

File created: 21-Mar-2024 10:21:14 {DSK}<home>larry>il>medley>sources>PACKAGE-STARTUP.;9

edit by: lmm

changes to: (VARIABLES CMLSYMBOLS.DECLARATORS CMLSYMBOLS.SHARED)

previous date: 20-Mar-2024 23:34:56 {DSK}<home>larry>il>medley>sources>PACKAGE-STARTUP.;8

Read Table: XCL

Package: INTERLISP

Format: XCCS

(RPAQQ **PACKAGE-STARTUPCOMS**

(

;;; Initialize the package system (LLPACKAGE must be loaded)

;; Simple definitions for the init. Improved in CMLPACKAGE

```
(FUNCTIONS RETURN-FIRST-OF-THREE ERROR-MISSING-EXTERNAL-SYMBOL)
(P (MOVD? 'ERROR-MISSING-EXTERNAL-SYMBOL 'RESOLVE-MISSING-EXTERNAL-SYMBOL)
  (MOVD? 'ERROR 'RESOLVE-MISSING-PACKAGE)
  (MOVD? 'ERROR 'RESOLVE-USE-PACKAGE-CONFLICT)
  (MOVD? 'ERROR 'RESOLVE-EXPORT-CONFLICT)
  (MOVD? 'ERROR 'RESOLVE-EXPORT-MISSING-CONFLICT)
  (MOVD? 'ERROR 'RESOLVE-IMPORT-CONFLICT)
  (MOVD? 'ERROR 'RESOLVE-UNINTERN-CONFLICT)
  (MOVD? 'RETURN-FIRST-OF-THREE 'RESOLVE-READER-CONFLICT)
  ; In pre-package init all symbols are prefixed, thus the
  ; INTERLISP symbol is always default
)
```

;; Reader changes

```
(FUNCTIONS CHECK-SYMBOL-NAMESTRING \\NEW.READ.SYMBOL \\NEW.MKATOM)
(VARIABLES LITATOM-PACKAGE-CONVERSION-ENABLED)
```

;; Initialization tables and functions

```
(VARIABLES CMLSYMBOLS.VARS CMLSYMBOLS.FNAMES CMLSYMBOLS.DECLARATORS CMLSYMBOLS.TYPENAMES
  CMLSYMBOLS.MACROS CMLSYMBOLS.SPECIALFORMS CMLSYMBOLS.LAMBDA.LIST.KEYWORDS)
(VARIABLES ; Be very careful with this.
  CMLSYMBOLS.SHARED)
(FUNCTIONS LITATOM.EXISTS)
(VARIABLES LITATOM-PACKAGE-CONVERSION-TABLE)
(FUNCTIONS NAMESTRING-CONVERSION-CLAUSE CONVERT-LITATOM CONCOCT-SYMBOL TRANSFER-SYMBOL INTERN-LITATOM
  \\LITATOM.EATCHARS)
(FUNCTIONS PACKAGE-INIT PACKAGE-CLEAR PACKAGE-MAKE PACKAGE-HIERARCHY-INIT PACKAGE-ENABLE PACKAGE-DISABLE
)
```

;; A hack for initialization

```
(FUNCTIONS ID)
(PROP (FILETYPE MAKEFILE-ENVIRONMENT)
  PACKAGE-STARTUP)
```

;; Initialize package system, plus functions needed in lpackage at init time

```
(DECLARE\ : DONTEVAL@LOAD DOCOPY (P (MOVD? 'EQ 'EQL)
  (MOVD? 'LENGTH 'CL:LENGTH)
  (MOVD? 'ID 'CL:IDENTITY)
  (MOVD? 'ID 'REMOVE-COMMENTS)
  (PACKAGE-INIT))) )
```

;;; Initialize the package system (LLPACKAGE must be loaded)

;; Simple definitions for the init. Improved in CMLPACKAGE

```
(CL:DEFUN RETURN-FIRST-OF-THREE (ONE TWO THREE)
  (DECLARE (IGNORE TWO THREE))
  ONE)
```

```
(CL:DEFUN ERROR-MISSING-EXTERNAL-SYMBOL (NAME PACKAGE)
  (ERROR (CONCAT "External symbol |" NAME "| not found in package " PACKAGE)))
```

(MOVD? 'ERROR-MISSING-EXTERNAL-SYMBOL 'RESOLVE-MISSING-EXTERNAL-SYMBOL)

(MOVD? 'ERROR 'RESOLVE-MISSING-PACKAGE)

(MOVD? 'ERROR 'RESOLVE-USE-PACKAGE-CONFLICT)

(MOVD? 'ERROR 'RESOLVE-EXPORT-CONFLICT)

(MOVD? 'ERROR 'RESOLVE-EXPORT-MISSING-CONFLICT)

(MOVD? 'ERROR 'RESOLVE-IMPORT-CONFLICT)

(MOVD? 'ERROR 'RESOLVE-UNINTERN-CONFLICT)

(MOVD? 'RETURN-FIRST-OF-THREE 'RESOLVE-READER-CONFLICT)

;; In pre-package init all symbols are prefixed, thus the INTERLISP symbol is always default

