

File created: 19-Feb-2025 13:31:28 {WMEDLEY}<library>tedit>TEDIT-STYLES.;4

edit by: rmk

changes to: (FNS \TEDIT.APPLY.STYLES \TEDIT.APPLY.PARASTYLES)

previous date: 12-Feb-2025 12:18:37 {WMEDLEY}<library>tedit>TEDIT-STYLES.;3

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

(RPAQQ TEDIT-STYLES COMS

```
(
; Style-sheet support
(FNS \TEDIT.APPLY.STYLES \TEDIT.APPLY.PARASTYLES TEDIT.STYLESHEET TEDIT.POP.STYLESHEET
TEDIT.PUSH.STYLESHEET TEDIT.ADD.STYLESHEET)
;; *TEDIT-PARASTYLE-CACHE* is an ALIST of original char/para looks to styled char/para looks. It is used to cache stylings, and is reset
;; when the main stylesheet changes, and when we change paragraph looks, given paras that have private char styles.
;; *TEDIT-CURRENTPARA-CACHE* is NIL if we're not in a para that has private char styles, or is the PARALOOKS (styled!) for that para, if
;; we are. Used to decide when we have to flush *TEDIT-PARASTYLE-CACHE* at paragraph boundaries. Mostly, this'll be NIL and not
;; interesting.
;; *TEDIT-STYLESHEET-SAVE-LIST* is a list of points inside TEDIT.STYLES, so we can "push" new style sheets on the front, and "pop"
;; them off sensibly. This is the push-stack, in effect. Used by TEDIT.ADD.STYLESHEET, TEDIT.PUSH.STYLESHEET, and
;; TEDIT.POP.STYLESHEET
(INITVARS (TEDIT.STYLES))
;; RMK 2023: Maybe this should be one of the later ones? Only partly implemented
(GLOBALVARS TEDIT.STYLES)
(INITVARS (*TEDIT-PARASTYLE-CACHE*
(*TEDIT-CURRENTPARA-CACHE*)
(*TEDIT-STYLESHEET-SAVE-LIST*)))
(GLOBALVARS *TEDIT-PARASTYLE-CACHE* *TEDIT-CURRENTPARA-CACHE* *TEDIT-STYLESHEET-SAVE-LIST*))
```

;; Style-sheet support

(DEFINEQ

(\TEDIT.APPLY.STYLES

```
[LAMBDA (LOOKS PC TSTREAM)
; Edited 19-Feb-2025 13:31 by rmk
; Edited 8-Feb-2025 21:07 by rmk
; Edited 12-Nov-2023 16:08 by rmk
; Edited 18-Mar-2023 21:45 by rmk
; Edited 25-Sep-2022 13:28 by rmk
; Edited 11-Sep-2022 14:45 by rmk
; Edited 4-Jul-93 01:02 by sybalsky:MV:ENVOS
```

;; Given a set of looks, return the looks with the proper styles expanded out.

```
(SETQ TSTREAM (TEXTSTREAM TSTREAM))
(OR (CDR (ASSOC LOOKS *TEDIT-CURRENTPARA-CACHE*))
(CDR (ASSOC LOOKS *TEDIT-PARASTYLE-CACHE*)))
(LET* ((TEXTOBJ (TEXTOBJ TSTREAM))
(STYLE (GETLOOKS LOOKS CLSTYLE))
(STYLE-SHEET (OR (FGETTOBJ TEXTOBJ TXTSTYLESHEET)
TEDIT.STYLES))
NOSTYLE CHARSTYLES CHARSTYLE IN-PARA)
(SETQ STYLE (COND
((NULL STYLE) ; STYLE of NIL means don't bother. Just use the looks we got.
(SETQ NOSTYLE T)
LOOKS)
((AND (SETQ CHARSTYLES (AND (GETTSTR TSTREAM CURRENTPARALOOKS)
(GETPLOOKS (GETTSTR TSTREAM CURRENTPARALOOKS)
FMTCHARSTYLES)))
(SETQ CHARSTYLE (FASSOC STYLE CHARSTYLES)))
; If the paragraph we're in has character styles, and this is one of
; them, use it.
(SETQ IN-PARA T)
CHARSTYLE)
(CDR (SASSOC STYLE STYLE-SHEET))))
((AND (LITATOM STYLE)
(DEFINEDP STYLE)) ; Call the guy's function to find the new looks
(APPLY* STYLE LOOKS PC TEXTOBJ))
; If all else fails, return the original set of looks
(T
(SETQ NOSTYLE T)
LOOKS)))
(SETQ STYLE (COND
((LISTP STYLE)
(\TEDIT.PARSE.CHARLOOKS.LIST (APPEND STYLE '(STYLE NIL))
LOOKS TEXTOBJ))
(T STYLE)))
```

