

File created: 17-Apr-2023 08:04:06 {DSK}<home>larry>il>medley>sources>LLDATATYPE.;2

edit by: lmm

changes to: (VARS LLDATATYPECOMS)
(FNS \SET.STORAGE.STATE \MAPMDS)

previous date: 28-Jun-99 16:57:50 {DSK}<home>larry>il>medley>sources>LLDATATYPE.;1

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

::
:: Copyright (c) 1982-1995, 1999 by VENUE, Oakland, CA.

(RPAQQ **LLDATATYPECOMS**

```
( (COMS                                     ; Because we use the UNLESSINew macro in this file, we need
                                         ; it when compiling.

(DECLARE%: EVAL@COMPILE DONTCOPY (FILES (SOURCE)
                                     RENAMEMACROS)))

(COMS                                     ; Storage management
(FNS NTYPX \TYPENAME.UFN \TYPE.UFN \ALLOCMDSPAGE \ALLOCPAGEBLOCK \ALLOCVIRTUALPAGEBLOCK \MAPMDS
 \CHECKFORSTORAGEFULL \DOSTORAGEFULLINTERRUPT \SET.STORAGE.STATE \SETTYPENAME
 \ADVANCE.STORAGE.STATE \NEW2PAGE \MAKEMDSENTRY \INITMDSPAGE \ASSIGNDATATYPE1
 \RESOLVE.TYPENUMBER \TYPENUMBERFROMNAME CREATECELL \CREATECELL)

;; For NEW_STORAGE option was set in Maiko, then \maiko.set.storage.state is active

SP
(INITVARS (CROSSCOMPILING)
          (ASSIGNDATATYPE.ASKUSERWAIT 300)
          (\STORAGEFULLSTATE)
          (\STORAGEFULL))
(GLOBALVARS CROSSCOMPILING \STORAGEFULLSTATE \STORAGEFULL \SYSTEMCACHEVARS \NxtArrayPage)
(SPECVARS ASSIGNDATATYPE.ASKUSERWAIT))

(COMS                                     ; fetch and replace
(FNS FETCHFIELD REPLACEFIELD BOXCOUNT CONSCOUNT \DTEST \TYPECHECK \DTEST.UFN \INSTANCEP.UFN
 \INSTANCE-P \TYPECHECK.UFN GETDESCRIPTORS GETSUPERTYPE GETFIELDSPECS NCREATE NCREATE2
 REPLACEFIELDVAL PUTBASEPTRX /REPLACEFIELD TYPENAME TYPENAMEP \TYPENAMEFROMNUMBER \BLOCKDATA
 USERDATATYPES DATATYPEP DATATYPES)
(P (MOVD? 'FETCHFIELD 'FFETCHFIELD NIL T)
 (MOVD? 'REPLACEFIELD 'FREPLACEFIELD NIL T)
 (MOVD? 'REPLACEFIELDVAL 'FREPLACEFIELDVAL NIL T))
(OPTIMIZERS TYPENAMEP \INSTANCE-P))

[COMS                                     ; STORAGE
(FNS STORAGE STORAGE.LEFT \STORAGE.TYPE \STLINP \STMDSTYPE \STMDS.APPROX \STORAGE.HUNKTYPE)
(DECLARE%: DONTCOPY (RECORDS HUNKSTAT))
(INITVARS (STORAGE.ARRAYSIZES '(4 16 64 256 1024 4096 16384 NIL])
(DECLARE%: (EXPORT (OPTIMIZERS PUTBASEPTRX)
                  (CONSTANTS \SMALLP \FIXP \FLOATP \LITATOM \LISTP \ARRAYP \STACKP \VMEMPAGEP \STREAM
 \NEW-ATOM)

;; This is the list of datatypes whos type #s must be known to microcode or to C.
;; It is used in \SETUP.HUNK.TYPENUMBERS (in LLARRAYELT) to create the list INITIALDTCNTENTS for
;; INITDATATYPES.
;; Changes to this list need to be reflected in C and maybe in microcode.

(VARS \BUILT-IN-SYSTEM-TYPES))

DONTCOPY
(EXPORT (RECORDS DTD)
        (MACROS \GETDTD)
        (OPTIMIZERS \TYPENAME.UFN)
        (CONSTANTS \GUARDSTORAGEFULL \GUARD1STORAGEFULL)
        (GLOBALVARS \NxtMDSPage \LeastMDSPage \SecondArrayPage \SecondMDSPage \MDSFREELISTPAGE
 \MaxSysTypeNum \MaxTypeNumber \STORAGEFULL \INTERRUPTSTATE \PENDINGINTERRUPT))
(CONSTANTS * STORAGEFULLSTATES))

[COMS                                     ; for MAKEINIT
(FNS CREATEMDSTYPETABLE INITDATATYPES INITDATATYPENAMES)
(DECLARE%: DONTCOPY
 (ADDVARS (INITVALUES (\NxtMDSPage \FirstMDSPage)
                    (\LeastMDSPage \FirstMDSPage)
                    (\SecondMDSPage \DefaultSecondMDSPage)
                    (\SecondArrayPage \DefaultSecondArrayPage)
                    (\MDSFREELISTPAGE)
                    (\MaxSysTypeNum 0)
                    (\MaxTypeNumber))
 (INITPTRS (\FINALIZATION.FUNCTIONS))
 (INewCOMS (FNS NTYPX \ALLOCMDSPAGE \MAKEMDSENTRY \INITMDSPAGE \ASSIGNDATATYPE1
 \TYPENUMBERFROMNAME \CREATECELL \NEW2PAGE)
 (FNS CREATEMDSTYPETABLE INITDATATYPES INITDATATYPENAMES)
 (VARS \BUILT-IN-SYSTEM-TYPES))
 (RDCOMS (FNS NTYPX TYPENAME \TYPENAMEFROMNUMBER))
 (RDVALS (\MaxTypeNumber))
 (RD.SUBFNS (\ARRAYTYPENAME LAMBDA (X)
```

```

' ARRAYP))
(EXPANDMACROFNS \GETDTD PUTBASEPTRX REPLACEFIELD FETCHFIELD \GETBITS \PUTBITS
 \TESTBITS GETBASEBITS PUTBASEBITS FFETCHFIELD FREPLACEFIELD FREPLACEFIELDVAL
 REPLACEFIELDVAL NCREATE)
(MKI.SUBFNS (\GCDISABLED . NIL)
 (CREATECELL . I.\CREATECELL)
 (\CHECKFORSTORAGEFULL . NIL)))
EVAL@COMPILE
(ADDVARS (DONTCOMPILEFNS CREATEMDSTYPETABLE INITDATATYPES INITDATATYPENAMES]
(LOCALVARS . T)
(PROP FILETYPE LLDATATYPE)
(DECLARE%: EVAL@COMPILE DONTCOPY (FILES (LOADCOMP)
DTDECLARE))))

```

:: Because we use the UNLESSINew macro in this file, we need it when compiling.

```

(DECLARE%: EVAL@COMPILE DONTCOPY
(FILESLoad (SOURCE)
RENAMEMACROS)
)

```

:: Storage management

```
(DEFINEQ
```

(NTYPX

```

[LAMBDA (X)
(* JonL "10-Nov-84 21:51"
; usually done in microcode --- this def used by MAKEINIT too
(LOGAND [\GETBASE \MDSTypeTable (FOLDLO (fetch (POINTER PAGE#) of X)
(CONSTANT (IQUOTIENT \MDSIncrement WORDSPERPAGE]
\TT.TYPEMASK]))