;; Reader changes

```

(CL:DEFUN CHECK-SYMBOL-NAMESTRING (BASE OFFSET LEN FATP)
  "Check whether a symbol would rather be in a package."
  (LET* ((CLAUSE (OR (NAMESTRING-CONVERSION-CLAUSE BASE OFFSET LEN FATP)
                     (CL:RETURN-FROM CHECK-SYMBOL-NAMESTRING NIL)))
        (PREFIX (CL:FIRST CLAUSE))
        (CL:PACKAGE-NAME (CL:THIRD CLAUSE))
        (WHERE (CL:FOURTH CLAUSE))
        (PREFIX-LENGTH (ffetch (STRINGP LENGTH)
                                PREFIX)))
    (COND
      (CL:PACKAGE-NAME (INTERN* BASE PREFIX-LENGTH (IDIFFERENCE LEN PREFIX-LENGTH)
                                FATP
                                (\\FATCHARSEENP BASE OFFSET LEN FATP)
                                (CL:FIND-PACKAGE CL:PACKAGE-NAME)
                                (EQ WHERE :EXTERNAL))))
      (T (UNINTERRUPTABLY
          (\\CREATE.SYMBOL BASE OFFSET LEN FATP (\\FATCHARSEENP BASE OFFSET LEN FATP))))))

(CL:DEFUN \\NEW.READ.SYMBOL (BASE OFFSET LEN FATP PACKAGE EXTERNALP NONNUMERICP)
  "Read a number or symbol from the string defined by BASE OFFSET LEN FATP PACKAGE is NIL if no package was
  specified, a package object or a string if an unknown package was typed (causes error). EXTERNALP is true if
  symbol was typed with one colon, which requires that the symbol exist and be external (unless it was a
  keyword). NONNUMERICP is true if we know the symbol is not a number, e.g., some characters in it were
  escaped."
  (DECLARE (CL:SPECIAL LITATOM-PACKAGE-CONVERSION-ENABLED *READTABLE* FILERDTBL CODERDTBL *PACKAGE*
              *LISP-PACKAGE* *INTERLISP-PACKAGE*))
  (OR (AND (NOT NONNUMERICP)
           (\\PARSE.NUMBER BASE OFFSET LEN FATP))
      (AND ;; The reader conversion feature is contained in this expression
          LITATOM-PACKAGE-CONVERSION-ENABLED
          (NULL PACKAGE)
          (OR (EQ *READTABLE* FILERDTBL)
              (EQ *READTABLE* CODERDTBL))
          (OR (CHECK-SYMBOL-NAMESTRING BASE OFFSET LEN FATP)
              (CL:MULTIPLE-VALUE-BIND (CLSYM CLSYMWHERE)
                (FIND-SYMBOL* BASE OFFSET LEN FATP *LISP-PACKAGE*)
                (LET ((ILSYM (FIND-SYMBOL* BASE OFFSET LEN FATP *INTERLISP-PACKAGE*)))
                  (COND
                    ((NULL ILSYM) ; No IL symbol, try CL
                     CLSYM)
                    ((NULL CLSYM) ; No CL symbol, use IL
                     ILSYM)
                    ((EQ ILSYM CLSYM) ; SAME
                     ILSYM)
                    (T ; Both symbols exist, resolve. During the INIT where packages
                     ; are turned off this is defined to return its first argument.
                     (RESOLVE-READER-CONFLICT ILSYM CLSYM CLSYMWHERE))))))
      (COND
        ((STRINGP PACKAGE)
         (RESOLVE-MISSING-PACKAGE PACKAGE (\\GETBASESTRING BASE OFFSET LEN FATP)
                                   EXTERNALP))
        ((OR (NOT EXTERNALP)
              (EQ PACKAGE *KEYWORD-PACKAGE*))
         (INTERN* BASE OFFSET LEN FATP (\\FATCHARSEENP BASE OFFSET LEN FATP)
                 (OR PACKAGE *PACKAGE*)
                 NIL))
        (T (CL:MULTIPLE-VALUE-BIND (CL:SYMBOL ACCESSIBLE)
            (FIND-SYMBOL* BASE OFFSET LEN FATP (OR PACKAGE *PACKAGE*))
            (COND
              ((EQ ACCESSIBLE :EXTERNAL)
               CL:SYMBOL)
              ((CL::%PACKAGE-EXTERNAL-ONLY PACKAGE) ; External only packages don't error creating external symbols on
               ; read
               (INTERN* BASE OFFSET LEN FATP (\\FATCHARSEENP BASE OFFSET LEN FATP)
                 (OR PACKAGE *PACKAGE*)
                 T))
              (T (RESOLVE-MISSING-EXTERNAL-SYMBOL (\\GETBASESTRING BASE OFFSET LEN FATP)
                PACKAGE))))))

(CL:DEFUN \\NEW.MKATOM (BASE OFFST LEN FATP)
  "A version of \\MKATOM which makes symbols in the Interlisp package instead of the old litatom table."
  (PROG ((FATCHARSEENP (\\FATCHARSEENP BASE OFFST LEN FATP))
        (FIRSTCHAR (UNLESSRDSYS (\\GETBASECHAR FATP BASE OFFST)
                                (NTHCHARCODE BASE OFFST)))
        TEMP)

```

```

(DECLARE (SPECVARS *INTERLISP-PACKAGE*))
(UNLESSRDSYS (COND
  ((AND (EQ LEN 1)
        (ILEQ FIRSTCHAR \\MAXTHINCHAR)
         |\\OneCharAtomBase|)
           ; The one-character atoms live in well known places, no need to
           ; hash
  (RETURN (COND
    ((IGREATERP FIRSTCHAR (CHARCODE "9"))
     (\\ADDBASE |\\OneCharAtomBase| (IDIFFERENCE FIRSTCHAR 10)))
    ((IGEQ FIRSTCHAR (CHARCODE "0"))
     ; These one-character atoms are integers
     (IDIFFERENCE FIRSTCHAR (CHARCODE "0"))
     (T (\\ADDBASE |\\OneCharAtomBase| FIRSTCHAR))))))
  ((AND (ILEQ FIRSTCHAR (CHARCODE "9"))
        (SETQ TEMP (\\PARSE.NUMBER BASE OFFST LEN FATP)))
   ;; \\PARSE.NUMBER returns a number or NIL
  (RETURN TEMP))))
(RETURN (CL:VALUES (INTERN* BASE OFFST LEN FATP FATCHARSEENP *INTERLISP-PACKAGE* T))))

```

```
(CL:DEFVAR LITATOM-PACKAGE-CONVERSION-ENABLED NIL)
```

```
;; Initialization tables and functions
```

```
(CL:DEFPARAMETER CMLSymbols.VARS
```

```

' ("*" "****" "APPLYHOOK*" "BREAK-ON-WARNINGS*" "DEBUG-IO*" "DEFAULT-PATHNAME-DEFAULTS*"
  "ERROR-OUTPUT*" "EVALHOOK*" "FEATURES*" "LOAD-VERBOSE*" "MACROEXPAND-HOOK*" "MODULES*"
  "PACKAGE*" "PRINT-ARRAY*" "PRINT-BASE*" "PRINT-CASE*" "PRINT-CIRCLE*" "PRINT-ESCAPE*"
  "PRINT-GENSYM*" "PRINT-LENGTH*" "PRINT-LEVEL*" "PRINT-PRETTY*" "PRINT-RADIX*" "QUERY-IO*"
  "RANDOM-STATE*" "READ-BASE*" "READ-DEFAULT-FLOAT-FORMAT*" "READ-SUPPRESS*" "READTABLE*"
  "STANDARD-INPUT*" "STANDARD-OUTPUT*" "TERMINAL-IO*" "TRACE-OUTPUT*" "+" "++" "+++" "-" "/" "///"
  "ARRAY-DIMENSION-LIMIT" "ARRAY-RANK-LIMIT" "ARRAY-TOTAL-SIZE-LIMIT" "BOOLE-1" "BOOLE-2"
  "BOOLE-AND" "BOOLE-ANDC1" "BOOLE-ANDC2" "BOOLE-C1" "BOOLE-C2" "BOOLE-CLR" "BOOLE-EQV" "BOOLE-IOR"
  "BOOLE-NAND" "BOOLE-NOR" "BOOLE-ORC1" "BOOLE-ORC2" "BOOLE-SET" "BOOLE-XOR" "CALL-ARGUMENTS-LIMIT"
  "CHAR-BITS-LIMIT" "CHAR-CODE-LIMIT" "CHAR-CONTROL-BIT" "CHAR-FONT-LIMIT" "CHAR-HYPER-BIT"
  "CHAR-META-BIT" "CHAR-SUPER-BIT" "DOUBLE-FLOAT-EPSILON" "DOUBLE-FLOAT-NEGATIVE-EPSILON"
  "INTERNAL-TIME-UNITS-PER-SECOND" "LAMBDA-LIST-KEYWORDS" "LAMBDA-PARAMETERS-LIMIT"
  "LEAST-NEGATIVE-DOUBLE-FLOAT" "LEAST-NEGATIVE-LONG-FLOAT" "LEAST-NEGATIVE-SHORT-FLOAT"
  "LEAST-NEGATIVE-SINGLE-FLOAT" "LEAST-POSITIVE-DOUBLE-FLOAT" "LEAST-POSITIVE-LONG-FLOAT"
  "LEAST-POSITIVE-SHORT-FLOAT" "LEAST-POSITIVE-SINGLE-FLOAT" "LONG-FLOAT-EPSILON"
  "LONG-FLOAT-NEGATIVE-EPSILON" "MOST-NEGATIVE-DOUBLE-FLOAT" "MOST-NEGATIVE-FIXNUM"
  "MOST-NEGATIVE-LONG-FLOAT" "MOST-NEGATIVE-SHORT-FLOAT" "MOST-NEGATIVE-SINGLE-FLOAT"
  "MOST-POSITIVE-DOUBLE-FLOAT" "MOST-POSITIVE-DOUBLE-FLOAT" "MOST-POSITIVE-FIXNUM"
  "MOST-POSITIVE-LONG-FLOAT" "MOST-POSITIVE-SHORT-FLOAT" "MOST-POSITIVE-SINGLE-FLOAT"
  "MULTIPLE-VALUES-LIMIT" "NIL" "OTHERWISE" "PI" "PRINT-ESCAPE*" "SHORT-FLOAT-EPSILON"
  "SHORT-FLOAT-NEGATIVE-EPSILON" "SINGLE-FLOAT-EPSILON" "SINGLE-FLOAT-NEGATIVE-EPSILON" "T"))