;; Cache the looks->styled-looks mapping, either in the cache for this kind of paragraph (which gets wiped when we hit a new para
;; type), or in the global cache.

[OR NOSTYLE (CL:IF IN-PARA

```

(push *TEDIT-CURRENTPARA-CACHE* (CONS LOOKS STYLE))
(push *TEDIT-PARASTYLE-CACHE* (CONS LOOKS STYLE)))
STYLE])

```

(\TEDIT.APPLY.PARASTYLES

```

[LAMBDA (PARALOOKS PC TEXTOBJ) ; Edited 19-Feb-2025 13:31 by rmk
; Edited 8-Feb-2025 22:07 by rmk
; Edited 4-Aug-2024 14:48 by rmk
; Edited 29-Apr-2024 11:06 by rmk
; Edited 4-Mar-2023 22:23 by rmk
; Edited 25-Sep-2022 13:26 by rmk
; Edited 3-Jul-93 23:15 by sybalskY:MV:ENVOS

```

:: Given a set of looks, return the looks with the proper styles expanded out.

```

(\TEDIT.CHECK (type? PARALOOKS PARALOOKS)) ; Incoming thing has to be a LOOKS.
(OR (CDR (ASSOC PARALOOKS *TEDIT-PARASTYLE-CACHE*))
(LET* [NOSTYLE (STYLE-SHEET (OR (fetch (TEXTOBJ TXTSTYLESHEET) of TEXTOBJ)
TEDIT.STYLES))
(STYLE (COND
((NULL (GETPLOOKS PARALOOKS FMTSTYLE))
(SETQ NOSTYLE T)
PARALOOKS)
((CDR (SASSOC (GETPLOOKS PARALOOKS FMTSTYLE)
STYLE-SHEET)))
((AND (LITATOM (GETPLOOKS PARALOOKS FMTSTYLE))
(DEFINEDP (GETPLOOKS PARALOOKS FMTSTYLE)))
; Call the guy's function to find the new looks
(APPLY* (GETPLOOKS PARALOOKS FMTSTYLE)
PARALOOKS PC TEXTOBJ)
(T (SETQ NOSTYLE T)
PARALOOKS]
(CL:WHEN (LISTP STYLE)
(SETQ STYLE (\TEDIT.PARSE.PARALOOKS.LIST (APPEND STYLE '(STYLE NIL))
PARALOOKS TEXTOBJ)))
(CL:UNLESS NOSTYLE
(push *TEDIT-PARASTYLE-CACHE* (CONS PARALOOKS STYLE)))
STYLE])

```

(\TEDIT.STYLESHEET

```

[LAMBDA (SHEET TEXTSTREAM) ; Edited 3-Jul-93 23:19 by sybalskY:MV:ENVOS

```

:: Put a new stylesheet into force. This REPLACES any existing style sheets, and forgets any pushed sheets.

```

(LET [(TEXTOBJ (AND TEXTSTREAM (TEXTOBJ TEXTSTREAM)
(COND
(TEXTOBJ (SETQ *TEDIT-PARASTYLE-CACHE* NIL) ; Clear the cache, to force reformatting
(replace (TEXTOBJ TXTSTYLESHEET) of TEXTOBJ with SHEET))
(T ; No specific document given; change the global style sheet TEDIT.STYLES
(SETQ *TEDIT-PARASTYLE-CACHE* NIL) ; Clear the cache, to force reformatting
(SETQ TEDIT.STYLES SHEET)
(SETQ *TEDIT-STYLESHEET-SAVE-LIST* (LIST TEDIT.STYLES))

