

File created: 6-Aug-2022 18:06:57 {DSK}<home>larry>medley>library>TABLEBROWSER.;8

changes to: (VARS TABLEBROWSERCOMS)
(RECORDS TABLEBROWSER TABLEITEM)

previous date: 4-Aug-2022 09:32:17 {DSK}<home>larry>medley>library>TABLEBROWSER.;7

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

::
:: Copyright (c) 1985-1988, 1990, 1993-1995, 1999, 2018, 2021-2022 by Venue & Xerox Corporation.

(RPAQQ TABLEBROWSERCOMS

```
((COMS ; Entries
(FNS TB.MAKE.BROWSER TB.REPLACE.ITEMS)
(FNS TB.DELETE.ITEM TB.UNDELETE.ITEM TB.INSERT.ITEM TB.REMOVE.ITEM TB.NORMALIZE.ITEM
TB.REDISPLAY.ITEMS TB.SELECT.ITEM TB.UNSELECT.ITEM TB.UNSELECT.ALL.ITEMS)
(FNS TB.NUMBER.OF.ITEMS TB.NTH.ITEM TB.COLLECT.ITEMS TB.MAP.ITEMS TB.MAP.DELETED.ITEMS
TB.MAP.SELECTED.ITEMS TB.FIND.ITEM TB.ITEM.SELECTED? TB.ITEM.DELETED?)
(FNS TB.CLEAR.LINE TB.USERDATA TB.WINDOW))
(COMS ; Display
(FNS TB.REPAINTFN TB.RESHAPEFN TB.SCROLLFN TB.DISPLAY.LINES TB.PRINT.LINE TB.FIRST.VISIBLE.ITEM#
TB.LAST.VISIBLE.ITEM# TB.ITEM.VISIBLE? TB.ITEM.FROM.YCOORD TB.BOTTOM.OF.ITEM TB.SHOW.DELETION
TB.SHOW.SELECTION TB.UPDATE.DISPLAY TB.ITEM.UPDATE?))
(COMS ; Selection
(FNS TB.BUTTONEVENTFN TB.DO.UNLESS.BUSY TB.DO.ITEM.SELECTION TB.CONTIGUOUS.SELP TB.DECONSIDERRANGE
TB.CONSIDERRANGE TB.DESELECTRANGE TB.RECONSIDERRANGE TB.SELECTRANGE TB.UNDOSELECTION
TB.FIND.SELECTED.ITEM TB.REV.FIND.SELECTED.ITEM)
(FNS TB.COPYBUTTONEVENTFN TB.SHOW.COPY.SELECTION))
(COMS ; Misc state change
(FNS TB.BROWSER.BUSY TB.CLOSE/SHRINK TB.CLOSEFN TB.FINISH.CLOSE TB.FLUSH.WINDOW TB.SET.FONT
TB.SHRINKFN TB.EXPANDFN TB.FIND.PREVIOUS.TAIL TB.RENUMBER.TAIL))
(COMS ; Misc
(FNS TB.PROCESS)
(INITVARS (TB.DELETEDLINEHEIGHT 1))
(VARS TB.SELECTION.BITMAP)
(CURSORS TB.CROSSCURSOR)
(DECLARE%: EVAL@COMPILE DONTCOPY (CONSTANTS TB.LEFT.MARGIN)
(CONSTANTS * TOCSTATES)
(MACROS .COPYKEYDOWNP.)
(GLOBALVARS TB.CROSSCURSOR TB.SELECTION.BITMAP TB.DELETEDLINEHEIGHT)
(LOCALVARS . T)
(RECORDS TABLEBROWSER TABLEITEM))
(DECLARE%: DONTVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVARS (ADDVARS (NLAMA)
(NLAML)
(LAMA TB.USERDATA)))
(INITRECORDS TABLEBROWSER TABLEITEM)
(SYSRECORDS TABLEBROWSER TABLEITEM))
```

:: Entries

(DEFINEQ

(TB.MAKE.BROWSER

[LAMBDA (ITEMS WINDOWSPEC PROPS) ; Edited 28-Jan-88 04:37 by bvm

::: Build a browser window, which consists of three attached windows: the main BROWSERWINDOW, the BROWSERMENUWINDOW containing the
::: menu, and a BROWSERPROMPTWINDOW for displaying random info

```
(PROG ((LINESPERITEM 1)
FONT PRINTFN COPYFN CLOSEFN AFTERCLOSEFN TITLE COLUMNS USERDATA WINDOW USERPROPS BROWSER ITEMHEIGHT
BASELINE HEADINGWINDOW LINETHICKNESS)
(DECLARE (SPECVARS FONT PRINTFN COPYFN CLOSEFN AFTERCLOSEFN TITLE COLUMNS USERDATA LINESPERITEM
ITEMHEIGHT BASELINE HEADINGWINDOW LINETHICKNESS))
; For SET below
[for TAIL on PROPS by (CDDR TAIL)
do (SELECTQ (CAR TAIL)
((FONT PRINTFN COPYFN CLOSEFN AFTERCLOSEFN TITLE COLUMNS USERDATA LINESPERITEM ITEMHEIGHT
BASELINE HEADINGWINDOW LINETHICKNESS)
(SET (CAR TAIL)
(CADR TAIL)))
(push USERPROPS (LIST (CAR TAIL)
(CADR TAIL]
(SETQ WINDOW (DECODE.WINDOW.ARG WINDOWSPEC NIL NIL TITLE))
[WINDOWPROP WINDOW 'TABLEBROWSER
(SETQ BROWSER
(create TABLEBROWSER
TBWINDOW _ WINDOW
TBFONT _ FONT
TBLOCK _ (CREATE.MONITORLOCK (OR (WINDOWPROP WINDOW 'TITLE)
"Table Browser"))
TB#LINESPERITEM _ (OR LINESPERITEM 1)
```

```

TBBASELINE _ (OR BASELINE 0)
TBCOLUMNS _ COLUMNS
TBPRINTFN _ PRINTFN
TBCOPYFN _ COPYFN
TBCLOSEFN _ CLOSEFN
TBATERCLOSERFN _ AFTERCLOSERFN
TBUSERDATA _ USERDATA
TBHEADINGWINDOW _ HEADINGWINDOW
TBLINETHICKNESS _ (OR LINETHICKNESS TB.DELETEDLINEHEIGHT)

(if ITEMHEIGHT
  then
    ; User explicitly controlling height variables.
    (replace (TABLEBROWSER TBITEMHEIGHT) of BROWSER with ITEMHEIGHT)
    (replace (TABLEBROWSER TBHEIGHTEXPLICIT) of BROWSER with T))
(DSPLEFTMARGIN TB.LEFT.MARGIN WINDOW)
(TB.REPLACE.ITEMS BROWSER ITEMS)
(WINDOWPROP WINDOW 'SCROLLFN (FUNCTION TB.SCROLLFN))
(WINDOWPROP WINDOW 'REPAINTFN (FUNCTION TB.REPAINTFN))
(WINDOWPROP WINDOW 'BUTTONEVENTFN (FUNCTION TB.BUTTONEVENTFN))
(WINDOWPROP WINDOW 'RIGHTBUTTONFN (FUNCTION TB.BUTTONEVENTFN))
(WINDOWPROP WINDOW 'COPYBUTTONEVENTFN (FUNCTION TB.COPYBUTTONEVENTFN))
(for PROP in ' (CLOSEFN SHRINKFN RESHAPEFN)
  do
    ;; This used to be (progn (windowaddprop window 'closefn (function tb.closefn)) (windowaddprop window 'shrinkfn (function
    ;; tb.shrinkfn)) (windowaddprop window 'reshapefn (function tb.reshapefn))). However, we want to be careful to put our stuff on
    ;; before any attached window stuff, so that we can reject a CLOSE, for example, before CLOSEATTACHEDWINDOWS has
    ;; already closed them. Could always put on front, but it's probably better to put our functions after any the user might have
    ;; explicitly put there already.
    (LET ((OLDP (WINDOWPROP WINDOW PROP))
          (FN (PACK* "TB." PROP)))
      (if (NULL OLDP)
        then (SETQ OLDP (LIST FN))
        else (for TAIL on (OR (LISTP OLDP)
                              (SETQ OLDP (LIST OLDP)))
              do (if (EQ (CAR TAIL)
                         FN)
                    then
                      ; Window already has our fn!
                      (RETURN)
                    elseif (STRPOS "ATTACHED" (CAR TAIL))
                      then
                        ; Insert before this attached window hacker
                        (RETURN (ATTACH FN TAIL)))
                    finally
                      ; Put at end
                      (NCONC1 OLDP FN)))
          (WINDOWPROP WINDOW PROP OLDP)))
    (replace (TABLEBROWSER TBREADY) of BROWSER with T)
  (RETURN BROWSER])

```

(TB.REPLACE.ITEMS

[LAMBDA (BROWSER NEWITEMS) ; Edited 27-Jan-88 16:27 by bvm

;; Completely replace the current items with the specified items

```

(LET ((N 0)
      FIRSTSEL)
  (SETQ BROWSER (\DTEST BROWSER 'TABLEBROWSER))
  (for ITEM in NEWITEMS do
    ; Number the items
    (freplace TI# of (\DTEST ITEM 'TABLEITEM) with (add N 1)))
  (freplace (TABLEBROWSER TTAILHINT) of BROWSER with NIL)
  (freplace (TABLEBROWSER TBITEMS) of BROWSER with NEWITEMS)
  (freplace (TABLEBROWSER TB#ITEMS) of BROWSER with N)
  (freplace (TABLEBROWSER TB#DELETED) of BROWSER with (for ITEM in NEWITEMS
                                                         count (ffetch TIDLETED of ITEM)))
  (COND
    ((SETQ FIRSTSEL (TB.FIND.SELECTED.ITEM BROWSER 1 N))
     (freplace (TABLEBROWSER TBFIRSTSELECTEDITEM) of BROWSER with FIRSTSEL)
     (freplace (TABLEBROWSER TBLASTSELECTEDITEM) of BROWSER with (TB.REV.FIND.SELECTED.ITEM BROWSER
                                                                                          FIRSTSEL N)))
    (T (freplace (TABLEBROWSER TBFIRSTSELECTEDITEM) of BROWSER with (ADD1 N))
       (freplace (TABLEBROWSER TBLASTSELECTEDITEM) of BROWSER with 0)))
  (TB.SET.FONT BROWSER)
  (LET [(REGION (DSPCLIPPINGREGION NIL (ffetch (TABLEBROWSER TBWINDOW) of BROWSER)
        (TB.DISPLAY.LINES BROWSER (TB.FIRST.VISIBLE.ITEM# BROWSER REGION)
        (TB.LAST.VISIBLE.ITEM# BROWSER REGION))