```

```
(CL:DEFPARAMETER CMLSymbols.FNAMES
```

```

' ("*" "+" "-" "/" "=" "1+" "1-" "<" "<=" "=" ">" ">=" "ABS" "ACONS" "ACOS" "ACOSH" "ADJOIN" "ADJUST-ARRAY"
  "ADJUSTABLE-ARRAY-P" "ALPHA-CHAR-P" "ALPHANUMERICP" "APPEND" "APPLY" "APPLYHOOK" "APROPOS"
  "APROPOS-LIST" "AREF" "ARRAY-DIMENSION" "ARRAY-DIMENSIONS" "ARRAY-ELEMENT-TYPE"
  "ARRAY-HAS-FILL-POINTER-P" "ARRAY-IN-BOUNDS-P" "ARRAY-RANK" "ARRAY-ROW-MAJOR-INDEX" "ARRAY-TOTAL-SIZE"
  "ARRAYP" "ASH" "ASIN" "ASINH" "ASSOC" "ASSOC-IF" "ASSOC-IF-NOT" "ATAN" "ATANH" "ATOM" "BIT" "BIT-AND"
  "BIT-ANDC1" "BIT-ANDC2" "BIT-EQV" "BIT-IOR" "BIT-NAND" "BIT-NOR" "BIT-NOT" "BIT-ORC1" "BIT-ORC2"
  "BIT-VECTOR-P" "BIT-XOR" "BOOLE" "BOTH-CASE-P" "BOUNDP" "BREAK" "BUTLAST" "BYTE" "BYTE-POSITION"
  "BYTE-SIZE" "CAR" "CDR" "CAAR" "CADR" "CDAR" "CDDR" "CAAAAR" "CAADR" "CADAR" "CADDR" "CDAAR" "CDADR"
  "CDAR" "CDDAR" "CAAAAR" "CAAADR" "CAADAR" "CAADDR" "CADAAR" "CADADR" "CADDAR" "CADDR" "CDAAR"
  "CDAADR" "CDADR" "CDDADR" "CDDAR" "CDDADR" "CDDDDR" "CEILING" "CERROR" "CHAR" "CHAR-BIT"
  "CHAR-BITS" "CHAR-CODE" "CHAR-DOWNCASE" "CHAR-EQUAL" "CHAR-FONT" "CHAR-GREATERP" "CHAR-INT"
  "CHAR-LESSP" "CHAR-NAME" "CHAR-NOT-EQUAL" "CHAR-NOT-GREATERP" "CHAR-NOT-LESSP" "CHAR-UPCASE" "CHAR/="
  "CHAR<" "CHAR<=" "CHAR=" "CHAR>" "CHAR>=" "CHARACTER" "CHARACTERP" "CIS" "CLEAR-INPUT" "CLEAR-OUTPUT"
  "CLOSE" "CLRHASH" "CODE-CHAR" "COERCE" "COMMONP" "COMPILE" "COMPILE-FILE" "COMPILED-FUNCTION-P"
  "COMPLEX" "COMPLEXP" "CONCATENATE" "CONJUGATE" "CONS" "CONSP" "CONSTANTP" "COPY-ALIST" "COPY-LIST"
  "COPY-READTABLE" "COPY-SEQ" "COPY-SYMBOL" "COPY-TREE" "COS" "COSH" "COUNT" "COUNT-IF" "COUNT-IF-NOT"
  "DECODE-FLOAT" "DECODE-UNIVERSAL-TIME" "DELETE" "DELETE-DUPPLICATES" "DELETE-FILE" "DELETE-IF"
  "DELETE-IF-NOT" "DENOMINATOR" "DEPOSIT-FIELD" "DESCRIBE" "DIGIT-CHAR" "DIGIT-CHAR-P" "DIRECTORY"
  "DIRECTORY-NAMESTRING" "DISASSEMBLE" "DOCUMENTATION" "DPB" "DRIBBLE" "ED" "EIGHTH" "ELT"
  "ENCODE-UNIVERSAL-TIME" "ENDP" "ENOUGH-NAMESTRING" "EQ" "EQL" "EQUAL" "EQUALP" "ERROR" "EVAL"
  "EVALHOOK" "EVENP" "EVERY" "EXP" "EXPORT" "EXPT" "FBOUNDP" "FCEILING" "FFLOOR" "FIFTH" "FILE-AUTHOR"
  "FILE-LENGTH" "FILE-NAMESTRING" "FILE-POSITION" "FILE-WRITE-DATE" "FILL" "FILL-POINTER" "FIND"
  "FIND-ALL-SYMBOLS" "FIND-IF" "FIND-IF-NOT" "FIND-PACKAGE" "FIND-SYMBOL" "FINISH-OUTPUT" "FIRST"
  "FLOAT" "FLOAT-DIGITS" "FLOAT-PRECISION" "FLOAT-RADIX" "FLOAT-SIGN" "FLOATP" "FLOOR" "FMAKUNBOUND"
  "FORCE-OUTPUT" "FORMAT" "FOURTH" "FRESH-LINE" "FROUND" "FTRUNCATE" "FUNCALL" "FUNCTIONP" "GCD"
  "GENSYM" "GENTEMP" "GET" "GET-DECODED-TIME" "GET-DISPATCH-MACRO-CHARACTER" "GET-INTERNAL-REAL-TIME"
  "GET-INTERNAL-RUN-TIME" "GET-MACRO-CHARACTER" "GET-OUTPUT-STREAM-STRING" "GET-PROPERTIES"
  "GET-SETF-METHOD" "GET-SETF-METHOD-MULTIPLE-VALUE" "GET-UNIVERSAL-TIME" "GETF" "GETHASH"
  "GRAPHIC-CHAR-P" "HASH-TABLE-COUNT" "HASH-TABLE-P" "HOST-NAMESTRING" "IDENTITY" "IMAGPART" "IMPORT"
  "IN-PACKAGE" "INPUT-STREAM-P" "INSPECT" "INT-CHAR" "INTEGER-DECODE-FLOAT" "INTEGER-LENGTH" "INTEGERP"
  "INTERN" "INTERSECTION" "ISQRT" "KEYWORDP" "LAST" "LCM" "LDB" "LDB-TEST" "LDIF" "LENGTH"
  "LISP-IMPLEMENTATION-TYPE" "LISP-IMPLEMENTATION-VERSION" "LIST" "LIST*" "LIST-ALL-PACKAGES"
  "LIST-LENGTH" "LISTEN" "LISTP" "LOAD" "LOG" "LOGAND" "LOGANDC1" "LOGANDC2" "LOGBITP" "LOGCOUNT"
  "LOGEQV" "LOGIOR" "LOGNAND" "LOGNOR" "LOGNOT" "LOGORC1" "LOGORC2" "LOGXOR" "LONG-SITE-NAME"
  "LOWER-CASE-P" "MACHINE-INSTANCE" "MACHINE-TYPE" "MACHINE-VERSION" "MACRO-FUNCTION" "MACROEXPAND"
  "MACROEXPAND-1" "MAKE-ARRAY" "MAKE-BROADCAST-STREAM" "MAKE-CHAR" "MAKE-CONCATENATED-STREAM"