```

(\TYPEMASK.UFN

```

[LAMBDA (X N)
(* Imm "22-Mar-85 16:37")
(COND
((NEQ 0 (LOGAND N (LRSH [\GETBASE \MDSTypeTable (FOLDLO (fetch (POINTER PAGE#) of X)
(CONSTANT (IQUOTIENT \MDSIncrement WORDSPERPAGE]
8))))
X]))

```

(\TYPEP.UFN

```

[LAMBDA (X N)
(* Imm "22-Mar-85 10:07")
(COND
((EQ (NTYPX X)
N)
X]))

```

(\ALLOCMDSPAGE

```

[LAMBDA (TYP)
; Edited 25-Apr-94 10:39 by jds
(PROG (VP VPTR)
BEG [COND
[(SETQ VP \MDSFREELISTPAGE)
(SETQ VPTR (create POINTER
PAGE# _ VP))
(PROG ((NXT (\GETBASEPTR VPTR 0)))
(COND
((AND NXT (NOT (SMALLP NXT)))
(\MP.ERROR \MP.BADMDSFREELIST "MDS Free Page link bad. ^N to continue"
(PROG1 \MDSFREELISTPAGE (SETQ \MDSFREELISTPAGE)))
(GO BEG))
(T (SETQ \MDSFREELISTPAGE NXT])
(T (\CHECKFORSTORAGEFULL)
(SETQ VP \NxtMDSPage)
[UNLESSINew (\PUTBASEFIXP \NxtMDSPage 0 (IDIFFERENCE VP (FOLDLO \MDSIncrement PAGESPERSEGMENT)))
(SETQ \NxtMDSPage (IDIFFERENCE VP (FOLDLO \MDSIncrement PAGESPERSEGMENT]
; Allocates 2 MDS pages
(SETQ VPTR (create POINTER
PAGE# _ VP))
(\NEWPAGE (\ADDBASE (\NEWPAGE VPTR)
WORDSPERPAGE]
(\MAKEMDSENTRY VP TYP)
(RETURN VPTR]))

```

(\ALLOCPAGEBLOCK

```

[LAMBDA (NPAGES)
(* ejs%: "11-Aug-85 15:02")
(UNINTERRUPTABLY
:: Allocates a contiguous chunk of NPAGES pages. Currently there is no provision for giving them back.
(LET ((RESULT (\ALLOCVIRTUALPAGEBLOCK NPAGES)))
(COND
(RESULT (to NPAGES as (BASE _ RESULT) by (\ADDBASE BASE WORDSPERPAGE) do

```

; Allocate the new pages. Leave them having the default type,
; namely type 0, don't refcnt

(\NEWPAGE BASE)

RESULT))))))

(\ALLOCVIRTUALPAGEBLOCK

; Edited 4-Jan-93 02:03 by jds

[LAMBDA (NPAGES)
(UNINTERRUPTABLY

;; Allocates a contiguous chunk of NPAGES virtual pages. Does not actually allocate the memory, just removes them from the set of pages
;; that the allocator will use

```
(PROG (FIRSTPAGE)
  (COND
    ([ILEQ (IPLUS \NxtArrayPage \GUARDSTORAGEFULL)
      (SETQ FIRSTPAGE (IDIFFERENCE (IPLUS \NxtMDSPage \PagesPerMDSUnit)
        (SETQ NPAGES (CEIL NPAGES \PagesPerMDSUnit)
          ; Plenty of space
        \PUTBASEFIXP \NxtMDSPage 0 (IDIFFERENCE FIRSTPAGE \PagesPerMDSUnit)))
      [(NEQ (OR \STORAGEFULLSTATE (\SET.STORAGE.STATE)
        \SFS.SWITCHABLE)
      (COND
        ([AND (EQ \STORAGEFULLSTATE \SFS.ARRAYSWITCHED)
          (ILESSP (IPLUS \SecondArrayPage \GUARDSTORAGEFULL)
            (SETQ FIRSTPAGE (IDIFFERENCE (IPLUS \SecondMDSPage \PagesPerMDSUnit)
              NPAGES])
          ; Arrays have been switched, but we're still allocating MDS in low space. Just bump the variable that says where MDS in
          ; high space will start
          (\PUTBASEFIXP \SecondMDSPage 0 (IDIFFERENCE FIRSTPAGE \PagesPerMDSUnit)))
        (T
          (RETURN NIL)
          ; Can't switch to the higher area
          ((ILESSP \NxtArrayPage FIRSTPAGE)
            ; Safe to go ahead anyway. We'll be pretty short of space in the
            ; first 8mb, but it's switchable
            (\PUTBASEFIXP \NxtMDSPage 0 (IDIFFERENCE FIRSTPAGE \PagesPerMDSUnit)))
            ((ILESSP (IPLUS (SETQ FIRSTPAGE \SecondArrayPage)
              NPAGES)
              \SecondMDSPage)
              ; There is space in upper area. So advance the pointer that says where array space will start when we switch later on
              (\PUTBASEFIXP \SecondArrayPage 0 (IPLUS FIRSTPAGE NPAGES))
              (replace (IFPAGE FullSpaceUsed) of \InterfacePage with 65535))
            (T (RETURN NIL)))
          (RETURN (create POINTER
            PAGE# _ FIRSTPAGE))))))
```

(\MAPMDS

[LAMBDA (TYPE FN)

; Edited 17-Apr-2023 07:49 by Imm
; Edited 19-Oct-94 09:29 by sybalsky

;;; Applies FN to each virtual page number that is of type TYPE, or to all MDS pages if TYPE is NIL

```
(OR (NULL TYPE)
  (FIXP TYPE)
  (SETQ TYPE (\TYPENUMBERFROMNAME TYPE)))
(LET* ((VP (\CREATECELL \FIXP)
  (END (IMAX \DefaultSecondArrayPage \MaxMDSPage))
  TYP)
  (\PUTBASEFIXP VP 0 (IMIN \NxtMDSPage \LeastMDSPage))
  (WHILE (ILEQ VP END) DO (COND
    ((OR (EQ (SETQ TYP (NTYPX (create POINTER
      PAGE# _ VP)))
      TYPE)
    (AND (NULL TYPE)
      (NEQ TYP 0)
      (NEQ TYP \SMALLP)))
    (SPREADAPPLY* FN VP)))
  (\BOXIPLUS VP 2])
```

(\CHECKFORSTORAGEFULL

; Edited 4-Jan-93 02:04 by jds

[LAMBDA (NPAGES)
(DECLARE (GLOBALVARS \INTERRUPTSTATE \PENDINGINTERRUPT))