```

(DEFINEQ

(TB.DELETE.ITEM

[LAMBDA (BROWSER ITEM) ; Edited 29-Jan-88 12:08 by bvm

```

(COND
  ([NOT (ffetch (TABLEITEM TIDLETED) of (\DTEST ITEM 'TABLEITEM)
    (freplace (TABLEITEM TIDLETED) of ITEM with T)
    (add (ffetch (TABLEBROWSER TB#DELETED) of (\DTEST BROWSER 'TABLEBROWSER))
        1)
    (if (TB.ITEM.UPDATABLE? BROWSER ITEM T)
      then (TB.SHOW.DELETION BROWSER ITEM (ffetch (TABLEBROWSER TBWINDOW) of BROWSER)
        'REPLACE])

```



```

      (T (RPLACD (SETQ TAIL (TB.FIND.PREVIOUS.TAIL BROWSER ITEM#))
                (SETQ TAIL (CDDR TAIL)
                      (TB.RENUMBER.TAIL BROWSER TAIL ITEM#)))
        (freplace (TABLEBROWSER TB#ITEMS) of BROWSER with (SUB1 LASTITEM#))
        (COND
          ((ffetch TIDLETED of ITEM)
           (add (ffetch (TABLEBROWSER TB#DELETED) of BROWSER)
                -1)))
        ;; Update first & last selected item if they fall after the deletion or if the old item is selected
        [COND
          ((>= (SETQ N (ffetch TBFIRSTSELECTEDITEM of BROWSER)
                ITEM#)
              (freplace TBFIRSTSELECTEDITEM of BROWSER with (COND
                ((EQ N ITEM#)
                 ; removed item was the first selected, so look for next one after it
                 (OR (TB.FIND.SELECTED.ITEM BROWSER ITEM#)
                     LASTITEM#))
                (T ; Item numbers are decremented
                  (SUB1 N)
                 )))
          [COND
            ((>= (SETQ N (ffetch TBLASTSELECTEDITEM of BROWSER)
                    ITEM#)
                (freplace TBLASTSELECTEDITEM of BROWSER with (COND
                  ((EQ N ITEM#)
                   ; removed item was the last selected, so look for next one before
                   ; it
                   (OR (TB.REV.FIND.SELECTED.ITEM BROWSER NIL
                       (SUB1 ITEM#))
                       0))
                  (T (SUB1 N)
                     )))
            (TB.UPDATE.DISPLAY BROWSER ITEM# 'REMOVE)])

```

(TB.NORMALIZE.ITEM

[LAMBDA (BROWSER ITEM) ; Edited 22-Jan-88 16:22 by bvm

;; Scroll, if necessary, so that ITEM is visible in browser.

```

(LET* ([WINDOW (ffetch (TABLEBROWSER TBWINDOW) of (\DTEST BROWSER 'TABLEBROWSER)
  (BOT (TB.BOTTOM.OF.ITEM BROWSER ITEM))
  (CLIP (DSPCLIPPINGREGION NIL WINDOW)))
  (COND
    ([OR (> (fetch (REGION BOTTOM) of CLIP)
          BOT)
         (< (fetch (REGION PTOP) of CLIP)
            (+ BOT (ffetch (TABLEBROWSER TBITEMHEIGHT) of BROWSER)
              ; Scroll so that item's midline is at midline of window
              (SCROLLBYREPAINTFN WINDOW 0 (- (+ (fetch (REGION BOTTOM) of CLIP)
                                                (IQUOTIENT (fetch (REGION HEIGHT) of CLIP)
                                                            2))
              (+ BOT (IQUOTIENT (ffetch (TABLEBROWSER TBITEMHEIGHT) of BROWSER)
                                2))
              )))

```

(TB.REDISPLAY.ITEMS

[LAMBDA (BROWSER FIRSTITEM LASTITEM) ; Edited 1-Dec-2018 17:25 by rmk:

; Edited 2-Feb-88 11:53 by bvm:

;; Force redisplay of all items from FIRSTITEM to LASTITEM, e.g., because their content or format changed. We'll only redisplay the visible ones,
;; of course. Also, if browser isn't open, we'll save the change until browser is expanded

```

(LET [(REGION (DSPCLIPPINGREGION NIL (ffetch (TABLEBROWSER TBWINDOW) of (\DTEST BROWSER 'TABLEBROWSER)
  (if (AND (NULL FIRSTITEM)
           (NULL LASTITEM))
      then ; We're being told to redisplay the whole browser, so recompute
          ; the extent while we're at it (it might have gotten smaller).
          (replace (TABLEBROWSER TBMAXXPOS) of BROWSER with 0))
  (SETQ FIRSTITEM (IMAX [COND
    ((NULL FIRSTITEM)
     1)
    ((FIXP FIRSTITEM)
     (T (ffetch TI# of (\DTEST FIRSTITEM 'TABLEITEM)
         (TB.FIRST.VISIBLE.ITEM# BROWSER REGION)))
    (SETQ LASTITEM (IMIN [COND
    ((NULL LASTITEM)
     (ffetch (TABLEBROWSER TB#ITEMS) of BROWSER))
    ((FIXP LASTITEM)
     (T (ffetch TI# of (\DTEST LASTITEM 'TABLEITEM)
         (TB.LAST.VISIBLE.ITEM# BROWSER REGION)))
    (if (AND (>= LASTITEM FIRSTITEM)
          (TB.ITEM.UPDATABLE? BROWSER FIRSTITEM))
      then
      ;; RMK: For whatever reason, on an FB recompute, this gets called after the items have first been displayed but not in proper
      ;; alignment. This redisplayes them to get the alignment, but the window is garbled if the old stuff isn't cleared first. So, added the
      ;; CLEARW
      (CLEARW (ffetch (TABLEBROWSER TBWINDOW) of BROWSER))
      (TB.DISPLAY.LINES BROWSER FIRSTITEM LASTITEM)])

```

(TB.SELECT.ITEM

```
[LAMBDA (BROWSER ITEM) ; Edited 29-Jan-88 12:08 by bvm
  (LET [(N (ffetch (TABLEITEM TI#) of (\DTEST ITEM 'TABLEITEM)
    (TB.SELECTRANGE (\DTEST BROWSER 'TABLEBROWSER)
      N N T)
    (if (TB.ITEM.UPDATABLE? BROWSER N T)
      then (TB.SHOW.SELECTION BROWSER N 'REPLACE]))
```

(TB.UNSELECT.ITEM

```
[LAMBDA (BROWSER ITEM) ; Edited 29-Jan-88 12:09 by bvm
  (if (ffetch (TABLEITEM TISELECTED) of (\DTEST ITEM 'TABLEITEM))
    then (LET ((N (ffetch (TABLEITEM TI#) of ITEM))
      (TB.DESELECTRANGE (\DTEST BROWSER 'TABLEBROWSER)
        N N)
      (if (TB.ITEM.UPDATABLE? BROWSER N T)
        then (TB.SHOW.SELECTION BROWSER N 'ERASE]))
```

(TB.UNSELECT.ALL.ITEMS

```
[LAMBDA (BROWSER) ; Edited 29-Jan-88 12:14 by bvm
  ;; User entry for unselecting all items in the browser.
  (LET (([START (ffetch (TABLEBROWSER TBFIRSTSELECTEDITEM) of (\DTEST BROWSER 'TABLEBROWSER)
    END (ffetch (TABLEBROWSER TBLASTSELECTEDITEM) of BROWSER))
    (if (<= START END)
      then [for I from START to END bind (UPDATABLE _ (TB.ITEM.UPDATABLE? BROWSER START))
        ITEM
        when (ffetch (TABLEITEM TISELECTED) of (SETQ ITEM (TB.NTH.ITEM BROWSER I)))
        do (freplace TISELECTED of ITEM with NIL)
        (if UPDATABLE
          then (TB.SHOW.SELECTION BROWSER I 'ERASE)
        (freplace TBFIRSTSELECTEDITEM of BROWSER with (ADD1 (ffetch (TABLEBROWSER TB#ITEMS) of BROWSER)))
        (freplace TBLASTSELECTEDITEM of BROWSER with 0])
    )
```

(DEFINEQ

(TB.NUMBER.OF.ITEMS

```
[LAMBDA (BROWSER TYPE) ; Edited 27-Jan-88 16:16 by bvm
  (SETQ BROWSER (\DTEST BROWSER 'TABLEBROWSER))
  (SELECTQ TYPE
    (NIL (ffetch (TABLEBROWSER TB#ITEMS) of BROWSER))
    (DELETED (ffetch (TABLEBROWSER TB#DELETED) of BROWSER))
    (SELECTED (for I from (ffetch (TABLEBROWSER TBFIRSTSELECTEDITEM) of BROWSER)
      to (ffetch (TABLEBROWSER TBLASTSELECTEDITEM) of BROWSER) count (ffetch (TABLEITEM TISELECTED)
        of (TB.NTH.ITEM BROWSER I))))
  )
  (\ILLEGAL.ARG TYPE))
```

(TB.NTH.ITEM

```
[LAMBDA (BROWSER N) ; Edited 27-Jan-88 16:18 by bvm
  ;; Return the Nth item of BROWSER, or NIL if N is out of range.
  ;; Browser items are currently stored as a simple list. To make most accesses reasonable, we save a hint to a recent tail of the list to speed up the
  ;; search.
  (\DTEST BROWSER 'TABLEBROWSER)
  (LET (TAIL TAILN)
    (if [AND (> N 0)
      (OR [AND (SETQ TAIL (ffetch (TABLEBROWSER TBTAILHINT) of BROWSER))
        (>= N (SETQ TAILN (ffetch (TABLEITEM TI#) of (CAR TAIL)
          (PROG1 (SETQ TAIL (ffetch (TABLEBROWSER TBITEMS) of BROWSER))
            ; Item is not in hint tail, have to search whole list
            (SETQ TAILN 1)))]
        then (while (< TAILN N) do (if (NULL (SETQ TAIL (CDR TAIL)))
          then ; Greater than last item. I could have done a comparison against
            ; #items, but it is rare to ask for this (and we never do internally).
            (RETURN NIL))
          (add TAILN 1)
        finally (freplace (TABLEBROWSER TBTAILHINT) of BROWSER with TAIL)
          ; Store away the new hint. This makes ascending iterations
          ; constant time, rather than n^2.
          (RETURN (CAR TAIL]))
```

(TB.COLLECT.ITEMS

```
[LAMBDA (BROWSER PREDFN) ; Edited 27-Jan-88 16:18 by bvm
  (SELECTQ PREDFN
    (DELETED (SETQ PREDFN (FUNCTION TB.ITEM.DELETED?)))
    (SELECTED (SETQ PREDFN (FUNCTION TB.ITEM.SELECTED?)))
    NIL)
  (for ITEM in (ffetch (TABLEBROWSER TBITEMS) of (\DTEST BROWSER 'TABLEBROWSER)) collect ITEM
    when (OR (NULL PREDFN)
      (CL:FUNCALL PREDFN BROWSER ITEM))
```

