

File created: 8-Apr-92 23:35:08 {DSK}<usr>local>lde>lispcore>sources>COMMON-LISP-PACKAGE.;4

changes to: (FUNCTIONS CREATE-CL-PACKAGE)

previous date: 8-Apr-92 16:51:03 {DSK}<usr>local>lde>lispcore>sources>COMMON-LISP-PACKAGE.;3

Read Table: XCL

Package: INTERLISP

Format: XCCS

; Copyright (c) 1991, 1992 by Venue Corporation. All rights reserved.

```
(RPAQQ COMMON-LISP-PACKAGECOMS ((VARIABLES *COMMON-LISP-PACKAGE* NEWCLSYMS OLDCLSYMS XCLCLSYMS SPLITCLSYMS
                                STRANGECLSYMS)
                                (FUNCTIONS CRUNCH-FILES CREATE-CL-PACKAGE FLIP-CL)
                                (PROP FILETYPE COMMON-LISP-PACKAGE)))
```

```
(DEFGLOBALVAR *COMMON-LISP-PACKAGE* NIL
 "Place holder for the COMMON-LISP package variable")
```

(CL:DEFPARAMETER **NEWCLSYMS**

```
' (*BREAK-ON-SIGNALS* " *COMPILE-FILE-PATHNAME*" " *COMPILE-FILE-TRUENAME*" " *COMPILE-PRINT*"
  " *COMPILE-VERBOSE*" " *DEBUGGER-HOOK*" " *LOAD-PATHNAME*" " *LOAD-PRINT*" " *LOAD-TRUENAME*"
  " *PRINT-LINES*" " *PRINT-MISER-WIDTH*" " *PRINT-PPRINT-DISPATCH*" " *PRINT-READABLY*"
  " *PRINT-RIGHT-MARGIN*" " *READ-EVAL*" " *ABORT" " *AUGMENT-ENVIRONMENT" " *BASE-CHARACTER" " *BASE-STRING"
  " *BROADCAST-STREAM" " *BROADCAST-STREAM-STREAMS" " *CELL-ERROR-NAME" " *COMPILE-FILE-PATHNAME"
  " *COMPILER-MACRO-FUNCTION" " *COMPILER-MACROEXPAND" " *COMPILER-MACROEXPAND-1" " *COMPLEMENT"
  " *COMPUTE-RESTARTS" " *CONCATENATED-STREAM" " *CONCATENATED-STREAM-STREAMS" " *CONDITION" " *CONTINUE"
  " *COPY-PPRINT-DISPATCH" " *DECLAIM" " *DECLARATION-INFORMATION" " *DEFINE-COMPILER-MACRO"
  " *DEFINE-CONDITION" " *DEFINE-DECLARATION" " *DESTRUCTURING-BIND" " *DIVISION-BY-ZERO" " *DYNAMIC-EXTENT"
  " *ECHO-STREAM" " *ECHO-STREAM-INPUT-STREAM" " *ECHO-STREAM-OUTPUT-STREAM" " *ENCLOSE" " *END-OF-FILE"
  " *EXTENDED-CHARACTER" " *FDEFINITION" " *FILE-ERROR" " *FILE-ERROR-PATHNAME" " *FILE-STREAM"
  " *FILE-STRING-LENGTH" " *FIND-RESTART" " *FLOATING-POINT-INEXACT" " *FLOATING-POINT-INVALID-OPERATION"
  " *FLOATING-POINT-OVERFLOW" " *FLOATING-POINT-UNDERFLOW" " *FORMATTER" " *FUNCTION-INFORMATION"