;;; Take appropriate action if storage is getting full. NPAGES is size of attempted allocation or NIL for MDS requests. Complications here because array
;;; space and MDS grow toward each other in two separate areas: the first 8MB of vmem and the remaining 24MB. Some machines cannot use the
;;; latter, so have to signal storage full when the first fills up. Other machines have to know when to switch over. Array space usually gets switched to
;;; the high segment before MDS, since MDS can eat the lo space in small increments all the way to the end --- Returns T if storage is ok, 0 if storage is
;;; ok but \NxtArrayPage changed, and NIL if storage is nearly full

```
(UNINTERRUPTABLY
  [PROG (PAGESLEFT)
    (RETURN (COND
      ((OR (ILESSP (SETQ PAGESLEFT (IPLUS (IDIFFERENCE \NxtMDSPage \NxtArrayPage)
        \PagesPerMDSUnit))
        \GUARDSTORAGEFULL)
```

```

NPAGES)
(SELECTC (OR \STORAGEFULLSTATE (\SET.STORAGE.STATE)
  (LIST \SFS.NOTSWITCHABLE \SFS.FULLYSWITCHED)
  (COND
    ((ILESSP PAGESLEFT 0)
      (while T do (\MP.ERROR \MP.MDSFULL "Storage completely full")))
    ((AND (ILEQ PAGESLEFT \GUARD1STORAGEFULL)
      (NEQ \STORAGEFULL 0))
      (SETQ \STORAGEFULL 0)
      (\MP.ERROR \MP.MDSFULLWARNING "Space getting VERY full. Please save and
        reload a.s.a.p. Type control-N to continue now.))
    ((NOT \STORAGEFULL)
      (SETQ \STORAGEFULL T) ; Note this is uninterruptable
      (replace STORAGEFULL of \INTERRUPTSTATE with T)
      (SETQ \PENDINGINTERRUPT T))
    (\DORECLAIM)
    NIL)
  (\SFS.SWITCHABLE ; We have verified that we can use the full 32MB, but haven't
    ; switched there yet
  (OR [COND
    [(NULL NPAGES) ; Want MDS
      (COND
        ((ILEQ PAGESLEFT 0)
          (\PUTBASEFIXP \LeastMDSPage 0 \NxtArrayPage)
          (\PUTBASEFIXP \NxtMDSPage 0 \SecondMDSPage)
          (\ADVANCE.STORAGE.STATE \SFS.FULLYSWITCHED)
          (\ADVANCE.ARRAY.SEGMENTS \SecondArrayPage]
        (T ; Want array space
          (COND
            ((IGREATERP NPAGES PAGESLEFT)
              ; Have to switch array space over, but leave MDS to fill the rest
              ; of the low pages
              (\PUTBASEFIXP \LeastMDSPage 0 \NxtArrayPage)
              (\ADVANCE.STORAGE.STATE \SFS.ARRAYSWITCHED)
              (\ADVANCE.ARRAY.SEGMENTS \SecondArrayPage]
            T))
          (\SFS.ARRAYSWITCHED
            (COND
              ((ILESSP \NxtMDSPage \LeastMDSPage)
                ; Finally used up lo MDS, so switch over to hi
                (\PUTBASEFIXP \NxtMDSPage 0 \SecondMDSPage)
                (\ADVANCE.STORAGE.STATE \SFS.FULLYSWITCHED)
                T)
              ((AND NPAGES (IGEQ (IPLUS NPAGES \GUARDSTORAGEFULL)
                (IDIFFERENCE \SecondMDSPage \NxtArrayPage)))
                ;; MDS still in lo area, arrays in hi area, and we're asking for too big an array! Unlikely, but handle it
                ;; as a storage full case
                NIL)
              (T T)))
            (SHOULDNT]))])

```

(\DOSTORAGEFULLINTERRUPT

```

[LAMBDA NIL ; (* bvm%: "13-Feb-85 16:28")
  (replace STORAGEFULL of \INTERRUPTSTATE with NIL)
  (PROG ((HELPFLAG 'BREAK!))
    (LISPERROR "STORAGE FULL" ' "save your work & reload a.s.a.p." T))

```

(\SET.STORAGE.STATE

```

[LAMBDA NIL ; Edited 24-May-90 19:11 by Takeshi
  (COND
    ((EQ (FETCH (IFPAGE DL24BitAddressable) OF \InterfacePage)
      0)
      (SETQ \STORAGEFULLSTATE \SFS.NOTSWITCHABLE))
    (T (SETQ \STORAGEFULLSTATE \SFS.SWITCHABLE)))
  (PUSH \SYSTEMCACHEVARS '\STORAGEFULLSTATE)
  \STORAGEFULLSTATE))

```

(\SETTYPEMASK

```

[LAMBDA (NTYPX BITS)
  (PROG ((DTD (\GETDTD NTYPX)))
    (change (fetch DTDTYPEENTRY of DTD)
      (LOGOR DATUM BITS))
    (\MAPMDS NTYPX (FUNCTION (LAMBDA (PAGE)
      (\PUTBASE \MDStypeTable (SETQ PAGE (FOLDLO PAGE (IQUOTIENT \MDSIncrement
        WORDSPERPAGE))))
      (LOGOR (\GETBASE \MDStypeTable PAGE)
        BITS))

```

(\ADVANCE.STORAGE.STATE

```

[LAMBDA (FLG) ; (* bvm%: " 9-Jan-85 15:30")
  ;; Bump the flag that tells what state storage allocation is in with respect to the 8MB -- 32MB distinction. Also remove flag from
  ;; \SYSTEMCACHEVARS since it can no longer get recomputed

```



```

(T "OK to deallocate DATATYPE ")
NAME]
; don't redeclare
(RETURN NTYPX)
((IGREATERP NTYPX \MaxSysTypeNum) ; Can redeclare 'user' types, i.e., anything not in the makeinit
(SETQ REDECLARED T)) ; can't mess with sys types
(T
(ERROR "ILLEGAL DATA TYPE" NAME]
;; If we get this far, we're about to create a for-real new datatype (we may need to deallocate the old version of this one...)
(COND
(NOT SIZE) ; only called to deallocate old datatype
)
(T (COND
((AND (EQ \MaxTypeNumber \EndTypeNumber)
(OR (NULL NTYPX)
REDECLARED))
(LISPERROR "DATA TYPES FULL" NAME)))
(UNINTERRUPTABLY
[COND
((OR (NULL NTYPX)
REDECLARED) ; Bump the global count of types assigned, and grab the latest.
(SETQ NEWTYPENUM (add \MaxTypeNumber 1)) ; Build a new DTD for it.
(SETQ NEWDTD (\GETDTD NEWTYPENUM))
(COND
((IGEQL (IPLUS (fetch WORDINPAGE of NEWDTD)
\DTDSIZE)
WORDSPERPAGE) ; if this is the last one which would fit on a page, create a new
; page
(\NEWPAGE (\ADDBASE NEWDTD \DTDSIZE)
T)))
(COND
[REDECLARED
;; When redeclaring a datatype, have to change the type of all old instances to be a new obsoleted type so
;; that the garbage collector will still collect them properly. Keep the original type number, because the
;; name -> type number mapping has already happened to compiled code
(LET ([NEWTPEENTRY (LOGOR NEWTYPENUM (LOGAND (fetch DTDTYPEENTRY of DTD)
(LOGNOT \TT.TYPEMASK)
FOUNDSOME)
[MAPMDS NTYPX (FUNCTION (LAMBDA (PAGE)
(\MAKEMDSEENTRY PAGE NEWTYPEENTRY)
(SETQ FOUNDSOME T)
(COND
((NOT FOUNDSOME)
;; Optimization: if no objects of the old type have been allocated (or all have been reclaimed and the
;; pages detyped), then don't need a new type number for them
(add \MaxTypeNumber -1))
(T (replace DTDESCRS of DTD with NIL)
(replace DTDTYPEESPECS of DTD with NIL)
(\BLT NEWDTD DTD \DTDSIZE)
;; Copy old DTD to new. Be careful about the pointer fields -- we haven't incremented their reference
;; counts. Those fields are DTDESCRS, DTDTYPEESPECS and DTDPTRS, the first two of which we
;; have conveniently smashed to NIL before copying.
(\ADDRESS (fetch DTDPTRS of NEWDTD))
(replace DTDOBSOLETE of NEWDTD with T)
(replace DTDTYPEENTRY of NEWDTD with NEWTYPEENTRY)
[replace DTDNAME of NEWDTD with (NEW-SYMBOL-CODE (PACK* "Obsolete-"
NAME)
(\ATOMPNAMINDEX (PACK*
ete-" NAME] "Obsol
(replace DTDFREE of DTD with NIL)
; Replacement type has no free list--just the old type, now in
; NEWDTD
]
)
; Normal case of a new type
(T
(SETQ NTYPX NEWTYPENUM)
(replace DTDNAME of (SETQ DTD NEWDTD) with (NEW-SYMBOL-CODE NAME (\ATOMPNAMINDEX
NAME]
(COND
((NEQL SIZE 0) ; If the datum takes up any space, remember what it looks like
; inside
(replace DTDSIZE of DTD with SIZE)
(replace DTDESCRS of DTD with (COPY DESCRIPTORS))
(replace DTDTYPEESPECS of DTD with (COPY SPECS))
(replace DTDPTRS of DTD with PTRFIELDS)
(replace DTDSUPERTYPE of DTD with SUPERTYPENUMBER)
(replace DTDTYPEENTRY of DTD with NTYPX) ; The type-masked type#, for fast type checking
))
;; NOTE: If the redeclared type has subtypes, we have to redeclare them, too!