(TB.MAP.ITEMS

[LAMBDA (BROWSER MAPFN NULLFN) ; Edited 27-Jan-88 16:18 by bvm

::: Apply MAPFN to each item in TABLEBROWSER -- args (TABLEBROWSER ITEM)

```
(LET [(ITEMS (ffetch (TABLEBROWSER TBITEMS) of (\DTEST BROWSER 'TABLEBROWSER)
(COND
  (ITEMS (for ITEM in ITEMS do (CL:FUNCALL MAPFN BROWSER ITEM)))
  (NULLFN (CL:FUNCALL NULLFN BROWSER]))
```

(TB.MAP.DELETED.ITEMS

[LAMBDA (BROWSER MAPFN NULLFN) ; Edited 27-Jan-88 16:18 by bvm

::: Apply MAPFN to each deleted item in TABLEBROWSER -- args (TABLEBROWSER ITEM)

```
(COND
  ((NEQ (ffetch TB#DELETED of (\DTEST BROWSER 'TABLEBROWSER)
  0)
  (for ITEM in (ffetch (TABLEBROWSER TBITEMS) of BROWSER) when (ffetch TIDELETED of ITEM)
  do (CL:FUNCALL MAPFN BROWSER ITEM)))
  (NULLFN ; Nothing deleted
  (CL:FUNCALL NULLFN BROWSER]))
```

(TB.MAP.SELECTED.ITEMS

[LAMBDA (BROWSER MAPFN NULLFN) ; Edited 27-Jan-88 16:19 by bvm

::: Apply MAPFN to each selected item in TABLEBROWSER -- args (TABLEBROWSER ITEM)

```
(LET ([ITEM# (SUB1 (ffetch (TABLEBROWSER TBFIRSTSELECTEDITEM) of (\DTEST BROWSER 'TABLEBROWSER)
(LASTITEM# (ffetch (TABLEBROWSER TBLASTSELECTEDITEM) of BROWSER))
ITEM)
(COND
  ((< ITEM# LASTITEM#)
  (until (> (add ITEM# 1)
  LASTITEM#)
  when (ffetch (TABLEITEM TISELECTED) of (SETQ ITEM (TB.NTH.ITEM BROWSER ITEM#)))
  do (CL:FUNCALL MAPFN BROWSER ITEM)))
  (NULLFN ; Nothing selected
  (CL:FUNCALL NULLFN BROWSER]))
```

(TB.FIND.ITEM

[LAMBDA (BROWSER PREDFN FIRST# LAST# BACKWARDSFLG) ; Edited 27-Jan-88 16:20 by bvm

::: Returns the first item in the designated range satisfying (PREDFN browser item); range defaults to whole browser

```
(\DTEST BROWSER 'TABLEBROWSER)
(LET ((LO (COND
  (FIRST# (IMAX FIRST# 1))
  (T 1)))
  [HI (COND
  (LAST# (IMIN LAST# (ffetch (TABLEBROWSER TB#ITEMS) of BROWSER)))
  (T (ffetch (TABLEBROWSER TB#ITEMS) of BROWSER]
  I END INCREMENT ITEM)
(COND
  ((<= LO HI)
  (COND
    (BACKWARDSFLG (SETQ I (ADD1 HI))
    (SETQ END LO)
    (SETQ INCREMENT -1))
    (T (SETQ I (SUB1 LO))
    (SETQ END HI)
    (SETQ INCREMENT 1))))
  (SELECTQ PREDFN
  (DELETED (SETQ PREDFN (FUNCTION TB.ITEM.DELETED?)))
  (SELECTED (SETQ PREDFN (FUNCTION TB.ITEM.SELECTED?)))
  NIL)
  (when [CL:FUNCALL PREDFN BROWSER (SETQ ITEM (TB.NTH.ITEM BROWSER (add I INCREMENT)
  do (RETURN ITEM) repeatuntil (EQ I END])
```

(TB.ITEM.SELECTED?

[LAMBDA (BROWSER ITEM) ; Edited 27-Jan-88 16:20 by bvm

```
(ffetch TISELECTED of (\DTEST ITEM 'TABLEITEM])
```

(TB.ITEM.DELETED?

[LAMBDA (BROWSER ITEM) ; Edited 27-Jan-88 16:20 by bvm

```
(ffetch TIDELETED of (\DTEST ITEM 'TABLEITEM])
```

)

(DEFINEQ

(TB.CLEAR.LINE

[LAMBDA (BROWSER ITEM LEFT WIDTH) ; Edited 22-Jan-88 16:06 by bvm

:: Clears the contents of ITEM's line starting at xpos LEFT for width WIDTH. Defaults to whole line

```
(BLTSHADE WHITESHADE (ffetch (TABLEBROWSER TBWINDOW) of (\DTEST BROWSER 'TABLEBROWSER))
  (OR LEFT 0)
  (TB.BOTTOM.OF.ITEM BROWSER ITEM)
  WIDTH
  (ffetch (TABLEBROWSER TBITEMHEIGHT) of BROWSER)
  'REPLACE])
```

(TB.USERDATA

[CL:LAMBDA (BROWSER &OPTIONAL (NEWDATA NIL NEWP)) ; Edited 27-Jan-88 16:25 by bvm

```
(PROG1 (ffetch (TABLEBROWSER TBUSERDATA) of (\DTEST BROWSER 'TABLEBROWSER))
  (COND
    (NEWP (freplace (TABLEBROWSER TBUSERDATA) of BROWSER with NEWDATA))))])
```

(TB.WINDOW

[LAMBDA (BROWSER) ; Edited 27-Jan-88 16:25 by bvm

```
(ffetch (TABLEBROWSER TBWINDOW) of (\DTEST BROWSER 'TABLEBROWSER])
```

)

:: Display

(DEFINEQ

(TB.REPAINTFN

[LAMBDA (WINDOW REGION) (* bvm%: "10-Sep-85 13:00")

```
(PROG [(BROWSER (WINDOWPROP WINDOW 'TABLEBROWSER)
  (AND (NEQ (fetch (TABLEBROWSER TB#ITEMS) of BROWSER)
    0)
  (RESETLST
    (COND
      ((OBTAIN.MONITORLOCK (fetch (TABLEBROWSER TBLOCK) of BROWSER)
        T T)
        (TB.DISPLAY.LINES BROWSER (TB.FIRST.VISIBLE.ITEM# BROWSER REGION)
          (TB.LAST.VISIBLE.ITEM# BROWSER REGION)))
      (T (TB.BROWSER.BUSY BROWSER))))])
```

(TB.RESHAPEFN

[LAMBDA (WINDOW OLDIMAGEBM OLDREGION) ; Edited 22-Jan-88 10:21 by bvm

```
(RESETLST
  (PROG ((BROWSER (WINDOWPROP WINDOW 'TABLEBROWSER))
    (REGION (DSPCLIPPINGREGION NIL WINDOW))
    ITEM#)
  [COND
    ((NOT (OBTAIN.MONITORLOCK (fetch (TABLEBROWSER TBLOCK) of BROWSER)
      T T))
      ; Browser is busy, have to wait until it is ready. But don't tie up
      ; mouse!
      (ALLOW.BUTTON.EVENTS)
      (OBTAIN.MONITORLOCK (fetch (TABLEBROWSER TBLOCK) of BROWSER)
        NIL T))
    ((NOT (fetch (TABLEBROWSER TBREADY) of BROWSER)) ; Browser not functional
      (RETURN (RESHAPEBYREPAINTFN WINDOW OLDIMAGEBM OLDREGION]
  (SETQ ITEM# (TB.FIRST.VISIBLE.ITEM# BROWSER REGION))
  (TB.SET.FONT BROWSER)
  (WYOFFSET (TIMES (SUB1 ITEM#)
    (fetch (TABLEBROWSER TBITEMHEIGHT) of BROWSER))
    WINDOW)
  (TB.DISPLAY.LINES BROWSER ITEM# (TB.LAST.VISIBLE.ITEM# BROWSER REGION))))])
```

(TB.SCROLLFN

[LAMBDA (WINDOW DX DY CONTINUOUSFLG) ; Edited 22-Jan-88 17:32 by bvm

:: only scroll if can get the monitor lock

```
(RESETLST
  [LET ((BROWSER (WINDOWPROP WINDOW 'TABLEBROWSER))
    HW)
  (COND
    ((OBTAIN.MONITORLOCK (fetch (TABLEBROWSER TBLOCK) of BROWSER)
      T T)
      (SCROLLBYREPAINTFN WINDOW DX DY CONTINUOUSFLG)
      (if (AND (EQ DY 0)
        (SETQ HW (fetch (TABLEBROWSER TBHEADINGWINDOW) of BROWSER)))
        then ; Horizontally scroll the header window together with it.
          (SCROLLW HW DX DY CONTINUOUSFLG)))
      (T (TB.BROWSER.BUSY BROWSER))))])
```

(TB.DISPLAY.LINES

```
[LAMBDA (BROWSER FIRST# LAST#) ; Edited 25-Jan-88 18:34 by bvm
  (for ITEM# from (IMAX FIRST# 1) to (IMIN LAST# (fetch (TABLEBROWSER TB#ITEMS) of BROWSER))
    bind (WINDOW _ (fetch (TABLEBROWSER TBWINDOW) of BROWSER))
          (MAXXPOS _ (fetch (TABLEBROWSER TBMAXXPOS) of BROWSER))
          (PRINTFN _ (fetch (TABLEBROWSER TBPRINTFN) of BROWSER))
          EXTENTCHANGED ITEM HERE EXTENT HWINDOW
    do (SETQ ITEM (TB.NTH.ITEM BROWSER ITEM#))
        (TB.PRINT.LINE BROWSER ITEM WINDOW PRINTFN) ; keep track of maximum width printed to, so window's EXTENT
                                                ; is always right
      (COND
        ((< MAXXPOS (SETQ HERE (DSPXPOSITION NIL WINDOW)))
         (SETQ MAXXPOS HERE)
         (SETQ EXTENTCHANGED T)))
      finally (COND
        (EXTENTCHANGED (replace (TABLEBROWSER TBMAXXPOS) of BROWSER with MAXXPOS)
                            (replace (REGION WIDTH) of (SETQ EXTENT (fetch (TABLEBROWSER TBEXTENT) of BROWSER))
                                     with MAXXPOS)
                            (WINDOWPROP WINDOW 'EXTENT EXTENT)
                            (if (SETQ HWINDOW (fetch (TABLEBROWSER TBHEADINGWINDOW) of BROWSER))
                                then ; Update heading window extent, too. Width has to account for
                                    ; the difference, if any, in borders.
                                    (replace (REGION WIDTH) of (SETQ EXTENT (WINDOWPROP HWINDOW 'EXTENT))
                                            with (+ MAXXPOS (TIMES 2 (- (WINDOWPROP WINDOW 'BORDER)
                                                                    (WINDOWPROP HWINDOW 'BORDER)))))
                                )
        )
      )
  )

