File created: 19-Jan-93 11:21:28 **{DSK}<python>lde>lispcore>sources>SPPDECLS.;4**

changes to: (RECORDS SPPCON SPPHEAD SPPXIP SPPSTREAM)

previous date: 5-Jan-93 02:32:12 {DSK}<python>lde>lispcore>sources>SPPDECLS.;3

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS


```
(RPAQQ SPPDECLSCOMS
       ((FILES (SOURCE)
               LLNSDECLS)
        (RECORDS SPPCON SPPHEAD SPPXIP)
        (CONSTANTS * SPPTYPES)
        (CONSTANTS * SPPSTATES)
        (CONSTANTS (\SPPHEAD.LENGTH 12)
               (\#WDS.SPPINFO (SUB1 (FOLDLO (IPLUS \XIPOVLEN \SPPHEAD.LENGTH)
                                            BYTESPERWORD)))
               (\SPP.INITIAL.ALLOCATION 5)
               (\SPP.INITIAL.ROUNDTRIP 1000)
               (\SPP.RETRANSMITQ.SIZE 8))
        (RECORDS SPPSTREAM)
        (MACROS GETSPPCON \FETCH.NSADDRESS \SPPINCFILEPTR GETWORD PUTWORD GETLONG PUTLONG SPP.STREAM.ERROR)
        (CONSTANTS * SPPEOFFLAGS)
        (GLOBALVARS \SPPDEVICE \SPP.BULKDATA.DEVICE)))

(FILESLOAD (SOURCE)
       LLNSDECLS)

(DECLARE%: EVAL@COMPILE

(DATATYPE SPPCON (;; First part of this record looks like the header of an SPP XIP filled in with defaults for this connection
                  (SPPXIPLENGTH WORD)
                  (NIL BYTE)                                    ; Transport control
                  (SPPXIPTYPE BYTE)                             ; Constant \XIPT.SPP
                  (SPPDESTNSADDRESS0 5 WORD)                    ; Destination address, maybe not filled in until connection
                                                                ; established
                  (SPPDESTSKT# WORD)
                  (SPPSOURCENSADDRESS0 5 WORD)                  ; My address and socket number
                  (SPPSOURCESKT# WORD)
                  (NIL BYTE)                                    ; Connection Control
                  (SPPDSTYPE BYTE)                              ; Current datastream type from our outgoing side.
                  (SPPSOURCEID WORD)                            ; Connection identification number for this side.
                  (SPPDESTID WORD)                              ; Connection identification number for the other side.
                  (SPPSEQNO WORD)                               ; Current sequence number -- next packet to go out will take this
                                                                ; and, if not a system packet, then increment it.
                  (SPPACKNO WORD)                               ; We've seen all seqno's up to but not including this one.
                  (SPPACCEPTNO WORD)                            ; The Allocation number we've sent -- I'll accept his sequence
                                                                ; numbers up to and including this.
                  ;; Remainder of record contains other interesting state not a part of the packet
                  (SPPESTABLISHEDP FLAG)                        ; True when connection is established.
                  (SPPDESTINATIONKNOWN FLAG)                    ; True if we initiate the connection, or once a passive connection
                                                                ; is established
                  (SPPTERMINATEDP FLAG)                         ; True when \TERMINATESPP wants this one to go away.
                  (SPPOUTPUTABORTEDP FLAG)                      ; Attempt to send output instead invokes the
                                                                ; SPPOUTPUTABORTEDFN -- typically used to handle Bulk
                                                                ; Data abort
                  (SPPOUTPUTABORTEDFN POINTER)
                  (SPPACKPENDING FLAG)                          ; True if we have been requested to send an Ack
                  (SPPEOMONFORCEOUT FLAG)                       ; True if we want each FORCEOUTPUT to cause an EOM on the
                                                                ; stream
                  (SPPSERVERFLAG FLAG)                          ; True if connection was opened as a server
                  (SPPINPUTBLOCKED FLAG)                        ; True if we have received packets filling our allocation, so that
