File created: 4-Jan-93 18:09:50 **{DSK}<python>lde>lispcore>sources>DEFSTRUCT-RUN-TIME.;2**

previous date: 16-May-90 15:32:24 {DSK}<python>lde>lispcore>sources>DEFSTRUCT-RUN-TIME.;1

Read Table: XCL

Package: LISP

Format: XCCS


```
(IL:RPAQQ IL:DEFSTRUCT-RUN-TIMECOMS
          ((IL:COMS
                 ;; Remembering parsed structures
                 (IL:VARIABLES *PARSED-DEFSTRUCTS*)
                 (IL:FUNCTIONS PARSED-STRUCTURE SET-PARSED-STRUCTURE)
                 (IL:SETFS PARSED-STRUCTURE))
           (IL:COMS
                 ;; Declaring storage for structures
                 (IL:FUNCTIONS SI::%STRUCTURE-DECLARE-DATATYPE)
                 (IL:DECLARE\: IL:DONTEVAL@LOAD IL:DOCOPY
                        ;; This defines the root of the defstruct type hierarchy.
                        (IL:P (IL:\\ASSIGNDATATYPE1 'STRUCTURE-OBJECT NIL 0))))
           (IL:COMS
                 ;; Support for setf expansions etc
                 (IL:VARIABLES *DEFSTRUCT-INFO-CACHE*)
                 (IL:FUNCTIONS ESTABLISH-SETFS-AND-OPTIMIZERS ESTABLISH-PREDICATE)
                 (IL:FUNCTIONS GET-PS-FROM-ACCESSOR GET-PS-FROM-PREDICATE GET-SLOT-DESCRIPTOR-FROM-PS)
                 (IL:FUNCTIONS CACHE-SETF-INFO))
           (IL:COMS
                 ;; defstruct IO
                 (IL:VARIABLES XCL:*PRINT-STRUCTURE*)
                 (IL:FUNCTIONS PRINT-STRUCTURE-INSTANCE DEFAULT-STRUCTURE-PRINTER STRUCTURE-SLOT-NAMES)
                 ;; For reading
                 (IL:FUNCTIONS IL:CREATE-STRUCTURE STRUCTURE-CONSTRUCTOR))
           (IL:PROP (IL:FILETYPE IL:MAKEFILE-ENVIRONMENT)
                  IL:DEFSTRUCT-RUN-TIME)
           (IL:DECLARE\: IL:DONTEVAL@LOAD IL:DOEVAL@COMPILE IL:DONTCOPY IL:COMPILERVARS (IL:ADDVARS (IL:NLAMA)
                                                                                                  (IL:NLAML)
                                                                                                  (IL:LAMA)))))
```

;; Remembering parsed structures

```
(DEFVAR *PARSED-DEFSTRUCTS* (IL:HASHARRAY 100)
                            ;; All declared structures
                            )
```

```
(DEFMACRO PARSED-STRUCTURE (NAME &OPTIONAL (NO-ERROR NIL))
    ;; Returns the parsed-structure corresponding to name
    (COND
       (NO-ERROR `(IL:GETHASH ,NAME *PARSED-DEFSTRUCTS*))
       (T `(OR (IL:GETHASH ,NAME *PARSED-DEFSTRUCTS*)
               (ERROR "~s is not a defined structure" ,NAME)))))
```

```
(DEFUN SET-PARSED-STRUCTURE (NAME PS &OPTIONAL (EXTRA NIL EXTRA-P))
    ;; SETF method for CL::PARSED-STRUCTURE. Extra arg is because CL::PARSED-STRUCTURE takes an optional, which we ignore here, but that
    ;; pushes the new value over one.
    (WHEN EXTRA-P (SETQ PS EXTRA))
    (IL:PUTHASH NAME PS *PARSED-DEFSTRUCTS*))
```

```
(DEFSETF PARSED-STRUCTURE SET-PARSED-STRUCTURE)
```

;; Declaring storage for structures

```
(DEFUN SI::%STRUCTURE-DECLARE-DATATYPE (NAME FIELD-SPECIFICATIONS FIELD-DESCRIPTORS WORD-LENGTH
                                        SUPERTYPE)
```

;;; analagous to declare-datatype, but does not prepend the supers descriptors. You must include all descs.

;;; N.B. descriptions and specs are for ALL slots, not just local-slots.