```

(TB.PRINT.LINE

```
[LAMBDA (BROWSER ITEM WINDOW PRINTFN) ; Edited 22-Jan-88 17:16 by bvm
  (MOVETO TB.LEFT.MARGIN (+ (TB.BOTTOM.OF.ITEM BROWSER ITEM)
                             (fetch (TABLEBROWSER TBBASELINE) of BROWSER))
   WINDOW) ; Move to item's baseline
  (POSITION WINDOW 0)
  (CL:FUNCALL PRINTFN BROWSER ITEM WINDOW)
  [TB.SHOW.SELECTION BROWSER ITEM (COND
    ((fetch (TABLEITEM TISELECTED) of ITEM)
     'REPLACE)
    (T 'ERASE]
  (COND
    ((fetch (TABLEITEM TIDLETED) of ITEM)
     (TB.SHOW.DELETION BROWSER ITEM WINDOW 'REPLACE])
  )

```

(TB.FIRST.VISIBLE.ITEM#

```
[LAMBDA (BROWSER REGION) ; Edited 22-Jan-88 16:59 by bvm
  ;; Computes number of the first item in TABLEBROWSER that is visible in REGION
  (IMAX 1 (ADD1 (IQUOTIENT [- (fetch (TABLEBROWSER TBORIGIN) of (\DTEST BROWSER 'TABLEBROWSER))
                             (fetch (REGION PTOP) of (OR REGION (DSPCLIPPINGREGION NIL (fetch (TABLEBROWSER
                                                                                               TBWINDOW)
                                                                                               of BROWSER))
                             (fetch (TABLEBROWSER TBITEMHEIGHT) of BROWSER])
  )

```

(TB.LAST.VISIBLE.ITEM#

```
[LAMBDA (BROWSER REGION) ; Edited 22-Jan-88 17:00 by bvm
  ;; Computes number of the last item in TABLEBROWSER that is visible in REGION
  (IMIN (ffetch (TABLEBROWSER TB#ITEMS) of (\DTEST BROWSER 'TABLEBROWSER))
   (CL:CEILING [- (fetch (TABLEBROWSER TBORIGIN) of BROWSER)
                 (fetch (REGION BOTTOM) of (OR REGION (DSPCLIPPINGREGION NIL (ffetch (TABLEBROWSER TBWINDOW)
                                                                                               of BROWSER))
                 (fetch (TABLEBROWSER TBITEMHEIGHT) of BROWSER])
  )

```

(TB.ITEM.VISIBLE?

```
[LAMBDA (BROWSER ITEM) ; Edited 22-Jan-88 16:12 by bvm
  ;; True if any part of ITEM is visible in window of BROWSER
  (LET ([CLIP (DSPCLIPPINGREGION NIL (ffetch (TABLEBROWSER TBWINDOW) of (\DTEST BROWSER 'TABLEBROWSER))
      (BOT (TB.BOTTOM.OF.ITEM BROWSER ITEM))
      ;; Check bottom of line is below top, and top of line is above the bottom
      (AND (< BOT (fetch (REGION PTOP) of CLIP))
           (< (fetch (REGION BOTTOM) of CLIP)
              (+ BOT (fetch (TABLEBROWSER TBITEMHEIGHT) of BROWSER))
            )
    )
  )

```

(TB.ITEM.FROM.YCOORD

```
[LAMBDA (BROWSER YPOS) ; Edited 22-Jan-88 16:41 by bvm
  (LET [(N (CL:CEILING (- (fetch (TABLEBROWSER TBORIGIN) of BROWSER)
                         YPOS)
                       (fetch (TABLEBROWSER TBITEMHEIGHT) of BROWSER))
        (TB.NTH.ITEM BROWSER (COND
          ((<= N 0)
           1)
        )
  )

```


(T (IMIN N (fetch (TABLEBROWSER TB#ITEMS) of BROWSER]))

(TB.BOTTOM.OF.ITEM

[LAMBDA (BROWSER ITEM)

; Edited 27-Jan-88 16:11 by bvm

:: Returns the y position of the bottom of specified item (number or tableitem). Add the font descent to get the baseline of the first line.

```
(- (fetch (TABLEBROWSER TBORIGIN) of BROWSER)
  (TIMES (fetch (TABLEBROWSER TBITEMHEIGHT) of BROWSER)
    (OR (FIXP ITEM)
        (ffetch (TABLEITEM TI#) of (\DTEST ITEM 'TABLEITEM))
```

(TB.SHOW.DELETION

[LAMBDA (BROWSER ITEM WINDOW OPERATION)

; Edited 27-Jan-88 17:00 by bvm

:: Draws or erases, for OPERATION = REPLACE or ERASE, the line indicating that ITEM is deleted

```
(LET ((THICKNESS (fetch (TABLEBROWSER TBLINETHICKNESS) of BROWSER))
      (BASELINE (fetch (TABLEBROWSER TBBASELINE) of BROWSER)))
  (BLTSHADE BLACKSHADE WINDOW TB.LEFT.MARGIN (PROGN
    ;; Center the deletion line between the baseline and the top of the item
    (+ (SUB1 BASELINE)
       (IQUOTIENT (- (fetch (TABLEBROWSER TBITEMHEIGHT) of BROWSER)
                     BASELINE THICKNESS)
                  2)
       (TB.BOTTOM.OF.ITEM BROWSER ITEM)))
    NIL THICKNESS OPERATION])
```

(TB.SHOW.SELECTION

[LAMBDA (BROWSER ITEM OPERATION)

; Edited 27-Jan-88 15:42 by bvm

:: Displays or erases, per OPERATION = REPLACE or ERASE, the mark indicating that ITEM is selected

```
(LET ((BASELINE (fetch (TABLEBROWSER TBBASELINE) of BROWSER))
      (BM TB.SELECTION.BITMAP))
  (BITBLT BM 0 0 (fetch (TABLEBROWSER TBWINDOW) of BROWSER)
    0
    (PROGN
      ;; Center the selection bitmap between the baseline and the top of the item, rounding down slightly on the grounds that
      ;; the top pixel of the line tends to be blank, so the center of gravity is lower than it might be.
      (+ (SUB1 BASELINE)
         (IQUOTIENT (- (fetch (TABLEBROWSER TBITEMHEIGHT) of BROWSER)
                       BASELINE
                       (fetch BITMAPHEIGHT of BM))
                    2)
         (TB.BOTTOM.OF.ITEM BROWSER ITEM)))
      NIL NIL 'INPUT OPERATION])
```

(TB.UPDATE.DISPLAY

[LAMBDA (BROWSER FROMITEM# TYPE)

; Edited 11-Feb-88 11:34 by bvm

:: Updates the display window appropriately after a TYPE operation (REMOVE or INSERT) on TABLEBROWSER that affects items starting at FROMITEM#

```
(PROG ((WINDOW (fetch (TABLEBROWSER TBWINDOW) of BROWSER))
      (EXTENT (fetch (TABLEBROWSER TBEXTENT) of BROWSER))
      (LASTITEM# (fetch (TABLEBROWSER TB#ITEMS) of BROWSER))
      (ITEMHEIGHT (fetch (TABLEBROWSER TBITEMHEIGHT) of BROWSER))
      (ITEMBOTTOM (TB.BOTTOM.OF.ITEM BROWSER FROMITEM#))
      DELTA HEIGHT LAST# CLIP WBOTTOM EXTENTBOTTOM)
  [add (fetch (REGION HEIGHT) of EXTENT)
    (SETQ DELTA (SELECTQ TYPE
      (REMOVE (- ITEMHEIGHT))
      (INSERT ITEMHEIGHT)
      (SHOULDNT])
    (SETQ CLIP (DSPCLIPPINGREGION NIL WINDOW))
    (COND
      (>= ITEMBOTTOM (fetch (REGION PTOP) of CLIP)) ; Changed item above top of window, so no visible change -- just
      ; cheat the origin appropriately
      (add (fetch (TABLEBROWSER TBORIGIN) of BROWSER)
        DELTA))
      (T ; Changed item visible or below bottom of window, so bottom of
      ; extent changes
      (replace (REGION BOTTOM) of EXTENT with (SETQ EXTENTBOTTOM (- (fetch (REGION BOTTOM) of EXTENT)
        DELTA)))
      (COND
        (<= (+ ITEMBOTTOM ITEMHEIGHT)
          (SETQ WBOTTOM (fetch (REGION BOTTOM) of CLIP))) ; Below bottom of window, so we're done
        )
      )
    ((TB.ITEM.UPDATABLE? BROWSER FROMITEM#) ; If window is visible, update it now
      (SELECTQ TYPE
```

```

(ININSERT                                     ; Push everything from line FROMITEM# down one line, then
                                           ; redisplay item FROMITEM#
(BITBLT WINDOW 0 (+ WBOTTOM ITEMHEIGHT)
 WINDOW 0 WBOTTOM NIL (- ITEMBOTTOM WBOTTOM)
 'INPUT
 'REPLACE)
(TB.DISPLAY.LINES BROWSER FROMITEM# FROMITEM#))
(REMOVE                                       ; Pull everything below line FROMITEM# up one line, then
                                           ; redisplay last visible item(s)
(BITBLT WINDOW 0 WBOTTOM WINDOW 0 (+ WBOTTOM ITEMHEIGHT)
 NIL
 (- ITEMBOTTOM WBOTTOM)
 'INPUT
 'REPLACE)
(TB.DISPLAY.LINES BROWSER (SETQ LAST# (+ FROMITEM# (IQUOTIENT (- ITEMBOTTOM
                                                                WBOTTOM)
                                                                ITEMHEIGHT)))
 (ADD1 LAST#))                               ; May have to display two lines if the bottom line of window was a
                                           ; half line
[COND
 (> EXTENTBOTTOM WBOTTOM)                   ; Clear everything below the extent
 (BLTSHADE WHITESHADE WINDOW 0 WBOTTOM NIL (- EXTENTBOTTOM WBOTTOM)
 'REPLACE)]
(SHOULDNT])

```

(TB.ITEM.UPDATABLE?)