```

```

"MAKE-DISPATCH-MACRO-CHARACTER" "MAKE-ECHO-STREAM" "MAKE-HASH-TABLE" "MAKE-LIST" "MAKE-PACKAGE"
"MAKE-PATHNAME" "MAKE-RANDOM-STATE" "MAKE-SEQUENCE" "MAKE-STRING" "MAKE-STRING-INPUT-STREAM"
"MAKE-STRING-OUTPUT-STREAM" "MAKE-SYMBOL" "MAKE-SYNONYM-STREAM" "MAKE-TWO-WAY-STREAM" "MAKUNBOUND"
"MAP" "MAPC" "MAPCAN" "MAPCAR" "MAPCON" "MAPHASH" "MAPL" "MAPLIST" "MASK-FIELD" "MAX" "MEMBER"
"MEMBER-IF" "MEMBER-IF-NOT" "MERGE" "MERGE-PATHNAMES" "MIN" "MINUSP" "MISMATCH" "MOD" "NAME-CHAR"
"NAMESTRING" "NBUTLAST" "NCONC" "NINTERSECTION" "NINTH" "NOT" "NOTANY" "NOTEVERY" "NRECONC" "NREVERSE"
"NSSET-DIFFERENCE" "NSSET-EXCLUSIVE-OR" "NSTRING-CAPITALIZE" "NSTRING-DOWNCASE" "NSTRING-UPCASE"
"NSUBLIS" "NSUBST" "NSUBST-IF" "NSUBST-IF-NOT" "NSUBSTITUTE" "NSUBSTITUTE-IF" "NSUBSTITUTE-IF-NOT"
"NTH" "NTHCDR" "NUMERATOR" "NULL" "NUMBERP" "NUNION" "ODDP" "OPEN" "OUTPUT-STREAM-P" "PACKAGE-NAME"
"PACKAGE-NICKNAMES" "PACKAGE-SHADOWING-SYMBOLS" "PACKAGE-USE-LIST" "PACKAGE-USED-BY-LIST" "PACKAGEP"
"PAIRLIS" "PARSE-INTEGER" "PARSE-NAMESTRING" "PATHNAME" "PATHNAME-DEVICE" "PATHNAME-DIRECTORY"
"PATHNAME-HOST" "PATHNAME-NAME" "PATHNAME-TYPE" "PATHNAME-VERSION" "PATHNAMEP" "PEEK-CHAR" "PHASE"
"PLUSP" "POSITION" "POSITION-IF" "POSITION-IF-NOT" "PPRINT" "PRIN1" "PRIN1-TO-STRING" "PRINC"
"PRINC-TO-STRING" "PRINT" "PROBE-FILE" "PROCLAIM" "PROVIDE" "RANDOM" "RANDOM-STATE-P" "RASSOC"
"RASSOC-IF" "RASSOC-IF-NOT" "RATIONAL" "RATIONALIZE" "RATIONALP" "READ" "READ-BYTE" "READ-CHAR"
"READ-CHAR-NO-HANG" "READ-DELIMITED-LIST" "READ-FROM-STRING" "READ-LINE" "READ-PRESERVING-WHITESPACE"
"READTABLEP" "REALPART" "REDUCE" "REM" "REMHASH" "REMOVE" "REMOVE-DUPPLICATES" "REMOVE-IF"
"REMOVE-IF-NOT" "REMPROP" "RENAME-FILE" "RENAME-PACKAGE" "REPLACE" "REQUIRE" "REST" "REVAPPEND"
"REVERSE" "ROOM" "ROUND" "RPLACA" "RPLACD" "SBIT" "SCALE-FLOAT" "SCHAR" "SEARCH" "SECOND" "SET"
"SET-CHAR-BIT" "SET-DIFFERENCE" "SET-DISPATCH-MACRO-CHARACTER" "SET-EXCLUSIVE-OR"
"SET-MACRO-CHARACTER" "SET-SYNTAX-FROM-CHAR" "SEVENTH" "SHADOW" "SHADOWING-IMPORT" "SHORT-SITE-NAME"
"SIGNUM" "SIMPLE-BIT-VECTOR-P" "SIMPLE-STRING-P" "SIMPLE-VECTOR-P" "SIN" "SINH" "SIXTH" "SLEEP"
"SOFTWARE-TYPE" "SOFTWARE-VERSION" "SOME" "SORT" "SPECIAL-FORM-P" "SQRT" "STABLE-SORT"
"STANDARD-CHAR-P" "STREAM-ELEMENT-TYPE" "STREAM-EXTERNAL-FORMAT" "STREAMP" "STRING"
"STRING-CAPITALIZE" "STRING-CHAR-P" "STRING-DOWNCASE" "STRING-EQUAL" "STRING-GREATERP"
"STRING-LEFT-TRIM" "STRING-LESSP" "STRING-NOT-EQUAL" "STRING-NOT-GREATERP" "STRING-NOT-LESSP"
"STRING-RIGHT-TRIM" "STRING-TRIM" "STRING-UPCASE" "STRING/= " "STRING< " "STRING<= " "STRING=" "STRING> "
"STRING>= " "STRINGP" "SUBLIS" "SUBSEQ" "SUBSETP" "SUBST" "SUBST-IF" "SUBST-IF-NOT" "SUBSTITUTE"
"SUBSTITUTE-IF" "SUBSTITUTE-IF-NOT" "SUBTYPEP" "SVREF" "SXHASH" "SYMBOL-FUNCTION" "SYMBOL-NAME"
"SYMBOL-PACKAGE" "SYMBOL-PLIST" "SYMBOL-VALUE" "SYMBOLP" "TAILP" "TANH" "TENTH" "TERPRI" "THIRD"
"TREE-EQUAL" "TRUENAME" "TRUNCATE" "TYPE-OF" "TYPEP" "UNEXPORT" "UNINTERN" "UNION" "UNREAD-CHAR"
"UNUSE-PACKAGE" "UPPER-CASE-P" "USE-PACKAGE" "USER-HOMEDIR-PATHNAME" "VALUES" "VALUES-LIST" "VECTOR"
"VECTOR-POP" "VECTOR-PUSH" "VECTOR-PUSH-EXTEND" "VECTORP" "WARN" "WRITE" "WRITE-BYTE" "WRITE-CHAR"
"WRITE-LINE" "WRITE-STRING" "WRITE-TO-STRING" "Y-OR-N-P" "YES-OR-NO-P" "ZEROP"))

