File created: 27-Feb-91 19:20:13 **{DSK}<medley>project2>lispcore>sources>CMLSMARTARGS.;2**

changes to: (VARS *CL-ARGINFO-LIST*)

previous date: 15-Jun-90 15:24:56 {DSK}<medley>project2>lispcore>sources>CMLSMARTARGS.;1

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

;;
;; Copyright (c) 1986, 1987, 1988, 1989, 1990, 1991 by Venue & Xerox Corporation. All rights reserved.

```
(RPAQQ CMLSMARTARGSCOMS
       ((VARS *CL-ARGINFO-LIST* *XCL-ARGINFO-LIST*)
        (FUNCTIONS ARGINFO-MUNG CLSMARTEN)
        [DECLARE%: DONTEVAL@LOAD DOCOPY (P (CLSMARTEN *CL-ARGINFO-LIST*)
                                           (CLSMARTEN *XCL-ARGINFO-LIST*)
                                           (SETQ *CL-ARGINFO-LIST* (SETQ *XCL-ARGINFO-LIST* 'NOBIND]
        (PROP FILETYPE CMLSMARTARGS)))

(RPAQQ *CL-ARGINFO-LIST*
       (((CL:* +)
         &REST NUMBERS)
        ((- / CL:/= < <= = > >= MAX MIN)
         CL:NUMBER &REST MORE-NUMBERS)
        ((CL:1+ CL:1- ABS CL:ACOS CL:ACOSH CL:ASIN CL:ASINH CL:ATANH CL:CONJUGATE CL:COSH CL:EXP CL:IMAGPART
                MINUSP CL:PHASE CL:PLUSP CL:RATIONAL CL:RATIONALIZE CL:REALPART CL:SIGNUM CL:SINH CL:SQRT
                CL:TANH CL:ZEROP)
         CL:NUMBER)
        (CL:ACONS KEY DATUM A-LIST)
        ((CL:ADJOIN CL:MEMBER)
         ITEM LIST &KEY :TEST :TEST-NOT :KEY)
        (CL:ADJUST-ARRAY CL:ARRAY NEW-DIMENSIONS &KEY :ELEMENT-TYPE :INITIAL-ELEMENT :INITIAL-CONTENTS
               :FILL-POINTER :DISPLACED-TO :DISPLACED-INDEX-OFFSET :FATP :DISPLACED-TO-BASE)
        ((CL:ADJUSTABLE-ARRAY-P CL:ARRAY-DIMENSIONS CL:ARRAY-ELEMENT-TYPE CL:ARRAY-HAS-FILL-POINTER-P
                CL:ARRAY-RANK CL:ARRAY-TOTAL-SIZE)
         CL:ARRAY)
        ((CL:ALPHA-CHAR-P CL:ALPHANUMERICP CL:BOTH-CASE-P CL:CHAR-BITS CL:CHAR-CODE CL:CHAR-DOWNCASE
                CL:CHAR-FONT CL:CHAR-INT CL:CHAR-NAME CL:CHAR-UPCASE CL:GRAPHIC-CHAR-P CL:LOWER-CASE-P
                CL:STANDARD-CHAR-P CL:STRING-CHAR-P CL:UPPER-CASE-P)
         CL:CHAR)
        ((AND OR PROGN)
         (CURLYLIST FORM)
         #\*)
        ((CL:APPEND NCONC)
         &REST LISTS)
        (CL:APPLY CL:FUNCTION ARG &REST MORE-ARGS)
        (CL:APPLYHOOK CL:FUNCTION ARGS EVALHOOKFN APPLYHOOKFN &OPTIONAL ENV)
        ((CL:APROPOS CL:APROPOS-LIST)
         STRING &OPTIONAL PACKAGE)
        ((CL:AREF CL:ARRAY-IN-BOUNDS-P CL:ARRAY-ROW-MAJOR-INDEX)
         CL:ARRAY &REST SUBSCRIPTS)
        (CL:ARRAY-DIMENSION CL:ARRAY AXIS-NUMBER)
        ((CL:ARRAYP CL:ATOM CL:BIT-VECTOR-P CL:CHARACTER CL:CHARACTERP CL:COMMONP CL:COMPILED-FUNCTION-P
                CL:COMPLEXP CL:CONSP CL:CONSTANTP CL:COPY-TREE CL:DESCRIBE CL:ENDP CL:FLOATP CL:FUNCTIONP
                CL:HASH-TABLE-P CL:IDENTITY INSPECT CL:INTEGERP CL:KEYWORDP CL:LISTP NULL CL:NUMBERP CL:PACKAGEP
                CL:PATHNAMEP CL:PRIN1-TO-STRING CL:PRINC-TO-STRING CL:RANDOM-STATE-P CL:RATIONALP READTABLEP
                CL:SIMPLE-BIT-VECTOR-P CL:SIMPLE-STRING-P CL:SIMPLE-VECTOR-P STREAMP CL:STRINGP CL:SXHASH
                CL:SYMBOLP CL:TYPE-OF CL:VECTORP)
         OBJECT)
        (CL:ASH INTEGER CL:COUNT)
        [CL:ASSERT TEST-FORM (SQUARELIST ((CURLYLIST* PLACE))
                                   (SQUARELIST STRING (CURLYLIST* ARG]
        ((CL:ASSOC CL:RASSOC)
         ITEM A-LIST &KEY :TEST :TEST-NOT :KEY)
        ((CL:ASSOC-IF CL:ASSOC-IF-NOT CL:RASSOC-IF CL:RASSOC-IF-NOT)
         PREDICATE A-LIST)
        (CL:ATAN Y &OPTIONAL X)
        (BIT BIT-ARRAY &REST SUBSCRIPTS)
        ((CL:BIT-AND CL:BIT-EQV CL:BIT-IOR CL:BIT-XOR)
         BIT-ARRAY1 BIT-ARRAY-2 &OPTIONAL RESULT-BIT-ARRAY)
        ((CL:BIT-ANDC1 CL:BIT-ANDC2 CL:BIT-NAND CL:BIT-NOR CL:BIT-ORC1 CL:BIT-ORC2)
         BIT-ARRAY1 BIT-ARRAY2 &OPTIONAL RESULT-BIT-ARRAY)
