

*File created:* 4-Feb-2022 14:14:40 {MM}<rooms>ROOMS-NOTES.;2

*changes to:* (IL:FUNCTIONS EDIT-NOTE-WINDOW-TEXT)

*previous date:* 5-Dec-2020 16:39:51 {MM}<rooms>ROOMS-NOTES.;1

*Read Table:* XCL

*Package:* ROOMS

*Format:* XCCS

; Copyright (c) 1987-1988, 1990, 2020 by Venue & Xerox Corporation.

```
(IL:RPAQQ IL:ROOMS-NOTESCOMS
  ((FILE-ENVIRONMENTS IL:ROOMS-NOTES)
   (IL:P (EXPORT ' (*DEFAULT-NOTE-WINDOW-FONT* MAKE-NOTE-WINDOW))
    (REQUIRE "ROOMS")))
  ;; provides note windows
  (IL:STRUCTURES NOTE)
  (IL:VARIABLES *DEFAULT-NOTE-WINDOW-FONT*)
  (IL:FUNCTIONS MAKE-NOTE-WINDOW NOTE-WINDOW-REPAINTFN PRINT-NOTE-STRING NOTE-WINDOW-BUTTONEVENTFN
    EDIT-NOTE-WINDOW-TEXT SET-NOTE-WINDOW-FONT SET-NOTE-WINDOW-TITLE)
  (IL:WINDOW-TYPES :NOTE-WINDOW)
  (IL:DECLARE\ IL:EVAL@COMPILE IL:DONTCOPY (IL:P (OR (IL:HASDEF 'STREAM 'IL:RECORDS)
    (IL:EVAL (IL:SYSRECLOOK1 'STREAM))))))
  (IL:GLOBALVARS IL:BOLDFONT)))

(DEFINE-FILE-ENVIRONMENT IL:ROOMS-NOTES :COMPILER :COMPILE-FILE
  :PACKAGE "ROOMS"
  :READTABLE "XCL")

(EXPORT ' (*DEFAULT-NOTE-WINDOW-FONT* MAKE-NOTE-WINDOW))

(REQUIRE "ROOMS")

;; provides note windows

(DEFSTRUCT NOTE
  ;;: a note for display in a note-window

  (STRING "" :TYPE STRING)
  (FONT NIL :TYPE FONT)
  (TITLE "Note:" :TYPE STRING)
  (READ-ONLY? NIL :TYPE (MEMBER T NIL)))

(DEFVAR *DEFAULT-NOTE-WINDOW-FONT* IL:BOLDFONT)

(DEFUN MAKE-NOTE-WINDOW (&KEY REGION (TITLE "Note:")
  (STRING "")
  (FONT *DEFAULT-NOTE-WINDOW-FONT*)
  (READ-ONLY? NIL))
  (LET ((WINDOW (IL:CREATEW REGION TITLE)))
    (IL:WINDOWPROP WINDOW 'NOTE (MAKE-NOTE :STRING STRING :FONT (IF (SYMBOLP FONT)
      FONT
      (IL:FONTCREATE FONT)))
      :TITLE TITLE :READ-ONLY? READ-ONLY?))
    (IL:WINDOWPROP WINDOW 'IL:REPAINTFN 'NOTE-WINDOW-REPAINTFN)
    (IL:WINDOWPROP WINDOW 'IL:RESHAPEFN 'NOTE-WINDOW-REPAINTFN)
    (IL:WINDOWPROP WINDOW 'IL:BUTTONEVENTFN 'NOTE-WINDOW-BUTTONEVENTFN)
    (IL:WINDOWPROP WINDOW 'IL:BUTTONEVENTFN 'NOTE-WINDOW-BUTTONEVENTFN)
    (NOTE-WINDOW-REPAINTFN WINDOW)
    WINDOW))

(DEFUN NOTE-WINDOW-REPAINTFN (WINDOW &REST IGNORE)
  (LET* ((NOTE (IL:WINDOWPROP WINDOW 'NOTE))
    (DSP (IL:GETSTREAM WINDOW))
    (FONT (NOTE-FONT NOTE)))
    (IL:WINDOWPROP WINDOW 'IL:TITLE (NOTE-TITLE NOTE))
    (IL:DSPFONT (IF (SYMBOLP FONT)
      (SYMBOL-VALUE FONT)
      FONT)
      DSP)
    (IL:CLEARW WINDOW)
    ;; why 8? that's what TEdit uses.
    (PRINT-NOTE-STRING (NOTE-STRING NOTE)
      DSP 8 (- (IL:WINDOWPROP WINDOW 'IL:WIDTH)
        8))))
```

```
(DEFUN PRINT-NOTE-STRING (STRING DSP LEFT-MARGIN RIGHT-MARGIN &OPTIONAL (LINE-LEADING 0))
```

```
;; print STRING to DSP within LEFT-MARGIN & RIGHT-MARGIN, breaking lines at spaces. I shouldn't have to write this, so it's ok if the code is ugly,
;; right?
```

```
(CHECK-TYPE DSP (SATISFIES IL:DISPLAYSTREAMP))
(PROG* ((CHAR)
  (FONT (IL:DSPFONT NIL DSP))
  (LINE-HEIGHT (+ (IL:FONTHEIGHT FONT)