[LAMBDA (BROWSER ITEM ONLYIFVISIBLE) ; Edited 29-Jan-88 12:08 by bvm

;;; True if window of BROWSER is open. If false, we update the TBUPDATEFROMHERE field, denoting that we should repaint window when it is opened. If ONLYIFVISIBLE is true, we do nothing and return NIL if the item is not currently visible.

```

(OR (FIXP ITEM)
 (SETQ ITEM (fetch TI# of ITEM)))
(COND
 ((AND ONLYIFVISIBLE (NOT (TB.ITEM.VISIBLE? BROWSER ITEM))) ; Item not visible, so no need to change display
 NIL)
 ((OPENWP (fetch (TABLEBROWSER TBWINDOW) of BROWSER)))
 (T [LET ((OLDN (fetch (TABLEBROWSER TBUPDATEFROMHERE) of BROWSER)))
 (COND
 ((OR (NULL OLDN)
 (< ITEM OLDN))
 (replace (TABLEBROWSER TBUPDATEFROMHERE) of BROWSER with ITEM)
 ; Mark browser for display update after being unshrunk
 NIL])
 )
 )
 )

```

;; Selection

(DEFINEQ

(TB.BUTTONEVENTFN

[LAMBDA (WINDOW) (* bvm%: " 6-Sep-85 15:23")

```

(TOTOPW WINDOW)
(LET (FN)
 (COND
 ((INSIDEP (DSPCLIPPINGREGION NIL WINDOW)
 (LASTMOUSEX WINDOW)
 (LASTMOUSEY WINDOW))
 (TB.DO.UNLESS.BUSY WINDOW (FUNCTION TB.DO.ITEM.SELECTION)))
 ((LASTMOUSESTATE (ONLY RIGHT))
 (DOWINDOWCOM WINDOW))
 ([AND (LASTMOUSESTATE (ONLY MIDDLE))
 (SETQ FN (fetch (TABLEBROWSER TBTITLEEVENTFN) of (WINDOWPROP WINDOW 'TABLEBROWSER)
 (TB.DO.UNLESS.BUSY WINDOW FN])
 )
 )
 )

```

(TB.DO.UNLESS.BUSY

[LAMBDA (WINDOW FN) ; Edited 20-Jan-88 23:30 by bvm

```

(RESETLST
 [LET [(BROWSER (WINDOWPROP WINDOW 'TABLEBROWSER)
 (COND
 ((AND (fetch (TABLEBROWSER TBREADY) of BROWSER)
 (OBTAIN.MONITORLOCK (fetch (TABLEBROWSER TBLOCK) of BROWSER)
 T T))
 (CL:FUNCALL FN WINDOW BROWSER)])
 )
 )

```

(TB.DO.ITEM.SELECTION

[LAMBDA (WINDOW) ; Edited 20-Jan-88 22:17 by bvm

```

(DECLARE (GLOBALVARS LASTMOUSEBUTTONS)
 (SPECVARS SELECTIONSTATE BROWSER FIRSTVISIBLE# LASTVISIBLE#))
(PROG ((BROWSER (WINDOWPROP WINDOW 'TABLEBROWSER))
 SELECTIONREGION FIRST# LAST# FIRSTVISIBLE# LASTVISIBLE# SELECTIONSTATE NEWSELECTION OLDSELECTION SEL#
 OLDSEL# CTRLDOWN OLDLASTMOUSEBUTTONS ITEM LASTX LASTY)
 (COND
 ((EQ (fetch (TABLEBROWSER TB#ITEMS) of BROWSER)

```

```

0) ; Nothing to select
(RETURN))
(SETQ SELECTIONREGION (DSPCLIPPINGREGION NIL WINDOW))
(SETQ LAST# (fetch TBLASTSELECTEDITEM of BROWSER))
(SETQ FIRST# (fetch TBFIRSTSELECTEDITEM of BROWSER))
(SETQ FIRSTVISIBLE# (TB.FIRST.VISIBLE.ITEM# BROWSER SELECTIONREGION))
(SETQ LASTVISIBLE# (TB.LAST.VISIBLE.ITEM# BROWSER SELECTIONREGION))
;; keep looping until all mouse buttons are up
(do (GETMOUSESTATE)
  (COND
    [[NOT (INSIDEP SELECTIONREGION (SETQ LASTX (LASTMOUSEX WINDOW))
      (SETQ LASTY (LASTMOUSEY WINDOW))
      ;; I would like to just return here and let the next window take over, but current mouse arrangement means I'll never get control
      ;; back unless user lets up on mouse
      (COND
        ((NEQ SELECTIONSTATE TS.IDLE)
          (TB.UNDOSELECTION) ; Forget what we were doing
          (SETQ OLDSELECTION))
        (COND
          ((LASTMOUSESTATE UP)
            (RETURN))
            (T (BLOCK]
          ((LASTMOUSESTATE UP) ; Make selection permanent
            (AND OLDSELECTION (SETQ OLDSEL# (fetch TI# of OLDSELECTION)))
            (SELECTC SELECTIONSTATE
              (TS.REPLACING (for I from FIRST# to LAST# do (replace TISELECTED of (TB.NTH.ITEM BROWSER I)
                with NIL))
                (replace TISELECTED of OLDSELECTION with T)
                (replace TBFIRSTSELECTEDITEM of BROWSER
                  with (replace TBLASTSELECTEDITEM of BROWSER with OLDSEL#)))
              (TS.ADDING (TB.SELECTRANGE BROWSER OLDSEL# OLDSEL# T))
              (TS.REMOVING (TB.DESELECTRANGE BROWSER OLDSEL# OLDSEL#))
              (TS.EXTENDING.HI
                (TB.SELECTRANGE BROWSER (ADD1 LAST#)
                  OLDSEL# CTRLDOWN))
              (TS.EXTENDING.LO
                (TB.SELECTRANGE BROWSER OLDSEL# (SUB1 FIRST#)
                  CTRLDOWN))
              (TS.SHRINKING.HI
                (TB.DESELECTRANGE BROWSER (ADD1 OLDSEL#)
                  LAST#))
              (TS.SHRINKING.LO
                (TB.DESELECTRANGE BROWSER FIRST# (SUB1 OLDSEL#)))
              NIL)
            (RETURN))
            [(AND NIL ; In a special column
              )
            (COND
              ((NEQ SELECTIONSTATE TS.IDLE)
                (TB.UNDOSELECTION)
                (SETQ OLDSELECTION]
              ((OR (NEQ (SETQ NEWSELECTION (TB.ITEM.FROM.YCOORD BROWSER LASTY))
                OLDSELECTION)
                (NEQ LASTMOUSEBUTTONS OLDLASTMOUSEBUTTONS))
                ; Something changed
              (COND
                [(AND (fetch TIUNSELECTABLE of NEWSELECTION)
                  (NOT (LASTMOUSESTATE RIGHT))) ; Can't select that item, so revert to idle
                  (COND
                    ((NEQ SELECTIONSTATE TS.IDLE)
                      (TB.UNDOSELECTION]
                    [(AND (LASTMOUSESTATE (OR LEFT MIDDLE))
                      (SHIFTDOWNP 'CTRL)) ; Deselect this item
                      (SELECTC SELECTIONSTATE
                        (TS.REMOVING ; we were deselecting, so reselect that guy
                          (TB.SHOW.SELECTION BROWSER OLDSELECTION 'REPLACE))
                          (TS.IDLE ; nothing going on
                            )
                          (TB.UNDOSELECTION))
                        (SETQ SELECTIONSTATE (COND
                          ((fetch TISELECTED of NEWSELECTION)
                            (TB.SHOW.SELECTION BROWSER NEWSELECTION 'ERASE)
                            TS.REMOVING)
                          (T TS.IDLE]
                        ((LASTMOUSESTATE LEFT) ; Set (change) the selection to this single item
                          (COND
                            ((EQ SELECTIONSTATE TS.REPLACING)
                              (TB.SHOW.SELECTION BROWSER OLDSELECTION 'ERASE))
                              (T (TB.DECONSIDERRANGE FIRSTVISIBLE# LASTVISIBLE#
                                (SETQ SELECTIONSTATE TS.REPLACING)))
                              (TB.SHOW.SELECTION BROWSER NEWSELECTION 'REPLACE))
                            [(LASTMOUSESTATE MIDDLE) ; Add this item to the selection
                              (SELECTC SELECTIONSTATE
                                (TS.ADDING (TB.SHOW.SELECTION BROWSER OLDSELECTION 'ERASE))
                                (TS.IDLE)

```

```

(TB.UNDOSELECTION)
(SETQ SELECTIONSTATE (COND
  ((NOT (fetch TISELECTED of NEWSELECTION))
    (TB.SHOW.SELECTION BROWSER NEWSELECTION 'REPLACE)
    TS.ADDING)
  (T TS.IDLE]
((LASTMOUSESTATE RIGHT) ; Extend: either up or down, or shrink a selection. This is messy
(SETQ SEL# (fetch TI# of NEWSELECTION))
(SETQ OLDSEL# (AND OLDSELECTION (fetch TI# of OLDSELECTION)))
(SELECTC SELECTIONSTATE
  (TS.EXTENDING.HI
    (COND
      (> SEL# OLDSEL#) ; Extend further
      (TB.CONSIDERRANGE (ADD1 OLDSEL#)
        SEL# CTRLDOWN))
      (T ; Shrinking back
        (TB.RECONSIDERRANGE (ADD1 (COND
          (> SEL# LAST#)
          SEL#)
          (T (SETQ SELECTIONSTATE TS.IDLE)
            LAST#)))
          OLDSEL#))))
  (TS.EXTENDING.LO
    [COND
      (< SEL# OLDSEL#) ; Extend further
      (TB.CONSIDERRANGE SEL# (SUB1 OLDSEL#)
        CTRLDOWN))
      (T ; Shrinking back
        (TB.RECONSIDERRANGE OLDSEL# (SUB1 (COND
          (< SEL# FIRST#)
          SEL#)
          (T (SETQ SELECTIONSTATE TS.IDLE)
            FIRST#)))
          OLDSEL#))))
  (TS.SHRINKING.HI
    (COND
      [(>= SEL# OLDSEL#) ; Shrinking less
        (TB.RECONSIDERRANGE (ADD1 OLDSEL#)
          (COND
            (< SEL# LAST#)
            SEL#)
          (T (SETQ SELECTIONSTATE TS.IDLE)
            LAST#]
          OLDSEL#))
        (>= SEL# FIRST#) ; Shrinking further
        (TB.DECONSIDERRANGE (ADD1 SEL#)
          OLDSEL#))
        (T ; Too far to shrink
          (TB.RECONSIDERRANGE FIRST# LAST#)
          (SETQ SELECTIONSTATE TS.IDLE)))]
  (TS.SHRINKING.LO
    (COND
      (<= SEL# OLDSEL#) ; Shrinking less
      (TB.RECONSIDERRANGE (COND
        (> SEL# FIRST#)
        SEL#)
        (T (SETQ SELECTIONSTATE TS.IDLE)
          FIRST#))
        (SUB1 OLDSEL#)))
      (<= SEL# LAST#) ; Shrinking further
      (TB.DECONSIDERRANGE OLDSEL# (SUB1 SEL#)))
      (T ; Too far to shrink
        (TB.RECONSIDERRANGE FIRST# LAST#)
        (SETQ SELECTIONSTATE TS.IDLE)))]
  (COND
    ((<= FIRST# LAST#) ; Something is already selected, so we can think about
      ; extending.
    (COND
      ((NEQ SELECTIONSTATE TS.IDLE) ; Cancel any selection we were thinking about
        (TB.UNDOSELECTION)))
      (SETQ CTRLDOWN (SHIFTDOWNP 'CTRL))
      (SETQ SELECTIONSTATE (COND
        (> SEL# LAST#)
        (TB.CONSIDERRANGE (ADD1 LAST#)
          SEL# CTRLDOWN)
        TS.EXTENDING.HI)
        (< SEL# FIRST#)
        (TB.CONSIDERRANGE SEL# (SUB1 FIRST#)
          CTRLDOWN)
        TS.EXTENDING.LO)
        (> SEL# (LRSH (+ LAST# FIRST#)
          1))
          ; we are closer to the high end, but inside. Shrink from the top,
          ; but only if we are pointing at a contiguous selection
          (if (TB.CONTIGUOUS.HELP BROWSER SEL# (SUB1 LAST#))
            then (TB.DECONSIDERRANGE (ADD1 SEL#)
              LAST#)
            TS.SHRINKING.HI
            else TS.IDLE))