```

```

(CL:DEFPARAMETER CMLSYMBOLS.DECLARATORS '("DECLARATION" "FTYPE" "FUNCTION" "IGNORE" "IGNORABLE"
                                           "INLINE" "NOTINLINE" "OPTIMIZE" "SPECIAL" "TYPE"))

```

(CL:DEFPARAMETER CMLSYMBOLS.TYPENAMES

```

'("ARRAY" "ATOM" "BIGNUM" "BIT" "BIT-VECTOR" "CHARACTER" "COMMON" "COMPILED-FUNCTION" "COMPLEX" "CONS"
  "DOUBLE-FLOAT" "FIXNUM" "FLOAT" "FUNCTION" "HASH-TABLE" "INTEGER" "KEYWORD" "LIST" "LONG-FLOAT"
  "NIL" "NUMBER" "PACKAGE" "PATHNAME" "RANDOM-STATE" "RATIO" "RATIONAL" "READTABLE" "SATISFIES"
  "SEQUENCE" "SHORT-FLOAT" "SIMPLE-ARRAY" "SIMPLE-BIT-VECTOR" "SIMPLE-STRING" "SIMPLE-VECTOR"
  "SIGNED-BYTE" "SINGLE-FLOAT" "STANDARD-CHAR" "STREAM" "STRING" "STRING-CHAR" "SYMBOL" "T"
  "UNSIGNED-BYTE" "VECTOR"))

```

(CL:DEFPARAMETER CMLSYMBOLS.MACROS

```

'("AND" "ASSERT" "CASE" "CCASE" "CHECK-TYPE" "COND" "CTYPECASE" "DECF" "DEFCONSTANT" "DEFINE-MODIFY-MACRO"
  "DEFINE-SETF-METHOD" "DEFMACRO" "DEFPARAMETER" "DEFSETF" "DEFSTRUCT" "DEFTYPE" "DEFUN" "DEFVAR" "DO"
  "DO*" "DO-ALL-SYMBOLS" "DO-EXTERNAL-SYMBOLS" "DO-SYMBOLS" "DOLIST" "DOTIMES" "ECASE" "ETYPECASE"
  "INCF" "LOCALLY" "LOOP" "LOOP-FINISH" "WITH-HASH-TABLE-ITERATOR" "WITH-PACKAGE-ITERATOR"
  "MULTIPLE-VALUE-BIND" "MULTIPLE-VALUE-LIST" "MULTIPLE-VALUE-SETQ" "OR" "POP" "PROG" "PROG*" "PROG1"
  "PROG2" "PSETF" "PSETQ" "PUSH" "PUSHNEW" "REMF" "RETURN" "ROTATEF" "SETF" "SHIFTF" "STEP" "TIME"
  "TRACE" "TYPECASE" "UNLESS" "UNTRACE" "WHEN" "WITH-INPUT-FROM-STRING" "WITH-OPEN-FILE"
  "WITH-OPEN-STREAM" "WITH-OUTPUT-TO-STRING"))

```

(CL:DEFPARAMETER CMLSYMBOLS.SPECIALFORMS

```

'("BLOCK" "CATCH" "COMPILER-LET" "DECLARE" "EVAL-WHEN" "FLET" "FUNCTION" "GO" "IF" "LABELS" "LAMBDA" "LET"
  "LET*" "MACROLET" "MULTIPLE-VALUE-CALL" "MULTIPLE-VALUE-PROG1" "PROGN" "PROGV" "QUOTE" "RETURN-FROM"
  "SETQ" "TAGBODY" "THE" "THROW" "UNWIND-PROTECT"))

```

```

(CL:DEFPARAMETER CMLSYMBOLS.LAMBDA.LIST.KEYWORDS '("&ALLOW-OTHER-KEYS" "&AUX" "&BODY" "&ENVIRONMENT"
                                                    "&KEY" "&OPTIONAL" "&REST" "&WHOLE"))

```

(CL:DEFPARAMETER CMLSYMBOLS.SHARED

```

'("+ " - " / " < " <= " = " > " >= " "&ALLOW-OTHER-KEYS" "&AUX" "&BODY" "&ENVIRONMENT" "&KEY" "&OPTIONAL"
  "&REST" "&WHOLE" "**APPLYHOOK*" "**BREAK-ON-WARNINGS*" "**DEBUG-IO*" "**DEFAULT-PATHNAME-DEFAULTS*"
  "**ERROR-OUTPUT*" "**EVALHOOK*" "**FEATURES*" "**LOAD-VERBOSE*" "**MACROEXPAND-HOOK*" "**MODULES*"
  "**PACKAGE*" "**PRINT-ARRAY*" "**PRINT-BASE*" "**PRINT-CASE*" "**PRINT-CIRCLE*" "**PRINT-ESCAPE*"
  "**PRINT-GENSYM*" "**PRINT-LENGTH*" "**PRINT-LEVEL*" "**PRINT-PRETTY*" "**PRINT-RADIX*" "**QUERY-IO*"
  "**RANDOM-STATE*" "**READ-BASE*" "**READ-DEFAULT-FLOAT-FORMAT*" "**READ-SUPPRESS*" "**READTABLE*"
  "**STANDARD-INPUT*" "**STANDARD-OUTPUT*" "**TERMINAL-IO*" "**TRACE-OUTPUT*" "ABS" "AND" "BIGNUM" "BIT"
  "BOUNDP" "BYTE" "BYTE-SIZE" "CAAAAR" "CAAADR" "CAAAAR" "CAADAR" "CAADDR" "CAADR" "CAAR" "CADAAR"
  "CADADR" "CADAR" "CADDAR" "CADDAR" "CADDR" "CADR" "CAR" "CASE" "CDAAR" "CDAADR" "CDAAR" "CDADAR"
  "CDADDR" "CDADR" "CDAR" "CDDAAR" "CDDADR" "CDDAR" "CDDAR" "CDDDR" "CDDDR" "CDDR" "CDR" "CLRHASH"
  "COERCE" "COMPLEX" "COND" "CONS" "DECLARE" "DEFMACRO" "DPB" "DRIBBLE" "ED" "EQ" "EQL" "EVENP" "EXPORT"
  "FLOAT" "GET" "GO" "IGNORE" "IGNORABLE" "IMPORT" "INSPECT" "INTEGER" "LAST" "LDB" "LET" "LET*" "LIST"
  "LIST*" "LOGAND" "LOGNOT" "LOGXOR" "MAX" "MIN" "MINUSP" "NCONC" "NIL" "NOT" "NULL" "ODDP" "OPEN" "OR"
  "PACKAGE" "PATHNAME" "PROG" "PROG*" "PROG1" "PROG2" "PROGN" "QUOTE" "RANDOM-STATE" "RATIO"
  "READTABLEP" "REMHASH" "REMPROP" "RETURN" "ROUND" "RPLACA" "RPLACD" "SATISFIES" "SEQUENCE" "SET"
  "STRING" "STRING-EQUAL" "STREAM" "STREAMP" "T" "TAILP" "THE" "TIME" "TRACE" "TYPE" "TYPEP" "UNTRACE"