```

```
)
(RETURN (CL:VALUES NTYPX REDECLARED])
```

(\RESOLVE.TYPENUMBER

```
[LAMBDA (TYPENAME)
```

(* bvm%: "13-Jun-86 16:11")

:: For the loader. Returns a type number for TYPENAME, possibly allocating a new type number (but not declaring it) if the type does not yet exist.

```
(COND
  ((AND TYPENAME (LITATOM TYPENAME))
   (OR (\TYPENUMBERFROMNAME TYPENAME)
        (\ASSIGNDATATYPE1 TYPENAME NIL 0)))
  (T (\ILLEGAL.ARG TYPENAME]))
```

(\TYPENUMBERFROMNAME

```
[LAMBDA (TYPE)
```

; Edited 2-Apr-91 15:48 by sybalsky

```
(AND TYPE (BIND (INDEX _ (NEW-SYMBOL-CODE TYPE (\ATOMNAMEINDEX TYPE))) for I from 1 to \MaxTypeNumber
  do (COND
      ((EQ INDEX (fetch DTDNAME of (\GETDTD I)))
       (RETURN I]))
```

(\CREATECELL

```
[LAMBDA (TYP)
```

(* lmm "10-DEC-82 15:49")

```
(\CREATECELL TYP))
```

(\CREATECELL

```
[LAMBDA (TYP)
```

; Edited 25-Apr-94 10:37 by jds

```
(COND
  ((AND (NEQ CDRCODING 0)
        (EQ TYP \LISTP))
   (RAID "CREATECELL \LISTP")))
```

:: For the real sysout, this must be the opcode CREATECELL, so we don't have to have the lisp versions of NEWPAGE &c track the C. JDS
;; 4/25/94

```
(UNLESSINEX (\CREATECELL TYP)
  (LET ((DTD (\GETDTD TYP))
        NEWCELL)
    (while (EQ (fetch DTDSIZE of DTD)
               0)
      do (ERROR "Attempt to CREATE a type not declared yet" (\TYPENAMEFROMNUMBER TYP)))
    (UNINTERRUPTABLY
     (COND
      ((SETQ NEWCELL (fetch DTDFREE of DTD))
       (CHECK (EQ TYP (NTYPX NEWCELL)))
       (replace DTDFREE of DTD with (\GETBASEPTR NEWCELL 0))
       (\StatsAdd1 (LOCF (fetch DTDOLDCNT of DTD)))
       (LET [(CNT (SUB1 (fetch DTDSIZE of DTD)) ; Clear object
             (\PUTBASE NEWCELL CNT 0)
             (\BLT NEWCELL (\ADDBASE NEWCELL 1)
                       CNT))
            (\CREATeref NEWCELL)
            NEWCELL)
        (T ; Free list exhausted. Replenish it, then do a CREATECELL, hopefully getting the microcode to do most of the work.
           ; Note: it is possible, albeit unlikely, that \ALLOCMDSPAGE will eventually cause a CREATECELL to occur. Hence,
           ; DTD:DTDFREE might possibly be non-NIL by the time we get back here, which is why it is included below.
           ; Don't understand this remark -- if CREATECELL gets called for this type before we have stored DTDFREE then are
           ; we just hoping the recursion eventually stops? Remark might apply for the old implementation where
           ; CREATECELL for a random type fixes everyone's free list, but again I'm not sure why. -bvm 5/86
           (replace DTDFREE of DTD with (\INITMDSPAGE (\ALLOCMDSPAGE (fetch DTDTYPEENTRY
                                                                     of DTD))
                                                       (fetch DTDSIZE of DTD)
                                                       (fetch DTDFREE of DTD)))
           (CREATECELL TYP))))))
```

:: For NEW_STORAGE option was set in Maiko, then \maiko.set.storage.state is active

```
(RPAQQ SP NOBIND)
```

```
(RPAQQ? CROSSCOMPILING )
```

```
(RPAQQ? ASSIGNDATATYPE.ASKUSERWAIT 300)
```

```
(RPAQQ? \STORAGEFULLSTATE )
```

```
(RPAQQ? \STORAGEFULL )
```

```
(DECLARE%: DOEVAL@COMPILE DONTCOPY
```

```
(GLOBALVARS CROSSCOMPILING \STORAGEFULLSTATE \STORAGEFULL \SYSTEMCACHEVARS \NxtArrayPage)
```

```

)
(DECLARE%: DOEVAL@COMPILE DONTCOPY
(SPECVARS ASSIGNDATATYPE.ASKUSERWAIT
)

```

:: fetch and replace

(DEFINEQ

(FETCHFIELD

[LAMBDA (DESCRIPTOR DATUM) (* edited%: " 7-JUN-83 10:23")

:: retrieves a data field from a user data structure.

```

(PROG ((TN (fetch fdTypeName of DESCRIPTOR))
(OFFSET (fetch fdOffset of DESCRIPTOR)))
(AND TN (SETQ DATUM (\DTEST DATUM TN)))
(RETURN (SELECTQ (fetch fdType of DESCRIPTOR)
((POINTER XPOINTER FULLPOINTER FULLXPOINTER)
(\GETBASEPTR DATUM OFFSET))
(FLOATP (MAKEFLOATNUMBER (\GETBASE DATUM OFFSET)
(\GETBASE (\ADDBASE DATUM 1)
OFFSET)))
(FIXP (\MAKENUMBER (\GETBASE DATUM OFFSET)
(\GETBASE (ADDBASE DATUM 1)
OFFSET)))
(SWAPPEDFIXP (\MAKENUMBER (\GETBASE (\ADDBASE DATUM 1)
OFFSET)
(\GETBASE DATUM OFFSET)))
(PROG ((FT (fetch fdType of DESCRIPTOR))
(OFF OFFSET))
(RETURN (SELECTQ (CAR FT)
(BITS (LOGAND (LRSH (\GETBASE DATUM OFF)
(BitFieldShift (CDR FT)))
(BitFieldMask (CDR FT))))
(SIGNEDBITS ([LAMBDA (N WIDTH)
(COND
[[IGREATERP N (SUB1 (LLSH 1 (SUB1 WIDTH)
(SUB1 (IDIFFERENCE N (SUB1 (LLSH 1 WIDTH)
(T N)
(LOGAND (LRSH (\GETBASE DATUM OFF)
(BitFieldShift (CDR FT)))
(BitFieldMask (CDR FT)))
(BitFieldWidth (CDR FT)))]
(LONGBITS (\MAKENUMBER (LOGAND (LRSH (\GETBASE DATUM OFF)
(BitFieldShift (CDR FT)))
(BitFieldMask (CDR FT)))
(\GETBASE (ADDBASE DATUM 1)
OFF)))
(FLAGBITS (NEQ (LOGAND (\GETBASE DATUM OFF)
(BitFieldShiftedMask (CDR FT)))
0))
(LISPERROR "ILLEGAL ARG" DESCRIPTOR])