```

```

(T ; We are closer to the low end, so shrink from bottom
  (if (TB.CONTIGUOUS.SELP BROWSER (ADD1 FIRST#)
      SEL#)
    then (TB.DECONSIDERRANGE FIRST# (SUB1 SEL#))
        TS.SHRINKING.LO
    else TS.IDLE]
  (SETQ OLDMOUSEBUTTONS LASTMOUSEBUTTONS)
  (SETQ OLDSELECTION NEWSELECTION])

```

(TB.CONTIGUOUS.SELP

[LAMBDA (BROWSER FIRST# LAST#) ; Edited 20-Jan-88 22:16 by bvm

;; true if all the elements of ITEMS from FIRST# to LAST# are selected (or deleted or unselectable)

```

(for I from FIRST# to LAST# bind ITEM always (OR (fetch TISELECTED of (SETQ ITEM (TB.NTH.ITEM BROWSER I)))
  (fetch TIDELETED of ITEM)
  (fetch TIUNSELECTABLE of ITEM))

```

(TB.DECONSIDERRANGE

[LAMBDA (FIRST# LAST#) ; Edited 20-Jan-88 22:08 by bvm

;;; Change display so that items from FIRST# to LAST# are marked as unselected.

```

(DECLARE (USEDFREE BROWSER FIRSTVISIBLE# LASTVISIBLE#))
(for I from (IMAX FIRST# FIRSTVISIBLE#) to (IMIN LAST# LASTVISIBLE#) do (TB.SHOW.SELECTION BROWSER
  (TB.NTH.ITEM BROWSER I)
  'ERASE])

```

(TB.CONSIDERRANGE

[LAMBDA (FIRST# LAST# EVENIFDELETED) ; Edited 20-Jan-88 22:08 by bvm

;;; Change display so that items from FIRST# to LAST# are marked as selected. Deleted items are not selected unless EVENIFDELETED is true

```

(DECLARE (USEDFREE BROWSER FIRSTVISIBLE# LASTVISIBLE#))
(for I from (IMAX FIRST# FIRSTVISIBLE#) to (IMIN LAST# LASTVISIBLE#) bind ITEM
  do (SETQ ITEM (TB.NTH.ITEM BROWSER I))
      (COND
        ([AND (NOT (fetch TIUNSELECTABLE of ITEM))
              (OR EVENIFDELETED (NOT (fetch TIDELETED of ITEM))
              (TB.SHOW.SELECTION BROWSER ITEM 'REPLACE])

```

(TB.DESELECTRANGE

[LAMBDA (BROWSER FIRST# LAST#) ; Edited 20-Jan-88 22:09 by bvm

;;; Mark internally items FIRST# thru LAST# as unselected. Keeps TBFIRSTSELECTEDITEM and TBLASTSELECTEDITEM up to date. Assumes display has already been appropriately modified--use TB.UNSELECT.ALL.ITEMS to do both

```

(LET ((FIRSTSEL (fetch TBFIRSTSELECTEDITEM of BROWSER))
      (LASTSEL (fetch TBLASTSELECTEDITEM of BROWSER)))
  (if (< FIRST# FIRSTSEL)
    then (SETQ FIRST# FIRSTSEL))
  (if (> LAST# LASTSEL)
    then (SETQ LAST# LASTSEL))
  (if (<= FIRST# LAST#)
    then (for I from FIRST# to LAST# do (replace TISELECTED of (TB.NTH.ITEM BROWSER I) with NIL))
      (COND
        [(EQ FIRST# FIRSTSEL)
         (replace TBFIRSTSELECTEDITEM of BROWSER with (COND
           ((TB.FIND.SELECTED.ITEM BROWSER (ADD1 LAST#)
            LASTSEL))
           (T (replace TBLASTSELECTEDITEM of BROWSER
             with 0)
             ; Null selection indicated by first GT last.
             (ADD1 (fetch (TABLEBROWSER TB#ITEMS)
               of BROWSER])
           ))
        [(EQ LAST# LASTSEL)
         (replace TBLASTSELECTEDITEM of BROWSER with (OR (TB.REV.FIND.SELECTED.ITEM BROWSER FIRSTSEL
           (SUB1 FIRST#))
           1])

```

(TB.RECONSIDERRANGE

[LAMBDA (FIRST# LAST#) ; Edited 20-Jan-88 22:09 by bvm

;;; Change display so that messages from FIRST# to LAST# are marked as selected or unselected according to the truth of the matter.

```

(DECLARE (USEDFREE BROWSER FIRSTVISIBLE# LASTVISIBLE#))
(for I from (IMAX FIRST# FIRSTVISIBLE#) to (IMIN LAST# LASTVISIBLE#) bind ITEM
  do (TB.SHOW.SELECTION BROWSER (SETQ ITEM (TB.NTH.ITEM BROWSER I))
      (COND
        ((fetch TISELECTED of ITEM)
         'REPLACE)
        (T 'ERASE])

```

(TB.SELECTRANGE

[LAMBDA (BROWSER FIRST# LAST# EVENIFDELETED) ; Edited 20-Jan-88 22:10 by bvm

;;; Mark internally items FIRST# thru LAST# as selected. Do not select deleted messages unless EVENIFDELETED is true. Keeps
 ;;; TBFIRSTSELECTEDITEM and TBLASTSELECTEDITEM up to date. Assumes display has already been appropriately modified

```
(PROG ((FIRSTSEL (fetch TBFIRSTSELECTEDITEM of BROWSER))
      (LASTSEL (fetch TBLASTSELECTEDITEM of BROWSER))
      ITEM)
  [for I from FIRST# to LAST# do (SETQ ITEM (TB.NTH.ITEM BROWSER I))
    (COND
      ([AND (NOT (fetch TIUNSELECTABLE of ITEM))
            (OR EVENIFDELETED (NOT (fetch TIDELETED of ITEM))
              (replace TISELECTED of ITEM with T])
      (COND
        ((OR (> FIRSTSEL LASTSEL)
              (< FIRST# (fetch TBFIRSTSELECTEDITEM of BROWSER)))
          (replace TBFIRSTSELECTEDITEM of BROWSER with FIRST#))
        (COND
          ((OR (> FIRSTSEL LASTSEL)
                (> LAST# (fetch TBLASTSELECTEDITEM of BROWSER)))
            (replace TBLASTSELECTEDITEM of BROWSER with LAST#))
```

(TB.UNDOSELECTION

[LAMBDA NIL (* bvm%: " 6-Sep-85 15:04")

;;; Restore browser to state before any selections were attempted

```
(DECLARE (USEDFREE FIRSTVISIBLE# LASTVISIBLE# SELECTIONSTATE))
(TB.RECONSIDERRANGE FIRSTVISIBLE# LASTVISIBLE#)
(SETQ SELECTIONSTATE TS.IDLE])
```

(TB.FIND.SELECTED.ITEM

[LAMBDA (BROWSER FIRST# LAST#) ; Edited 20-Jan-88 22:11 by bvm
 (find I from (OR FIRST# 1) to (OR LAST# (fetch TB#ITEMS of BROWSER)) suchthat (fetch TISELECTED of (TB.NTH.ITEM BROWSER I]))

(TB.REV.FIND.SELECTED.ITEM

[LAMBDA (BROWSER FIRST# LAST#) ; Edited 20-Jan-88 22:11 by bvm
 (find I from (OR LAST# (fetch TB#ITEMS of BROWSER)) to (OR FIRST# 1) by -1 suchthat (fetch TISELECTED of (TB.NTH.ITEM BROWSER I]))

(DEFINEQ

(TB.COPYBUTTONEVENTFN

[LAMBDA (WINDOW) ; Edited 22-Jan-88 12:08 by bvm

;;; copy select an item from the window.

```
(PROG ((BROWSER (WINDOWPROP WINDOW 'TABLEBROWSER))
      SELECTIONREGION COPYFN CURRENTITEM NEWITEM LASTX LASTY)
  [COND
    ((OR (NULL (SETQ COPYFN (fetch (TABLEBROWSER TBCOPYFN) of BROWSER)))
          (NULL (fetch (TABLEBROWSER TBITEMS) of BROWSER)))
      (RETURN (TOTOPW WINDOW))
    (SETQ SELECTIONREGION (DSPCLIPPINGREGION NIL WINDOW))
  LP (TOTOPW WINDOW)
    (COND
      ((AND (SETQ NEWITEM (AND (INSIDEP SELECTIONREGION (SETQ LASTX (LASTMOUSEX WINDOW))
                                                                    (SETQ LASTY (LASTMOUSEY WINDOW)))
                            (TB.ITEM.FROM.YCOORD BROWSER LASTY)))
            (fetch TIUNCOPYSELECTABLE of NEWITEM))
        (SETQ NEWITEM NIL)))
    [COND
      ((NEQ CURRENTITEM NEWITEM)
        (COND
          (CURRENTITEM ; turn off old selection.
            (TB.SHOW.COPY.SELECTION BROWSER CURRENTITEM)))
        (COND
          ((SETQ CURRENTITEM NEWITEM)
            (TB.SHOW.COPY.SELECTION BROWSER CURRENTITEM) ; wait for a button up or move out of region
          LP2 (BLOCK)
            (COND
              ((NOT (.COPYKEYDOWNP.)) ; Finished, copy selected item
                (COND
                  (CURRENTITEM (TB.SHOW.COPY.SELECTION BROWSER CURRENTITEM)
                    (CL:FUNCALL COPYFN BROWSER CURRENTITEM)))
                (RETURN))
              ((MOUSESTATE UP) ; button up, no action
                (GO LP2))
```

(T (GO LP])

(TB.SHOW.COPY.SELECTION

[LAMBDA (BROWSER ITEM)

; Edited 22-Jan-88 16:38 by bvm

;;; underline this item in browser

(BLTSHADE GRAYSHADE (**fetch** (TABLEBROWSER TBWINDOW) **of** BROWSER)
TB.LEFT.MARGIN
(**TB.BOTTOM.OF.ITEM** BROWSER ITEM)
NIL 2 'INVERT])

)

;; Misc state change

(DEFINEQ

(TB.BROWSER.BUSY

[LAMBDA (BROWSER)
(RESETFORM (CURSOR TB.CROSSCURSOR)
(BLOCK 1000])

(* bvm%: " 8-Sep-85 16:42")