```

```
"WRITE")
```

```
;;; Symbols shared by the Interlisp and Lisp packages.
```

```
)
```

```
(CL:DEFUN LITATOM.EXISTS (STRING)
  (AND (ATOMHASH#PROBES STRING)
    T))
```

```
(CL:DEFVAR LITATOM-PACKAGE-CONVERSION-TABLE
```

```
  ' ((("CL::" NIL "LISP" :INTERNAL)
    ("CL:" ("CL:FLG" "CL:MAKE-SYMBOL" "CL:COPY-SYMBOL" "CL:INTERN" "CL:MAKE-KEYWORD" "CL:GENTEMP"
      "CL:KEYWORDP")
      "LISP" :EXTERNAL)
    (":" NIL "KEYWORD" :EXTERNAL)
    ("CONDITIONS::" NIL "CONDITIONS" :INTERNAL)
    ("CONDITIONS:" NIL "CONDITIONS" :EXTERNAL)
    ("XCL::" NIL "XCL" :INTERNAL)
    ("XCL:" NIL "XCL" :EXTERNAL)
    ("SI::" NIL "SI" :INTERNAL)
    ("SI:" NIL "SI" :EXTERNAL)
    ("COMPILER::" NIL "COMPILER" :INTERNAL)
    ("COMPILER:" NIL "COMPILER" :EXTERNAL)
    ("FASL::" NIL "FASL" :INTERNAL)
    ("FASL:" NIL "FASL" :EXTERNAL)))
```

```
(CL:DEFUN NAMESTRING-CONVERSION-CLAUSE (BASE OFFSET LEN FATP)
```

```
;;; Check whether a given namestring has a prefix that would indicate membership in a package. If so, return the first clause out of the conversion table
;;; that matched. Otherwise, return NIL.
```

```
(DECLARE (CL:SPECIAL LITATOM-PACKAGE-CONVERSION-TABLE))
(CL:DOLIST (CONVERSION-LIST LITATOM-PACKAGE-CONVERSION-TABLE NIL)
  (LET* ((PREFIX (CL:FIRST CONVERSION-LIST))
    (EXCEPTIONS (CL:SECOND CONVERSION-LIST))
    (PREFIX-LENGTH (|ffetch| (STRINGP LENGTH)
      PREFIX)))
    (COND
      ((AND (IGREATERP LEN PREFIX-LENGTH)
        (\\STRING-EQUALBASE PREFIX BASE OFFSET PREFIX-LENGTH FATP)
        (NOT (|for| X |in| EXCEPTIONS |suchthat| (\\STRING-EQUALBASE X BASE OFFSET LEN FATP)))))
        (RETURN CONVERSION-LIST))))))
```

```
(CL:DEFUN CONVERT-LITATOM (ATOM)
```

```
;; Conditionally move an INTERLISP litatom into a package based on the naming conventions in LITATOM-PACKAGE-CONVERSION-TABLE.
```

```
(LET* ((BASE (|ffetch| (CL:SYMBOL PNAMEBASE) |of| ATOM))
  (LEN (|ffetch| (CL:SYMBOL PNAMELENGTH) |of| ATOM))
  (FATP (|ffetch| (CL:SYMBOL FATPNAMEP) |of| ATOM))
  (CLAUSE (OR (NAMESTRING-CONVERSION-CLAUSE BASE 1 LEN FATP)
    (CL:RETURN-FROM CONVERT-LITATOM NIL)))
  (PREFIX (CL:FIRST CLAUSE))
  (CL:PACKAGE-NAME (CL:THIRD CLAUSE))
  (WHERE (CL:FOURTH CLAUSE))
  (PREFIX-LENGTH (|ffetch| (STRINGP LENGTH)
    PREFIX)))
  (\\LITATOM.EATCHARS ATOM PREFIX-LENGTH) ; Take off the pseudo-package prefix. This makes the symbol
; inaccessible in INTERLISP (because not rehashed).

  (COND
    (CL:PACKAGE-NAME ; Symbol is interned, put it in the package.
      (INTERN-LITATOM ATOM (CL:FIND-PACKAGE CL:PACKAGE-NAME)
        :WHERE WHERE)))
    T))
```

```
(CL:DEFUN CONCOCT-SYMBOL (STRING)
```

```
;; Create a symbol in the LISP package. Conflicting symbols must already have been converted and defined by CONVERT-LITATOM. Given a
;; string, if a symbol by that name exists in INTERLISP (and doesn't conflict) we INTERN-LITATOM it into the LISP package, making that its home.
;; Otherwise, we create a new one.
```

```
(DECLARE (CL:SPECIAL *LISP-PACKAGE* *INTERLISP-PACKAGE* CMLSYMBOLS.SHARED))
(LET (ILSYM CLSYM)
  (COND
    ((CL:MULTIPLE-VALUE-BIND (SYM WHERE)
      (CL:FIND-SYMBOL STRING *LISP-PACKAGE*)
      (CL:WHEN (EQ WHERE :INTERNAL)
        (EXPORT SYM *LISP-PACKAGE*)))
      (SETQ CLSYM SYM)
      WHERE) ; The CL symbol already exists. Make it external. If the symbol
; is shared, import it into IL.
    (CL:WHEN (CL:MEMBER STRING CMLSYMBOLS.SHARED :TEST 'STREQUAL)
      (IMPORT CLSYM *INTERLISP-PACKAGE*)))
```

;; From this point down, the CL symbol doesn't yet exist.