  " *FUNCTION-LAMBDA-EXPRESSION" " *HANDLER-CASE" " *HANDLER-BIND" " *HASH-TABLE-REHASH-SIZE"
  " *HASH-TABLE-REHASH-THRESHOLD" " *HASH-TABLE-SIZE" " *HASH-TABLE-TEST" " *IGNORE-ERRORS"
  " *INTERACTIVE-STREAM-P" " *INVOKE-DEBUGGER" " *INVOKE-RESTART" " *INVOKE-RESTART-INTERACTIVELY"
  " *LEAST-NEGATIVE-NORMALIZED-DOUBLE-FLOAT" " *LEAST-NEGATIVE-NORMALIZED-LONG-FLOAT"
  " *LEAST-NEGATIVE-NORMALIZED-SHORT-FLOAT" " *LEAST-NEGATIVE-NORMALIZED-SINGLE-FLOAT"
  " *LEAST-POSITIVE-NORMALIZED-DOUBLE-FLOAT" " *LEAST-POSITIVE-NORMALIZED-LONG-FLOAT"
  " *LEAST-POSITIVE-NORMALIZED-SHORT-FLOAT" " *LEAST-POSITIVE-NORMALIZED-SINGLE-FLOAT"
  " *LOAD-LOGICAL-PATHNAME-TRANSLATIONS" " *LOAD-TIME-EVAL" " *LOAD-TIME-VALUE" " *LOGICAL-PATHNAME"
  " *LOGICAL-PATHNAME-TRANSLATIONS" " *MAKE-CONDITION" " *MAKE-LOAD-FORM" " *MAKE-LOAD-FORM-SAVING-SLOTS"
  " *MAP-INTO" " *MUFFLE-WARNING" " *NTH-VALUE" " *OPEN-STREAM-P" " *PACKAGE-ERROR" " *PARSE-ERROR" " *PARSE-MACRO"
  " *PATHNAME-MATCH-P" " *PPRINT-DISPATCH" " *PPRINT-EXIT-IF-LIST-EXHAUSTED" " *PPRINT-FILL" " *PPRINT-INDENT"
  " *PPRINT-LINEAR" " *PPRINT-LOGICAL-BLOCK" " *PPRINT-NEWLINE" " *PPRINT-POP" " *PPRINT-TAB" " *PPRINT-TABULAR"
  " *PRINT-NOT-READABLE" " *PRINT-UNREADABLE-OBJECT" " *PROGRAM-ERROR" " *READER-ERROR" " *READTABLE-CASE"
  " *READTABLE-CASE" " *REAL" " *REALP" " *RESTART" " *RESTART-BIND" " *RESTART-CASE" " *SET-PPRINT-DISPATCH"
  " *SIMPLE-BASE-STRING" " *SIMPLE-CONDITION" " *SIMPLE-CONDITION-FORMAT-ARGUMENTS" " *SIMPLE-CONDITION-FORMAT-STRING"
  " *STORE-VALUE" " *STREAM-ERROR-STREAM" " *STREAM-EXTERNAL-FORMAT" " *STRING-STREAM" " *STYLE-WARNING"
  " *SYNONYM-STREAM" " *SYNONYM-STREAM-SYMBOL" " *TRANSLATE-LOGICAL-PATHNAME" " *TRANSLATE-PATHNAME"
  " *TWO-WAY-STREAM" " *TWO-WAY-STREAM-INPUT-STREAM" " *TWO-WAY-STREAM-OUTPUT-STREAM" " *TYPE-ERROR-DATUM"
  " *UNBOUND-SLOT" " *UNBOUND-VARIABLE" " *UPGRADED-ARRAY-ELEMENT-TYPE" " *UPGRADED-COMPLEX-PART-TYPE"
  " *USE-VALUE" " *VARIABLE-INFORMATION" " *WILD-PATHNAME-P" " *WITH-COMPILATION-UNIT"
  " *WITH-CONDITION-RESTARTS" " *WITH-HASH-TABLE-ITERATOR" " *WITH-PACKAGE-ITERATOR" " *WITH-SIMPLE-RESTART"
  " *WITH-STANDARD-IO-SYNTAX"))
```