(TB.CLOSE/SHRINK

[LAMBDA (WINDOW FLG)
(RESETLST

; Edited 20-Jan-88 23:36 by bvm

[LET ((BROWSER (WINDOWPROP WINDOW 'TABLEBROWSER))
HOW?)
(COND
((OBTAIN.MONITORLOCK (**fetch** (TABLEBROWSER TBLOCK) **of** BROWSER)
T T)
(COND
((AND (SETQ HOW? (**fetch** (TABLEBROWSER TBCLOSEFN) **of** BROWSER))
(SETQ HOW? (CL:FUNCALL HOW? BROWSER WINDOW FLG)))
[COND
((NEQ HOW? 'DON'T)
(**TB.PROCESS** ` (,HOW? ',BROWSER ',WINDOW ',FLG)
' TB.UPDATE]
' DON'T)
(T (**TB.FINISH.CLOSE** BROWSER WINDOW FLG T)
NIL))
(T (printout PROMPTWINDOW T "Browser is busy, can't close")
'DON'T]]))

(TB.CLOSEFN

[LAMBDA (WINDOW)
(**TB.CLOSE/SHRINK** WINDOW 'CLOSE])

(* bvm%: " 6-Sep-85 12:25")

(TB.FINISH.CLOSE

[LAMBDA (BROWSER WINDOW CLOSEFLG DONTCLOSE)

(* bvm%: " 9-Sep-85 00:42")

;;; Takes care of closing/shrinking WINDOW after an update or expunge. DONTCLOSE is true if neither occurred, in which case we are being called
;;; directly from the CLOSEFN and should not close/shrink the window ourselves

(WITH.MONITOR (**fetch** (TABLEBROWSER TBLOCK) **of** BROWSER)
(SELECTQ CLOSEFLG
(CLOSE (SETQ WINDOW (**TB.FLUSH.WINDOW** BROWSER WINDOW))
(OR DONTCLOSE (CLOSEW WINDOW)))
(SHRINK (WINDOWADDPROP WINDOW 'EXPANDFN (FUNCTION TB.EXPANDFN))
(WINDOWDELPROP WINDOW 'SHRINKFN (FUNCTION TB.SHRINKFN))
(OR DONTCLOSE (SHRINKW WINDOW)))
NIL)))

(TB.FLUSH.WINDOW

[LAMBDA (BROWSER WINDOW)
(WINDOWDELPROP WINDOW 'CLOSEFN (FUNCTION TB.CLOSEFN))
[ERSETQ (LET ((FN (**fetch** (TABLEBROWSER TBAFTERCLOSEFN) **of** BROWSER))
(AND FN (CL:FUNCALL FN BROWSER WINDOW)
(**replace** (TABLEBROWSER TBITEMS) **of** BROWSER **with** (**replace** (TABLEBROWSER TBWINDOW) **of** BROWSER
with (**replace** (TABLEBROWSER TBTAILHINT) **of** BROWSER
with NIL)))
(WINDOWPROP WINDOW 'TABLEBROWSER NIL)
(OR (OPENWP WINDOW)
(OPENWP (WINDOWPROP WINDOW 'ICONWINDOW]))

; Edited 20-Jan-88 22:42 by bvm

(TB.SET.FONT

[LAMBDA (BROWSER FONT)

; Edited 10-Feb-88 11:07 by bvm:

;;; Sets/changes font of TABLEBROWSER to be FONT. Clears window. Caller is responsible for repainting window

(LET ((FONTGIVEN FONT)

```

(WINDOW (fetch (TABLEBROWSER TBWINDOW) of BROWSER))
WIDTH HEIGHT ASCENT TOTALHEIGHT ORIGIN FN EXTENT HW)
(CLEARW WINDOW)
[SETQ FONT (FONTCREATE (OR FONT (fetch (TABLEBROWSER TBFONT) of BROWSER)
(DSPFONT NIL WINDOW)
(DSPFONT FONT WINDOW)
(DSPRIGHTMARGIN MAX.SMALLP WINDOW)
(LINELENGTH T WINDOW)
(replace (TABLEBROWSER TBFONT) of BROWSER with FONT)
[replace (TABLEBROWSER TBFONTHEIGHT) of BROWSER with (SETQ HEIGHT (FONTPROP FONT 'HEIGHT)
(if (NOT (fetch (TABLEBROWSER TBHEIGHTEXPLICIT) of BROWSER))
then ; Compute item heights. Don't do this if user gave an explicit
; height.
[replace (TABLEBROWSER TBITEMHEIGHT) of BROWSER with (SETQ HEIGHT (TIMES HEIGHT
(fetch (TABLEBROWSER
TB#LINESPERITEM)
of BROWSER]
[replace (TABLEBROWSER TBFONTASCENT) of BROWSER with (SETQ ASCENT (FONTPROP FONT 'ASCENT)
(replace (TABLEBROWSER TBBASELINE) of BROWSER with (- HEIGHT ASCENT))
else (SETQ HEIGHT (fetch (TABLEBROWSER TBITEMHEIGHT) of BROWSER))
[replace (TABLEBROWSER TBORIGIN) of BROWSER with (SETQ ORIGIN (fetch (REGION PTOP) of (DSPCLIPPINGREGION
NIL WINDOW)
(SETQ TOTALHEIGHT (TIMES (fetch (TABLEBROWSER TB#ITEMS) of BROWSER)
HEIGHT))
(WINDOWPROP WINDOW 'EXTENT (replace (TABLEBROWSER TBEXTENT) of BROWSER
with (create REGION
LEFT _ 0
BOTTOM _ (- ORIGIN TOTALHEIGHT)
WIDTH _ 0
HEIGHT _ TOTALHEIGHT)))
; Let extent width be zero until we print something
(replace (TABLEBROWSER TBMAXXPOS) of BROWSER with 0)
[if (SETQ HW (fetch (TABLEBROWSER TBHEADINGWINDOW) of BROWSER))
then ; Fix extent of header window, too. Be sure to account for
; different size of borders, if any
(LET [(HWIDTH (TIMES 2 (- (WINDOWPROP WINDOW 'BORDER)
(WINDOWPROP HW 'BORDER)
(if (SETQ EXTENT (WINDOWPROP HW 'EXTENT))
then (replace (REGION WIDTH) of EXTENT with HWIDTH)
else (WINDOWPROP HW 'EXTENT
(create REGION
LEFT _ 0
BOTTOM _ 0
WIDTH _ HWIDTH
HEIGHT _ -1]
(COND
((AND FONTGIVEN (SETQ FN (fetch (TABLEBROWSER TBFONTCHANGEFN) of BROWSER)))
; Notify application program of font change
(CL:FUNCALL FN BROWSER WINDOW])

```

(TB.SHRINKFN

```

[LAMBDA (WINDOW)
(TB.CLOSE/SHRINK WINDOW 'SHRINK)]

```

(* bvm%: " 6-Sep-85 12:14")

(TB.EXPANDFN

```

[LAMBDA (WINDOW)

```

; Edited 27-Jan-88 16:53 by bvm

;;; If browser changed while it was shrunk, update display accordingly

```

(LET [(BROWSER (WINDOWPROP WINDOW 'TABLEBROWSER)
(WITH.MONITOR (fetch (TABLEBROWSER TBLOCK) of BROWSER)
[LET ((FIRSTCHANGEDITEM# (fetch (TABLEBROWSER TBUPDATEFROMHERE) of BROWSER))
REGION FN ; Restore SHRINKFN prop if necessary
(WINDOWADDPROP WINDOW 'SHRINKFN (FUNCTION TB.SHRINKFN)
T)
(COND
(FIRSTCHANGEDITEM# ; Browser has changed since shrinking
(TB.DISPLAY.LINES BROWSER [IMAX FIRSTCHANGEDITEM# (TB.FIRST.VISIBLE.ITEM#
BROWSER
(SETQ REGION (DSPCLIPPINGREGION
NIL WINDOW)
(TB.LAST.VISIBLE.ITEM# BROWSER REGION))
(replace (TABLEBROWSER TBUPDATEFROMHERE) of BROWSER with NIL]]))

```

(TB.FIND.PREVIOUS.TAIL

```

[LAMBDA (BROWSER ITEM#)

```

; Edited 20-Jan-88 23:23 by bvm

;; Return the tail of BROWSER's items whose CADR is ITEM#. Assumes ITEM# at least 2 and not greater than number of items

```

(LET (TAIL TAILN)
(if [OR (NULL (SETQ TAIL (fetch (TABLEBROWSER TBTAILHINT) of BROWSER))
(< ITEM# (SETQ TAILN (ADD1 (fetch (TABLEITEM TI#) of (CAR TAIL]
then ; Can't use the hint
(SETQ TAIL (fetch (TABLEBROWSER TBITEMS) of BROWSER))
(SETQ TAILN 2))

```



```

;; TAILN is the number of (CADR TAIL). Want to get TAIL pointing to one before the requested number
(while (< TAILN ITEM#) do (SETQ TAIL (CDR TAIL))
      (add TAILN 1))
(if (OR (NULL TAIL)
      (NEQ TAILN ITEM#))
    then (HELP "Failed to find item tail" ITEM#))
TAIL])

```

(TB.RENUMBER.TAIL

```

[LAMBDA (BROWSER TAIL FIRST#) ; Edited 20-Jan-88 23:22 by bvm

```

;; Renumbers all of BROWSER's items from TAIL onward, giving (CAR TAIL) the number FIRST#. Also updates tail hint.

```

(for ITEM in TAIL as I from FIRST# do (replace TI# of ITEM with I))
(replace (TABLEBROWSER TBTAILHINT) of BROWSER with TAIL])

```

)

;; Misc

```

(DEFINEQ

```

(TB.PROCESS

```

[LAMBDA (FORM NAME ALLOWLOGOUT RESTARTABLE) (* bvm%: "25-Mar-84 17:16")

```

;;; Creates a process running FORM which by default is not restartable and will not permit LOGOUT while it is running

```

(ADD.PROCESS FORM 'NAME NAME 'RESTARTABLE (OR RESTARTABLE 'NO)
 'BEFOREEXIT
 (COND
  (ALLOWLOGOUT NIL)
  (T 'DON'T)))

```

)

```

(RPAQ? TB.DELETEDLINEHEIGHT 1)

```

```

(RPAQQ TB.SELECTION.BITMAP )

```

```

(RPAQ TB.CROSSCURSOR (CURSORCREATE 'X
 'NIL 8 8))

```

```

(DECLARE%: EVAL@COMPILE DONTCOPY

```

```

(DECLARE%: EVAL@COMPILE

```

```

(RPAQQ TB.LEFT.MARGIN 8)

```

```

(CONSTANTS TB.LEFT.MARGIN)

```

)

```

(RPAQQ TOCSTATES ((TS.IDLE 0)
 (TS.REPLACING 1)
 (TS.ADDING 2)
 (TS.REMOVING 3)
 (TS.EXTENDING.HI 4)
 (TS.EXTENDING.LO 5)
 (TS.SHRINKING.HI 6)
 (TS.SHRINKING.LO 7)))

```

```

(DECLARE%: EVAL@COMPILE

```

```

(RPAQQ TS.IDLE 0)

```

```

(RPAQQ TS.REPLACING 1)