;; field-specifications is a list of the form '(pointer pointer (bits 3) (bits 5) word fixp). See p. 8.21 IRM

```
    ;; field-descriptors is the list returned from translate.datatype when given the above FIELD-SPECIFICATIONS.  They are legal to pass to fetchfield.
    ;; word-length is the car of the result of translate.datatype.
    ;; supertype is the typename of the supertype.
    (IF (NOT (AND (SYMBOLP NAME)
                  (IL:SMALLPOSP WORD-LENGTH)))
        (ERROR "Illegal arguments: ~s ~s" NAME WORD-LENGTH))
    (LET ((REFERENCE-COUNTED-POINTERS (MAPCAN #'(LAMBDA (DESCRIPTOR)
                                                  (CASE (CADDR DESCRIPTOR)
                                                      ((IL:POINTER IL:FULLPOINTER) (LIST (CADR DESCRIPTOR)))
))
                                              FIELD-DESCRIPTORS)))
          (MULTIPLE-VALUE-BIND (TYPE-NUMBER REDECLARED?)
              (IL:\\ASSIGNDATATYPE1 NAME FIELD-DESCRIPTORS WORD-LENGTH FIELD-SPECIFICATIONS
                     REFERENCE-COUNTED-POINTERS SUPERTYPE)
            ;; set the magic global to the allocated type number
            (IL:SETTOPVAL (IL:\\TYPEGLOBALVARIABLE NAME T)
                   TYPE-NUMBER)
            (VALUES FIELD-DESCRIPTORS REDECLARED?))))

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

(IL:\\ASSIGNDATATYPE1 'STRUCTURE-OBJECT NIL 0)
)

;; Support for setf expansions etc

(DEFVAR *DEFSTRUCT-INFO-CACHE* (IL:HASHARRAY 100)
                               ;; Used to cache slots and predicates

                               )

(DEFUN ESTABLISH-SETFS-AND-OPTIMIZERS (PS-NAME)
    ;; Caches shared setf expanders and accessor optimizers where appropriate
    (LET* ((PS (PARSED-STRUCTURE PS-NAME))
           (INLINE (PS-INLINE PS)))
          (MAPC #'(LAMBDA (SLOT)
                          ;; function-defining-form decides whether or not the accessors should be defun, definline, etc.
                          (LET ((ACCESSOR (PSLOT-ACCESSOR SLOT)))
                               (WHEN ACCESSOR
                                   (REMHASH ACCESSOR *DEFSTRUCT-INFO-CACHE*)
                                   (IF (NOT (PSLOT-READ-ONLY SLOT))
                                       ;; install the setf method expander that is shared for all accessors
                                       (SET-SHARED-SETF-INVERSE ACCESSOR 'DEFSTRUCT-SHARED-SETF-EXPANDER))
                                   (COND
                                      ((EQ INLINE :ONLY)
                                       (SETF (MACRO-FUNCTION ACCESSOR)
                                             'DEFSTRUCT-SHARED-ACCESSOR-OPTIMIZER))
                                      ((MEMBER :ACCESSOR INLINE :TEST #'EQ)
                                       (SETF (GET ACCESSOR 'COMPILER:OPTIMIZER-LIST)
                                             (LIST 'DEFSTRUCT-SHARED-ACCESSOR-OPTIMIZER)))
                                      (T (REMPROP ACCESSOR 'COMPILER:OPTIMIZER-LIST))))))
                  (PS-ALL-SLOTS PS))))

(DEFUN ESTABLISH-PREDICATE (PS-NAME)
    ;; Establishes a shared a shared optimizer for a defstruct predicate
    (LET* ((PS (PARSED-STRUCTURE PS-NAME))
           (PREDICATE (PS-PREDICATE PS)))
          (REMHASH PREDICATE *DEFSTRUCT-INFO-CACHE*)
          (IF (EQ (PS-INLINE PS)
                  :ONLY)
              (SETF (MACRO-FUNCTION PREDICATE)
                    'DEFSTRUCT-SHARED-PREDICATE-OPTIMIZER)
              (SETF (GET PREDICATE 'COMPILER:OPTIMIZER-LIST)
                    (LIST 'DEFSTRUCT-SHARED-PREDICATE-OPTIMIZER)))))

(DEFUN GET-PS-FROM-ACCESSOR (ACCESSOR &OPTIONAL (NO-ERROR-P NIL))
   (OR (CATCH 'FIND-PS
          (MAPHASH #'(LAMBDA (KEY VALUE)
                             (DOLIST (SLOT (PS-ALL-SLOTS VALUE)
                                           NIL)
                                 (IF (EQ ACCESSOR (PSLOT-ACCESSOR SLOT))
                                     (THROW 'FIND-PS VALUE))))
                   *PARSED-DEFSTRUCTS*))
       (IF (NULL NO-ERROR-P)
           (ERROR "No such slot: ~s" ACCESSOR))))
```

```
(DEFUN GET-PS-FROM-PREDICATE (PREDICATE &OPTIONAL (NO-ERROR-P NIL))
   (OR (CATCH 'FIND-PS
           (MAPHASH #'(LAMBDA (KEY VALUE)
                            (IF (EQ PREDICATE (PS-PREDICATE VALUE))
                                (THROW 'FIND-PS VALUE)))
                    *PARSED-DEFSTRUCTS*))
       (IF (NULL NO-ERROR-P)
           (ERROR "No such predicate: ~s" PREDICATE))))

(DEFUN GET-SLOT-DESCRIPTOR-FROM-PS (ACCESSOR PS &OPTIONAL (NO-ERROR-P NIL))
   (OR (DOLIST (SLOT (PS-ALL-SLOTS PS)
                     NIL)
           (IF (EQ ACCESSOR (PSLOT-ACCESSOR SLOT))
               (RETURN SLOT)))
       (IF (NULL NO-ERROR-P)
           (ERROR "No such slot: ~s" ACCESSOR))))

(DEFUN CACHE-SETF-INFO (PS-NAME)
   ;; For compatability with the old defstruct
   (LET ((PS (PARSED-STRUCTURE PS-NAME)))
        (MAPC #'(LAMBDA (SLOT)
                        ;; function-defining-form decides whether or not the accessors should be defun, definline, etc.
                        (LET ((ACCESSOR (PSLOT-ACCESSOR SLOT)))
                             (WHEN ACCESSOR
                                 (REMHASH ACCESSOR *DEFSTRUCT-INFO-CACHE*)
                                 (IF (NOT (PSLOT-READ-ONLY SLOT))
                                     ;; install the setf method expander that is shared for all accessors
                                     (SET-SHARED-SETF-INVERSE ACCESSOR 'DEFSTRUCT-SHARED-SETF-EXPANDER)))))
              (PS-ALL-SLOTS PS))))
```