```

(REPLACEFIELD

[LAMBDA (DESCRIPTOR DATUM NEWVALUE) (* Imm " 1-Jan-85 23:09")

; replace a field in a user data structure. return coerced value.

```

(PROG ((OFFSET (fetch fdOffset of DESCRIPTOR))
(FT (fetch fdType of DESCRIPTOR))
(TN (fetch fdTypeName of DESCRIPTOR))
SHIFT MASK)
(AND TN (SETQ DATUM (\DTEST DATUM TN)))
(RETURN
(SELECTQ FT
((POINTER FULLPOINTER)
(\RPLPTR DATUM OFFSET NEWVALUE))
(XPOINTER (PUTBASEPTR DATUM OFFSET NEWVALUE)) ; no ref count, hi bits used
(FULLXPOINTER (\PUTBASEPTR DATUM OFFSET NEWVALUE))
(FLOATP (\PUTBASEFLOATP DATUM OFFSET NEWVALUE))
(FIXP (\PUTFIXP (\ADDBASE DATUM OFFSET)
NEWVALUE)
NEWVALUE)
(SWAPPEDFIXP (\PUTSWAPPEDFIXP (\ADDBASE DATUM OFFSET)
NEWVALUE)
NEWVALUE)
(SELECTQ (CAR FT)
(BITS (LOGAND (LRSH (\PUTBASE DATUM OFFSET (LOGOR [LOGAND
(\GETBASE DATUM OFFSET)
(LOGXOR 65535
(LLSH (SETQ MASK
(BitFieldMask (CDR FT)))
(SETQ SHIFT
(BitFieldShift (CDR FT]
(LLSH (LOGAND NEWVALUE MASK)

```



```

                                SHIFT)))
                                MASK))
(SIGNEDBITS ([LAMBDA (X)
              (COND
                [[IGREATERP X (SUB1 (LLSH 1 (SUB1 (BitFieldWidth (CDR FT]
                (SUB1 (IDIFFERENCE X (SUB1 (LLSH 1 (BitFieldWidth (CDR FT]
                (T X]
              (LOGAND
                (LRSH (\PUTBASE DATUM OFFSET
                  (LOGOR [LOGAND (\GETBASE DATUM OFFSET)
                    (LOGXOR 65535 (LSH (SETQ MASK (BitFieldMask
                      (CDR FT)))
                    (SETQ SHIFT (BitFieldShift
                      (CDR FT]
                    (LSH (LOGAND [LOGAND NEWVALUE
                      (SUB1 (LLSH 1 (BitFieldWidth
                        (CDR FT]
                      MASK)
                    SHIFT)))
              MASK)))
              SHIFT))
              MASK)))
(FLAGSBITS (\PUTBASE DATUM OFFSET (LOGOR [LOGAND (\GETBASE DATUM OFFSET)
      (LOGXOR 65535 (LSH (SETQ MASK
        (BitFieldMask
          (CDR FT)))
        (SETQ SHIFT
          (BitFieldShift
            (CDR FT]
        (LSH (LOGAND (COND
          (NEWVALUE 65535)
          (T 0))
        MASK)
      SHIFT)))
      (AND NEWVALUE T))
(LONGBITS (PROG (LO HI)
  (.UNBOX. NEWVALUE HI LO)
  (UNINTERRUPTABLY
    (\PUTBASE DATUM OFFSET (LOGOR [LOGAND
      (\GETBASE DATUM OFFSET)
      (LOGXOR 65535
        (LSH (SETQ MASK
          (BitFieldMask (CDR FT)))
        (SETQ SHIFT
          (BitFieldShift (CDR FT]
        (LSH (LOGAND HI MASK)
          SHIFT)))
      (\PUTBASE DATUM (ADD1 OFFSET
        LO)))
      NEWVALUE)
(LISPERROR "ILLEGAL ARG" DESCRIPTOR])

```

(BOXCOUNT

```

[LAMBDA (TYPE N) (*/ Imm "20-OCT-81 20:27")
  (PROG [(DTD (\GETDTD (OR (SMALLP TYPE)
    (COND
      ((NULL TYPE)
        \FIXP)
      (T (\TYPENUMBERFROMNAME TYPE]
    (RETURN (PROG1 (fetch DTCNT of DTD)
      (AND (NUMBERP N)
        (replace DTCNT of DTD with N) ]])

```

(CONSCOUNT

```

[LAMBDA (N) (*/ Imm "13-MAY-80 23:02")
  (BOXCOUNT \LISTP N])

```

(\DTEST

```

[LAMBDA (OBJ TYPE) (*/ Imm "22-Mar-85 12:29")
  (\DTEST.UFN OBJ TYPE])

```

(\TYPECHECK

```

[LAMBDA (OBJ TYPE) (*/ Imm "22-Mar-85 12:29")
  (\DTEST.UFN OBJ TYPE])

```

(\DTEST.UFN

```

[LAMBDA (OBJ TYPEN) (*/ gbn " 3-Oct-86 10:49")
  ;; ufn for DTEST opcode
  ;; coerce into desired type
  (PROG ((N (NTYPX OBJ)))
    LP (COND

```

```
(EQ (fetch DTDNAME of (\GETDTD N))
  TYPEN) ; should be happening in microcode
(RETURN OBJ)
([NEQ 0 (SETQ N (fetch DTDSUPERTYPE of (\GETDTD N)
  (GO LP))
(T (RETURN (SELECTQ (\INDEXATOMPNAME TYPEN)
  (FLOATP (\FLOAT OBJ)) ; Should be able to get at the INPUT/OUTPUT flg--a second arg
  (STREAM) ; to \DTEST ?
  (\GETSTREAM OBJ (SELECTQ (STKNTHNAME -1 '\DTEST.UFN)
  ((\BINS \BIN BIN)
  'INPUT)
  ((\BOOTS \BOUT BOUT)
  'OUTPUT)
  NIL)))
(HARRAYP (DECLARE (GLOBALVARS SYSHASHARRAY))
  (COND
  [(NULL OBJ)
  (COND
  (SYSHASHARRAY (\DTEST SYSHASHARRAY 'HARRAYP))
  (T (LISPERROR "ARG NOT HARRAY" OBJ T])
  ((AND (LISTP OBJ)
  (TYPENAMEP (CAR OBJ)
  'HARRAYP))
  (CAR OBJ))
  (T (LISPERROR "ARG NOT HARRAY" OBJ T])))
(FONTDESCRIPTOR
  (\COERCEFONTDESC OBJ))
(SMALLP [PROG (HI LO)
  (.UNBOX. OBJ HI LO)
  (RETURN (OR (SMALLP (\MAKENUMBER HI LO))
  (LISPERROR "ILLEGAL ARG" OBJ T])
(LISTP (LISPERROR "ARG NOT LIST" OBJ T))
(LITATOM (LISPERROR "ARG NOT LITATOM" OBJ T))
(STACKP (LISPERROR "ILLEGAL STACK ARG" OBJ T))
(READTABLEP (LISPERROR "ILLEGAL READTABLE" OBJ T))
(TERMTABLEP (LISPERROR "ILLEGAL TERMINAL TABLE" OBJ T))
(ARRAYP (LISPERROR "ARG NOT ARRAY" OBJ T))
(\DISPLAYDATA ; Should be able to get at the stream--a second arg to \DTEST ?
  (ERROR "ARG NOT DISPLAY STREAM" NIL))
(\LISPERROR OBJ (CONCAT "ARG NOT " (\INDEXATOMPNAME TYPEN)
  T])
```

