by| |Mark| |Stefik.|)
```

(DEFINEQ

RS.INTTY

```
(LAMBDA (|promptStr| |goodList| |helpStr| |noShiftFlg| |initialAtom|)
; Edited 10-May-88 15:37 by hwl
```

```
(* RS.INTTY |is| |a| |routine| |which| |returns| |an| |atom| |typed| |by| |the| |User| |at| |a| |terminal.|
|If| |the| |User| |has| |not| |yet| |typed| |anything| |and| |a| |promptStr| |has| |been| |supplied| |it| |is| |typed| |as| |a| |prompt| |to|
|the| |User|. (|If| |the| |User| |has| |typed| |ahead| |the| |prompt| |is| |suppressed|.)| |goodList|
(|if| |given|) |is| |a| |list| |of| |acceptable| |answers|. |If| |the| |first| |characters| |typed| |by| |the| |User| |correspond| |to| |the|
|first| |characters| |of| |any| |element| |of| |goodlist| |that| |element| |is| |returned| |as| |the| |value| |of| |INTTY|.
|If| |goodList| |is| |supplied| |and| |the| |User's| |reply| |matches| |no| |entry|. |he| |is| |warned| |and| |reprompted|.
|If| |the| |User| |types| |?| |to| |INTTY\, |the| |helpStr| |is| |typed| |out| |and| |he| |is| |re-prompted|.
|Note:| |If| |the| |User| |types| |a| |semicolon| |the| |entire| |message| |after| |it| |will| |be| |treated| |as| |a| |comment|.
MJS)
```

```
(PROG (|reply| (CRLF (CHARACTER 13))
(SPACE (CHARACTER 32))
|promptedInitial|)
(COND
((AND |goodList| (NOT |noShiftFlg|)) (* |Capitalize| |Goodlist| |Items|.))
(SETQ |goodList| (|for| |Item| |in| |goodList| |collect| (U-CASE |Item|))))
|ReadLoop|
(COND
((READP |tty|)
(COND
((EQ SPACE (PEEKC |tty|)) (* |Eliminate| |leading| |blanks|.))
(READC |tty|)
(GO |ReadLoop|))))
(|promptStr| (PRIN1 |promptStr| |tty|)
(COND
((AND |initialAtom| (NOT |promptedInitial|))
(SETQ |promptedInitial| T)
(BKSYSEBUF |initialAtom|))))
(SETQ |reply| (PROG (|chars| |char| |doneflg|)
(|do| (PROGN (SETQ |char| (READC |tty|))
(COND
((AND (NULL |chars|)
|promptStr|
(EQ |char| CRLF))
(* |Re-prompt| |if| |just| CRLF |typed|.))
(PRIN1 |promptStr| T))
((OR (EQ |char| SPACE)
(EQ |char| CRLF))
(* |Done| |if| SPACE |or| |Return| |Typed| |after| |some| CHARS.))
(COND
(|chars| (SETQ |doneflg| T))))
(T (SETQ |chars| (CONS |char| |chars|))))
|until| |doneflg|)
(RETURN (PACK (DREVERSE |chars|))))))
(COND
((EQ |reply| '?')
(COND
(|helpStr| (RS.WRITE |helpStr|)
(COND
(|goodList| (RS.WRITE "Expecting one of: " |goodList|))))
(|goodList| (RS.WRITE "Expecting one of: " |goodList|))
(T (RS.WRITE "Sorry -- No Help Provided"))))
((STRPOS ";" |reply| 1 NIL T))
((NOT (ATOM |reply|))
```

```

(RS.WRITE "Invalid response. Expecting one of: " |goodList|)
(T (COND
  ((NOT |noShiftFlg|)
   (SETQ |reply| (U-CASE |reply|))))
 (COND
  ((SETQ |reply| (RS.CHECKREPLY |reply| |goodList|))
   (RETURN |reply|))))
(CLEARBUF)
(GO |ReadLoop|)))

```

(RS.INTTYL

```

(LAMBDA (|promptStr| |goodList| |helpStr| |noShiftFlg| |initialList|)
; Edited 10-May-88 15:37 by hwl

```

```

(* RS.INTTYL returns a list typed by the User at the terminal.
If the User has not yet typed anything and a promptStr has been supplied, it is typed as a prompt to
the User. (If the User has typed ahead, the prompt is suppressed.) |goodList|
(optional) is a list of acceptable answers. If the leading characters correspond to any element of |goodList|,
that element is added to the list returned. If |goodList| is supplied, an the User's reply matches no entry,
he is warned and reprompted. |helpStr| is a help message supplied to the User if he types a question
mark. Note: If the User types a semicolon, the entire message after it will be treated as a comment.
If the User just types a carriage return, then NIL will be returned.
If |noShiftFlg| is NIL, type-in will be returned in upper case;
it is T, it will be returned as typed.)

```