;; defstruct IO

```
(DEFVAR XCL:*PRINT-STRUCTURE* T
   "Flag indicating whether the contents of structures are to be printed.")

(DEFUN PRINT-STRUCTURE-INSTANCE (OBJECT STREAM DEPTH)
   ;; Looks up the print function for the structure instance and calls it
   (FUNCALL (OR (PS-PRINT-FUNCTION (PARSED-STRUCTURE (TYPE-OF OBJECT)))
                %DEFAULT-PRINT-FUNCTION)
          OBJECT STREAM (OR DEPTH 0)))

(DEFUN DEFAULT-STRUCTURE-PRINTER (STRUC STREAM &OPTIONAL (PRINT-LEVEL 0))
   (IF (NOT XCL:*PRINT-STRUCTURE*)
       (IL:\\PRINT-USING-ADDRESS STRUC STREAM 0)
       (LET
        ((TYPE (IL:TYPENAME STRUC))
         LABEL
         (FIRST-TIME? T))
        (WHEN IL:*PRINT-CIRCLE-HASHTABLE*
            ;; only true if *print-circle* is true and the structure is circular.
            (MULTIPLE-VALUE-SETQ (LABEL FIRST-TIME?)
                   (IL:PRINT-CIRCLE-LOOKUP STRUC)))
        ;; (cl:format t "label: ~S firsttime ~S~%" label first-time?)
        (WHEN LABEL
            ;; this guy needs to be flagged for circle-printing
            (IL:PRIN3 LABEL STREAM))
        (WHEN (OR (NOT LABEL)
                  FIRST-TIME?)
            (LET ((*PRINT-LEVEL* (AND *PRINT-LEVEL* (1- *PRINT-LEVEL*))))
                 (IF (OR (AND *PRINT-LEVEL* (<= *PRINT-LEVEL* PRINT-LEVEL))
                         (AND *PRINT-LENGTH* (<= *PRINT-LENGTH* 0)))
                     (IL:\\ELIDE.PRINT.ELEMENT STREAM)
                     (LET ((LENGTHSOFAR (IF *PRINT-LENGTH* 0)))
                          (IL:\\OUTCHAR STREAM (IL:|fetch| (READTABLEP IL:HASHMACROCHAR) IL:|of| *READTABLE*))
                          (WRITE-STRING "S(" STREAM)
                          (IF (AND LENGTHSOFAR (> (INCF LENGTHSOFAR)
                                                  *PRINT-LENGTH*))
                              (IL:\\ELIDE.PRINT.TAIL STREAM T)
                              (PROGN (IF *PRINT-ESCAPE*
                                         (PRIN1 TYPE STREAM)
                                         (PRINC TYPE STREAM))
                                     (DO ((FIELD (STRUCTURE-SLOT-NAMES TYPE)
                                                 (CDR FIELD))
```

```
                                        (DESCRIPTOR (IL:GETDESCRIPTORS TYPE)
                                               (CDR DESCRIPTOR)))
                                      ((NULL FIELD))
                                    (WHEN (EQ (CAR FIELD)
                                              'SI::--STRUCTURE-DUMMY-SLOT--)
                                          (GO SKIP))
                                    (IL:\\OUTCHAR STREAM (IL:CONSTANT (CHAR-CODE #\Space)))
                                    (IF (AND LENGTHSOFAR (> (INCF LENGTHSOFAR)
                                                            *PRINT-LENGTH*))
                                        (PROGN (IL:\\ELIDE.PRINT.TAIL STREAM T)
                                               (RETURN NIL))
                                        (PROGN (PRINC (CAR FIELD)
                                                      STREAM)
                                               (IF (AND LENGTHSOFAR (> (INCF LENGTHSOFAR)
                                                                       *PRINT-LENGTH*))
                                                   (PROGN (IL:\\ELIDE.PRINT.TAIL STREAM T)
                                                          (RETURN NIL))
                                                   (PROGN (IL:\\OUTCHAR STREAM (IL:CONSTANT (CHAR-CODE
                                                                                            #\Space))
                                                                        )
                                                          (IL:\\PRINDATUM (IL:FETCHFIELD (CAR DESCRIPTOR)
                                                                                         STRUC)
                                                                 STREAM
                                                                 (1+ PRINT-LEVEL))))))
                                    SKIP)))
                              (WRITE-STRING ")" STREAM)))))
       T)))

(DEFUN STRUCTURE-SLOT-NAMES (STRUCTURE-NAME &OPTIONAL (DONT-COPY NIL))
   (LET* ((PS (PARSED-STRUCTURE STRUCTURE-NAME))
          NAMES)
         (SETQ NAMES (PS-ALL-SLOT-NAMES PS))
         (IF DONT-COPY
             NAMES
             (COPY-LIST NAMES))))
```