(CL:DEFPARAMETER **OLDCLSYMS**

```
' ("COMMON" "COMMONP" "STRING-CHAR-P" "INT-CHAR" "COMPILER-LET" "CHAR-BIT" "SET-CHAR-BIT"
  " *MODULES*" " *PROVIDE" " *REQUIRE" " *CHAR-FONT-LIMIT" " *CHAR-BITS-LIMIT" " *CHAR-BITS" " *CHAR-FONT"
  " *MAKE-CHAR" " *CHAR-CONTROL-BIT" " *CHAR-META-BIT" " *CHAR-SUPER-BIT" " *CHAR-HYPER-BIT"
  " *BREAK-ON-WARNINGS*")
"Symbols in LISP and not in COMMON-LISP")
```

(CL:DEFPARAMETER **XCLCLSYMS**

```
' ("ABORT" "ARITHMETIC-ERROR" "ARITHMETIC-ERROR-OPERANDS" "ARITHMETIC-ERROR-OPERATION"
  "BROADCAST-STREAM-STREAMS" "CELL-ERROR" "CELL-ERROR-NAME" "CONCATENATED-STREAM-STREAMS" "CONDITION"
  "CONTROL-ERROR" "DEFINE-CONDITION" "DEFPACKAGE" "DESTRUCTURING-BIND" "DELETE-PACKAGE"
  "ECHO-STREAM-INPUT-STREAM" "ECHO-STREAM-OUTPUT-STREAM" "END-OF-FILE" "HANDLER-BIND" "IGNORE-ERRORS"
  "MAKE-CONDITION" "OPEN-STREAM-P" "PACKAGE-ERROR" "PACKAGE-ERROR-PACKAGE" "SERIOUS-CONDITION"
  "SIGNAL" "SIMPLE-CONDITION" "SIMPLE-CONDITION-FORMAT-ARGUMENTS" "SIMPLE-CONDITION-FORMAT-STRING"
  "SIMPLE-ERROR" "SIMPLE-TYPE-ERROR" "SIMPLE-WARNING" "STORAGE-CONDITION" "STORE-VALUE" "STREAM-ERROR"
  "STREAM-ERROR-STREAM" "SYNONYM-STREAM-SYMBOL" "TWO-WAY-STREAM-INPUT-STREAM"
  "TWO-WAY-STREAM-OUTPUT-STREAM" "TYPE-ERROR" "TYPE-ERROR-EXPECTED-TYPE" "UNBOUND-VARIABLE"
  "UNDEFINED-FUNCTION" "USE-VALUE" "WARNING"))
```

(CL:DEFPARAMETER **SPLITCLSYMS** ' ("LOCALLY" "IN-PACKAGE"))

(CL:DEFPARAMETER **STRANGECLSYMS** ' (("LISP" "SIMPLE-STRING" " *GENSYM-COUNTER*")
 ("XCL" "ROW-MAJOR-AREF"))

"Symbols in CL that are predefined in the loadup in another package"

```
(CL:DEFUN CRUNCH-FILES (FL)
  (CL:WHEN (AND FL (CL:SYMBOLP FL))
    (CL:SETQ FL (LIST FL)))
  (CL:DOLIST (F FL)
    (CL:FORMAT T "Crunching ~a~%" F)
    (FLIP-CL :LISP)
    (LOAD F 'ALLPROP)
    (FLIP-CL :NOWHERE)
    (MAKEFILE F 'NEW)
    (CL:IF (CL:PROBE-FILE (CONCAT F ".DFASL"))
      (CL:COMPILE-FILE F)
      (FAKE-COMPILE-FILE F))
    (CL:FORMAT T "Done crunching ~a~%" F)))
```

```
(CL:DEFUN CREATE-CL-PACKAGE () ; Edited 8-Apr-92 20:15 by jrb:
```

;; First, rename the LISP package to get its nicknames out of our way

```
(CL:RENAME-PACKAGE (CL:FIND-PACKAGE "LISP")
  "LISP" NIL NIL)
```