```

```

(RPAQQ TS.ADDING 2)

```

```

(RPAQQ TS.REMOVING 3)

```

```

(RPAQQ TS.EXTENDING.HI 4)

```

```

(RPAQQ TS.EXTENDING.LO 5)

```

```

(RPAQQ TS.SHRINKING.HI 6)

```

```

(RPAQQ TS.SHRINKING.LO 7)

```

```

(CONSTANTS (TS.IDLE 0)
 (TS.REPLACING 1)
 (TS.ADDING 2)
 (TS.REMOVING 3)
 (TS.EXTENDING.HI 4)
 (TS.EXTENDING.LO 5)

```

```

(TS.SHRINKING.HI 6)
(TS.SHRINKING.LO 7))
)
(DECLARE%: EVAL@COMPILE
(PUTPROPS .COPYKEYDOWNP. MACRO [NIL (OR (KEYDOWNP 'LSHIFT)
                                         (KEYDOWNP 'RSHIFT)
                                         (KEYDOWNP 'COPY))]
)
(DECLARE%: DOEVAL@COMPILE DONTCOPY
(GLOBALVARS TB.CROSSCURSOR TB.SELECTION.BITMAP TB.DELETEDLINEHEIGHT)
)
(DECLARE%: DOEVAL@COMPILE DONTCOPY
(LOCALVARS . T)
)
(DECLARE%: EVAL@COMPILE
(DATATYPE TABLEBROWSER ((TBREADY FLAG)
                          (TBHEIGHTEXPLICIT FLAG)
                          (TBITEMS POINTER)
                          (TB#ITEMS WORD)
                          (TB#DELETED WORD)
                          (TB#LINESPERITEM WORD)
                          (TBFIRSTSELECTEDITEM WORD)
                          (TBLASTSELECTEDITEM WORD)
                          (TBITEMHEIGHT WORD)
                          (TBMAXXPOS WORD)
                          (TBFONTHEIGHT WORD)
                          (TBFONTASCENT WORD)
                          (TBBASELINE WORD)
                          (TBWINDOW POINTER)
                          (TBLOCK POINTER)
                          (TBUSERDATA POINTER)
                          (TBFONT POINTER)
                          (TBEXTENT POINTER)
                          (TBUPDATEFROMHERE POINTER)
                          (TBCOLUMNS POINTER)
                          (TBPRINTFN POINTER)
                          (TBCOPYFN POINTER)
                          (TBFONTCHANGEFN POINTER)
                          (TBCLOSEFN POINTER)
                          (TBAFTERCLOSEFN POINTER)
                          (TBTITLEEVENTFN POINTER)
                          (TBLINETHICKNESS POINTER)
                          (TBORIGIN POINTER)
                          (TBTAILHINT POINTER)
                          (TBHEADINGWINDOW POINTER)
                          (NIL POINTER)))
                          ; True if creator set explicit item height or baseline
                          ; List of items in this browser
                          ; Number of items
                          ; Number of items marked deleted
                          ; Number of lines occupied by each item, normally 1 (dunno if
                          ; any other values work)
                          ; Number of first selected item. If none selected, is > TB#ITEMS
                          ; Number of last selected item. If none selected, is 0
                          ; Height of an item, i.e., fontheight*linesperitem
                          ; The largest x-position a user printfn has printed to
                          ; Height, ascent, descent of font
                          ; Pointer to the display window. Need to snap this link when
                          ; browser is closed
                          ; Monitor lock guarding some browser operations
                          ; Arbitrary user storage
                          ; Pointer to font used by display
                          ; Window's extent, updated as items are added, deleted, or
                          ; printfn prints farther to right
                          ; If changes have occurred while shrunk, this gives the # of first
                          ; item that needs redisplay
                          ; Number of columns--not yet implemented
                          ; (Browser Item Window) -- displays Item at current line position
                          ; in window
                          ; (Browser Item) -- copy selects Item
                          ; (Browser Window) -- called when tb.set.font changes the font
                          ; (Browser Window Close/Shrink) -- called when you try to close
                          ; or shrink window
                          ; (Browser Window) -- called to cleanup AFTER a closew
                          ; (Window Browser) -- handles button event in browser's title
                          ; Thickness of line for deletions (normally 1)
                          ; Y position of the top of the first item
                          ; A tail of TBITEMS, used to speed up TB.NTH.ITEM
                          ; An optional "header window" that should be horizontally
                          ; scrolled in synchrony with this one

(DATATYPE TABLEITEM ((TISELECTED FLAG)
                      (TIDELETED FLAG)
                      (TIUNDELETABLE FLAG)
                      (TIUNSELECTABLE FLAG)
                      (TIUNCOPYSELECTABLE FLAG)
                      (TIDATA POINTER)
                      (TI# WORD)))
)
(/DECLAREDATATYPE 'TABLEBROWSER
' (FLAG FLAG POINTER WORD WORD WORD WORD WORD WORD WORD WORD WORD POINTER POINTER POINTER POINTER
  POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER
  POINTER POINTER)
;; ---field descriptor list elided by lister---
' 48)
(/DECLAREDATATYPE 'TABLEITEM ' (FLAG FLAG FLAG FLAG FLAG POINTER WORD)
;; ---field descriptor list elided by lister---
' 4)
)
(DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVARS

```

(ADDTOVAR **NLAMA**)

(ADDTOVAR **NLAML**)

(ADDTOVAR **LAMA** TB.USERDATA)
)

(/DECLAREDATATYPE 'TABLEBROWSER
' (FLAG FLAG POINTER WORD WORD WORD WORD WORD WORD WORD WORD WORD POINTER POINTER POINTER POINTER
POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER
POINTER POINTER)
;; ---field descriptor list elided by lister---
' 48)

(/DECLAREDATATYPE 'TABLEITEM ' (FLAG FLAG FLAG FLAG FLAG POINTER WORD)
;; ---field descriptor list elided by lister---
' 4)

(ADDTOVAR **SYSTEMRECLST**
(DATATYPE TABLEBROWSER ((TBREADY FLAG)
(TBHEIGHTEXPLICIT FLAG)
(TBITEMS POINTER)
(TB#ITEMS WORD)
(TB#DELETED WORD)
(TB#LINESPERITEM WORD)
(TBFIRSTSELECTEDITEM WORD)
(TBLASTSELECTEDITEM WORD)
(TBITEMHEIGHT WORD)
(TBMAXXPOS WORD)
(TBFONTHEIGHT WORD)
(TBFONTASCENT WORD)
(TBBASELINE WORD)
(TBWINDOW POINTER)
(TBLOCK POINTER)
(TBUSERDATA POINTER)
(TBFONT POINTER)
(TBEXTENT POINTER)
(TBUPDATEFROMHERE POINTER)
(TBCOLUMNS POINTER)
(TBPRINTFN POINTER)
(TBCOPYFN POINTER)
(TBFONTCHANGEFN POINTER)
(TBCLOSEFN POINTER)
(TBAFTERCLOSEFN POINTER)
(TBTITLEEVENTFN POINTER)
(TBLINETHICKNESS POINTER)
(TBORIGIN POINTER)
(TBTAILHINT POINTER)
(TBHEADINGWINDOW POINTER)
(NIL POINTER)))
(DATATYPE TABLEITEM ((TISELECTED FLAG)
(TIDELETED FLAG)
(TIUNDELETABLE FLAG)
(TIUNSELECTABLE FLAG)
(TIUNCOPYSELECTABLE FLAG)
(TIDATA POINTER)
(TI# WORD))))

(PUTPROPS **TABLEBROWSER COPYRIGHT** ("Venue & Xerox Corporation" 1985 1986 1987 1988 1990 1993 1994 1995 1999
2018 2021 2022))

FUNCTION INDEX

TB.BOTTOM.OF.ITEM	9	TB.FIRST.VISIBLE.ITEM#	8	TB.RENUMBER.TAIL	17
TB.BROWSER.BUSY	15	TB.FLUSH.WINDOW	15	TB.REPAINTFN	7
TB.BUTTONEVENTFN	10	TB.INSERT.ITEM	3	TB.REPLACE.ITEMS	2
TB.CLEAR.LINE	7	TB.ITEM.DELETED?	6	TB.RESHAPEFN	7
TB.CLOSE/SHRINK	15	TB.ITEM.FROM.YCOORD	8	TB.REV.FIND.SELECTED.ITEM	14
TB.CLOSEFN	15	TB.ITEM.SELECTED?	6	TB.SCROLLFN	7
TB.COLLECT.ITEMS	5	TB.ITEM.UPDATABLE?	10	TB.SELECT.ITEM	5
TB.CONSIDERRANGE	13	TB.ITEM.VISIBLE?	8	TB.SELECTRANGE	14
TB.CONTIGUOUS.SELP	13	TB.LAST.VISIBLE.ITEM#	8	TB.SET.FONT	15
TB.COPYBUTTONEVENTFN	14	TB.MAKE.BROWSER	1	TB.SHOW.COPY.SELECTION	15
TB.DECONSIDERRANGE	13	TB.MAP.DELETED.ITEMS	6	TB.SHOW.DELETION	9
TB.DELETE.ITEM	2	TB.MAP.ITEMS	6	TB.SHOW.SELECTION	9
TB.DESELECTRANGE	13	TB.MAP.SELECTED.ITEMS	6	TB.SHRINKFN	16
TB.DISPLAY.LINES	7	TB.NORMALIZE.ITEM	4	TB.UNDELETE.ITEM	3
TB.DO.ITEM.SELECTION	10	TB.NTH.ITEM	5	TB.UNDOSELECTION	14
TB.DO.UNLESS.BUSY	10	TB.NUMBER.OF.ITEMS	5	TB.UNSELECT.ALL.ITEMS	5
TB.EXPANDFN	16	TB.PRINT.LINE	8	TB.UNSELECT.ITEM	5
TB.FIND.ITEM	6	TB.PROCESS	17	TB.UPDATE.DISPLAY	9
TB.FIND.PREVIOUS.TAIL	16	TB.RECONSIDERRANGE	13	TB.USERDATA	7
TB.FIND.SELECTED.ITEM	14	TB.REDISPLAY.ITEMS	4	TB.WINDOW	7
TB.FINISH.CLOSE	15	TB.REMOVE.ITEM	3		

CONSTANT INDEX

TB.LEFT.MARGIN	17	TS.EXTENDING.HI	17	TS.IDLE	17	TS.REPLACING	17	TS.SHRINKING.LO	17
TS.ADDING	17	TS.EXTENDING.LO	17	TS.REMOVING	17	TS.SHRINKING.HI	17		

VARIABLE INDEX

SYSTEMRECLST	19	TB.DELETEDLINEHEIGHT	17	TOCSTATES	17
TB.CROSSCURSOR	17	TB.SELECTION.BITMAP	17		

RECORD INDEX

TABLEBROWSER	18	TABLEITEM	18
--------------	----	-----------	----

MACRO INDEX

.COPYKEYDOWNP.	18
----------------	----