```

(\TEDIT.POP.STYLESHEET

```

[LAMBDA NIL ; Edited 3-Jul-93 17:42 by sybalskY:MV:ENVOS

```

:: Go back to an earlier stylesheet, by popping the stack of saved sheets. You can't pop back to no sheet -- you'll always bottom out at the original style sheet.

```

(SETQ *TEDIT-PARASTYLE-CACHE* NIL) ; Clear the cache, to force reformatting
(SETQ TEDIT.STYLES (OR (CL:POP *TEDIT-STYLESHEET-SAVE-LIST*)
TEDIT.STYLES])

```

(\TEDIT.PUSH.STYLESHEET

```

[LAMBDA (SHEET) ; Edited 3-Jul-93 17:40 by sybalskY:MV:ENVOS

```

:: Add more style definitions to the current style sheet, and remember how to get back to the old one. Think of this as PUSHING onto a stack of stylesheets, with the new sheet being a composition of SHEET and the existing styles.

```

(SETQ *TEDIT-PARASTYLE-CACHE* NIL) ; Clear the cache, to force reformatting
(SETQ TEDIT.STYLES (APPEND SHEET TEDIT.STYLES))
(CL:PUSH TEDIT.STYLES *TEDIT-STYLESHEET-SAVE-LIST*)

```

(\TEDIT.ADD.STYLESHEET

```

[LAMBDA (SHEET) ; Edited 3-Jul-93 17:38 by sybalskY:MV:ENVOS

```

:: Add more style definitions to the current style sheet. This ADDS entries, without remembering that there was an earlier sheet.

```

(SETQ *TEDIT-PARASTYLE-CACHE* NIL) ; Clear the cache, to force reformatting
(SETQ TEDIT.STYLES (APPEND SHEET TEDIT.STYLES))
(SETQ *TEDIT-STYLESHEET-SAVE-LIST* (LIST TEDIT.STYLES])

```

)

:: *TEDIT-PARASTYLE-CACHE* is an ALIST of original char/para looks to styled char/para looks. It is used to cache stylings, and is reset when the main stylesheet changes, and when we change paragraph looks, given paras that have private char styles.

:: *TEDIT-CURRENTPARA-CACHE* is NIL if we're not in a para that has private char styles, or is the PARALOOKS (styled!) for that para, if we are.
:: Used to decide when we have to flush *TEDIT-PARASTYLE-CACHE* at paragraph boundaries. Mostly, this'll be NIL and not interesting.
:: *TEDIT-STYLE SHEET-SAVE-LIST* is a list of points inside TEDIT.STYLES, so we can "push" new style sheets on the front, and "pop" them off
:: sensibly. This is the push-stack, in effect. Used by TEDIT.ADD.STYLESHEET, TEDIT.PUSH.STYLESHEET, and TEDIT.POP.STYLESHEET

(RPAQ? **TEDIT.STYLES**)

:: RMK 2023: Maybe this should be one of the later ones? Only partly implemented

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS TEDIT.STYLES)

)

(RPAQ? ***TEDIT-PARASTYLE-CACHE***)

(RPAQ? ***TEDIT-CURRENTPARA-CACHE***)

(RPAQ? ***TEDIT-STYLE SHEET-SAVE-LIST***)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS ***TEDIT-PARASTYLE-CACHE*** ***TEDIT-CURRENTPARA-CACHE*** ***TEDIT-STYLE SHEET-SAVE-LIST***)

)

FUNCTION INDEX

TEDIT.ADD.STYLESHEET	2	TEDIT.PUSH.STYLESHEET	2	\TEDIT.APPLY.PARASTYLES	2
TEDIT.POP.STYLESHEET	2	TEDIT.STYLESHEET	2	\TEDIT.APPLY.STYLES	1

VARIABLE INDEX

TEDIT-CURRENTPARA-CACHE	3	*TEDIT-STYLE-SHEET-SAVE-LIST*	3
TEDIT-PARASTYLE-CACHE	3	TEDIT.STYLES	3