;; Then create the COMMON-LISP package and friends

```
(CL:UNLESS (CL:FIND-PACKAGE "COMMON-LISP")
```

;; For the moment, no nicknames for COMMON-LISP; FLIP-CL can be used to fix this later.

```
(SETQ *COMMON-LISP-PACKAGE* (CL:MAKE-PACKAGE "COMMON-LISP" :USE NIL))
(CL:MAKE-PACKAGE "COMMON-LISP-USER" :USE '("COMMON-LISP" "XCL"))
```

```
(LET ((WEIRDTAG (CONS NIL NIL))
      (OLDPROP (CONS NIL NIL))
      (UNSHAREDPROP (CONS NIL NIL))
      I)
```

;; Flag the atoms in LISP that are not going to be shared into COMMON-LISP

```
(CL:DOLIST (I OLDCLSYMS)
  (PUT (CL:FIND-SYMBOL I *LISP-PACKAGE*)
    WEIRDTAG OLDPROP))
(CL:DOLIST (I SPLITCLSYMS)
  (PUT (CL:FIND-SYMBOL I *LISP-PACKAGE*)
    WEIRDTAG UNSHAREDPROP))
```

;; OK, crunch the external symbols in LISP. We may eventually rehome these symbols into COMMON-LISP

```
(CL:DO-EXTERNAL-SYMBOLS (I *LISP-PACKAGE*)
  (LET ((WEIRD? (GET I WEIRDTAG))
        S)
    (COND
      ((EQ WEIRD? OLDPROP) ; Just leave it alone
       (REMPROP I WEIRDTAG))
      ((EQ WEIRD? UNSHAREDPROP) ; Export a new, unshared symbol
       (EXPORT (CL:INTERN (CL:SYMBOL-NAME I)
                          *COMMON-LISP-PACKAGE*)
              *COMMON-LISP-PACKAGE*)
       (REMPROP I WEIRDTAG))
      ((NULL WEIRD?) ; Share symbol; if it's already there, shadow it
       (CL:IF (SETQ S (CL:FIND-SYMBOL (CL:SYMBOL-NAME I)
                                      *COMMON-LISP-PACKAGE*))
              (CL:UNLESS (EQ S I)
                (CL:SHADOWING-IMPORT I *COMMON-LISP-PACKAGE*))
              (IMPORT I *COMMON-LISP-PACKAGE*))
              (EXPORT I *COMMON-LISP-PACKAGE*))
              ; VERY unlikely...
              (T (ERROR "Garbage on property list during LISP->COMMON-LISP import" (CONS I WEIRD?))))))
```

;; Handle the few strange CLsymbols (ones that for one reason or another already exist in another package).

```
(CL:DOLIST (SL STRANGECLSYMS)
  (LET ((P (CL:FIND-PACKAGE (CAR SL)))
        OLDS)
    (CL:DOLIST (S (CDR SL))
      (IF (SETQ OLDS (CL:FIND-SYMBOL S P))
        THEN (IMPORT OLDS *COMMON-LISP-PACKAGE*)
              (EXPORT OLDS *COMMON-LISP-PACKAGE*))))))
```

;; And snarf the XCL symbols that need to be shared with COMMON-LISP

```
(LET ((XCLP (CL:FIND-PACKAGE "XCL")))
  (CL:DOLIST (I XCLCLSYMS)
    (IMPORT (CL:FIND-SYMBOL I XCLP)
      *COMMON-LISP-PACKAGE*)
    (EXPORT (CL:FIND-SYMBOL I *COMMON-LISP-PACKAGE*)
      *COMMON-LISP-PACKAGE*)))
```

;; If these other packages are around, grab their symbols

```
(LET (P S)
  (CL:WHEN (SETQ P (CL:FIND-PACKAGE "XP"))
    (CL:DOLIST (I '(" *PRINT-PPRINT-DISPATCH*" " *PPRINT-RIGHT-MARGIN*" " *PPRINT-MISER-WIDTH*"
                  " *PPRINT-NEWLINE*" " *PPRINT-LOGICAL-BLOCK*" " *PPRINT-EXIT-IF-LIST-EXHAUSTED"
```

