

File created: 2-Feb-91 14:33:36 {DSK}<usr>local>lde>lispcore>sources2>COMMON-LISP-PACKAGE.;7

changes to: (VARIABLES OLDCLSYMS *COMMON-LISP-PACKAGE*)
(VARS COMMON-LISP-PACKAGECOMS)
(FUNCTIONS CRUNCH-FILES CREATE-CL-PACKAGE)

previous date: 2-Feb-91 13:17:24 {DSK}<usr>local>lde>lispcore>sources2>COMMON-LISP-PACKAGE.;5

Read Table: XCL

Package: INTERLISP

Format: XCCS

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

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

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

```
(CL:DEFPARAMETER NEWCLSYMS
 ' ("REAL" "BASE-CHARACTER" "EXTENDED-CHARACTER" "READTABLE-CASE" "SIMPLE-STRING" "BASE-STRING"
 "SIMPLE-BASE-STRING" "BROADCAST-STREAM" "CONCATENATED-STREAM" "ECHO-STREAM" "SYNONYM-STREAM"
 "STRING-STREAM" "FILE-STREAM" "TWO-WAY-STREAM" "UPGRADED-ARRAY-ELEMENT-TYPE"
 "UPGRADED-COMPLEX-PART-TYPE" "LOAD-TIME-EVAL" "REALP" "FDEFINITION" "NTH-VALUE" "DESTRUCTURING-BIND"
 "DEFINE-COMPILER-MACRO" "COMPILER-MACRO-FUNCTION" "COMPILER-MACROEXPAND" "COMPILER-MACROEXPAND-1"
 "VARIABLE-INFORMATION" "FUNCTION-INFORMATION" "DECLARATION-INFORMATION" "AUGMENT-ENVIRONMENT"
 "DEFINE-DECLARATION" "PARSE-MACRO" "ENCLOSE" "DECLAIM" "DYNAMIC-EXTENT" "GENSYM-COUNTER*"
 "DELETE-PACKAGE" "DEFPACKAGE" "WITH-PACKAGE-ITERATOR" "LEAST-POSITIVE-NORMALIZED-SHORT-FLOAT"
 "LEAST-POSITIVE-NORMALIZED-SINGLE-FLOAT" "LEAST-POSITIVE-NORMALIZED-DOUBLE-FLOAT"
 "LEAST-NEGATIVE-NORMALIZED-LONG-FLOAT" "LEAST-NEGATIVE-NORMALIZED-SHORT-FLOAT"
 "LEAST-NEGATIVE-NORMALIZED-SINGLE-FLOAT" "LEAST-NEGATIVE-NORMALIZED-DOUBLE-FLOAT"
 "LEAST-NEGATIVE-NORMALIZED-LONG-FLOAT" "COMPLEMENT" "MAP-INTO" "WITH-HASH-TABLE-ITERATOR"
 "HASH-TABLE-REHASH-SIZE" "HASH-TABLE-REHASH-THRESHOLD" "HASH-TABLE-SIZE" "HASH-TABLE-TEST"
 "ROW-MAJOR-AREF" "OPEN-STREAM-P" "BROADCAST-STREAM-STREAMS" "CONCATENATED-STREAM-STREAMS"
 "ECHO-STREAM-INPUT-STREAM" "ECHO-STREAM-OUTPUT-STREAM" "SYNONYM-STREAM-SYMBOL"
 "TWO-WAY-STREAM-INPUT-STREAM" "TWO-WAY-STREAM-OUTPUT-STREAM" "INTERACTIVE-STREAM-P"
 "STREAM-EXTERNAL-FORMAT" "READ-EVAL*" "READTABLE-CASE" "PRINT-READABLY*" "WITH-STANDARD-IO-SYNTAX"
 "PRINT-UNREADABLE-OBJECT" "WILD-PATHNAME-P" "PATHNAME-MATCH-P" "TRANSLATE-PATHNAME"
 "LOGICAL-PATHNAME" "TRANSLATE-LOGICAL-PATHNAME" "LOGICAL-PATHNAME-TRANSLATIONS"
 "LOAD-LOGICAL-PATHNAME-TRANSLATIONS" "COMPILE-FILE-PATHNAME" "FILE-STRING-LENGTH" "LOAD-PRINT*"
 "LOAD-PATHNAME*" "LOAD-TRUE-NAME*" "MAKE-LOAD-FORM" "MAKE-LOAD-FORM-SAVING-SLOTS"
 "COMPILE-VERBOSE" "COMPILE-PRINT*" "COMPILE-FILE-PATHNAME*" "COMPILE-FILE-TRUE-NAME*"
 "LOAD-TIME-VALUE" "FUNCTION-LAMBDA-EXPRESSION" "WITH-COMPILATION-UNIT"))
```

```
(CL:DEFPARAMETER OLDCLSYMS
 ' ("COMMON" "COMMONP" "STRING-CHAR" "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 SPLITCLSYMS ' ("LOCALLY" "IN-PACKAGE"))
```

```
(CL:DEFUN CRUNCH-FILES (FL)
 (CL:WHEN (AND FL (CL:SYMBOLP FL))
 (CL:SETQ FL (LIST FL)))
 (CL:DO-LIST (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 FLIP-CL (WHERE)
 (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)))))
```

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

;; 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))
```

;; We probably want to have COMMON-LISP-USER use XCL; this needs to be discussed

```
(CL:MAKE-PACKAGE "COMMON-LISP-USER" :USE '("COMMON-LISP"))
```

```
(LET ((WEIRDTAG (CONS NIL NIL))
      (OLDPROP (CONS NIL NIL))
      (UNSHAREDPROP (CONS NIL NIL))
      I)
  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*))
       (T ; VERY unlikely...
        (ERROR "Garbage on property list during LISP->COMMON-LISP import" (CONS I WEIRD?))))))
```

;; Hose out the new COMMON-LISP symbols

```
(CL:DOLIST (I NEWCLSYMS)
  (EXPORT (CL:INTERN 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:DO-EXTERNAL-SYMBOLS (I P)
    (IMPORT I *COMMON-LISP-PACKAGE*)
    (EXPORT I *COMMON-LISP-PACKAGE*)))
(FLIP-CL :COMMON-LISP))
```

FUNCTION INDEX

CREATE-CL-PACKAGE2 CRUNCH-FILES1 FLIP-CL1

VARIABLE INDEX

COMMON-LISP-PACKAGE ...1 NEWCLSYMS1 OLDCLSYMS1 SPLITCLSYMS1
