

File created: 7-Jan-88 15:36:10 {ERINYES}<LispUsers>Lyric>DUMpload.;4

changes to: (FNS \DUMP.COPY \DUMP.PURGE DUMPREAD)  
(VARS DUMploadCOMs)  
(PROPS (DUMPREAD ARGNAMEs))

previous date: 4-Dec-83 17:53:41 {ERINYES}<LISP>KOTO>LISPUSERS>DUMpload.;1

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

::  
:: Copyright (c) 1983, 1988 by Xerox Corporation. All rights reserved.

### (RPAQQ DUMploadCOMs

```
[ (FNS DUMPREAD \DUMP.PARSEDATE \DUMP.PARSENAME \DUMP.COPY \DUMP.PURGE)
(DECLARE%: EVAL@COMPILE DONTCOPY (CONSTANTS * DUMPTYPES))
(PROP ARGNAMEs DUMPREAD)
(PROP FILETYPE DUMpload)
(DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILEVARS (ADDVARS (NLAMA)
(NLAML)
(LAMA DUMPREAD]))
```

(DEFINEQ

### (DUMPREAD

```
[CL:LAMBDA (FILE &KEY ((:COPY-TO COPYFLG)
T)
((:ASK ASKFLG))
((:PRINT PRINTFLG)
T)
((:DATES DATEFLG)
T)
(COLLECT NIL COLLECTP)) ; Edited 7-Jan-88 14:28 by bvm:
(LET (STREAM OUTSTREAM)
(CL:UNWIND-PROTECT
(PROG ((*UPPER-CASE-FILE-NAMES* NIL)
(*STANDARD-OUTPUT* (GETSTREAM T 'OUTPUT))
SEEFLG RESULT NAME CREATIONDATE KEYWORD TYPE)
(IF (NULL COPYFLG)
ELSEIF (EQ COPYFLG T)
THEN (SETQ SEEFLG T)
(SETQ ASKFLG T)
(SETQ COPYFLG NIL)
ELSEIF (NOT COLLECTP)
THEN (SETQ COLLECT T))
[SETQ KEYWORD (COND
(SEEFLG (SETQ OUTSTREAM (GETSTREAM T 'OUTPUT))
"See")
(COPYFLG "Copy")
(T (SETQ ASKFLG NIL)
[SETQ STREAM (OPENSTREAM FILE 'INPUT 'OLD '((SEQUENTIAL T)
LP (SETQ TYPE (BIN STREAM))
TYPELP
(SELECTC TYPE
(\DUMP.END ; End of file, return accumulated names, if we've been collecting
(RETURN RESULT))
(\DUMP.NAME (SETQ NAME (\DUMP.PARSENAME STREAM)))
(\DUMP.DATE (SETQ CREATIONDATE (\DUMP.PARSEDATE STREAM))
; Date is in alto format
[SETQ CREATIONDATE (AND (OR DATEFLG COPYFLG)
(GDATE (ALTO.TO.LISP.DATE CREATIONDATE)
(DATEFORMAT TIME.ZONE)))
(\DUMP.ERROR (ERROR "Error block encountered in dump file" (FULLNAME STREAM)))
(\DUMP.DATA [COND
[ (COND
(NULL NAME)
(printout T "[Skipping nameless data...]" T)
NIL)
(ASKFLG (CL:Y-OR-N-P "~A ~A-@[ [~A~]? " KEYWORD NAME
CREATIONDATE))
(T [COND
(PRINTFLG (CL:FORMAT T "~A-@[ [~A~]" NAME CREATIONDATE)
(COND
(SEEFLG (printout T '%: T))
(COPYFLG (printout T "-> ")))
(T (TERPRI T)
T))
[SETQ TYPE (COND
[SEEFLG (PROG1 (\DUMP.COPY STREAM OUTSTREAM T)
(COND
(PRINTFLG (printout T .TAB0 0 T)))]
[ COPYFLG [SETQ OUTSTREAM
```

```

                                (OPENSTREAM (PACKFILENAME.STRING
                                                'DIRECTORY COPYFLG
                                                'BODY NAME)
                                'OUTPUT NIL NIL
                                `((SEQUENTIAL T)
                                   (CREATIONDATE ,CREATIONDATE])
                                (PROG1 (\DUMP.COPY STREAM OUTSTREAM)
                                       (SETQ NAME (CLOSEF OUTSTREAM))
                                       (SETQ OUTSTREAM NIL)
                                       (COND
                                        (PRINTFLG (printout T NAME T))))]
                                (T (\DUMP.PURGE STREAM]
                                (COND
                                 (COLLECT (push RESULT NAME]
                                 (T (SETQ TYPE (\DUMP.PURGE STREAM]
                                    (SETQ NAME (SETQ CREATIONDATE NIL))
                                    (GO TYPELP))
                                    (GO TYPELP))
                                 (ERROR "Bad Block Type in dump file" (FULLNAME STREAM))
                                 (GO LP))
;; Be sure to clean up this stream on the way out
(IF (AND OUTSTREAM (NEQ COPYFLG T))
    THEN
    (DELFILE (CLOSEF OUTSTREAM)) ; Flush partially written copy
    (AND STREAM (CLOSEF STREAM))))