(INSTANCEP.UFN

[LAMBDA (OBJ TYPEN)

; Edited 2-Apr-91 00:40 by sybalsky

;;; ufn for INSTANCEP opcode

```
(PROG ((N (NTYPX OBJ)))
  LP (NEW-SYMBOL-CODE (COND
  ((AND (FIXP TYPEN)
  (EQ (\VAG2 \AtomHI TYPEN)
  (fetch DTDNAME of (\GETDTD N)
  (RETURN T))
  ((EQ (fetch DTDNAME of (\GETDTD N)
  TYPEN)
  (RETURN T))
  ([NEQ 0 (SETQ N (fetch DTDSUPERTYPE of (\GETDTD N)
  ;; recur on the supertype
  (GO LP))
  (T (RETURN NIL))))
(COND
  ((IEQP (fetch DTDNAME of (\GETDTD N)
  TYPEN)
  (RETURN T))
  ([NEQ 0 (SETQ N (fetch DTDSUPERTYPE of (\GETDTD N)
  ;; recur on the supertype
  (GO LP))
  (T (RETURN NIL]))
```

(INSTANCE-P

[LAMBDA (OBJECT TYPE)

(* gbn "26-Sep-86 17:07")

;; should be phased out in favor of calls to typenamp, which shares the definition.

(INSTANCEP.UFN OBJECT (\ATOMPNAMEINDEX TYPE])

(TYPECHECK.UFN

[LAMBDA (OBJ TYPEN)

(* gbn "23-Sep-86 20:06")

;;; ufn for TYPECHECK opcode --- cause error if not of right type

```
(PROG ((N (NTYPX OBJ)))
  LP (COND
```

```

(EQ (fetch DTDNAME of (\GETDTD N))
  TYPEN)
(RETURN OBJ)
([NEQ 0 (SETQ N (fetch DTDSUPERTYPE of (\GETDTD N)
  (GO LP))
  (T (RETURN (\LISPERROR OBJ (CONCAT "ARG NOT " (\INDEXATOMPNAME TYPEN)
  T]))

```

(GETDESCRIPTORS

```

[LAMBDA (TYPENAME) (* Imm "21-Apr-85 15:10")
  (PROG NIL
    (RETURN (fetch DTDESCRS of (\GETDTD (COND
      ((LITATOM TYPENAME)
        (OR (\TYPENUMBERFROMNAME TYPENAME)
          (RETURN)))
      (T (NTYPX TYPENAME]))

```

(GETSUPERTYPE

```

[LAMBDA (TYPENAME) (* Imm "13-Mar-86 14:36")
  ;; return the name of the supertype (i.e., the :INCLUDEd type) of a datatype if it has one, NIL otherwise
  (LET ((NX (\TYPENUMBERFROMNAME TYPENAME)))
    (COND
      (NX (LET [(N (fetch DTDSUPERTYPE of (\GETDTD NX)
        (COND
          ((NEQ N 0)
            (\TYPENAMEFROMNUMBER N))
          (T NIL]))

```

(GETFIELDSPECS

```

[LAMBDA (TYPENAME) (* rmk%: "28-OCT-81 17:42")
  (PROG NIL
    (RETURN (COPY (fetch DTDTYPESECS of (\GETDTD (COND
      ((LITATOM TYPENAME)
        (OR (\TYPENUMBERFROMNAME TYPENAME)
          (RETURN)))
      (T (NTYPX TYPENAME]))

```

(NCREATE

```

[LAMBDA (TYPE OLDOBJ) (* Imm "14-MAY-80 08:33")
  (NCREATE2 (\TYPENUMBERFROMNAME TYPE)
    OLDOBJ)]

```

(NCREATE2

```

[LAMBDA (NTYPX OLDOBJ) (* bvm%: " 5-Feb-85 16:43")
  ;; a version of NCREATE which has is compiled from calls to NCREATE which have a quoted first arg and an old object. These can use the TYPE
  ;; number variable in stead of having to look it up.
  (PROG ((DTD (\GETDTD NTYPX))
    (NEW (CREATECELL NTYPX)))
    [COND
      ((EQ (NTYPX OLDOBJ)
        NTYPX)
        (UNINTERRUPTABLY
          (\BLT NEW OLDOBJ (fetch DTDSIZE of DTD)
            (for P in (fetch DTDPTRS of DTD) do (\ADDRESS (\GETBASEPTR NEW P))))))
      (RETURN NEW)]

```

(REPLACEFIELDVAL

```

[LAMBDA (DESCRIPTOR DATUM NEWVALUE) (* Imm%: "22-AUG-76 04:18:20")
  ;; used by the record package-- compiles open better than saving datum
  (REPLACEFIELD DESCRIPTOR DATUM NEWVALUE)
  DATUM)]

```

(PUTBASEPTRX

```

[LAMBDA (DATUM OFFSET NEWVALUE) ; Edited 13-Jan-93 00:13 by jds
  ;; Put the new value into an XPOINTER field. As of Medley 2.1/3.0, this is a 28-bit quantity.
  (UNINTERRUPTABLY
    (PUTBASE DATUM OFFSET (LOGOR (LOGAND 61440 (GETBASE DATUM OFFSET))
      (HILOC NEWVALUE)))
    (PUTBASE DATUM (ADD1 OFFSET)
      (LOLOC NEWVALUE))
    NEWVALUE)])

```

(/REPLACEFIELD

```

[LAMBDA (DESCRIPTOR DATUM NEWVALUE) (* Imm%: "23-AUG-76 00:01:53")
  [AND LISPXHIST (UNDOSAVE (LIST '/REPLACEFIELD DESCRIPTOR DATUM (FETCHFIELD DESCRIPTOR DATUM)

```

(REPLACEFIELD DESCRIPTOR DATUM NEWVALUE])

(TYPENAME

[LAMBDA (X)

; Edited 28-Jun-99 16:56 by rmk;
; Edited 28-Jun-99 16:55 by rmk;
; Edited 11-Nov-98 12:14 by rmk:

(LET ((N (N (NTYPX X)))

(COND

((EQ N \ARRAYP)

(\ARRAYTYPENAME X))

((%STRINGP X)

'STRINGP)

; Common lisp strings report as STRINGP's.

