```
File created:    28-Jun-93 22:43:05  {DSK}<python>RELEASE>loops>2.0>src>LOOPSPRINT.;4

  changes to:    (ALISTS (PRETTYPRINTMACROS @)
                         (PRETTYPRINTMACROS _@)
                         (PRETTYPRINTMACROS _)
                         (PRETTYPRINTMACROS SEND)
                         (PRETTYPRINTMACROS _IV)
                         (PRETTYPRINTMACROS _Proto)
                         (PRETTYPRINTMACROS _New)
                         (PRETTYPRINTMACROS _Super)
                         (PRETTYPRINTMACROS _!)
                         (PRETTYPRINTMACROS _Process)
                         (PRETTYPRINTMACROS _Process!)
                         (PRETTYPRINTMACROS \LOOPS-PRINT-EVAL-MARKER))
                 (FNS PrintLoopsEvalAtLoad)
                 (VARS LOOPSPRINTCOMS)

previous date:   14-Jun-93 23:50:45  {DSK}<python>RELEASE>loops>2.0>src>LOOPSPRINT.;3

  Read Table:    INTERLISP

     Package:    INTERLISP

      Format:     XCCS


;;
;; Copyright (c) 1983, 1984, 1985, 1986, 1987, 1988, 1990, 1991, 1993 by Venue & Xerox Corporation.  All rights reserved.


(RPAQQ LOOPSPRINTCOMS
       [

;;; Functions for handling READ macros

        (DECLARE%: DONTCOPY (PROP MAKEFILE-ENVIRONMENT LOOPSPRINT)
               (PROP FILETYPE LOOPSPRINT))
        (INITVARS (LoopsReadMacroChar '%#))
        (GLOBALVARS LoopsReadMacroChar)
        (ADDVARS (GLOBALVARS LoopsReadMacroChar))
        (FNS HashMacro PPDefault SETALLSYNTAX $& $ $C)
        ;; Used to include the following: (COMS (P (SETALLSYNTAX LoopsReadMacroChar '(MACRO FIRST HashMacro))) (ADVICE
        ;; HPINITRDTBL)


;;; Printing of classes and instances

        (FNS PrintClass PrintInstance)
        (DECLARE%: EVAL@COMPILE DONTCOPY (EXPORT (FUNCTIONS TTYStream?))
               (FILES (LOADCOMP)
                      LOOPSDATATYPES))
        (P (DEFPRINT 'class 'PrintClass)
           (DEFPRINT 'instance 'PrintInstance))


;;; Prettyprinting functions

        ;; PPObj is Called by the PreetyPrinter on instances.  ObjectAlwaysPPFlag and ObjectDontPPFlag are used to determine if the PrettyPrint
        ;; macro should be applied recursively within the scope of a pretty printing operation.  ObjectDontPPFlag is bound to NIL in Object.PP to
        ;; supress recursive prettyprinting on files and if objects don't have special PP Methods of there own.
        (INITVARS (ObjectAlwaysPPFlag)
               (ObjectDontPPFlag)
               (WritingToEtherNet))
        (FUNCTIONS WITH.PP.OUTPUT)
        (SPECVARS ObjectAlwaysPPFlag ObjectDontPPFlag WritingToEtherNet)
        (FNS EntityAddress PrettyPrintLoopsForm PrettyPrintClass PrintDefInstances PrettyPrintInstance
             PrttyClasses PrttyInstances PrintLoopsEvalAtLoad PPC PPObj)
        (INITVARS (UnnamedInstances NIL))
        (GLOBALVARS UnnamedInstances)
        (ALISTS (PRETTYPRINTYPEMACROS instance)
               (PRETTYPRINTMACROS @ _@ _ SEND _IV _Proto _New _Super _! _Process _Process!
                      \LOOPS-PRINT-EVAL-MARKER))


;;; Set up macros so that references to classes and instances seen by HPRINT don't recur into their structure.