                  LINE-LEADING))
  (LENGTH (VECTOR-LENGTH STRING))
  (DD (IL:FETCH (STREAM IL:IMAGEDATA) IL:OF DSP))
  (LAST-SPACE 0)
  (LINE-START 0)
  (I -1)
  (X LEFT-MARGIN)
  (X-AT-LAST-SPACE LEFT-MARGIN)
  )
  (IL:MOVETO LEFT-MARGIN (- (IL:DSPYPOSITION NIL DSP)
                           LINE-LEADING)
    DSP)
```

```
; offset in string where we'll break
; offset into string where this line starts
; current offset into string
; x position of char at I in pixels
; x position of char at LAST-SPACE in pixels
```

```
LOOP
```

```
(INCF I)
(WHEN (>= I LENGTH)
  (SETQ LAST-SPACE LENGTH)
  (GO DUMP-LINE))
(SETQ CHAR (AREF STRING I))
(CASE CHAR
  (#\Space
    (DO ((N (1+ I)
            (1+ N)))
      ;; skip through multiple spaces without checking for line breaks so that line breaks are always forced after a group of spaces
      ((OR (= N LENGTH)
            (NOT (EQL (AREF STRING N)
                      #\Space))))
      (INCF I)
      (INCF X (IL:CHARWIDTH (CHAR-CODE #\Space)
                            FONT)))
    (SETQ LAST-SPACE I)
    (SETQ X-AT-LAST-SPACE X))
  (#\Newline
    (SETQ LAST-SPACE I)
    (SETQ X-AT-LAST-SPACE X)
    (GO DUMP-LINE)))
(INCF X (IL:CHARWIDTH (CHAR-CODE CHAR)
                      FONT))
(WHEN (> X RIGHT-MARGIN)
```

```
; force line break
```

```
;; check if line needs breaking
```

```
(WHEN (AND (<= LAST-SPACE LINE-START))
```

```
;; if we've had no spaces on this line, just break it where we are. we actually lose a character here, as DUMP-LINE always skips
;; the character we're on, presuming it's a space or CR.
```

```
(SETQ LAST-SPACE I)
(SETQ X-AT-LAST-SPACE X))
(GO DUMP-LINE))
(GO LOOP)
```

```
DUMP-LINE
```

```
;; dump chars from LINE-START up to (but not including) LAST-SPACE.
```

```
(DO ((N LINE-START (1+ N)))
  ((OR (= N LAST-SPACE)
        (= N LENGTH))
    ;; move to the next line
    (IL:MOVETO LEFT-MARGIN (- (IL:DSPYPOSITION NIL DSP)
                              LINE-HEIGHT)
      DSP)
    ;; adjust X & LINE-START
    (SETQ X (+ LEFT-MARGIN (- X X-AT-LAST-SPACE)))
    (SETQ LINE-START (1+ LAST-SPACE)))
```

```
;; this is soooo much faster than calling WRITE-CHAR. the down side is that this code will now only work on display streams.
```

```
(IL:\BLTCHAR (CHAR-CODE (AREF STRING N))
  DSP DD))
(IF (>= I LENGTH)
  (RETURN)
  (GO LOOP)))
```

```
(DEFUN NOTE-WINDOW-BUTTONEVENTFN (WINDOW)
  (IL:TOTOPW WINDOW)
```

```

(WHEN (AND (IL:MOUSESTATE (IL:ONLY IL:MIDDLE))
  (NOT (NOTE-READ-ONLY? (IL:WINDOWPROP WINDOW 'NOTE))))
(CASE (MENU ' ("Edit Text" :EDIT "Edit the text of this note window with TEdit.")
  ("Set Font" :FONT "Set the font of this note window.")
  ("Set Title" :TITLE "Set the title of this note window.")))
(:EDIT (IL:ADD.PROCESS '(EDIT-NOTE-WINDOW-TEXT ',WINDOW)))
(:FONT (IL:ADD.PROCESS '(SET-NOTE-WINDOW-FONT ',WINDOW)))
(:TITLE (IL:ADD.PROCESS '(SET-NOTE-WINDOW-TITLE ',WINDOW))))

