File created: 31-Oct-2023 16:16:39 **{WMEDLEY}<sources>LLSYMBOL.;2**

edit by: rmk

previous date: 11-Jun-90 17:56:50 {WMEDLEY}<sources>LLSYMBOL.;1

Read Table: XCL

Package: LISP

Format: XCCS

```
(IL:RPAQQ IL:LLSYMBOLCOMS
          (;; Symbol functions.
           ;; SET ,  BOUNDP and REMPROP are the same as and shared with Interlisp-D
           ;; Where is the optimizer for CL:GETF?
           (IL:FUNCTIONS MAKUNBOUND SYMBOL-NAME SYMBOL-VALUE GET GETF GET-PROPERTIES)
           (IL:DECLARE\: IL:DOCOPY IL:DONTEVAL@LOAD IL:DONTEVAL@COMPILE (IL:P (IL:MOVD 'IL:GETPROPLIST
                                                                                'SYMBOL-PLIST)))
           (IL:FUNCTIONS FBOUNDP FMAKUNBOUND SYMBOL-FUNCTION IL:SETF-SYMBOL-FUNCTION)
           (IL:COMS
                  ;; GENSYM Code
                  (IL:VARIABLES *GENSYM-COUNTER* *GENSYM-PREFIX* *GENTEMP-COUNTER*)
                  (IL:FUNCTIONS GENSYM GENTEMP))
           (IL:FUNCTIONS COPY-SYMBOL IL:MAKE-KEYWORD KEYWORDP)
           (IL:PROP (IL:FILETYPE IL:MAKEFILE-ENVIRONMENT)
                  IL:LLSYMBOL)))
```

;; Symbol functions.

