File created: 16-May-90 18:22:09 **{DSK}<usr>local>lde>lispcore>sources>IMPLICIT-KEY-HASH.;2**

changes to: (IL:VARS IL:IMPLICIT-KEY-HASHCOMS)

previous date: 24-Jan-88 16:54:16 {DSK}<usr>local>lde>lispcore>sources>IMPLICIT-KEY-HASH.;1

Read Table: XCL

Package: XEROX-COMMON-LISP

Format: XCCS


```
(IL:RPAQQ IL:IMPLICIT-KEY-HASHCOMS ((IL:STRUCTURES IMPLICIT-KEY-HASH-TABLE)
                                    (IL:VARIABLES *DELETED-IMPLICIT-HASH-SLOT*)
                                    (IL:FUNCTIONS MAKE-IMPLICIT-KEY-HASH-TABLE GET-IMPLICIT-KEY-HASH
                                           PUT-IMPLICIT-KEY-HASH IMPLICIT-KEY-MAP-HASH CLEAR-IMPLICIT-KEY-HASH
                                           IMPLICIT-KEY-REHASH ADJUST-IMPLICIT-KEY-HASH)
                                    (IL:FUNCTIONS GET-IK-VALUE PUT-IK-VALUE GET-IK-KEY REPROBE 16BIT-+)
                                    (IL:SETFS GET-IMPLICIT-KEY-HASH GET-IK-VALUE)
                                    (FILE-ENVIRONMENTS "IMPLICIT-KEY-HASH")))

(DEFSTRUCT (IMPLICIT-KEY-HASH-TABLE (:CONC-NAME IK-HASH-)
                                    (:CONSTRUCTOR %MAKE-IK-HASH-TABLE)
                                    (:COPIER NIL)
                                    (:PREDICATE NIL)
                                    (:FAST-ACCESSORS T))
   BASE
   (LAST-INDEX 0 :TYPE (UNSIGNED-BYTE 16))
   (NUM-SLOTS 0 :TYPE (UNSIGNED-BYTE 16))
   (NUM-KEYS 0 :TYPE (UNSIGNED-BYTE 16))
   (NULL-SLOTS 0 :TYPE (UNSIGNED-BYTE 16))
   KEY-ACCESSOR)

(DEFVAR *DELETED-IMPLICIT-HASH-SLOT* "Unique string")

(DEFUN MAKE-IMPLICIT-KEY-HASH-TABLE (&OPTIONAL (MIN-KEYS 20)
                                               (KEY-ACCESSOR :FIRST))
   ;; Does eq hashing
   (LET* ((NUM-SLOTS
               ;; num-slots is always a power of two
               (DO ((IDEAL-SIZE (ASH (TRUNCATE (1- MIN-KEYS)
                                               3)
                                     2))
                    (I 8 (+ I I)))
                   ((> I IDEAL-SIZE)
                    I)))
          (LOGICAL-SLOTS
               ;; 75% of NUM-SLOTS
               (+ (ASH NUM-SLOTS -1)
                  (ASH NUM-SLOTS -2))))
         (%MAKE-IK-HASH-TABLE :BASE (IL:\\ALLOCBLOCK NUM-SLOTS IL:PTRBLOCK.GCT)
                :LAST-INDEX
                (1- NUM-SLOTS)
                :NUM-SLOTS LOGICAL-SLOTS :NUM-KEYS 0 :NULL-SLOTS LOGICAL-SLOTS :KEY-ACCESSOR KEY-ACCESSOR)))

(DEFUN GET-IMPLICIT-KEY-HASH (ITEM IK-HASH-TABLE)
   (IF (NOT (TYPEP IK-HASH-TABLE 'IMPLICIT-KEY-HASH-TABLE))
       (ERROR "Not an implicit key hash table: ~s" IK-HASH-TABLE))
   ;; Do first index outside of loop, so don't have to do setup on fast case
   (PROG* ((BITS (IL:\\EQHASHINGBITS ITEM))
           (LIMIT (IK-HASH-LAST-INDEX IK-HASH-TABLE))
           (INDEX (LOGAND BITS LIMIT))
           (BASE (IK-HASH-BASE IK-HASH-TABLE))
           (VALUE (GET-IK-VALUE BASE INDEX))
           (KEY-ACCESSOR (IK-HASH-KEY-ACCESSOR IK-HASH-TABLE))
           (DELETED-INDICATOR *DELETED-IMPLICIT-HASH-SLOT*)
           REPROBE)
          (COND
             ((EQ VALUE DELETED-INDICATOR)
              ;; Deleted slot -- continue

              )