```

```

(\DUMP.PARSEDATE
 [LAMBDA (INSTREAM)
  (PROG1 (\MAKENUMBER (\WIN INSTREAM)
                     (\WIN INSTREAM))
         (BIN INSTREAM)
         (BIN INSTREAM))]
 (* bvm%: " 2-Dec-83 18:22")
 ; Ignore two bytes

```

```

(\DUMP.PARSENAME
 [LAMBDA (INSTREAM)
  (BIN INSTREAM)
  (BIN INSTREAM)
  (PROG ((CHARS (bind CH until (EQ 0 (SETQ CH (BIN INSTREAM))) collect CH))
         RESULT)
         (SETQ RESULT (ALLOCSTRING (LENGTH CHARS)))
         (for CH in CHARS as I from 1 do (RPLCHARCODE RESULT I CH))
         (RETURN RESULT))]
 (* bvm%: " 2-Dec-83 18:26")
 ; Skip two bytes

```

```

(\DUMP.COPY
 [LAMBDA (INSTREAM OUTSTREAM TYP)
  ;; Copy data blocks byte-by-byte from INSTREAM to OUTSTREAM. Return the type of the next block. TYP is true if OUTSTREAM is the
  ;; terminal.
  (bind TYPE do (COPYBYTES INSTREAM OUTSTREAM (PROG1 (\WIN INSTREAM)
                                                       (BIN INSTREAM)
                                                       (BIN INSTREAM)))
              ; Length in bytes. Now skip checksum
              (BIN INSTREAM)
              (BIN INSTREAM)))
  repeatwhile (EQ (SETQ TYPE (BIN INSTREAM))
                 \DUMP.DATA)
  finally (RETURN TYPE)

```

```

(\DUMP.PURGE
 [LAMBDA (STREAM)
  (do (SETFILEPTR STREAM (+ (PROG1 (\WIN STREAM)
                                   (BIN STREAM)
                                   (BIN STREAM))
                            (GETFILEPTR STREAM)))
      repeatwhile (EQ (SETQ $$VAL (BIN STREAM))
                    \DUMP.DATA])

```

```

)
(DECLARE%: EVAL@COMPILE DONTCOPY
(RPAQQ DUMPTYPES ((\DUMP.DATE 251)
                 (\DUMP.END 252)
                 (\DUMP.ERROR 253)
                 (\DUMP.DATA 254)
                 (\DUMP.NAME 255)))

```

```

(DECLARE%: EVAL@COMPILE
(RPAQQ \DUMP.DATE 251)
(RPAQQ \DUMP.END 252)
(RPAQQ \DUMP.ERROR 253)

```

```
{MEDLEY}<lispusers>DUMpload.;1
(RPAQQ \DUMP.DATA 254)
(RPAQQ \DUMP.NAME 255)
(CONSTANTS (\DUMP.DATE 251)
           (\DUMP.END 252)
           (\DUMP.ERROR 253)
           (\DUMP.DATA 254)
           (\DUMP.NAME 255))
)
)
(PUTPROPS DUMPREAD ARGNAMES (FILE &KEY :ASK :COPY-TO :DATES :PRINT :COLLECT))
(PUTPROPS DUMpload FILETYPE :COMPILE-FILE)
(DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVERS
(ADDTOVAR NLAMA )
(ADDTOVAR NLAML )
(ADDTOVAR LAMA DUMPREAD)
)
(PUTPROPS DUMpload COPYRIGHT ("Xerox Corporation" 1983 1988))
```

---

**FUNCTION INDEX**

DUMPREAD .....1    \DUMP.COPY .....2    \DUMP.PARSEDATE ...2    \DUMP.PARSENAME ...2    \DUMP.PURGE .....2

---

**CONSTANT INDEX**

\DUMP.DATA .....3    \DUMP.DATE .....3    \DUMP.END .....3    \DUMP.ERROR .....3    \DUMP.NAME .....3

---

**PROPERTY INDEX**

DUMpload .....3    DUMPREAD .....3

---

**VARIABLE INDEX**

DUMPTYPES .....2

---