((CL:MEMBER STRING CMLSYMBOLS.SHARED :TEST 'STREQUAL) ; The symbol is shared. Create it in CL and import it to IL.
; NOTE that the symbol should never be found in IL.

```
(COND
  ((CL:FIND-SYMBOL STRING *INTERLISP-PACKAGE*)
   (CL:ERROR "Shared symbol found in IL: ~S" STRING)
   ;; (intern-litatom ilsym *lisp-package* :where :external)
```

```
)
(T (LET ((SYM (CL:INTERN STRING *LISP-PACKAGE*)))
    (EXPORT SYM *LISP-PACKAGE*)
    (IMPORT SYM *INTERLISP-PACKAGE*))))
```

```
(T (EXPORT (CL:INTERN STRING *LISP-PACKAGE*)
            *LISP-PACKAGE*))) ; Symbol doesn't exist, so just create it in LISP.
```

(CL:DEFUN **TRANSFER-SYMBOL (FROM TO)**

"Move the function and plist definition cells of a symbol onto another, leaving name and value alone."

```
(CL:SETF (CL:SYMBOL-PLIST TO)
          (CL:SYMBOL-PLIST FROM))
(CL:SETF (CL:SYMBOL-FUNCTION TO)
          (CL:SYMBOL-FUNCTION FROM))
```

(CL:DEFUN **INTERN-LITATOM (ATOM PACKAGE &KEY WHERE)**

"Tag a litatom with a package. Add it to the package hashtable. Handle keywords appropriately. Return the symbol."

```
(CL:WHEN (AND (CL::%PACKAGE-EXTERNAL-ONLY PACKAGE)
              (EQ WHERE :INTERNAL))
  (ERROR (CONCAT "Attempting to INTERN-LITATOM " ATOM "internal in external-only package " PACKAGE)))
(ADD-SYMBOL (CL:IF (EQ WHERE :INTERNAL)
                  (CL::%PACKAGE-INTERNAL-SYMBOLS PACKAGE)
                  (CL::%PACKAGE-EXTERNAL-SYMBOLS PACKAGE))
            ATOM)
(CL:SETF (CL:SYMBOL-PACKAGE ATOM)
          PACKAGE)
(CL:IF (EQ *KEYWORD-PACKAGE* PACKAGE)
      (SET ATOM ATOM)
      ATOM)
```

(CL:DEFUN **\\LITATOM.EATCHARS (LITATOM N)**

```
(LET* ((PNBASE (|fetch| (LITATOM PNAMEBASE) |of| LITATOM))
       (LEN (- (|fetch| (PNAMEBASE PNAMELENGTH) |of| PNBASE)
               N)))
  (COND
    ((|fetch| (LITATOM FATPNAMEP) |of| LITATOM)
     (ERROR (CONCAT "Can't move fat LITATOM |" LITATOM "|" into LISP package")))
    (T (|for| I |from| 0 |to| LEN |as| J |from| N |do| (\\PUTBASETHIN PNBASE I (\\GETBASETHIN PNBASE J))))))
  (|replace| (PNAMEBASE PNAMELENGTH) |of| PNBASE |with| LEN))
LITATOM)
```

(CL:DEFUN **PACKAGE-INIT (&OPTIONAL (CONVERT? T))**

"Clear, make structures of, initialize & convert symbols to, and enable use of the symbol package system."

```
(PACKAGE-CLEAR)
(PACKAGE-MAKE)
(PACKAGE-HIERARCHY-INIT CONVERT?)
(PACKAGE-ENABLE)
T)
```

(CL:DEFUN **PACKAGE-CLEAR ()**

"Clear the global package data (used by FIND-PACKAGE) and reset the globals that hold the existing packages."

```
(DECLARE (CL:SPECIAL *PACKAGE-FROM-NAME* *PACKAGE-FROM-INDEX* *PACKAGE* *LISP-PACKAGE* *KEYWORD-PACKAGE*
                    *INTERLISP-PACKAGE*))
(CLRHASH *PACKAGE-FROM-NAME*)
(CL:DOTIMES (I (ADD1 *TOTAL-PACKAGES-LIMIT*))
  (CL:SETF (CL:AREF *PACKAGE-FROM-INDEX* I)
            NIL))
(SETQ *PACKAGE* NIL)
(SETQ *LISP-PACKAGE* NIL)
(SETQ *KEYWORD-PACKAGE* NIL)
(SETQ *INTERLISP-PACKAGE* NIL)
T)
```

(CL:DEFUN **PACKAGE-MAKE ()**

"Create, but do not fill with symbols, the base packages that need to exist. Also enables the package qualifier characters in the readtables and saves the old definitions of \\READ.SYMBOL and \\MKATOM."

```
(DECLARE (CL:SPECIAL *LISP-PACKAGE* *KEYWORD-PACKAGE* *INTERLISP-PACKAGE* *PACKAGE* HASHTABLE-SIZE-LIMIT))
(SETQ *INTERLISP-PACKAGE* (CL:MAKE-PACKAGE "INTERLISP" :USE NIL :NICKNAMES '("IL")
                                           :PREFIX-NAME "IL" :EXTERNAL-ONLY T :EXTERNAL-SYMBOLS 32749))
(SETQ *LISP-PACKAGE* (CL:MAKE-PACKAGE "LISP" :USE NIL :NICKNAMES '("CL" "COMMON-LISP"))
```

```

                                :PREFIX-NAME "CL" :EXTERNAL-SYMBOLS 1173))
(CL:MAKE-PACKAGE "CONDITIONS" :USE "LISP" :PREFIX-NAME "CONDITIONS")
(CL:MAKE-PACKAGE "XEROX-COMMON-LISP" :USE '("LISP" "CONDITIONS")
:NICKNAMES
'("XCL")
:PREFIX-NAME "XCL")
(CL:MAKE-PACKAGE "SYSTEM" :USE "LISP" :NICKNAMES '("SYS" "SI")
:PREFIX-NAME "SI")
(CL:MAKE-PACKAGE "USER" :USE "LISP")
(SETQ *KEYWORD-PACKAGE* (CL:MAKE-PACKAGE "KEYWORD" :USE NIL :EXTERNAL-ONLY T :EXTERNAL-SYMBOLS 96))
(CL:MAKE-PACKAGE "COMPILER" :USE "LISP")
(CL:MAKE-PACKAGE "FASL" :USE "LISP")
(CL:MAKE-PACKAGE "XCL-USER" :USE '("LISP" "XCL"))
(MOVD '\\READ.SYMBOL '\\OLD.READ.SYMBOL)
(MOVD '\\MKATOM '\\OLD.MKATOM)
T)

```

```
(CL:DEFUN PACKAGE-HIERARCHY-INIT (&OPTIONAL (CONVERT? NIL))
```

```

;;; Fill all the initial system packages with their proper symbols, moving litatoms into appropriate places and such. If convert? is non-nil then symbols
;;; whose pnames have fake package qualifiers, like cl:length, will be converted IN PLACE to remove the qualifier. If conversion takes place you cannot
;;; fully disable the package system.