             (VALUE ;; Slot is occupied
                    (IF (EQ ITEM (GET-IK-KEY VALUE KEY-ACCESSOR))
                        (GO FOUND)
                        ;; Else try again
```

```
                    ))
          (T ;; Null slot
             (RETURN NIL)))
    ;; Compute reprobe interval
        (SETQ REPROBE (REPROBE BITS LIMIT))
  LP
    ;; Since table size is a power of two, any wraparound in the IPLUS16 will be consistent with the LOGAND
        (SETQ INDEX (LOGAND (16BIT-+ INDEX REPROBE)
                            LIMIT))
        (SETQ VALUE (GET-IK-VALUE BASE INDEX))
        (COND
           ((EQ VALUE DELETED-INDICATOR)
            ;; Deleted slot -- continue

            )
           (VALUE ;; Slot is occupied
                  (IF (EQ ITEM (GET-IK-KEY VALUE KEY-ACCESSOR))
                      (GO FOUND)
                      ;; Else try again

                      ))
           (T ;; Null slot
              (RETURN NIL)))
        (GO LP)
   FOUND
        (RETURN VALUE)))

(DEFUN PUT-IMPLICIT-KEY-HASH (ITEM IK-HASH-TABLE NEW-VALUE)
   ;; Puthash nil is equivalent to remhash for these tables
   (IF (NOT (TYPEP IK-HASH-TABLE 'IMPLICIT-KEY-HASH-TABLE))
       (ERROR "Not an implicit key hash table: ~s" IK-HASH-TABLE))
   (PROG ((BITS (IL:\\EQHASHINGBITS ITEM))
          (LIMIT (IK-HASH-LAST-INDEX IK-HASH-TABLE))
          (BASE (IK-HASH-BASE IK-HASH-TABLE))
          (KEY-ACCESSOR (IK-HASH-KEY-ACCESSOR IK-HASH-TABLE))
          (DELETED-INDICATOR *DELETED-IMPLICIT-HASH-SLOT*)
          INDEX VALUE FIRST-INDEX REPROBE DELETED-SLOT-INDEX)
    PHTOP

    ;; Handle first probe outside loop in case it wins
         (SETQ INDEX (LOGAND BITS LIMIT))
         (SETQ VALUE (GET-IK-VALUE BASE INDEX))
         (COND
            ((EQ VALUE DELETED-INDICATOR)
             ;; Found a deleted slot -- continue lookup
             (SETQ DELETED-SLOT-INDEX INDEX))
            (VALUE ;; Slot is occupied
                   (IF (EQ ITEM (GET-IK-KEY VALUE KEY-ACCESSOR))
                       (GO FOUND)
                       ;; else try again

                       ))
            (T ;; Empty slot
               (GO ADDNEWENTRY)))
    ;; Chase reprobe chain
         (SETQ FIRST-INDEX INDEX)
         (SETQ REPROBE (REPROBE BITS LIMIT))
    LP   (SETQ INDEX (LOGAND (16BIT-+ INDEX REPROBE)
                             LIMIT))
         (WHEN (EQ INDEX FIRST-INDEX)
             ;; We don't allow full occupancy, so if we get to the beginning without finding an empty slot, we must have found a deleted one
             (SETQ INDEX (OR DELETED-SLOT-INDEX (ERROR "No vacant slot in Implicit key hash table: ~s"
                                                       IK-HASH-TABLE)))
             (GO ADDNEWENTRY))
         (SETQ VALUE (GET-IK-VALUE BASE INDEX))
         (COND
            ((EQ VALUE DELETED-INDICATOR)
             ;; Found a deleted slot -- continue lookup
             (SETQ DELETED-SLOT-INDEX INDEX))
            (VALUE ;; Slot is occupied
```

```
                (IF (EQ ITEM (GET-IK-KEY VALUE KEY-ACCESSOR))
                    (GO FOUND)
                    ;; else try again
                    ))
           (T ;; Empty slot