        (CL:BIT-NOT BIT-ARRAY &OPTIONAL RESULT-BIT-ARRAY)
        (CL:BLOCK NAME
            (CURLYLIST FORM)
            #\*)
        (CL:BOOLE OP INTEGER1 INTEGER2)
        ((BOUNDP CL:FBOUNDP CL:FMAKUNBOUND CL:MACRO-FUNCTION CL:MAKE-SYNONYM-STREAM CL:MAKUNBOUND
                CL:SPECIAL-FORM-P CL:SYMBOL-FUNCTION CL:SYMBOL-PLIST CL:SYMBOL-VALUE)
         CL:SYMBOL)
        (CL:BREAK &OPTIONAL FORMAT-STRING &REST ARGS)
        ((CL:BUTLAST CL:NBUTLAST)
```

```
 LIST &OPTIONAL N)
(BYTE SIZE CL:POSITION)
((CL:BYTE-POSITION BYTE-SIZE)
 BYTESPEC)
((CAAAAR CAAADR CAAAR CAADAR CAADDR CAADR CAAR CADAAR CADADR CADAR CADDAR CADDDR CADDR CADR CAR CDAAAR
        CDAADR CDAAR CDADAR CDADDR CDADR CDAR CDDAAR CDDADR CDDAR CDDDAR CDDDDR CDDDR CDDR CDR CL:EIGHTH
        CL:FIFTH CL:FIRST CL:FOURTH LAST CL:LIST-LENGTH CL:NINTH CL:REST CL:SECOND CL:SEVENTH CL:SIXTH
        CL:TENTH CL:THIRD)
 LIST)
[(CASE CL:ECASE)
 KEYFORM
 (CURLYLIST* ((CURLYLIST ((CURLYLIST* KEY))
                         #\| KEY)
              (CURLYLIST* FORM]
(CL:CATCH TAG
    (CURLYLIST FORM)
    #\*)
[CL:CCASE KEYPLACE (CURLYLIST* ((CURLYLIST ((CURLYLIST* KEY))
                                           #\| KEY)
                                (CURLYLIST* FORM]
((CL:CEILING CL:FCEILING CL:FFLOOR CL:FLOOR CL:FROUND CL:FTRUNCATE ROUND CL:TRUNCATE)
 CL:NUMBER &OPTIONAL DIVISOR)
(CL:CERROR CONTINUE-FORMAT-STRING ERROR-FORMAT-STRING &REST ARGS)
(CL:CHAR STRING INDEX)
(CL:CHAR-BIT CL:CHAR NAME)
((CL:CHAR-EQUAL CL:CHAR-GREATERP CL:CHAR-LESSP CL:CHAR-NOT-EQUAL CL:CHAR-NOT-GREATERP CL:CHAR-NOT-LESSP
        CL:CHAR/= CL:CHAR< CL:CHAR<= CL:CHAR= CL:CHAR> CL:CHAR>=)
 CL:CHARACTER &REST MORE-CHARACTERS)
(CL:CHECK-TYPE PLACE TYPESPEC &OPTIONAL STRING)
((CL:CIS CL:COS CL:SIN CL:TAN)
 RADIANS)
((CL:CLEAR-INPUT CL:LISTEN)
 &OPTIONAL INPUT-STREAM)
((CL:CLEAR-OUTPUT CL:FINISH-OUTPUT CL:FORCE-OUTPUT CL:FRESH-LINE CL:TERPRI)
 &OPTIONAL OUTPUT-STREAM)
(CL:CLOSE STREAM &KEY :ABORT)
((CLRHASH CL:HASH-TABLE-COUNT)
 CL:HASH-TABLE)
(CL:CODE-CHAR CODE &OPTIONAL BITS FONT)
(COERCE OBJECT RESULT-TYPE)
(CL:COMPILE NAME &OPTIONAL DEFINITION &KEY :LAP)
(CL:COMPILE-FILE INPUT-PATHNAME &KEY :OUTPUT-FILE :ERROR-FILE :ERRORS-TO-TERMINAL :LAP-FILE :LOAD
       :FILE-MANAGER-FORMAT :PROCESS-ENTIRE-FILE)
(CL:COMPILER-LET ((CURLYLIST VAR #\| (VAR VALUE))
                  #\*)
       (CURLYLIST FORM)
       #\*)
(COMPLEX CL:REALPART &OPTIONAL CL:IMAGPART)
(CL:CONCATENATE RESULT-TYPE &REST SEQUENCES)
(COND (CURLYLIST (TEST (CURLYLIST FORM)
                       #\*))
      #\*)
((CONS CL:NRECONC CL:REVAPPEND RPLACA RPLACD)
 X Y)
((CL:COPY-ALIST CL:COPY-LIST CL:VALUES-LIST)
 LIST)
(CL:COPY-READTABLE &OPTIONAL FROM-READTABLE TO-READTABLE)
((CL:COPY-SEQ CL:LENGTH CL:NREVERSE CL:REVERSE)
 SEQUENCE)
(CL:COPY-SYMBOL SYM &OPTIONAL COPY-PROPS)
((CL:COUNT CL:FIND CL:POSITION)
 ITEM SEQUENCE &KEY :FROM-END :TEST :TEST-NOT :START :END :KEY)
((CL:COUNT-IF CL:COUNT-IF-NOT CL:FIND-IF CL:FIND-IF-NOT CL:POSITION-IF CL:POSITION-IF-NOT)
 TEST SEQUENCE &KEY :FROM-END :START :END :KEY)
[CL:CTYPECASE KEYPLACE (CURLYLIST* (TYPE (CURLYLIST* FORM]
((CL:DECF CL:INCF)
 PLACE
 (SQUARELIST DELTA))
(DECLARE (CURLYLIST DECL-SPEC)
       #\*)
((CL:DECODE-FLOAT CL:FLOAT-DIGITS CL:FLOAT-PRECISION CL:FLOAT-RADIX CL:INTEGER-DECODE-FLOAT)
 FLOAT)
(CL:DECODE-UNIVERSAL-TIME UNIVERSAL-TIME &OPTIONAL TIME-ZONE)
((CL:DEFCONSTANT CL:DEFPARAMETER)
 NAME INITIAL-VALUE (SQUARELIST CL:DOCUMENTATION))
(CL:DEFINE-MODIFY-MACRO NAME LAMBDA-LIST CL:FUNCTION (SQUARELIST DOC-STRING))
(CL:DEFINE-SETF-METHOD ACCESS-FN LAMBDA-LIST (CURLYLIST CL:DECLARATION #\| DOC-STRING)
       #\*
       (CURLYLIST FORM)
       #\*)
((DEFMACRO CL:DEFTYPE CL:DEFUN)