                                                                ; further input is blocked until we consume some
                  (SPPINPUTQ POINTER)                           ; Packets that have arrived wait in this queue. The packets are
                                                                ; in order but some may be missing.
                  (SPPRETRANSMITQ POINTER)                      ; Packets which have been to SENDXIP but have not yet been
                                                                ; acknowledged.
                  (SPPRETRANSMITTING POINTER)                   ; Queue of packets that we get back from the driver after
                                                                ; transmission. These have to be merged into the retransmit
                                                                ; queue.
                  (SPPLOCK POINTER)                             ; Monitor lock for connection.
                  (SPPMYNSOCKET POINTER)                        ; NS socket for sending and receiving XIPs.
                  (SPPACKEDSEQNO WORD)                          ; The most recent Acknowledge number we have received; i.e.
                                                                ; the SEQNO he expects to receive next.
                  (SPPOUTPUTALLOCNO WORD)                       ; The most recent Allocation number we've received.
                  (SPPRETRANSMITTIMER POINTER)                  ; Time at which the next Acknowledgement request or
                                                                ; retransmission should occur.
```

```
                (SPPACKREQUESTED POINTER)                    ; Will be set to a seqno when an ACK request has been sent but
                                                             ; not acknowledged.
                (SPPACKREQTIME POINTER)                      ; Whenever an ACK request is sent, this is set to the current
                                                             ; time.  When a response arrives, the round trip time is updated.
                (SPPACKREQTIMEOUT POINTER)                   ; Time at which an ACK request should be considered hopeless.
                (SPPROUNDTRIPTIME POINTER)                   ; Estimate of (twice) the round trip delay on this connection.
                (SPPACTIVITYTIMER POINTER)                   ; If non-NIL, the time for the next probe to see if the other end is
                                                             ; still there.
                (SPPATTENTIONFN POINTER)                     ; Fn to call when attention packet is received
                (SPPINPKT POINTER)                           ; Packet currently being read from, for BIN.
                (SPPOUTPKT POINTER)                          ; Packet currently being written to, for BOUT.
                (SPPSYSPKT POINTER)                          ; Cached System packet for probing and answering
                                                             ; Acknowledgement requests.
                (SPPINPUTSTREAM POINTER)                     ; Stream interface for this connection.
                (SPPSUBSTREAM POINTER)                       ; Bulk data substream for connection.
                (SPPPROCESS POINTER)                         ; Process managing this connection.
                (SPPALLOCATIONEVENT POINTER)                 ; Event which occurs when the allocation increases.
                (SPPINPUTEVENT POINTER)                      ; Event which occurs when the next data packet arrives.
                (SPPOUTPUTSTREAM POINTER)                    ; Stream for output side
                (SPPWHENCLOSEDFN POINTER)
                (SPPSTATE POINTER)
                (SPPERRORHANDLER POINTER)                    ; Fn to call when stream is in abnormal input state
                (SPPSERVERFN POINTER)                        ; Function to use as toplevel function for connections opened as
                                                             ; servers