"PPRINT-POP" "PPRINT-INDENT" "PPRINT-TAB" "PPRINT-FILL" "PPRINT-LINEAR"
"PPRINT-TABULAR" "FORMATTER" "COPY-PPRINT-DISPATCH" "PPRINT-DISPATCH"
"SET-PPRINT-DISPATCH"))

(SETQ S (CL:FIND-SYMBOL I P))
(IMPORT S *COMMON-LISP-PACKAGE*)
(EXPORT S *COMMON-LISP-PACKAGE*))

;; This will have to be changed somewhat as we change the CONDITIONS system to comply with CLtL2

(CL:WHEN (SETQ P (CL:FIND-PACKAGE "CONDITIONS"))
(CL:UNUSE-PACKAGE *LISP-PACKAGE* P)
(CL:USE-PACKAGE *COMMON-LISP-PACKAGE* P)
(CL:DO-EXTERNAL-SYMBOLS (I P)
(LET ((S (CL:FIND-SYMBOL (CL:SYMBOL-NAME I)
COMMON-LISP-PACKAGE))
(IF S
THEN (CL:SHADOWING-IMPORT I *COMMON-LISP-PACKAGE*)
ELSE (IMPORT I *COMMON-LISP-PACKAGE*))
(EXPORT I *COMMON-LISP-PACKAGE*)))))

;; Finally, hose out the new COMMON-LISP symbols

(CL:DOLIST (I NEWCLSYMS)
(EXPORT (CL:INTERN I *COMMON-LISP-PACKAGE*)
COMMON-LISP-PACKAGE))
(FLIP-CL :COMMON-LISP)))

(CL:DEFUN FLIP-CL (WHERE)

(LET ((WHERE-WAS-IT (CL:FIND-PACKAGE "CL")))
(SETQ WHERE-WAS-IT (COND
(EQ WHERE-WAS-IT *COMMON-LISP-PACKAGE*)
:COMMON-LISP)
(EQ WHERE-WAS-IT *LISP-PACKAGE*)
:LISP)
(NULL WHERE-WAS-IT)
:NOWHERE)
(T (ERROR "CL nickname in odd package" WHERE-WAS-IT))))

(CL:ECASE WHERE
(:LISP
(CL:RENAME-PACKAGE *COMMON-LISP-PACKAGE* "COMMON-LISP" NIL NIL)
(CL:RENAME-PACKAGE *LISP-PACKAGE* "LISP" '("CL")
"CL"))
(:COMMON-LISP
(CL:RENAME-PACKAGE *LISP-PACKAGE* "LISP" NIL NIL)
(CL:RENAME-PACKAGE *COMMON-LISP-PACKAGE* "COMMON-LISP" '("CL")
"CL"))
(:NOWHERE
(CL:RENAME-PACKAGE *LISP-PACKAGE* "LISP" NIL NIL)
(CL:RENAME-PACKAGE *COMMON-LISP-PACKAGE* "COMMON-LISP" NIL NIL)))
WHERE-WAS-IT))

(PUTPROPS COMMON-LISP-PACKAGE FILETYPE :COMPILE-FILE)

(PUTPROPS COMMON-LISP-PACKAGE COPYRIGHT ("Venue Corporation" 1991 1992))

FUNCTION INDEX

CREATE-CL-PACKAGE2 CRUNCH-FILES2 FLIP-CL3

VARIABLE INDEX

COMMON-LISP-PACKAGE1 OLDCLSYMS1 STRANGECLSYMS1
NEWCLSYMS1 SPLITCLSYMS1 XCLCLSYMS1

PROPERTY INDEX

COMMON-LISP-PACKAGE3