```

```

(DECLARE (CL:SPECIAL *INTERLISP-PACKAGE* *KEYWORD-PACKAGE* CMLSYMBOLS.LAMBDA.LIST.KEYWORDS
CMLSYMBOLS.SPECIALFORMS CMLSYMBOLS.MACROS CMLSYMBOLS.TYPENAMES CMLSYMBOLS.FNAMES
CMLSYMBOLS.DECLARATORS CMLSYMBOLS.VARS))

```

```
;; Fill the INTERLISP package with its symbols.
```

```

(MAPATOMS #'(CL:LAMBDA (ATOM)
              (CL:IF (OR (NULL CONVERT?)
                        (NULL (CONVERT-LITATOM ATOM)))
                    (INTERN-LITATOM ATOM *INTERLISP-PACKAGE* :WHERE :EXTERNAL))))

```

```
;; Fill the LISP package with its symbols.
```

```

(CL:DOLIST (I (APPEND CMLSYMBOLS.VARS CMLSYMBOLS.FNAMES CMLSYMBOLS.DECLARATORS CMLSYMBOLS.TYPENAMES
CMLSYMBOLS.MACROS CMLSYMBOLS.SPECIALFORMS CMLSYMBOLS.LAMBDA.LIST.KEYWORDS))
  (CONCOCT-SYMBOL I))
T)

```

```
(CL:DEFUN PACKAGE-ENABLE (&OPTIONAL (PACKAGE *INTERLISP-PACKAGE*))
```

```
"Turn on the package system, making PACKAGE the current one and redefining \\READ.SYMBOL and \\MKATOM appropriately."
```

```

(DECLARE (CL:SPECIAL *INTERLISP-PACKAGE* *PACKAGE* *OLD-INTERLISP-READ-ENVIRONMENT* *PER-EXEC-VARIABLES*))
(|replace| REPACKAGE |of| *OLD-INTERLISP-READ-ENVIRONMENT* |with| *INTERLISP-PACKAGE*)
(|replace| REPACKAGE |of| *DEFINE-FILE-INFO-ENV* |with| *INTERLISP-PACKAGE*)
(COND

```

```

  ((FIND-READTABLE "LISP")
   (READTABLEPROP (FIND-READTABLE "LISP")
    'PACKAGECHAR
    (CHARCODE ":"))))

```

```

(COND
  ((FIND-READTABLE "INTERLISP")
   (READTABLEPROP (FIND-READTABLE "INTERLISP")
    'PACKAGECHAR
    (CHARCODE ":"))))

```

```

(COND
  ((FIND-READTABLE "XCL")
   (READTABLEPROP (FIND-READTABLE "XCL")
    'PACKAGECHAR
    (CHARCODE ":"))))

```

```
(RPAQ? *PER-EXEC-VARIABLES* NIL)
```

```

(CL:PUSHNEW '(*PACKAGE* (COND
  ((CL:PACKAGEP *PACKAGE*)
   *PACKAGE*)
  (T (PROMPTPRINT "Invalid package, reset to LISP")
     (SETQ *PACKAGE* (CL:FIND-PACKAGE "LISP")))))
  *PER-EXEC-VARIABLES* :TEST 'CL:EQUAL)
(CL:SETF *DEFAULT-MAKEFILE-ENVIRONMENT* '(:READTABLE "INTERLISP" :PACKAGE "INTERLISP" :FORMAT :XCCS))
(MOVD '\\NEW.READ.SYMBOL '\\READ.SYMBOL)
(MOVD '\\NEW.MKATOM '\\MKATOM)
(CL:SETF *PACKAGE* PACKAGE)
T)

```

```
(CL:DEFUN PACKAGE-DISABLE ()
```

```
"Turn off the package system and restore the old definitions of \\\\READ.SYMBOL and \\MKATOM. After disabling, symbols interned under the package system will not be EQ to symbols of the same name reread."
```

```

(MOVD '\\OLD.READ.SYMBOL '\\READ.SYMBOL)
(MOVD '\\OLD.MKATOM '\\MKATOM)
(SETQ *PACKAGE* NIL)
(|replace| REPACKAGE |of| *OLD-INTERLISP-READ-ENVIRONMENT* |with| NIL)
(READTABLEPROP (FIND-READTABLE "LISP")
 'PACKAGECHAR 0)
(READTABLEPROP (FIND-READTABLE "INTERLISP")
 'PACKAGECHAR 0)

```

```
{MEDLEY}<sources>PACKAGE-STARTUP.;1 (PACKAGE-DISABLE cont.)
```

Page 8

```
(READTABLEPROP (FIND-READTABLE "XCL")
 'PACKAGECHAR 0)
T)
```

;; A hack for initialization

```
(CL:DEFUN ID (X)
 X)
```

```
(PUTPROPS PACKAGE-STARTUP FILETYPE CL:COMPILE-FILE)
```

```
(PUTPROPS PACKAGE-STARTUP MAKEFILE-ENVIRONMENT (:READTABLE "XCL" :PACKAGE "INTERLISP"))
```

;; Initialize package system, plus functions needed in lpackage at init time

```
(DECLARE\ : DONTEVAL@LOAD DOCOPY
```

```
(MOVD? 'EQ 'EQL)
```

```
(MOVD? 'LENGTH 'CL:LENGTH)
```

```
(MOVD? 'ID 'CL:IDENTITY)
```

```
(MOVD? 'ID 'REMOVE-COMMENTS)
```

```
(PACKAGE-INIT)
)
```

FUNCTION INDEX

CHECK-SYMBOL-NAMESTRING	2	NAMESTRING-CONVERSION-CLAUSE	5	RETURN-FIRST-OF-THREE	1
CONCOCT-SYMBOL	5	PACKAGE-CLEAR	6	TRANSFER-SYMBOL	6
CONVERT-LITATOM	5	PACKAGE-DISABLE	7	\\LITATOM.EATCHARS	6
ERROR-MISSING-EXTERNAL-SYMBOL	1	PACKAGE-ENABLE	7	\\NEW.MKATOM	2
ID	8	PACKAGE-HIERARCHY-INIT	7	\\NEW.READ.SYMBOL	2
INTERN-LITATOM	6	PACKAGE-INIT	6		
LITATOM.EXISTS	5	PACKAGE-MAKE	6		

VARIABLE INDEX

CMLSYMBOLS.DECLARATORS	4	CMLSYMBOLS.SPECIALFORMS	4
CMLSYMBOLS.FNAMES	3	CMLSYMBOLS.TYPENAMES	4
CMLSYMBOLS.LAMBDA.LIST.KEYWORDS	4	CMLSYMBOLS.VARS	3
CMLSYMBOLS.MACROS	4	LITATOM-PACKAGE-CONVERSION-ENABLED	3
CMLSYMBOLS.SHARED	4	LITATOM-PACKAGE-CONVERSION-TABLE	5

PROPERTY INDEX

PACKAGE-STARTUP	8
-----------------------	---