 NAME LAMBDA-LIST (CURLYLIST* CL:DECLARATION #\| DOC-STRING)
 (CURLYLIST* FORM))
(CL:DEFSETF ACCESS-FN (CURLYLIST UPDATE-FN (SQUARELIST DOC-STRING)
                              #\| LAMBDA-LIST (STORE-VARIABLE)
                              (CURLYLIST CL:DECLARATION #\| DOC-STRING)
                              #\*
```

```
                                    (CURLYLIST FORM)
                                    #\*))
        (CL:DEFSTRUCT NAME-AND-OPTIONS (SQUARELIST DOC-STRING)
               (CURLYLIST SLOT-DESCRIPTION)
               #\+)
        (CL:DEFVAR NAME (SQUARELIST INITIAL-VALUE (SQUARELIST CL:DOCUMENTATION)))
        ((CL:DELETE CL:REMOVE)
         ITEM SEQUENCE &KEY :FROM-END :TEST :TEST-NOT :START :END :COUNT :KEY)
        ((CL:DELETE-DUPLICATES CL:REMOVE-DUPLICATES)
         SEQUENCE &KEY :FROM-END :TEST :TEST-NOT :START :END :KEY)
        ((CL:DELETE-FILE CL:FILE-AUTHOR CL:FILE-WRITE-DATE CL:PROBE-FILE)
         FILE)
        ((CL:DELETE-IF CL:DELETE-IF-NOT CL:REMOVE-IF CL:REMOVE-IF-NOT)
         TEST SEQUENCE &KEY :FROM-END :START :END :COUNT :KEY)
        ((CL:DENOMINATOR CL:NUMERATOR)
         CL:RATIONAL)
        ((CL:DEPOSIT-FIELD DPB)
         NEWBYTE BYTESPEC INTEGER)
        (CL:DIGIT-CHAR WEIGHT &OPTIONAL RADIX FONT)
        (CL:DIGIT-CHAR-P CL:CHAR &OPTIONAL RADIX)
        ((CL:DIRECTORY CL:DIRECTORY-NAMESTRING CL:FILE-NAMESTRING CL:HOST-NAMESTRING CL:NAMESTRING PATHNAME
                CL:PATHNAME-DEVICE CL:PATHNAME-DIRECTORY CL:PATHNAME-HOST CL:PATHNAME-NAME CL:PATHNAME-TYPE
                CL:PATHNAME-VERSION CL:TRUENAME)
         PATHNAME)
        (CL:DISASSEMBLE NAME-OR-COMPILED-FUNCTION)
        ((CL:DO CL:DO*)
         [(CURLYLIST* (VAR (SQUARELIST INIT (SQUARELIST CL:STEP]
         (END-TEST (CURLYLIST* RESULT))
         (CURLYLIST* CL:DECLARATION)
         (CURLYLIST* TAG #\| STATEMENT))
        (CL:DO-ALL-SYMBOLS (VAR (SQUARELIST RESULT-FORM))
               (CURLYLIST CL:DECLARATION)
               #\*
               (CURLYLIST TAG #\| STATEMENT)
               #\*)
        ((CL:DO-EXTERNAL-SYMBOLS CL:DO-SYMBOLS)
         (VAR (SQUARELIST PACKAGE (SQUARELIST RESULT-FORM)))
         (CURLYLIST CL:DECLARATION)
         #\*
         (CURLYLIST TAG #\| STATEMENT)
         #\*)
        (CL:DOCUMENTATION CL:SYMBOL DOC-TYPE)
        (CL:DOLIST (VAR LISTFORM (SQUARELIST RESULTFORM))
            (CURLYLIST CL:DECLARATION)
            #\*
            (CURLYLIST TAG #\| STATEMENT)
            #\*)
        (CL:DOTIMES (VAR COUNTFORM (SQUARELIST RESULTFORM))
            (CURLYLIST CL:DECLARATION)
            #\*
            (CURLYLIST TAG #\| STATEMENT)
            #\*)
        (DRIBBLE &OPTIONAL PATHNAME)
        (ED &OPTIONAL NAME OPTIONS #\= ((CURLYLIST "FILEPKGTYPE" #\| :DISPLAY #\| :NEW)
                                        #\*))
        (CL:ELT SEQUENCE INDEX)
        (CL:ENCODE-UNIVERSAL-TIME CL:SECOND MINUTE HOUR DATE MONTH YEAR &OPTIONAL TIME-ZONE)
        (CL:ENOUGH-NAMESTRING PATHNAME &OPTIONAL DEFAULTS)
        ((EQ EQL CL:EQUAL CL:EQUALP)
         X Y)
        ((CL:ERROR CL:WARN)
         FORMAT-STRING &REST ARGS)
        ((CL:ETYPECASE CL:TYPECASE)
         KEYFORM
         (CURLYLIST (TYPE (CURLYLIST FORM)
                          #\*))
         #\*)
        ((CL:EVAL CL:GET-SETF-METHOD CL:GET-SETF-METHOD-MULTIPLE-VALUE)
         FORM)
        (CL:EVAL-WHEN ((CURLYLIST SITUATION)
                       #\*)
               (CURLYLIST FORM)
               #\*)
        (CL:EVALHOOK FORM EVALHOOKFN APPLYHOOKFN &OPTIONAL ENV)
        ((EVENP CL:INT-CHAR CL:INTEGER-LENGTH CL:ISQRT CL:LOGCOUNT LOGNOT ODDP)
         INTEGER)
        ((CL:EVERY CL:NOTANY CL:NOTEVERY CL:SOME)