                (SPPOTHERXIPHANDLER POINTER)                 ; Function to call when non-SPP, non-ERROR XIP received on
                                                             ; socket
                (SPPINACTIVECOUNT POINTER)
                (SPPINPUTDSTYPE BYTE)
                (SPPDSTYPECHANGEFN POINTER))
        [ACCESSFNS SPPCON ([SPPSOURCENSADDRESS (\FETCH.NSADDRESS (LOCF (fetch SPPSOURCENSADDRESS0 of DATUM]
                           [SPPDESTNSADDRESS (\FETCH.NSADDRESS (LOCF (fetch SPPDESTNSADDRESS0 of DATUM]
                           (SPPDESTNSNET (\GETBASEFIXP (LOCF (fetch SPPDESTNSADDRESS0 of DATUM))
                                                       0]
        SPPINPUTQ _ (create SYSQUEUE)
        SPPRETRANSMITQ _ (ARRAY \SPP.RETRANSMITQ.SIZE 'POINTER NIL 0)
        SPPALLOCATIONEVENT _ (CREATE.EVENT "SPP Allocation")
        SPPRETRANSMITTIMER _ (SETUPTIMER 0)
        SPPERRORHANDLER _ (FUNCTION \SPP.DEFAULT.ERRORHANDLER))

[BLOCKRECORD SPPHEAD ((CC BYTE)
                      (DSTYPE BYTE)
                      (SOURCECONID WORD)
                      (DESTCONID WORD)
                      (SEQNO WORD)
                      (ACKNO WORD)
                      (ALLOCNO WORD)
                      (FIRSTSPPDATABYTE BYTE)
                      (NIL BYTE))
        (BLOCKRECORD SPPHEAD ((SYSTEMPACKET FLAG)            ; Interpretation of Connection Control bits
                              (SENDACK FLAG)
                              (ATTENTION FLAG)
                              (ENDOFMESSAGE FLAG)
                              (NIL BITS 4)
                              (NIL BYTE)))
        (BLOCKRECORD SPPHEAD ((NIL FLAG)
                              (NIL FLAG)
                              (EOMBITS BITS 2)               ; End of message or Attention
                              (NIL BITS 4)
                              (NIL BYTE)))
        (ACCESSFNS SPPHEAD ((SPPCONTENTS (LOCF (fetch (SPPHEAD FIRSTSPPDATABYTE) of DATUM)))
                            (EOMP (NEQ 0 (fetch (SPPHEAD EOMBITS) of DATUM]

[ACCESSFNS SPPXIP ((SPPHEAD (fetch XIPCONTENTS of DATUM]
)

(/DECLAREDATATYPE 'SPPCON
        '(WORD BYTE BYTE WORD WORD WORD WORD WORD WORD WORD WORD WORD WORD WORD BYTE BYTE WORD WORD WORD
               WORD WORD FLAG FLAG FLAG FLAG POINTER FLAG FLAG FLAG FLAG POINTER POINTER POINTER POINTER POINTER
               WORD WORD POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER
               POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER BYTE
               POINTER)
        ;; ---field descriptor list elided by lister---
        '82)

(RPAQQ SPPTYPES (\SPPHEAD.CC.SYSTEM \SPPHEAD.CC.ACKNOWLEDGE \SPPHEAD.CC.ATTENTION \SPPHEAD.CC.EOM
                        \SPPDSTYPE.COURIER \SPPDSTYPE.BULKDATA \SPPDSTYPE.END \SPPDSTYPE.ENDREPLY))

(DECLARE%: EVAL@COMPILE

(RPAQQ \SPPHEAD.CC.SYSTEM 128)

(RPAQQ \SPPHEAD.CC.ACKNOWLEDGE 64)

(RPAQQ \SPPHEAD.CC.ATTENTION 32)

(RPAQQ \SPPHEAD.CC.EOM 16)
```

```
(RPAQQ \SPPDSTYPE.COURIER 0)

(RPAQQ \SPPDSTYPE.BULKDATA 1)

(RPAQQ \SPPDSTYPE.END 254)

(RPAQQ \SPPDSTYPE.ENDREPLY 255)

(CONSTANTS \SPPHEAD.CC.SYSTEM \SPPHEAD.CC.ACKNOWLEDGE \SPPHEAD.CC.ATTENTION \SPPHEAD.CC.EOM \SPPDSTYPE.COURIER
       \SPPDSTYPE.BULKDATA \SPPDSTYPE.END \SPPDSTYPE.ENDREPLY)
)

(RPAQQ SPPSTATES ((\SPS.INIT 0)
                  (\SPS.LISTENING 1)
                  (\SPS.OPEN 2)
                  (\SPS.ENDSENT 3)
                  (\SPS.ENDRECEIVED 4)
                  (\SPS.DALLYING 5)
                  (\SPS.CLOSED 6)
                  (\SPS.ABORTED 7)))

(DECLARE%: EVAL@COMPILE

(RPAQQ \SPS.INIT 0)

(RPAQQ \SPS.LISTENING 1)

(RPAQQ \SPS.OPEN 2)

(RPAQQ \SPS.ENDSENT 3)

(RPAQQ \SPS.ENDRECEIVED 4)

(RPAQQ \SPS.DALLYING 5)

(RPAQQ \SPS.CLOSED 6)

(RPAQQ \SPS.ABORTED 7)

(CONSTANTS (\SPS.INIT 0)
       (\SPS.LISTENING 1)
       (\SPS.OPEN 2)
       (\SPS.ENDSENT 3)
       (\SPS.ENDRECEIVED 4)
       (\SPS.DALLYING 5)
       (\SPS.CLOSED 6)
       (\SPS.ABORTED 7))
)

(DECLARE%: EVAL@COMPILE

(RPAQQ \SPPHEAD.LENGTH 12)

(RPAQ \#WDS.SPPINFO (SUB1 (FOLDLO (IPLUS \XIPOVLEN \SPPHEAD.LENGTH)
                                  BYTESPERWORD)))

(RPAQQ \SPP.INITIAL.ALLOCATION 5)

(RPAQQ \SPP.INITIAL.ROUNDTRIP 1000)