;; SET , BOUNDP and REMPROP are the same as and shared with Interlisp-D

;; Where is the optimizer for CL:GETF?

```
(DEFUN MAKUNBOUND (SYMBOL)
    ;; Make a symbol unbound.
    ;;  Unbound symbols are set to IL:NOBIND
    (IF (CONSTANTP SYMBOL)
        (PROGN (XCL::SET-CONSTANTP SYMBOL NIL)
               (PROCLAIM `(SPECIAL ,SYMBOL))))
    (SET SYMBOL 'IL:NOBIND)
    SYMBOL)
```

```
(DEFUN SYMBOL-NAME (SYMBOL)
    (IF (SYMBOLP SYMBOL)
        ;; Make a read-only string header displaced to the pname base
        (IL:%MAKE-ONED-ARRAY (IL:|ffetch| (IL:LITATOM IL:PNAMELENGTH) IL:|of| SYMBOL)
               'STRING-CHAR NIL (IL:|ffetch| (IL:LITATOM IL:FATPNAMEP) IL:|of| SYMBOL)
               T NIL (IL:|ffetch| (IL:LITATOM IL:PNAMEBASE) IL:|of| SYMBOL)
               1)
        (ERROR 'CONDITIONS:SIMPLE-TYPE-ERROR :EXPECTED-TYPE 'SYMBOL :CULPRIT SYMBOL)))
```

```
(DEFUN SYMBOL-VALUE (SYMBOL)
    ;; Like EVALV, but must give error if unbound - uses fact that \eval has an opcode which hooks into free variable microcode
    (IF (SYMBOLP SYMBOL)
        (IL:\\EVAL SYMBOL)
        (ERROR 'CONDITIONS:SIMPLE-TYPE-ERROR :EXPECTED-TYPE 'SYMBOL :CULPRIT SYMBOL)))
```

```
(DEFUN GET (SYMBOL INDICATOR &OPTIONAL (DEFAULT NIL))
    ;; Look on the property list of SYMBOL for the specified INDICATOR. If this is found, return the associated value, else return DEFAULT.
    (GETF (IL:GETPROPLIST SYMBOL)
          INDICATOR DEFAULT))
```

```
(DEFUN GETF (PLACE INDICATOR &OPTIONAL (DEFAULT NIL))
    ;; Searches the property list stored in Place for an indicator EQ to Indicator.  If one is found, the corresponding value is returned, else the Default is
    ;; returned.
    (DO ((PLIST PLACE (CDDR PLIST)))
        ((NULL PLIST)
         DEFAULT)
      (WHEN (EQ (CAR PLIST)
                INDICATOR)
          (IF (NOT (CONSP (CDR PLIST)))
              (ERROR "Malformed property list: ~s" PLACE)
              (RETURN (CADR PLIST))))))
```

```
(DEFUN GET-PROPERTIES (PLACE INDICATOR-LIST)
   (DO ((PLIST PLACE (CDDR PLIST)))
       ((NULL PLIST)
        (VALUES NIL NIL NIL))
     (WHEN (MEMBER (CAR PLIST)
                   INDICATOR-LIST :TEST #'EQ)
       (IF (NOT (CONSP (CDR PLIST)))
           (ERROR "Malformed p-list: ~s" PLACE)
           (RETURN (VALUES (CAR PLIST)
                           (CADR PLIST)
                           PLIST))))))

(IL:DECLARE\: IL:DOCOPY IL:DONTEVAL@LOAD IL:DONTEVAL@COMPILE

(IL:MOVD 'IL:GETPROPLIST 'SYMBOL-PLIST)
)


(DEFUN FBOUNDP (FN)
   (AND (SYMBOLP FN)
        (OR (IL:ARGTYPE FN)
            (MACRO-FUNCTION FN)
            (SPECIAL-FORM-P FN))
        T))


(DEFUN FMAKUNBOUND (SYMBOL)
   ;; Has lots of special knowledge of prop list names
   (SETF (SYMBOL-FUNCTION SYMBOL)
         NIL)
   (SETF (MACRO-FUNCTION SYMBOL)
         NIL)
   (REMPROP SYMBOL 'IL:SPECIAL-FORM)
   (REMPROP SYMBOL 'IL:CODE)
   (REMPROP SYMBOL 'IL:EXPR)
   SYMBOL)


(DEFUN SYMBOL-FUNCTION (SYMBOL &AUX (DEF (IL:GETD SYMBOL)))
   ;; this function is preformance-critical, as it is used in the compilation of #'FOO => (CL:SYMBOL-FUNCTION 'FOO). Thus, this definition checks for
   ;; the GETD definition first. It might even be reasonable to open-code the GETD here. It *is* unreasonable to call MACRO-FUNCTION and
   ;; SPECIAL-FORM-P first.
   (COND
      (DEF)                                                    ; GETD returned non-NIL
      ((SETQ DEF (MACRO-FUNCTION SYMBOL))                      ; Return something representing the macro's implementation.
       (CONS ':MACRO DEF))
      ((SETQ DEF (SPECIAL-FORM-P SYMBOL))                      ; Return something representing the special-form's
                                                               ; implementation.
       (CONS ':SPECIAL-FORM DEF))
      (T (ERROR 'XCL:UNDEFINED-FUNCTION :NAME SYMBOL))))


(DEFUN IL:SETF-SYMBOL-FUNCTION (SYMBOL DEFINITION)
   ;; NOTE: If you change this, be sure to change the undoable version on CMLUNDO!
   ;;  inverse of SYMBOL-FUNCTION
   (IL:VIRGINFN SYMBOL T)
   (COND
      ((CONSP DEFINITION)
       ;; Either it's a LAMBDA form or one of the special lists put together by SYMBOL-FUNCTION for macros and special forms.
       (CASE (CAR DEFINITION)
           (:MACRO (SETF (MACRO-FUNCTION SYMBOL)
                         (CDR DEFINITION)))
           (:SPECIAL-FORM (SETF (GET SYMBOL 'IL:SPECIAL-FORM)
                                (CDR DEFINITION)))
           (T (IL:PUTD SYMBOL DEFINITION T))))
      ;; If it's (SETF (SYMBOL-FUNCTION 'FOO) 'BAR) then we give FOO the same definition as BAR.  This isn't quite like Lucid and Symbolics, but
      ;; it will do for now.
      ((AND (SYMBOLP DEFINITION)
            (NOT (NULL DEFINITION)))
       (IL:PUTD SYMBOL (IL:GETD DEFINITION)
              T))
      ;; It's probably a compiled-code object or an interpreted closure.  In any case, go ahead and put it in there; if it's illegal, we'll find out when we try
      ;; to apply it.
      (T (IL:PUTD SYMBOL DEFINITION T)))
   ;; (SETF (SYMBOL-FUNCTION ...) ...) is supposed to remove macro definitions.  We only remove the ones that could come from DEFMACRO.
   (UNLESS (OR (NULL DEFINITION)
               (AND (CONSP DEFINITION)
                    (EQ (CAR DEFINITION)
                        :MACRO)))
```

```
       (REMPROP SYMBOL 'IL:MACRO-FN))
   DEFINITION)
```