         PREDICATE SEQUENCE &REST MORE-SEQUENCES)
        ((EXPORT IMPORT CL:SHADOW CL:SHADOWING-IMPORT CL:UNEXPORT)
         SYMBOLS &OPTIONAL PACKAGE)
        (CL:EXPT BASE-NUMBER POWER-NUMBER)
        (CL:FILE-LENGTH FILE-STREAM)
        (CL:FILE-POSITION FILE-STREAM &OPTIONAL CL:POSITION)
        (CL:FILL SEQUENCE ITEM &KEY :START :END)
        ((CL:FILL-POINTER CL:VECTOR-POP)
         CL:VECTOR)
        (CL:FIND-ALL-SYMBOLS STRING-OR-SYMBOL)
```

```
        ((CL:FIND-PACKAGE CL:NAME-CHAR)
         NAME)
        ((CL:FIND-SYMBOL CL:INTERN)
         STRING &OPTIONAL PACKAGE)
        ((CL:FLET CL:LABELS)
         ((CURLYLIST (NAME LAMBDA-LIST (CURLYLIST CL:DECLARATION #\| DOC-STRING)
                           #\*
                           (CURLYLIST FORM)
                           #\*))
          #\*)
         (CURLYLIST FORM)
         #\*)
        (FLOAT CL:NUMBER &OPTIONAL OTHER)
        (CL:FLOAT-SIGN FLOAT1 &OPTIONAL FLOAT2)
        (CL:FORMAT DESTINATION CONTROL-STRING &REST ARGUMENTS)
        (CL:FUNCALL FN &REST ARGUMENTS)
        #'FN
        ((CL:GCD LOGAND CL:LOGEQV CL:LOGIOR LOGXOR)
         &REST INTEGERS)
        (CL:GENSYM &OPTIONAL X)
        (CL:GENTEMP &OPTIONAL PREFIX PACKAGE)
        (GET CL:SYMBOL INDICATOR &OPTIONAL DEFAULT)
        ((CL:GET-DECODED-TIME CL:GET-INTERNAL-REAL-TIME CL:GET-INTERNAL-RUN-TIME CL:GET-UNIVERSAL-TIME
                CL:LISP-IMPLEMENTATION-TYPE CL:LISP-IMPLEMENTATION-VERSION CL:LIST-ALL-PACKAGES
                CL:LONG-SITE-NAME CL:MACHINE-INSTANCE CL:MACHINE-TYPE CL:MACHINE-VERSION
                CL:MAKE-STRING-OUTPUT-STREAM CL:SHORT-SITE-NAME CL:SOFTWARE-TYPE CL:SOFTWARE-VERSION))
        (CL:GET-DISPATCH-MACRO-CHARACTER DISP-CHAR SUB-CHAR &OPTIONAL CL:READTABLE)
        (CL:GET-MACRO-CHARACTER CL:CHAR &OPTIONAL CL:READTABLE)
        (CL:GET-OUTPUT-STREAM-STRING STRING-OUTPUT-STREAM)
        (CL:GET-PROPERTIES PLACE INDICATOR-LIST)
        (CL:GETF PLACE INDICATOR &OPTIONAL DEFAULT)
        (CL:GETHASH KEY CL:HASH-TABLE &OPTIONAL DEFAULT)
        (GO TAG)
        (CL:IF TEST
            THEN
            (SQUARELIST ELSE))
        (CL:IN-PACKAGE CL:PACKAGE-NAME &KEY :NICKNAMES :USE)
        ((CL:INPUT-STREAM-P CL:OUTPUT-STREAM-P CL:STREAM-ELEMENT-TYPE)
         STREAM)
        ((CL:INTERSECTION CL:NINTERSECTION CL:NSET-DIFFERENCE CL:NSET-EXCLUSIVE-OR CL:NUNION CL:SET-DIFFERENCE
                CL:SET-EXCLUSIVE-OR CL:SUBSETP CL:UNION)
         LIST1 LIST2 &KEY :TEST :TEST-NOT :KEY)
        (CL:LCM INTEGER &REST MORE-INTEGERS)
        ((LDB CL:LDB-TEST CL:MASK-FIELD)
         BYTESPEC INTEGER)
        (CL:LDIFF LIST SUBLIST)
        ((LET LET*)
         ((CURLYLIST VAR #\| (VAR VALUE))
          #\*)
         (CURLYLIST CL:DECLARATION)
         #\*
         (CURLYLIST FORM)
         #\*)
        ((LIST CL:VALUES)
         &REST ARGS)
        (LIST* ARG &REST OTHERS)
        (CL:LOAD FILENAME &KEY :VERBOSE :PRINT :IF-DOES-NOT-EXIST :PACKAGE :LOADFLG)
        (CL:LOCALLY (CURLYLIST CL:DECLARATION)
               #\*
               (CURLYLIST FORM)
               #\*)
        (CL:LOG CL:NUMBER &OPTIONAL BASE)
        ((CL:LOGANDC1 CL:LOGANDC2 CL:LOGNAND CL:LOGNOR CL:LOGORC1 CL:LOGORC2 CL:LOGTEST)
         INTEGER1 INTEGER2)
        (CL:LOGBITP INDEX INTEGER)
        (CL:LOOP (CURLYLIST FORM)
               #\*)
        ((CL:MACROEXPAND CL:MACROEXPAND-1)
         FORM &OPTIONAL ENV)
        (CL:MACROLET ((CURLYLIST (NAME VARLIST (CURLYLIST CL:DECLARATION #\| DOC-STRING)
                                       #\*
                                       (CURLYLIST FORM)
                                       #\*))
                      #\*)
               (CURLYLIST FORM)
               #\*)