```

(PROG (|reply| (CRLF (CHARACTER 13))
      (SPACE (CHARACTER 32))
      |promptedInitial|)
(COND
  ((AND |goodList| (NOT |noShiftFlg|))
   (SETQ |goodList| (|for| |Item| |in| |goodList| |collect| (* |Capitalize| the |Goodlist| items.) (U-CASE |Item|))))
 |ReadLoop|
(COND
  ((READP |tty|)
   (COND
    ((EQ SPACE (PEEKC |tty|))
     (READC |tty|)
     (GO |ReadLoop|))))
  (|promptStr| (PRIN1 |promptStr| |tty|)
   (COND
    ((AND |initialList| (NOT |promptedInitial|))
     (SETQ |promptedInitial| T)
     (SETQ |initialList| (SUBSTRING (MKSTRING |initialList|
                                         2 -2))
                        (BKSYSBUF |initialList|))))
   (SETQ |reply| (COND
    ((EQ CRLF (PEEKC |tty|))
     (READC |tty|)
     NIL)
    (T (|for| |char| |collect| (PROG (|chars| |doneFlg|)
                                   (|do| (PROGN (SETQ |char| (READC |tty|))
                                                (COND
                                                 ((OR (EQ |char| SPACE)
                                                         (EQ |char| CRLF))
                                                  (* |Done| if |SPACE| or |Return| Typed--
even| if |no| |chars| typed.)
                                                 (SETQ |doneFlg| T))
                                                (T (SETQ |chars| (CONS |char| |chars|))))))
                                   |until| |doneFlg|)
                                   (RETURN (PACK (DREVERSE |chars|))))
    (|until| (EQ |char| CRLF))))))
  (COND
   ((NOT |reply|)
    (RETURN)))
  (COND
   ((FMEMB '? |reply|)
    (COND
     (|helpStr| (RS.WRITE |helpStr|)
      (COND
       (|goodList| (RS.WRITE "Expecting subset of: " |goodList|)))
      (|goodList| (RS.WRITE "Expecting subset of: " |goodList|))
      (T (RS.WRITE "(Sorry -- No Help Provided)"))))
    ((STRPOS ";" |reply| 1 NIL T))
    (T (COND
      ((NOT |noShiftFlg|)
       (SETQ |reply| (U-CASE |reply|)))
      (SETQ |reply| (|for| x |in| (MKLIST |reply|) |collect| (RS.CHECKREPLY x |goodList|)))
      (COND
       ((NOTANY |reply| (FUNCTION NULL))
        (RETURN |reply|))))
      (CLEARBUF)
      (GO |ReadLoop|))))))

```

(RS.WRITE

```

(LAMBDA |nargs|
; Edited 10-May-88 13:09 by hwl

```

```

(PROG ((I 0))
  TOP (|while| (ILESSP I |nargs|) |do| (PRIN1 (ARG |nargs| (SETQ I (ADD1 I))))
    (TERPRI))))
)

(* * |Internal| |fns| |not| |called| |directly| |by| |User|.)

(DEFINEQ
(RS.CHECKREPLY
  (LAMBDA (|reply| |goodList| |possibilities|)
    ; Edited 10-May-88 12:57 by hwl

    (* RS.CHECKREPLY |is| \a |subroutine| |of| RS.INTTY |and| RS.INTTYL.
      |t| |tries| |to| |match| |the| |reply| |against| |the| |items| |in| |goodList| |as| |described| |below|.))

    (COND
      ((NULL |goodList|) (* |If| |goodList|=NIL,| |no| |work| |to| |do|.))
      (|reply|)
      ((MEMBER |reply| |goodList|) (* |If| |spelling| |is| |correct,| |no| |work| |to| |do|.))
      (|reply|)
      ((SETQ |possibilities| (|for| X |in| |goodList| |when| (STRPOS |reply| X 1 NIL T) |collect| X))
        (* |Match| |if| |an| |exact| |leading| |substring| |of| 1 |item| |in|
          |goodList|.))

      ((COND
        ((CDR |possibilities|)
          (RS.WRITE "Ambiguous: " |possibilities|)
          NIL)
        (T (CAR |possibilities|))))
      ((FIXSPELL |reply| 70 |goodList| T))
      ((SETQ |possibilities| (|for| X |in| |goodList| |when| (STRPOS X |reply| 1 NIL T) |collect| X))
        (* |Match| |if| |one| |goodList| |item| |is| |leading| |substring| |of|
          |reply|.))

      ((COND
        ((CDR |possibilities|)
          (RS.WRITE "Ambiguous: " |possibilities|))
        (T (CAR |possibilities|))))
      (T (RS.WRITE "Response not recognized. Type ? for help."))))
)

(RPAQQ |tty| T)

(DECLARE\; DONTVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILEVARs

(ADDTovar NLAMA )

(ADDTovar NLAML )

(ADDTovar LAMA RS.WRITE)
)

(PUTPROPS LOOPSRULESTTY COPYRIGHT ("XEROX Corporation" 1988))

```

FUNCTION INDEX

RS.CHECKREPLY3 RS.INTTY1 RS.INTTYL2 RS.WRITE2

VARIABLE INDEX

|tty|3