        (FNS HPRINTCLASS HPRINTINSTANCE HREADCLASS HREADINSTANCE)
        (ADDVARS (HPRINTMACROS (class . HPRINTCLASS)
                        (instance . HPRINTINSTANCE)))
        (DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVARS (ADDVARS (NLAMA $&)
                                                                              (NLAML $C $)
                                                                              (LAMA])


;;; Functions for handling READ macros

(DECLARE%: DONTCOPY
(PUTPROPS LOOPSPRINT MAKEFILE-ENVIRONMENT (:PACKAGE "IL" :READTABLE "INTERLISP" :BASE 10))

(PUTPROPS LOOPSPRINT FILETYPE :COMPILE-FILE)
)

(RPAQ? LoopsReadMacroChar '%#)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS LoopsReadMacroChar)
)

(ADDTOVAR GLOBALVARS LoopsReadMacroChar)

(DEFINEQ

(HashMacro
  [LAMBDA (fileHandle readTable)                                 (* AAA "24-Feb-86 18:41")
    (LET* [(char (PEEKC fileHandle))
           (charFn (GETPROP char 'HASHREADMACRO]
          (COND
             (charFn (if (NEQ char '%()
                         then (READCCODE fileHandle))
                    (APPLY* charFn fileHandle readTable))
             (T (SELCHARQ (CHCON1 char)
                     ("."                                        (* Following CommonLisp, %#. means read in the value of the
                                                                 next expression)
                          (READCCODE fileHandle)
                          (EVAL (READ fileHandle readTable)))
                     (-                                          (* a datatype dumped)
                        (READCCODE fileHandle)
                        (APPLY* (FUNCTION $A)
                               (HREAD fileHandle)
                               'DataType))
                     (SELECTQ (GETSYNTAX (CHCON1 char)
                                     readTable)
                         ((LEFTBRACKET RIGHTPAREN RIGHTBRACKET STRINGDELIM BREAKCHAR SEPRCHAR)
                              '%#)
                         (PACK* '%# (RATOM fileHandle readTable])

(PPDefault
  [LAMBDA (file)                                                 (* smL "31-Oct-86 10:56")

;;;

;;; Return the default stream to use for output

;;;

    (OR file (AND (TTYStream? NIL)
                  PPDefault])

(SETALLSYNTAX
  [LAMBDA (CHAR FORM)                                            (* smL "15-Oct-86 16:44")
    (DECLARE (GLOBALVARS CODERDTBL DEDITRDTBL EDITRDTBL FILERDTBL))
    (SETSYNTAX CHAR FORM)
    (SETSYNTAX CHAR FORM T)
    (SETSYNTAX CHAR FORM FILERDTBL)
    (SETSYNTAX CHAR FORM EDITRDTBL)                              ; In Lyric, the DEdit readtable is not defined...
    (AND (BOUNDP 'DEDITRDTBL)
         (SETSYNTAX CHAR FORM DEDITRDTBL))
    (SETSYNTAX CHAR FORM CODERDTBL])

($&
  [NLAMBDA nameOrUID                                             (* smL " 5-Sep-86 13:35")

          (* * Used to read back in printed instances - 
          "(className (name1 ... nameN uid) (ivName ivValue prop val ...)...)" is a posibility - 
          Archaic! Ugh! nameOrUID = "(mumble number ...)" is a pointer to location number!)

    (COND
       ([AND (LISTP (CADR nameOrUID))
             (FIXP (CAR (CADR nameOrUID)))
             (FIXP (CDR (CADR nameOrUID]
        (VAG (CADR nameOrUID)))
       (T (ModifyInstance nameOrUID])

($
  [NLAMBDA (name)                                                (* smL " 4-Apr-86 15:21")

          (* * Get object given name)
    (GetObjectRec name])

($C
  [NLAMBDA (name)                                                (* smL " 4-Apr-86 15:25")
          (* * Get class, given the name. If the class doesn't exist, create it.
          -
          This is an attempt to take care of forward references in files. See the fn PrintClass to see how classes print out.)

    (OR (GetClassRec name)
        (NewClass name])
)

;; Used to include the following: (COMS (P (SETALLSYNTAX LoopsReadMacroChar '(MACRO FIRST HashMacro))) (ADVICE HPINITRDTBL)

;;; Printing of classes and instances

(DEFINEQ

(PrintClass
  [LAMBDA (classRec file)                                        ; Edited 15-Aug-90 12:49 by jds

;;; This is called by the LISP PRINT function when it sees an instance of the data type class

    (CONS (CONCAT LoopsReadMacroChar ",")
          (LIST (if (OR (NULL file)
                        (TTYStream? file)
                        (DISPLAYSTREAMP file))
                    then                                         ; for local human consumption
                         '$
                  else                                           ; Dumping to a file
                       '$C)
                (fetch (class className) of classRec])

(PrintInstance
  [LAMBDA (instanceRec file)                                     ; Edited  4-Jun-87 18:33 by smL

;;; This is called by the LISP PRINT function when it sees an instance of the data type instance