        (CL:MAKE-ARRAY DIMENSIONS &KEY :ELEMENT-TYPE :INITIAL-ELEMENT :INITIAL-CONTENTS :ADJUSTABLE
               :FILL-POINTER :DISPLACED-TO :DISPLACED-INDEX-OFFSET :FATP :EXTENDABLE :READ-ONLY-P
               :DISPLACED-TO-BASE)
        ((CL:MAKE-BROADCAST-STREAM CL:MAKE-CONCATENATED-STREAM)
         &REST STREAMS)
        (CL:MAKE-CHAR CL:CHAR &OPTIONAL BITS FONT)
        (CL:MAKE-DISPATCH-MACRO-CHARACTER CL:CHAR &OPTIONAL NON-TERMINATING-P CL:READTABLE)
        ((CL:MAKE-ECHO-STREAM CL:MAKE-TWO-WAY-STREAM)
         INPUT-STREAM OUTPUT-STREAM)
        (CL:MAKE-HASH-TABLE &KEY :TEST :SIZE :REHASH-SIZE :REHASH-THRESHOLD)
        ((CL:MAKE-LIST CL:MAKE-STRING)
```

```
 SIZE &KEY :INITIAL-ELEMENT)
(CL:MAKE-PACKAGE CL:PACKAGE-NAME &KEY :NICKNAMES :USE :PREFIX-NAME :INTERNAL-SYMBOLS :EXTERNAL-SYMBOLS
       :EXTERNAL-ONLY)
(CL:MAKE-PATHNAME &KEY :HOST :DEVICE :DIRECTORY :NAME :TYPE :VERSION :DEFAULTS)
(CL:MAKE-RANDOM-STATE &OPTIONAL STATE)
(CL:MAKE-SEQUENCE TYPE SIZE &KEY :INITIAL-ELEMENT)
(CL:MAKE-STRING-INPUT-STREAM STRING &OPTIONAL START END)
(CL:MAKE-SYMBOL PRINT-NAME)
(MAKE-VECTOR CL:LENGTH &OPTIONAL TYPE INITIAL-VALUE)
(CL:MAP RESULT-TYPE CL:FUNCTION SEQUENCE &REST MORE-SEQUENCES)
((CL:MAPC CL:MAPCAN CL:MAPCAR CL:MAPCON CL:MAPL CL:MAPLIST)
 CL:FUNCTION LIST &REST MORE-LISTS)
(CL:MAPHASH CL:FUNCTION CL:HASH-TABLE)
((CL:MEMBER-IF CL:MEMBER-IF-NOT)
 PREDICATE LIST &KEY :KEY)
(CL:MERGE RESULT-TYPE SEQUENCE1 SEQUENCE2 PREDICATE &KEY :KEY)
(CL:MERGE-PATHNAMES PATHNAME &OPTIONAL DEFAULTS DEFAULT-VERSION)
((CL:MISMATCH CL:SEARCH)
 SEQUENCE1 SEQUENCE2 &KEY :FROM-END :TEST :TEST-NOT :KEY :START1 :START2 :END1 :END2)
((CL:MOD CL:REM)
 CL:NUMBER DIVISOR)
(CL:MULTIPLE-VALUE-BIND ((CURLYLIST VAR)
                         #\*)
    VALUES-FORM
  (CURLYLIST CL:DECLARATION)
  #\*
  (CURLYLIST FORM)
  #\*)
(CL:MULTIPLE-VALUE-CALL CL:FUNCTION (CURLYLIST FORM)
       #\*)
((CL:MULTIPLE-VALUE-LIST CL:STEP)
 FORM)
(CL:MULTIPLE-VALUE-PROG1 FORM (CURLYLIST FORM)
       #\*)
(CL:MULTIPLE-VALUE-SETQ VARIABLES FORM)
((NOT STRING)
 X)
((CL:NSTRING-CAPITALIZE CL:NSTRING-DOWNCASE CL:NSTRING-UPCASE CL:STRING-CAPITALIZE CL:STRING-DOWNCASE
        CL:STRING-UPCASE)
 STRING &KEY :START :END)
((CL:NSUBLIS CL:SUBLIS)
 ALIST TREE &KEY :TEST :TEST-NOT :KEY)
((CL:NSUBST CL:SUBST)
 NEW OLD TREE &KEY :TEST :TEST-NOT :KEY)
((CL:NSUBST-IF CL:NSUBST-IF-NOT CL:SUBST-IF CL:SUBST-IF-NOT)
 NEW TEST TREE &KEY :KEY)
((CL:NSUBSTITUTE CL:SUBSTITUTE)
 NEWITEM OLDITEM SEQUENCE &KEY :FROM-END :TEST :TEST-NOT :START :END :COUNT :KEY)
((CL:NSUBSTITUTE-IF CL:NSUBSTITUTE-IF-NOT CL:SUBSTITUTE-IF CL:SUBSTITUTE-IF-NOT)
 NEWITEM TEST SEQUENCE &KEY :FROM-END :START :END :COUNT :KEY)
((CL:NTH CL:NTHCDR)
 N LIST)
(OPEN FILENAME &KEY :DIRECTION :ELEMENT-TYPE :IF-EXISTS :IF-DOES-NOT-EXIST :EXTERNAL-FORMAT)
((CL:PACKAGE-NAME CL:PACKAGE-NICKNAMES CL:PACKAGE-SHADOWING-SYMBOLS CL:PACKAGE-USE-LIST
        CL:PACKAGE-USED-BY-LIST)
 PACKAGE)
(CL:PAIRLIS KEYS DATA &OPTIONAL A-LIST)
(CL:PARSE-INTEGER STRING &KEY :START :END :RADIX :JUNK-ALLOWED)
(CL:PARSE-NAMESTRING THING &OPTIONAL HOST DEFAULTS &KEY :START :END :JUNK-ALLOWED)
(CL:PEEK-CHAR &OPTIONAL PEEK-TYPE INPUT-STREAM EOF-ERROR-P EOF-VALUE RECURSIVE-P)
(CL:POP PLACE)
((CL:PPRINT CL:PRIN1 CL:PRINC CL:PRINT)
 OBJECT &OPTIONAL OUTPUT-STREAM)
(CL:PROCLAIM DECL-SPEC)
((PROG PROG*)
 ((CURLYLIST VAR #\| (VAR (SQUARELIST INIT)))
  #\*)
 (CURLYLIST CL:DECLARATION)
 #\*
 (CURLYLIST TAG #\| STATEMENT)
 #\*)
(PROG1 CL:FIRST
    (CURLYLIST FORM)
    #\*)
(PROG2 CL:FIRST CL:SECOND (CURLYLIST FORM)
       #\*)
(CL:PROGV SYMBOLS CL:VALUES
    (CURLYLIST FORM)
    #\*)
(CL:PROVIDE MODULE-NAME)
((CL:PSETF CL:SETF)
 (CURLYLIST PLACE NEWVALUE)
 #\*)
((CL:PSETQ CL:SETQ)
 (CURLYLIST VAR FORM)