;; For reading

```
(DEFUN IL:CREATE-STRUCTURE (STRUCTURE-FORM)
   (APPLY (STRUCTURE-CONSTRUCTOR (CAR STRUCTURE-FORM))
          (XCL:WITH-COLLECTION (DO ((TAIL (CDR STRUCTURE-FORM)
                                          (CDDR TAIL)))
                                   ((NULL TAIL))
                                 (XCL:COLLECT (IL:MAKE-KEYWORD (CAR TAIL)))
                                 (XCL:COLLECT (CADR TAIL))))))

(DEFUN STRUCTURE-CONSTRUCTOR (STRUCTURE-NAME)
   (OR (GET STRUCTURE-NAME 'IL:STRUCTURE-CONSTRUCTOR)
       (LET* ((PS (PARSED-STRUCTURE STRUCTURE-NAME))
              (CONSTRUCTOR (PS-STANDARD-CONSTRUCTOR PS)))
             (OR CONSTRUCTOR (ERROR "~S is a structure with no standard constructor." (PS-NAME PS))))))

(IL:PUTPROPS IL:DEFSTRUCT-RUN-TIME IL:FILETYPE COMPILE-FILE)

(IL:PUTPROPS IL:DEFSTRUCT-RUN-TIME IL:MAKEFILE-ENVIRONMENT (:READTABLE "XCL" :PACKAGE "LISP"))

(IL:DECLARE\: IL:DONTEVAL@LOAD IL:DOEVAL@COMPILE IL:DONTCOPY IL:COMPILERVARS

(IL:ADDTOVAR IL:NLAMA )

(IL:ADDTOVAR IL:NLAML )

(IL:ADDTOVAR IL:LAMA )
)
```

FUNCTION INDEX

VARIABLE INDEX

PROPERTY INDEX

SETF INDEX

MACRO INDEX