              (IF DELETED-SLOT-INDEX (SETQ INDEX DELETED-SLOT-INDEX))
              (GO ADDNEWENTRY)))
        (GO LP)
     FOUND
        (IL:UNINTERRUPTABLY
            (SETF (GET-IK-VALUE BASE INDEX)
                  (OR NEW-VALUE DELETED-INDICATOR))
            (IF (NULL NEW-VALUE)
                (DECF (IK-HASH-NUM-KEYS IK-HASH-TABLE))))
        (RETURN NEW-VALUE)
     ADDNEWENTRY

     ;; Didn't find this item in table.
        (IF (NULL NEW-VALUE)
            ;; Nothing to add
            (RETURN NEW-VALUE))
        (WHEN (EQ 0 (IK-HASH-NULL-SLOTS IK-HASH-TABLE))
            (IL:UNINTERRUPTABLY
                (LET* ((NUM-SLOTS (IK-HASH-NUM-SLOTS IK-HASH-TABLE))
                       (NEW-ARRAY (IMPLICIT-KEY-REHASH IK-HASH-TABLE (MAKE-IMPLICIT-KEY-HASH-TABLE
                                                                      ;; 1.5 times NUM-SLOTS
                                                                      (+ NUM-SLOTS (ASH (1+ NUM-SLOTS)
                                                                                        -1))
                                                                      KEY-ACCESSOR))))
                    (SETQ IK-HASH-TABLE (ADJUST-IMPLICIT-KEY-HASH IK-HASH-TABLE NEW-ARRAY))
                    ;; update local state
                    (SETQ LIMIT (IK-HASH-LAST-INDEX IK-HASH-TABLE))
                    (SETQ BASE (IK-HASH-BASE IK-HASH-TABLE))
                    ;; Non-NIL DELSLOT is an index into the old array
                    (SETQ DELETED-SLOT-INDEX NIL)))
            (GO PHTOP))
        (IL:UNINTERRUPTABLY
            (IF (NOT (EQ INDEX DELETED-SLOT-INDEX))
                (DECF (IK-HASH-NULL-SLOTS IK-HASH-TABLE)))
            (INCF (IK-HASH-NUM-KEYS IK-HASH-TABLE))
            (SETF (GET-IK-VALUE BASE INDEX)
                  NEW-VALUE))
        (RETURN NEW-VALUE)))

(DEFUN IMPLICIT-KEY-MAP-HASH (FN IK-HASH-TABLE)
   (IF (NOT (TYPEP IK-HASH-TABLE 'IMPLICIT-KEY-HASH-TABLE))
       (ERROR "Not an implicit key hash table: ~s" IK-HASH-TABLE))
   (LET* ((BASE (IK-HASH-BASE IK-HASH-TABLE))
          (LAST-INDEX (1+ (IK-HASH-LAST-INDEX IK-HASH-TABLE)))
          (LAST-ADDRESS (IL:\\ADDBASE (IL:\\ADDBASE BASE LAST-INDEX)
                                LAST-INDEX))
          (KEY-ACCESSOR (IK-HASH-KEY-ACCESSOR IK-HASH-TABLE))
          (NULL-SLOT-INDICATOR *DELETED-IMPLICIT-HASH-SLOT*)
          VALUE)
         (LOOP (IF (EQ BASE LAST-ADDRESS)
                   (RETURN NIL))
               (SETQ VALUE (IL:\\GETBASEPTR BASE 0))
               (IF (AND VALUE (NOT (EQ VALUE NULL-SLOT-INDICATOR)))
                   (FUNCALL FN VALUE (GET-IK-KEY VALUE KEY-ACCESSOR)))
               (SETQ BASE (IL:\\ADDBASE BASE 2)))))

(DEFUN CLEAR-IMPLICIT-KEY-HASH (IK-HASH-TABLE)
   (IF (NOT (TYPEP IK-HASH-TABLE 'IMPLICIT-KEY-HASH-TABLE))
       (ERROR "Not an implicit key hash table: ~s" IK-HASH-TABLE))
   (LET* ((BASE (IK-HASH-BASE IK-HASH-TABLE))
          (LAST-INDEX (1+ (IK-HASH-LAST-INDEX IK-HASH-TABLE)))
          (LAST-ADDRESS (IL:\\ADDBASE (IL:\\ADDBASE BASE LAST-INDEX)
                                LAST-INDEX)))
         (IL:UNINTERRUPTABLY