 #\*)
(CL:PUSH ITEM PLACE)
```

```
        (CL:PUSHNEW ITEM LIST &KEY :TEST :TEST-NOT :KEY)
        'OBJECT
        (CL:RANDOM CL:NUMBER &OPTIONAL STATE)
        ((CL:READ CL:READ-CHAR CL:READ-CHAR-NO-HANG CL:READ-LINE)
         &OPTIONAL INPUT-STREAM EOF-ERROR-P EOF-VALUE RECURSIVE-P)
        (CL:READ-BYTE BINARY-INPUT-STREAM &OPTIONAL EOF-ERROR-P EOF-VALUE)
        (CL:READ-DELIMITED-LIST CL:CHAR &OPTIONAL INPUT-STREAM RECURSIVE-P)
        (CL:READ-FROM-STRING STRING &OPTIONAL EOF-ERROR-P EOF-VALUE &KEY :START :END :PRESERVE-WHITESPACE)
        (CL:READ-PRESERVING-WHITESPACE &OPTIONAL IN-STREAM EOF-ERROR-P EOF-VALUE RECURSIVE-P)
        (CL:REDUCE CL:FUNCTION SEQUENCE &KEY :FROM-END :START :END :INITIAL-VALUE)
        (CL:REMF PLACE INDICATOR)
        (REMHASH KEY CL:HASH-TABLE)
        (REMPROP CL:SYMBOL INDICATOR)
        (CL:RENAME-FILE FILE NEW-NAME)
        (CL:RENAME-PACKAGE PACKAGE NEW-NAME &OPTIONAL NEW-NICKNAMES)
        (CL:REPLACE SEQUENCE1 SEQUENCE2 &KEY :START1 :END1 :START2 :END2)
        (CL:REQUIRE MODULE-NAME &OPTIONAL PATHNAME)
        (RETURN (SQUARELIST RESULT))
        (CL:RETURN-FROM NAME (SQUARELIST RESULT))
        (CL:ROTATEF (CURLYLIST PLACE)
               #\*)
        (CL:SBIT SIMPLE-BIT-ARRAY &REST SUBSCRIPTS)
        (CL:SCALE-FLOAT FLOAT INTEGER)
        (CL:SCHAR CL:SIMPLE-STRING INDEX)
        (SET CL:SYMBOL VALUE)
        (CL:SET-CHAR-BIT CL:CHAR NAME NEWVALUE)
        (CL:SET-DISPATCH-MACRO-CHARACTER DISP-CHAR SUB-CHAR CL:FUNCTION &OPTIONAL CL:READTABLE)
        (CL:SET-MACRO-CHARACTER CL:CHAR CL:FUNCTION &OPTIONAL NON-TERMINATING-P CL:READTABLE)
        (CL:SET-SYNTAX-FROM-CHAR TO-CHAR FROM-CHAR &OPTIONAL TO-READTABLE FROM-READTABLE)
        (CL:SHIFTF (CURLYLIST PLACE)
               #\+ NEWVALUE)
        (CL:SLEEP SECONDS)
        ((CL:SORT CL:STABLE-SORT)
         SEQUENCE PREDICATE &KEY :KEY)
        (CL:STREAM-EXTERNAL-FORMAT STREAM)
        ((STRING-EQUAL CL:STRING-GREATERP CL:STRING-LESSP CL:STRING-NOT-EQUAL CL:STRING-NOT-GREATERP
                CL:STRING-NOT-LESSP CL:STRING/= CL:STRING< CL:STRING<= CL:STRING= CL:STRING> CL:STRING>=)
         STRING1 STRING2 &KEY :START1 :END1 :START2 :END2)
        ((CL:STRING-LEFT-TRIM CL:STRING-RIGHT-TRIM CL:STRING-TRIM)
         CHARACTER-BAG STRING)
        (CL:SUBSEQ SEQUENCE START &OPTIONAL END)
        (CL:SUBTYPEP TYPE1 TYPE2)
        (CL:SVREF CL:SIMPLE-VECTOR INDEX)
        ((CL:SYMBOL-NAME CL:SYMBOL-PACKAGE)
         SYM)
        (CL:TAGBODY (CURLYLIST TAG #\| STATEMENT)
               #\*)
        (TAILP SUBLIST LIST)
        (THE VALUE-TYPE FORM)
        (CL:THROW TAG RESULT)
        (TIME FORM &KEY :REPEAT :OUTPUT :DATA-TYPES)
        ((TRACE UNTRACE)
         (CURLYLIST FUNCTION-NAME)
         #\*)
        (CL:TREE-EQUAL X Y &KEY :TEST :TEST-NOT)
        (TYPEP OBJECT TYPE)
        (CL:UNINTERN CL:SYMBOL &OPTIONAL PACKAGE)
        ((CL:UNLESS CL:WHEN)
         TEST
         (CURLYLIST FORM)
         #\*)
        (CL:UNREAD-CHAR CL:CHARACTER &OPTIONAL INPUT-STREAM)
        (CL:UNUSE-PACKAGE PACKAGES-TO-UNUSE &OPTIONAL PACKAGE)
        (CL:UNWIND-PROTECT
            PROTECTED-FORM
            (CURLYLIST CLEANUP-FORM)
            #\*)
        (CL:USE-PACKAGE PACKAGES-TO-USE &OPTIONAL PACKAGE)
        (CL:USER-HOMEDIR-PATHNAME &OPTIONAL HOST)
        (CL:VECTOR &REST OBJECTS)
        (CL:VECTOR-PUSH NEW-ELEMENT CL:VECTOR)
        (CL:VECTOR-PUSH-EXTEND NEW-ELEMENT CL:VECTOR &OPTIONAL EXTENSION)
        (CL:WITH-INPUT-FROM-STRING (VAR STRING (CURLYLIST CL:KEYWORD VALUE)
                                           #\*)
               (CURLYLIST CL:DECLARATION)
               #\*
               (CURLYLIST FORM)
               #\*)