```

```

(DEFUN EDIT-NOTE-WINDOW-TEXT (WINDOW) ; Edited 4-Feb-2022 14:14 by rmk
  (LET ((NOTE (IL:WINDOWPROP WINDOW 'NOTE))
    (IF (FBOUNDP 'IL:TEDIT)
      (LET ((TEXT-STREAM (IL:OPENTEXTSTREAM (IL:OPENSTRINGSTREAM (NOTE-STRING NOTE))
        NIL NIL NIL
        '(IL:FONT , (NOTE-FONT NOTE)
          IL:NOTITLE T IL:PROMPTWINDOW ,IL:PROMPTWINDOW IL:MENU
          (IL:|Find| IL:|Substitute| IL:|Quit|)
          IL:QUITFN
          ,#' (LAMBDA (WINDOW STREAM TEXTOBJ IL:PROPS)
            (IL:|replace| IL:EDITFINISHEDFLG IL:|of| TEXTOBJ IL:|with| T)
            'IL:DON\`T)
          IL:AFTERQUITFN
          ,#' (LAMBDA (WINDOW STREAM)
            (IL:OPENW WINDOW))))))
        (IL:TTY.PROCESS (IL:THIS.PROCESS))
        (SETF (NOTE-STRING NOTE)
          (IL:TEDIT TEXT-STREAM WINDOW T))
        (IL:BLOCK 200)
        (IL:WINDOWPROP WINDOW 'IL:REPAINTFN 'NOTE-WINDOW-REPAINTFN)
        (IL:WINDOWPROP WINDOW 'IL:RESHAPEFN 'NOTE-WINDOW-REPAINTFN)
        (IL:WINDOWPROP WINDOW 'IL:BUTTONEVENTFN 'NOTE-WINDOW-BUTTONEVENTFN)
        (NOTE-WINDOW-REPAINTFN WINDOW))))))

```

```

(DEFUN SET-NOTE-WINDOW-FONT (WINDOW)
  (LET* ((NOTE (IL:WINDOWPROP WINDOW 'NOTE))
    (OLD-FONT (NOTE-FONT NOTE))
    (NEW-FONT (SEDIT::SEDITE (IF (SYMBOLP OLD-FONT)
      OLD-FONT
      (EXTERNALIZE-FONT OLD-FONT))
      NIL NIL NIL NIL ' (:CLOSE-ON-COMPLETION))))
    (SETF (NOTE-FONT NOTE)
      (IF (SYMBOLP NEW-FONT)
        NEW-FONT
        (IL:FONTCREATE NEW-FONT)))
    (NOTE-WINDOW-REPAINTFN WINDOW)))