             (LOOP (IF (EQ BASE LAST-ADDRESS)
                       (RETURN NIL))
                   (IL:\\RPLPTR BASE 0 NIL)
                   (SETQ BASE (IL:\\ADDBASE BASE 2)))
             (SETF (IK-HASH-NULL-SLOTS IK-HASH-TABLE)
                   (IK-HASH-NUM-SLOTS IK-HASH-TABLE))
             (SETF (IK-HASH-NUM-KEYS IK-HASH-TABLE)
                   0))
         IK-HASH-TABLE))
```

```
(DEFUN IMPLICIT-KEY-REHASH (FROM-TABLE TO-TABLE)
   (IF (NOT (TYPEP FROM-TABLE 'IMPLICIT-KEY-HASH-TABLE))
       (ERROR "Not an implicit key hash table: ~s" FROM-TABLE))
   (CLEAR-IMPLICIT-KEY-HASH TO-TABLE)
   (IF (NOT (< (IK-HASH-NUM-SLOTS FROM-TABLE)
               (IK-HASH-NUM-SLOTS TO-TABLE)))
       (ERROR "To table too small: ~s" TO-TABLE))
   (LET* ((FROM-BASE (IK-HASH-BASE FROM-TABLE))
          (FROM-LAST-INDEX (1+ (IK-HASH-LAST-INDEX FROM-TABLE)))
          (LAST-ADDRESS (IL:\\ADDBASE (IL:\\ADDBASE FROM-BASE FROM-LAST-INDEX)
                                      FROM-LAST-INDEX))
          (KEY-ACCESSOR (IK-HASH-KEY-ACCESSOR FROM-TABLE))
          (NULL-SLOT-INDICATOR *DELETED-IMPLICIT-HASH-SLOT*)
          VALUE)
         (LOOP (IF (EQ FROM-BASE LAST-ADDRESS)
                   (RETURN TO-TABLE))
               (SETQ VALUE (IL:\\GETBASEPTR FROM-BASE 0))
               (IF (AND VALUE (NOT (EQ VALUE NULL-SLOT-INDICATOR)))
                   (PUT-IMPLICIT-KEY-HASH (GET-IK-KEY VALUE KEY-ACCESSOR)
                          TO-TABLE VALUE))
               (SETQ FROM-BASE (IL:\\ADDBASE FROM-BASE 2)))))

(DEFUN ADJUST-IMPLICIT-KEY-HASH (OLD-IK-TABLE NEW-IK-TABLE)
   (IL:UNINTERRUPTABLY
       (SETF (IK-HASH-BASE OLD-IK-TABLE)
             (IK-HASH-BASE NEW-IK-TABLE))
       (SETF (IK-HASH-LAST-INDEX OLD-IK-TABLE)
             (IK-HASH-LAST-INDEX NEW-IK-TABLE))
       (SETF (IK-HASH-NUM-SLOTS OLD-IK-TABLE)
             (IK-HASH-NUM-SLOTS NEW-IK-TABLE))
       (SETF (IK-HASH-NUM-KEYS OLD-IK-TABLE)
             (IK-HASH-NUM-KEYS NEW-IK-TABLE))
       (SETF (IK-HASH-NULL-SLOTS OLD-IK-TABLE)
             (IK-HASH-NULL-SLOTS NEW-IK-TABLE))
       (SETF (IK-HASH-KEY-ACCESSOR OLD-IK-TABLE)
             (IK-HASH-KEY-ACCESSOR NEW-IK-TABLE)))
   OLD-IK-TABLE)

(DEFMACRO GET-IK-VALUE (BASE INDEX)
   `(IL:\\GETBASEPTR ,BASE (IL:LLSH ,INDEX 1)))

(DEFMACRO PUT-IK-VALUE (BASE INDEX NEW-VALUE)
   `(IL:\\RPLPTR ,BASE (IL:LLSH ,INDEX 1)
          ,NEW-VALUE))

(DEFMACRO GET-IK-KEY (VALUE KEY-ACCESSOR)
   (ONCE-ONLY (VALUE KEY-ACCESSOR)
          `(IF (EQ KEY-ACCESSOR :FIRST)
               (IL:\\GETBASEPTR ,VALUE 0)
               (FUNCALL ,KEY-ACCESSOR ,VALUE))))

(DEFMACRO REPROBE (BITS LAST-INDEX)
   `(LOGIOR (LOGAND (LOGXOR ,BITS (IL:LRSH ,BITS 8))
                    (MIN 63 ,LAST-INDEX))
            1))

(DEFMACRO 16BIT-+ (A B)
   `(IL:\\LOLOC (IL:\\ADDBASE ,A ,B)))

(DEFSETF GET-IMPLICIT-KEY-HASH PUT-IMPLICIT-KEY-HASH)

(DEFSETF GET-IK-VALUE PUT-IK-VALUE)

(DEFINE-FILE-ENVIRONMENT "IMPLICIT-KEY-HASH" :READTABLE "XCL"
   :PACKAGE "XCL"
   :COMPILER :COMPILE-FILE)

(IL:PUTPROPS IL:IMPLICIT-KEY-HASH IL:COPYRIGHT ("Venue & Xerox Corporation" 1987 1988 1990))
```

FUNCTION INDEX

MACRO INDEX

SETF INDEX

FILE-ENVIRONMENT INDEX

STRUCTURE INDEX

VARIABLE INDEX