([EQ 'NEW-ATOM (SETQ N (\INDEXATOMPNAME (fetch DTDNAME of (\GETDTD N]

;; Large atom space returns NEW-ATOM instead of LITATOM

'LITATOM)

(T N])

(TYPENAMEP

[LAMBDA (DATUM TYPE)

; Edited 18-Dec-86 16:33 by jop

(COND

((EQ TYPE 'STRINGP)

(%STRINGP DATUM))

(T (\INSTANCEP.UFN DATUM TYPE])

(\TYPENAMEFROMNUMBER

[LAMBDA (N)

(* Imm "13-FEB-83 14:13")

(COND

((ILESSP N (ADD1 \MaxTypeNumber))

(\INDEXATOMPNAME (fetch DTDNAME of (\GETDTD N])

(BLOCKDATAP

[LAMBDA (X)

(* JonL "22-Sep-84 23:15")

(PROG ((TYPENO (NTYPX X)))

(RETURN (COND

((EQ 0 TYPENO)

(type? ARRAYBLOCK X))

(T (fetch DTDHUNKP of (\GETDTD TYPENO])

(USERDATATYPES

[LAMBDA NIL

(* rrb "16-JUL-80 13:17")

(DATATYPES T])

(DATATYPEP

[LAMBDA (DATATYPESPEC)

(* bvm%: "12-Feb-85 17:29")

;; returns the type name of a data type spec if it is a datatype.

(COND

[(SMALLP DATATYPESPEC)

(PROG ((DTD (\GETDTD DATATYPESPEC))

NAME)

(RETURN (AND (NOT (fetch DTDHUNKP of DTD))

(SETQ NAME (\INDEXATOMPNAME (fetch DTDNAME of DTD)))

(NEQ NAME '**DEALLOC**)

NAME]

((NOT (LITATOM DATATYPESPEC))

NIL)

((FMEMB DATATYPESPEC '(CCODEP HARRAYP))

; handle subtypes of arrayp specially.

DATATYPESPEC)

((for I from 1 to \MaxTypeNumber thereis (EQ (\INDEXATOMPNAME (fetch DTDNAME of (\GETDTD I)))

DATATYPESPEC))

DATATYPESPEC])

(DATATYPES

[LAMBDA (USERSFLG)

(* rrb "16-JUL-80 13:20")

(bind N for I from (COND

(USERSFLG (ADD1 \MaxSysTypeNum))

(T 1))

to \MaxTypeNumber when (SETQ N (DATATYPEP I)) collect N])

)

(MOVD? 'FETCHFIELD 'FFETCHFIELD NIL T)

(MOVD? 'REPLACEFIELD 'FREPLACEFIELD NIL T)

(MOVD? 'REPLACEFIELDVAL 'FREPLACEFIELDVAL NIL T)

(DEFOPTIMIZER TYPENAMEP (DATUM TYPE &ENVIRONMENT ENV)

```
(LET [(TYPE-NAME (CL:IF (AND (CL:CONSP TYPE)
                             (EQ (CAR TYPE)
                                 'QUOTE)
                             (CL:SYMBOLP (CADR TYPE)))
      (CADR TYPE)
      (CL:IF [AND TYPE-NAME (NOT (EQ TYPE-NAME 'STRINGP))
             [COND
              [(FMEMB :4-BYTE (COMPILER::ENV-TARGET-ARCHITECTURE ENV))
               `((OPCODES INSTANCEP 0 0 0 (ATOM \, (CADR TYPE)))
                 ,DATUM]
              [(FMEMB :3-BYTE (COMPILER::ENV-TARGET-ARCHITECTURE ENV))
               `((OPCODES INSTANCEP 0 0 (ATOM \, (CADR TYPE)))
                 ,DATUM]
              (T `((OPCODES INSTANCEP 0 (ATOM \, (CADR TYPE)))
                  ,DATUM]
               'COMPILER:PASS)))
```

(DEFOPTIMIZER **INSTANCE-P** (&BODY BODY &ENVIRONMENT ENV)

```
(COND
  [[AND (EQ (CAADR BODY)
            'QUOTE)
        (CL:SYMBOLP (CADR (CADR BODY))
        (COND
          [(FMEMB :4-BYTE (COMPILER::ENV-TARGET-ARCHITECTURE ENV))
           `([OPCODES INSTANCEP 0 0 0 (ATOM \, (CADR (CADR BODY))
             , (CAR BODY)
           [(FMEMB :3-BYTE (COMPILER::ENV-TARGET-ARCHITECTURE ENV))
           `([OPCODES INSTANCEP 0 0 (ATOM \, (CADR (CADR BODY))
             , (CAR BODY)
           (T `([OPCODES INSTANCEP 0 (ATOM \, (CADR (CADR BODY))
              , (CAR BODY)
           (T 'IGNOREMACRO)))
```