        (CL:WITH-OPEN-FILE (STREAM FILENAME (CURLYLIST OPTIONS)
                                   #\*)
               (CURLYLIST CL:DECLARATION)
               #\*
               (CURLYLIST FORM)
               #\*)
        (CL:WITH-OPEN-STREAM (VAR STREAM)
               (CURLYLIST CL:DECLARATION)
               #\*
```

```
              (CURLYLIST FORM)
              #\*)
       (CL:WITH-OUTPUT-TO-STRING (VAR (SQUARELIST STRING))
              (CURLYLIST CL:DECLARATION)
              #\*
              (CURLYLIST FORM)
              #\*)
       (WRITE OBJECT &KEY :STREAM :ESCAPE :RADIX :BASE :CIRCLE :PRETTY :LEVEL :LENGTH :CASE :GENSYM :ARRAY)
       (CL:WRITE-BYTE INTEGER BINARY-OUTPUT-STREAM)
       (CL:WRITE-CHAR CL:CHARACTER &OPTIONAL OUTPUT-STREAM)
       ((CL:WRITE-LINE CL:WRITE-STRING)
        STRING &OPTIONAL OUTPUT-STREAM &KEY :START :END)
       (CL:WRITE-TO-STRING OBJECT &KEY :ESCAPE :RADIX :BASE :CIRCLE :PRETTY :LEVEL :LENGTH :CASE :GENSYM :ARRAY
              )
       ((CL:Y-OR-N-P CL:YES-OR-NO-P)
        &OPTIONAL FORMAT-STRING &REST ARGUMENTS)))
```

(RPAQQ ***XCL-ARGINFO-LIST***

```
      ((ADD-EXEC &KEY :PROFILE :REGION :TTY :EXEC :ID)
       (ASET NEWVALUE ARRAY &REST INDICES)
       (CATCH-ABORT PRINT-FORM &BODY FORMS)
       (CONDITION-CASE FORM (CURLYLIST (TYPE ((SQUARELIST VAR))
                                             (CURLYLIST FORM)
                                             #\*))
              #\*)
       ((CONDITION-HANDLER CONDITION-REPORTER)
        TYPE)
       (COMPILER:COPY-ENV-WITH-FUNCTION ENVIRONMENT FUNCTION &OPTIONAL KIND EXP-FN)
       (COMPILER:COPY-ENV-WITH-VARIABLE ENVIRONMENT VARIABLE &OPTIONAL KIND)
       (DEBUG &OPTIONAL DATUM &REST ARGUMENTS)
       (DEF-DEFINE-TYPE NAME DESCRIPTION-STRING &KEY :UNDEFINER)
       (DEFAULT-PROCEED-TEST PROCEED-CASE-NAME)
       (DEFCOMMAND NAME ARGUMENT-LIST &REST BODY)
       (DEFDEFINER (CURLYLIST NAME #\| (NAME (CURLYLIST OPTION-CLAUSE)
                                             #\*))
              TYPE ARGLIST &BODY BODY)
       (DEFGLOBALPARAMETER NAME INITIAL-VALUE &OPTIONAL DOC-STRING)
       (DEFGLOBALVAR NAME &OPTIONAL INITIAL-VALUE DOC-STRING)
       (DEFINE-CONDITION NAME PARENT-TYPE SLOT-LIST (SQUARELIST KEYWORD VALUE)
              #\*)
       (DEFINE-PROCEED-FUNCTION NAME (SQUARELIST KEYWORD VALUE)
              #\* &REST VARIABLES)
       (DEFINLINE NAME ARG-LIST &BODY BODY)
       (DEFOPTIMIZER FORM-NAME (SQUARELIST OPT-NAME)
              (SQUARELIST ARG-LIST (SQUARELIST DECL #\| DOC-STRING)
                     #\*)
              BODY)
       (DEFPACKAGE NAME &REST OPTION-CLAUSES)
       (DESTRUCTURING-BIND BIND-PATTERN VALUE &BODY BODY)
       ((XCL:DO-INTERNAL-SYMBOLS DO-LOCAL-SYMBOLS)
        (VAR (SQUARELIST PACKAGE (SQUARELIST RESULT-FORM)))
        (CURLYLIST CL:DECLARATION)
        #\*
        (CURLYLIST TAG #\| STATEMENT)
        #\*)
       (EXEC &KEY :TOP-LEVEL-P :WINDOW :TITLE :COMMAND-TABLES :ENVIRONMENT :PROMPT :FUNCTION :PROFILE :ID)
       (EXEC-EVAL FORM &OPTIONAL ENVIRONMENT &KEY :PROMPT :ID :TYPE)
       (EXEC-FORMAT CONTROL-STRING &REST ARGUMENTS)
       ((EXTENDABLE-ARRAY-P READ-ONLY-ARRAY-P)
        ARRAY)
       (FILL-VECTOR VECTOR VALUE &KEY :START :END)
       (GLOBALIZE NAMESTRINGS &OPTIONAL PACKAGE)
       (HANDLER-BIND ((CURLYLIST (TYPE HANDLER))
                      #\*)
              (CURLYLIST FORM)
              #\*)
       (IGNORE-ERRORS &BODY FORMS)
       (INVOKE-PROCEED-CASE PROCEED-CASE &REST VALUES)
       (MAKE-CONDITION TYPE &REST SLOT-INITIALIZATIONS)
       (COMPILER:MAKE-CONTEXT &KEY :TOP-LEVEL-P :VALUES-USED :PREDICATE-P)
       (PARSE-BODY BODY ENVIRONMENT &OPTIONAL DOC-STRING-ALLOWED?)