```

```

(DEFUN SET-NOTE-WINDOW-TITLE (WINDOW)
  (LET* ((NOTE (IL:WINDOWPROP WINDOW 'NOTE))
    (TITLE (PROMPT-USER "Title:" "Type title (CR to abort)"))
    (WHEN TITLE
      (SETF (NOTE-TITLE NOTE)
        TITLE)
      (NOTE-WINDOW-REPAINTFN WINDOW))))

```

```

(DEF-WINDOW-TYPE :NOTE-WINDOW :RECOGNIZER (LAMBDA (WINDOW)
  (NOTE-P (IL:WINDOWPROP WINDOW 'NOTE)))
  :ABTRACTER (LAMBDA (WINDOW)
    (LET* ((NOTE (IL:WINDOWPROP WINDOW 'NOTE))
      (FONT (NOTE-FONT NOTE))
      ' (:REGION , (EXTERNALIZE-REGION (WINDOW-REGION WINDOW))
        :TITLE
        , (NOTE-TITLE NOTE)
        :STRING
        , (NOTE-STRING NOTE)
        :FONT
        , (IF (SYMBOLP FONT)
          FONT
          (EXTERNALIZE-FONT FONT))
        :READ-ONLY?
        , (NOTE-READ-ONLY? NOTE))))
    :RECONSTITUTER (LAMBDA (ARGS)
      (LET ((REST (COPY-LIST ARGS))
        (REMF REST :REGION)
        (APPLY #'MAKE-NOTE-WINDOW :REGION (INTERNALIZE-REGION (GETF ARGS :REGION)
          ' (0 0 100 100)))
        REST)))
    :TITLE (LAMBDA (PLACEMENT REGION DSP)
      (LET ((NOTE (IL:WINDOWPROP (PLACEMENT-WINDOW PLACEMENT)
        'NOTE)))
        (PRINT-PEP-TITLE-STRING (IF (AND NOTE (NOTE-TITLE NOTE))
          (STRING (NOTE-TITLE NOTE))
          "Note:")
          REGION DSP :NO-TITLE-BAR? (PLACEMENT-SHRUNKEN? PLACEMENT))))))

```

{MEDLEY}<rooms>ROOMS-NOTES.;1

Page 4

(IL:DECLARE\ : IL:EVAL@COMPILE IL:DONTCOPY

(OR (IL:HASDEF 'STREAM 'IL:RECORDS)  
 (IL:EVAL (IL:SYSRECLOOK1 'STREAM)))  
)

(IL:DECLARE\ : IL:DOEVAL@COMPILE IL:DONTCOPY

(IL:GLOBALVARS IL:BOLDFONT)  
)

(IL:PUTPROPS **IL:ROOMS-NOTES IL:COPYRIGHT** ("Venue & Xerox Corporation" 1987 1988 1990 2020))

FUNCTION INDEX

EDIT-NOTE-WINDOW-TEXT .....	3	NOTE-WINDOW-REPAINTFN .....	1	SET-NOTE-WINDOW-TITLE .....	3
MAKE-NOTE-WINDOW .....	1	PRINT-NOTE-STRING .....	2		
NOTE-WINDOW-BUTTONEVENTFN .....	2	SET-NOTE-WINDOW-FONT .....	3		

WINDOW-TYPE INDEX

:NOTE-WINDOW .....	3
--------------------	---

VARIABLE INDEX

*DEFAULT-NOTE-WINDOW-FONT* .....	1
----------------------------------	---

STRUCTURE INDEX

NOTE .....	1
------------	---

FILE-ENVIRONMENT INDEX

IL:ROOMS-NOTES .....	1
----------------------	---