    (DECLARE (SPECVARS WritingToEtherNet FileEntities OutInstances))
    (PROG (entity uid uidString)                                 ; (NOT (EQP FILERDTBL (GETREADTABLE)))
          [COND
             ((OR WritingToEtherNet OutInstances)                ; Force items being saved to have a uid
              (SETQ uid (UID instanceRec)))
             ((OR (NULL file)
                  (TTYStream? file)
                  (DISPLAYSTREAMP file))                         ; for local human consumption
              (RETURN (_ instanceRec PrintOn file)))
             (T                                                  ; this case is for a non display stream file
                (SETQ uid (UID instanceRec]
          (RETURN (COND
                     (OutInstances                               ; This case when dumping instances to a file with INSTANCES
                            (AND (_ instanceRec SaveInstance? file (CAR OutInstances))
                                 (TCONC OutInstances instanceRec))
                                                                 ; Recursive dump of instances referred to
                            (LIST (CONCAT LoopsReadMacroChar ",")
                                  '$&
                                  (ClassName instanceRec)
                                  uid))
                     (T

;;; Internal or ethernetPrinting printing * WritingToEtherNet set when writing to EtherNet

                        (LIST (CONCAT LoopsReadMacroChar ",")
                              '$&
                              (ClassName instanceRec)
                              uid])
)

(DECLARE%: EVAL@COMPILE DONTCOPY

;; FOLLOWING DEFINITIONS EXPORTED


(DEFMACRO TTYStream? (s)
   `(EQ (GETSTREAM T 'OUTPUT)
        (GETSTREAM ,s 'OUTPUT)))

;; END EXPORTED DEFINITIONS

(FILESLOAD (LOADCOMP)
       LOOPSDATATYPES)
```

```
)

(DEFPRINT 'class 'PrintClass)

(DEFPRINT 'instance 'PrintInstance)


;;; Prettyprinting functions

;; PPObj is Called by the PreetyPrinter on instances.  ObjectAlwaysPPFlag and ObjectDontPPFlag are used to determine if the PrettyPrint macro should
;; be applied recursively within the scope of a pretty printing operation.  ObjectDontPPFlag is bound to NIL in Object.PP to supress recursive
;; prettyprinting on files and if objects don't have special PP Methods of there own.

(RPAQ? ObjectAlwaysPPFlag )

(RPAQ? ObjectDontPPFlag )

(RPAQ? WritingToEtherNet )


(DEFMACRO WITH.PP.OUTPUT (stream &BODY forms)
   [Once-Only (stream)
          `(RESETLST
               [if (WINDOWP ,stream)
                   then [RESETSAVE NIL (LIST (FUNCTION WINDOWPROP)
                                             ,stream
                                             'PROCESS
                                             (WINDOWPROP ,stream 'PROCESS]
                        (RESETSAVE (TTYDISPLAYSTREAM ,stream))
                        (if (NOT (TTY.PROCESSP))
                            then (RESETSAVE (TTY.PROCESS (THIS.PROCESS]
               ,@forms)])

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(SPECVARS ObjectAlwaysPPFlag ObjectDontPPFlag WritingToEtherNet)
)

(DEFINEQ

(EntityAddress
  [LAMBDA (E)                                                    (* dgb%: "11-NOV-82 02:29")
    (LOC E])

(PrettyPrintLoopsForm
  [LAMBDA (form)                                                 (* smL " 2-May-86 16:30")

          (* * Pretty-print the Loops form)

    (PRIN1 "(")
    (PRIN2 (CAR form))
    (SPACES 1)
    (PRINTDEF (CDR form)
           (POSITION)
           T T)
    (PRIN1 ")")
    NIL])

(PrettyPrintClass
  [LAMBDA (className file)                                       ; Edited 25-Jun-87 13:57 by smL

;;; Called to prettyPrint a class definition on a file by the FILEPKGTYPE CLASSES and by the PP: method in Class

    (DECLARE (SPECVARS className))
    (LET ((class (GetClassRec className)))
         (COND
            ((NULL class)
             (HELPCHECK className " is not defined as a class.
                    Type OK to ignore this class and go on."))
            (T (_ class FileOut file)
               className])

(PrintDefInstances
  [LAMBDA (instances file)                                       (* dgb%: " 5-OCT-82 10:50")

          (* * Does the INSTANCES command for the LISP file pkg.)

    (TERPRI file)
    (PRIN1 "[DEFINSTANCES " file)
    (for inst instRec in instances do (SETQ instRec (GetObjectRec inst))
                                      (PRIN2 (CONS (ClassName instRec)
                                                   (GetObjectNames instRec))
                                             file)
                                      (SPACES 1 file))
    (PRIN1 "]" file)
    (TERPRI file)
    (TERPRI file)
    instances])


(PrettyPrintInstance
  [LAMBDA (self file)                                             ; Edited  9-Mar-88 11:55 by jrb:
                                                                  (* PrettyPrint an self definition on file.)