       (PROCEED-CASE FORM (CURLYLIST (PROCEED-FUNCTION-NAME ARGLIST (SQUARELIST KEYWORD VALUE)
                                                            #\*
                                                            (CURLYLIST BODY-FORM)
                                                            #\*))
              #\*)
       ((XCL:SET-DEFAULT-EXEC-TYPE XCL:SET-EXEC-TYPE)
        NAME)
       (SIGNAL DATUM &REST ARGUMENTS)
       ((STORE-VALUE USE-VALUE)
        &OPTIONAL NEW-VALUE)
       (UNDOABLY (CURLYLIST FORMS))
       (UNDOABLY-SETF (CURLYLIST PLACE VALUE)
              #\*)))
```

(CL:DEFUN **ARGINFO-MUNG** (LST)

```
    ;; Flattens list elements of LST into a single top-level list of characters and words, recognizing special directives (SQUARELIST . things) and
    ;; (CURLYLIST . things) to mean turn it into [things] and {things}, respectively.

    [FOR THING IN LST JOIN (COND
                               [(CL:CONSP THING)
                                (CASE (CAR THING)
                                     (SQUARELIST (CONS #\[ (NCONC1 (ARGINFO-MUNG (CDR THING))
                                                                   #\])))
                                     (CURLYLIST (CONS #\{ (NCONC1 (ARGINFO-MUNG (CDR THING))
                                                                  #\})))
                                     (CURLYLIST* (CONS #\{ (NCONC (ARGINFO-MUNG (CDR THING))
                                                                  (LIST #\} #\*))))
                                     (CL:OTHERWISE (CONS #\( (NCONC1 (ARGINFO-MUNG THING)
                                                                     #\))))))]
                               (T (LIST THING])


(CL:DEFUN CLSMARTEN (FNLIST)
    ;; Transfer arg info from entries in FNLIST to the ARGNAMES props of those fns that need it.  Format of an entry in FNLIST is (Functions .
    ;; StylizedArgInfo), where Functions can be a symbol or list of symbols.

    [LET ((NOSPELLFLG T))                                          ; Tell SMARTARGLIST not to try too hard
         (DECLARE (CL:SPECIAL NOSPELLFLG))
         (CL:DOLIST (PAIR FNLIST)
             [LET (NEWARGS KNOWNARGS)
                  (CL:DOLIST [FN (OR (LISTP (CAR PAIR))
                                     (LIST (CAR PAIR]
                      (CL:UNLESS (AND [SETQ KNOWNARGS (NLSETQ (SMARTARGLIST FN (MEMB (ARGTYPE FN)
                                                                                     '(0 2]
                                      (CL:LISTP (SETQ KNOWNARGS (CAR KNOWNARGS)))
                                      (NOT (CL:MACRO-FUNCTION FN)))
                          ;; Only do this for fns for which SMARTARGLIST doesn't know the answer (something other than an atomic arglist)
                          ;; already.  Also ignore macros to override arglists provided by DEFMACRO.  The ARGTYPE check means try
                          ;; EXPLAINFLG=T in the case where the function is already defined as a lambda (don't want to do that for macros,
                          ;; since SMARTARGLIST would then fake something out of a macro/dmacro prop).  Format of ARGNAMES prop for
                          ;; this kind of guy is (NIL PrettyArgs . InterlispArgs).
                          (CL:SETF (GET FN 'ARGNAMES)
                                 (LIST* NIL [OR NEWARGS (SETQ NEWARGS (ARGINFO-MUNG (CDR PAIR]
                                        KNOWNARGS))))])])


(DECLARE%: DONTEVAL@LOAD DOCOPY

(CLSMARTEN *CL-ARGINFO-LIST*)

(CLSMARTEN *XCL-ARGINFO-LIST*)

(SETQ *CL-ARGINFO-LIST* (SETQ *XCL-ARGINFO-LIST* 'NOBIND))
)

(PUTPROPS CMLSMARTARGS FILETYPE :COMPILE-FILE)

(PUTPROPS CMLSMARTARGS COPYRIGHT ("Venue & Xerox Corporation" 1986 1987 1988 1989 1990 1991))
```

FUNCTION INDEX

VARIABLE INDEX

PROPERTY INDEX