(RPAQQ \SPP.RETRANSMITQ.SIZE 8)

(CONSTANTS (\SPPHEAD.LENGTH 12)
       (\#WDS.SPPINFO (SUB1 (FOLDLO (IPLUS \XIPOVLEN \SPPHEAD.LENGTH)
                                    BYTESPERWORD)))
       (\SPP.INITIAL.ALLOCATION 5)
       (\SPP.INITIAL.ROUNDTRIP 1000)
       (\SPP.RETRANSMITQ.SIZE 8))
)

(DECLARE%: EVAL@COMPILE

[ACCESSFNS SPPSTREAM ((SPP.CONNECTION (fetch F1 of DATUM)
                              (replace F1 of DATUM with NEWVALUE))
                      (BULK.DATA.CONTINUATION (fetch F2 of DATUM)
                              (replace F2 of DATUM with NEWVALUE))
                      (SPPEOFBITS (fetch FW8 of DATUM)
                              (replace FW8 of DATUM with NEWVALUE))
                      (SPPFILEPTRHI (fetch FW6 of DATUM)
                              (replace FW6 of DATUM with NEWVALUE))
                      (SPPFILEPTRLO (fetch FW7 of DATUM)
                              (replace FW7 of DATUM with NEWVALUE)))
       (ACCESSFNS SPPSTREAM ([SPPEOFP (SELECTC (fetch SPPEOFBITS of DATUM)
                                              (0 NIL)
                                              (\SPPFLAG.END 'END)
                                              (\SPPFLAG.ATTENTION
```

```
                                        'ATTENTION)
                                    (\SPPFLAG.EOM 'EOM)
                                    NIL)
                                (replace SPPEOFBITS of DATUM with (SELECTQ NEWVALUE
                                                                          (NIL 0)
                                                                          (EOM \SPPFLAG.EOM)
                                                                          (END \SPPFLAG.END)
                                                                          (ATTENTION \SPPFLAG.ATTENTION)
                                                                          (\ILLEGAL.ARG NEWVALUE]
                            (SPPFILEPTR (\MAKENUMBER (fetch SPPFILEPTRHI of DATUM)
                                                     (fetch SPPFILEPTRLO of DATUM]
)

(DECLARE%: EVAL@COMPILE

(PUTPROPS GETSPPCON MACRO ((X)
                           (fetch SPP.CONNECTION of X)))

(PUTPROPS \FETCH.NSADDRESS MACRO ((BASE)
                                  (PROG ((ADDRESS (create NSADDRESS)))
                                        (\BLT ADDRESS BASE \#WDS.NSADDRESS)
                                        (RETURN ADDRESS))))

(PUTPROPS \SPPINCFILEPTR MACRO [OPENLAMBDA (STREAM NBYTES)
                                (COND
                                   ((ILESSP (replace SPPFILEPTRLO of STREAM
                                               with (\LOLOC (\ADDBASE (fetch SPPFILEPTRLO of STREAM)
                                                                      NBYTES)))
                                            NBYTES)
                                    (add (fetch SPPFILEPTRHI of STREAM)
                                         1])

(PUTPROPS GETWORD MACRO (= . \WIN))

(PUTPROPS PUTWORD MACRO (= . \WOUT))

(PUTPROPS GETLONG MACRO (OPENLAMBDA (STREAM)
                          (\MAKENUMBER (\WIN STREAM)
                                 (\WIN STREAM))))

(PUTPROPS PUTLONG MACRO [OPENLAMBDA (STREAM FIXP)
                          (PROGN (\WOUT STREAM (\HINUM FIXP))
                                 (\WOUT STREAM (LOGAND FIXP 65535])

(PUTPROPS SPP.STREAM.ERROR MACRO (OPENLAMBDA (STREAM ERRCODE)
                                   (SPREADAPPLY* (fetch SPPERRORHANDLER of (GETSPPCON STREAM))
                                          STREAM ERRCODE)))
)

(RPAQQ SPPEOFFLAGS ((\SPPFLAG.EOM 1)
                    (\SPPFLAG.END 2)
                    (\SPPFLAG.ATTENTION 3)))

(DECLARE%: EVAL@COMPILE

(RPAQQ \SPPFLAG.EOM 1)

(RPAQQ \SPPFLAG.END 2)

(RPAQQ \SPPFLAG.ATTENTION 3)

(CONSTANTS (\SPPFLAG.EOM 1)
       (\SPPFLAG.END 2)
       (\SPPFLAG.ATTENTION 3))
)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS \SPPDEVICE \SPP.BULKDATA.DEVICE)
)

(PUTPROPS SPPDECLS COPYRIGHT ("Venue & Xerox Corporation" 1986 1987 1990 1992 1993))
```

CONSTANT INDEX

MACRO INDEX

RECORD INDEX

VARIABLE INDEX