;; GENSYM Code

```
(DEFVAR *GENSYM-COUNTER* 0)

(DEFVAR *GENSYM-PREFIX* "G")

(DEFVAR *GENTEMP-COUNTER* 0)

(DEFUN GENSYM (&OPTIONAL (X NIL X-P))
   (IF X-P
       (ETYPECASE X
           (STRING (SETQ *GENSYM-PREFIX* X))
           (INTEGER (SETQ *GENSYM-COUNTER* X))))
   (PROG1 (MAKE-SYMBOL (CONCATENATE 'STRING *GENSYM-PREFIX* (IL:MKSTRING *GENSYM-COUNTER*)))
       (SETQ *GENSYM-COUNTER* (1+ *GENSYM-COUNTER*))))

(DEFUN GENTEMP (&OPTIONAL (PREFIX "T")
                          (PACKAGE *PACKAGE*))
   ;; *gentemp-counter* holds a good guess for the suffix
   (LET ((COUNTER *GENTEMP-COUNTER*)
         NAMESTRING)                                                    ; Use IL:MKSTRING rather than princ-to-string, since
                                                                        ; princ-to-string occurs late in the loadup
        (LOOP (SETQ NAMESTRING (CONCATENATE 'STRING PREFIX (IL:MKSTRING COUNTER)))
              (WHEN (NULL (FIND-SYMBOL NAMESTRING PACKAGE))
                  (SETQ *GENTEMP-COUNTER* (1+ COUNTER))
                  (RETURN (INTERN NAMESTRING PACKAGE)))
              (SETQ COUNTER (1+ COUNTER)))))

(DEFUN COPY-SYMBOL (SYM &OPTIONAL COPY-PROPS)
   (LET ((NEW-SYM (MAKE-SYMBOL (SYMBOL-NAME SYM))))
        (WHEN COPY-PROPS
            (IF (BOUNDP SYM)
                (SETF (SYMBOL-VALUE NEW-SYM)
                      (SYMBOL-VALUE SYM)))
            (IF (FBOUNDP SYM)
                (SETF (SYMBOL-FUNCTION NEW-SYM)
                      (SYMBOL-FUNCTION SYM)))
            (SETF (SYMBOL-PLIST NEW-SYM)
                  (COPY-LIST (SYMBOL-PLIST SYM))))
        NEW-SYM))

(DEFUN IL:MAKE-KEYWORD (SYMBOL)
   (DECLARE (SPECIAL IL:*KEYWORD-PACKAGE*))
   (VALUES (INTERN (SYMBOL-NAME SYMBOL)
                   IL:*KEYWORD-PACKAGE*)))

(DEFUN KEYWORDP (OBJECT)
   (AND (SYMBOLP OBJECT)
        (EQ (SYMBOL-PACKAGE OBJECT)
            IL:*KEYWORD-PACKAGE*)))

(IL:PUTPROPS IL:LLSYMBOL IL:FILETYPE COMPILE-FILE)

(IL:PUTPROPS IL:LLSYMBOL IL:MAKEFILE-ENVIRONMENT (:READTABLE "XCL" :PACKAGE "LISP"))
```

FUNCTION INDEX

VARIABLE INDEX

PROPERTY INDEX