    (if self
        then (if (CL:SYMBOLP self)
                 then (SETQ self ($! self)))
             (_ self FileOut file))
    self])


(PrttyClasses
  [LAMBDA (classes)                                               (* smL "15-Aug-86 10:21")

          (* * Used by CLASSES command to print out rereadable versions of class definitions on files)

    (PRINT (CONS 'DEFCLASSES classes))
    (TERPRI)
    (for className in classes do (PrettyPrintClass className (OUTPUT])


(PrttyInstances
  [LAMBDA (instances)                                             (* smL "15-Aug-86 10:23")

          (* * Prettyprint a list of named instances to a file. OutInstances is bound here, although usually a free variable when printing
          an instance)

    (LET [(OutInstances (LCONC NIL (for name obj in instances collect obj
                                     when (SETQ obj (OR (GetObjectRec name)
                                                        (NULL (HELPCHECK name " not defined as an instance.
                                                                     Type OK to ignore and go on."]
         (DECLARE (SPECVARS OutInstances))
         (for obj in (CAR OutInstances) do (PrettyPrintInstance obj (OUTPUT))
                                           (TERPRI)
                                           (TERPRI])


(PrintLoopsEvalAtLoad
  [LAMBDA (FORM)                                                  ; Edited 28-Jun-93 16:20 by sybalsky:mv:envos
    (PRIN1 "#,")
    (PRIN2 (CADR FORM))
    NIL])


(PPC
  [LAMBDA (className)                                             (* smL "11-Apr-86 17:11")

          (* * PrettyPrint the class)

    (LET ((class (GetClassRec className)))
         (if class
             then (_ class PP)
           else (ERROR "No such class" className])


(PPObj
  [LAMBDA (instance)                                              ; Edited  2-Dec-87 16:22 by jrb:

;;; Called by the PrettyPrinter on instances.

    [LET ((FOO *STANDARD-OUTPUT*))
         (COND
            (ObjectAlwaysPPFlag                                   ; Don't suppress PPing
                    (_ instance PP *STANDARD-OUTPUT*))
            (ObjectDontPPFlag                                     ; Suppress PPing
                    (PRIN2 instance))
            ((EQ *STANDARD-OUTPUT* *TERMINAL-IO*)                 ; Displaying for the user, so PP it
             (_ instance PP *STANDARD-OUTPUT*))
            (T                                                    ; No reason to PP to a file, so dont
               (PRIN2 instance]
    NIL])

)

(RPAQ? UnnamedInstances NIL)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS UnnamedInstances)
)

(ADDTOVAR PRETTYPRINTYPEMACROS (instance . PPObj))
```

```
(ADDTOVAR PRETTYPRINTMACROS (@ . PrettyPrintLoopsForm)
                            (_@ . PrettyPrintLoopsForm)
                            (_ . PrettyPrintLoopsForm)
                            (SEND . PrettyPrintLoopsForm)
                            (_IV . PrettyPrintLoopsForm)
                            (_Proto . PrettyPrintLoopsForm)
                            (_New . PrettyPrintLoopsForm)
                            (_Super . PrettyPrintLoopsForm)
                            (_! . PrettyPrintLoopsForm)
                            (_Process . PrettyPrintLoopsForm)
                            (_Process! . PrettyPrintLoopsForm)
                            NIL)

;;; Set up macros so that references to classes and instances seen by HPRINT don't recur into their structure.

(DEFINEQ

(HPRINTCLASS
  [LAMBDA (EXPR FILE)                                          (* dgb%: "18-MAR-83 16:35")

            (* Used by HPRINT to print out an expression which will be read back in as a class)

    (PRIN1 '(HREADCLASS)
           FILE)
    (PRIN2 (ClassName EXPR])

(HPRINTINSTANCE
  [LAMBDA (EXPR FILE)                                          (* dgb%: "18-MAR-83 16:36")

            (* Used by HPRINT to print out an expression referring to an instance, provided instance is dumped eleewhere)

    (PRIN1 '(HREADINSTANCE)
           FILE)
    (PRIN2 (UID EXPR])

(HREADCLASS
  [LAMBDA (FILE)                                               (* dgb%: "13-DEC-82 15:13")
    (GetObjectRec (HREAD FILE])

(HREADINSTANCE
  [LAMBDA (FILE)                                               (* dgb%: " 8-Oct-84 13:28")
    (GetObjectRec (HREAD FILE])
)

(ADDTOVAR HPRINTMACROS (class . HPRINTCLASS)
                       (instance . HPRINTINSTANCE))

(DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVARS

(ADDTOVAR NLAMA $&)

(ADDTOVAR NLAML $C $)

(ADDTOVAR LAMA )
)

(PUTPROPS LOOPSPRINT COPYRIGHT ("Venue & Xerox Corporation" 1983 1984 1985 1986 1987 1988 1990 1991 1993))
```

{MEDLEY}<loops>system>LOOPSPRINT.;1   30-Jun-2024 08:54:28

-- Listed on 30-Jun-2024 09:23:31 --

FUNCTION INDEX

VARIABLE INDEX

MACRO INDEX

PROPERTY INDEX