:: STORAGE

(DEFINEQ

(STORAGE

```
[LAMBDA (TYPES PAGE-THRESHOLD IN-USE-THRESHOLD) ; Edited 8-Jan-88 14:39 by bvm
  (PROG ((TOTALALLOCMDS (CREATECELL \FIXP))
        (TOTALHUNKS (CREATECELL \FIXP))
        (FREE (CREATECELL \FIXP))
        (HUNKSTATS (from 0 to 2 collect (create HUNKSTAT)))
        TYPE TYPENAME DOBLOCKSFLG)
  (DECLARE (SPECVARS HUNKSTATS))
  (printout NIL "Type" 17 "Assigned" 30 "Free items" 45 "In use" 55 "Total alloc" T 15 "pages [items]" T
  )
  (COND
    [(AND TYPES (NEQ TYPES T))
     (for TYPE HFLG inside TYPES when [COND
      ((FIXP TYPE)
       (COND
         ((OR (< TYPE 0)
              (> TYPE \MaxTypeNumber))
          ; An explicit type number ought to be 'right'
          (ERROR "Not a type number" TYPE))
         (EQ TYPE 0)
         (SETQ DOBLOCKSFLG T)
         NIL
         (T T)))
      (T (SETQ TYPE (\TYPENUMBERFROMNAME TYPE)

  do (COND
    ((fetch DTDHUNKP of (\GETDTD TYPE))
     (SETQ HFLG T)))
    (\STORAGE.TYPE TYPE FREE TOTALALLOCMDS PAGE-THRESHOLD IN-USE-THRESHOLD)
  finally (COND
    (HFLG (\STORAGE.HUNKTYPE TOTALALLOCMDS PAGE-THRESHOLD IN-USE-THRESHOLD]
  (T (for I from 1 to \MaxTypeNumber do (\STORAGE.TYPE I FREE TOTALALLOCMDS PAGE-THRESHOLD
                                           IN-USE-THRESHOLD))
    (\STORAGE.HUNKTYPE TOTALHUNKS PAGE-THRESHOLD IN-USE-THRESHOLD)
    (printout NIL T "TOTAL" 15 .I5 (+ TOTALALLOCMDS TOTALHUNKS)
              T T)
    (printout NIL "Data Spaces Summary" T)
    (printout NIL 30 "Allocated" 50 "Remaining" T)
    (printout NIL 32 "Pages" 52 "Pages" T)
    (printout NIL "Datatypes (incl. LISTP etc.)" 30 .I8 TOTALALLOCMDS 50 "\ " T)
    ; Arrayspace and MDS come out of the same pot, so lump their
    ; 'remaining' pages together
    (printout NIL "ArrayBlocks" (COND
      ((NOT (= TOTALHUNKS 0))
       " (variable)"
       (T ""))
    30 .I8 (SELECTC \STORAGEFULLSTATE
      ((LIST \SFS.FULLYSWITCHED \SFS.ARRAYSWITCHED)
       (+ (- \LeastMDSPage \FirstArrayPage)
          (- \NxtArrayPage \SecondArrayPage)))
```



```

(IQUOTIENT \MDSIncrement SIZE))
(T (CONSTANT (FIX (FQUOTIENT \MDSIncrement 2.2]))
(COND
  ((EQ SIZE 0) ; Undeclared, or not allocated
  (RETURN))
  (T (IQUOTIENT \MDSIncrement SIZE]
[MAPMDS TYPE (FUNCTION (LAMBDA NIL
  (add ALLOCMDS 1]
(SETQ NPAGESALLOCATED (TIMES ALLOCMDS \PagesPerMDSUnit))
(COND
  ((SETQ HUNKP (fetch DTDHUNKP of DTD))
  (add [fetch (HUNKSTAT NPAGES) of (SETQ STAT (CAR (NTH HUNKSTATS (ADD1 (fetch DTDGCTYPE of DTD]
  NPAGESALLOCATED))
  (T (\BOXIPLUS TOTALALLOCMDS NPAGESALLOCATED)))
(COND
  ((< NPAGESALLOCATED (OR PAGE-THRESHOLD 1))
  (RETURN))
(\PUTBASEFIXP (\DTEST FREE 'FIXP)
  0 0)
[COND
  [(AND (NEQ CDCODING 0)
  (EQ TYPE \LISTP)) ; CONS pages have a different kind of free list
  (for (LSTPAG _ (create POINTER
  PAGE# _ (fetch DTDNEXTPAGE of \LISTPDTD)))
  by (create POINTER
  PAGE# _ (fetch (CONSPAGE NEXTPAGE) of LSTPAG))
  while LSTPAG do (\BOXIPLUS FREE (fetch (CONSPAGE CNT) of LSTPAG]
  (T (for (PTR _ (fetch DTDFREE of DTD)) by (\GETBASEPTR PTR 0) while PTR
  do (CHECK (EQ (NTYPX PTR)
  TYPE))
  (\BOXIPLUS FREE 1]
(SETQ INUSE (- (SETQ ALLOC (TIMES ALLOCMDS ITEMSPERMDS)
  FREE))
(COND
  ((fetch DTDHUNKP of DTD) ; Keep a cumulative table to be printed out at the end of this all
  ; by \STORAGE.HUNKTYPE
  (add (fetch (HUNKSTAT NITEMS) of STAT)
  ALLOC)
  (add (fetch (HUNKSTAT NFREE) of STAT)
  FREE)
  (add (fetch (HUNKSTAT NINUSE) of STAT)
  INUSE)
  (add (fetch (HUNKSTAT NALLOCATED) of STAT)
  (BOXCOUNT TYPE))
  ((OR (NOT IN-USE-THRESHOLD)
  (>= INUSE IN-USE-THRESHOLD))
  (\STMDSTYPE NAME NPAGESALLOCATED ALLOC FREE INUSE (BOXCOUNT TYPE]))

```

```

(\STLINP
[LAMBDA (STR ALLOC TOT) ; (* bvm%: " 9-Feb-85 15:23")
  (printout NIL STR 30 .I8 ALLOC 50 .I8 (IDIFFERENCE TOT ALLOC)
  T])

```

```

(\STMDSTYPE
[LAMBDA (NAME NPAGESALLOCATED ALLOC FREE INUSE BOXCOUNT) ; Edited 8-Jan-88 14:33 by bvm
  (PRIN2 NAME)
  (LET ((COL (POSITION)
  NC)
  (if (AND (>= COL 15)
  (< COL 19)
  (> (SETQ COL (- 20 COL (NCHARS NPAGESALLOCATED)))
  0))
  then ; Past the point we allocated for starting the #pages field, but
  ; #pages is small, so we can squeak in.
  (SPACES COL)
  (printout NIL .I1 NPAGESALLOCATED)
  else (printout NIL 15 .I5 NPAGESALLOCATED)))
  (if (EQ NAME 'LISTP) ; Indicate that LISTP numbers for total & in use are approximate
  then
  (\STMDS.APPROX ALLOC)
  else (printout NIL .I8 ALLOC))
  (printout NIL 30 .I8 FREE 43)
  (if (EQ NAME 'LISTP)
  then (\STMDS.APPROX INUSE)
  else (printout NIL .I8 INUSE))
  (printout NIL 56 .I10 BOXCOUNT T])

```

```

(\STMDS.APPROX
[LAMBDA (N) ; Edited 8-Jan-88 14:33 by bvm
  ;; Print n in an 8-col field preceded by a ~ to indicate approximation
  (SPACES (- 7 (NCHARS N)))
  (printout NIL "~" .I1 N])

```

(\STORAGE.HUNKTYPE

[LAMBDA (TOTAL PAGE-THRESHOLD IN-USE-THRESHOLD) ; Edited 8-Jan-88 14:39 by bvm

```
(DECLARE (USEDFREE HUNKSTATS))
(PROG (NPAGESALLOCATED STAT)
  (for GCTYPE.NAME in [CONSTANT (LIST (LIST UNBOXEDBLOCK.GCT 'UNBOXEDHUNK)
                                     (LIST PTRBLOCK.GCT 'PTRHUNK)
                                     (LIST CODEBLOCK.GCT 'CODEHUNK)]
    do [SETQ STAT (CAR (NTH HUNKSTATS (ADD1 (CAR GCTYPE.NAME)
                                         (SETQ NPAGESALLOCATED (fetch (HUNKSTAT NPAGES) of STAT))
                                         (\BOXIPLUS TOTAL NPAGESALLOCATED)
                                         (COND
                                           ((AND (NEQ NPAGESALLOCATED 0)
                                                  (OR (NOT PAGE-THRESHOLD)
                                                       (>= NPAGESALLOCATED PAGE-THRESHOLD))
                                                  (OR (NOT IN-USE-THRESHOLD)
                                                       (>= (fetch (HUNKSTAT NINUSE) of STAT)
                                                            IN-USE-THRESHOLD)))
                                           (\STMDSTYPE (CADR GCTYPE.NAME)
                                                       NPAGESALLOCATED
                                                       (fetch (HUNKSTAT NITEMS) of STAT)
                                                       (fetch (HUNKSTAT NFREE) of STAT)
                                                       (fetch (HUNKSTAT NINUSE) of STAT)
                                                       (fetch (HUNKSTAT NALLOCATED) of STAT]))
      ]
  )
)
```

(DECLARE%: DONTCOPY

(DECLARE%: EVAL@COMPILE

```
(RECORD HUNKSTAT (NPAGES NITEMS NFREE NINUSE NALLOCATED)
  NPAGES _ 0 NITEMS _ 0 NFREE _ 0 NINUSE _ 0 NALLOCATED _ 0)
)
```

(RPAQ? STORAGE.ARRAYSIZES ' (4 16 64 256 1024 4096 16384 NIL))

(DECLARE%:

:: FOLLOWING DEFINITIONS EXPORTED

```
(DEFOPTIMIZER PUTBASEPTRX (&REST ARGS)
  (CONS ' (OPENLAMBDA (DATUM OFFSET NEWVALUE)
    (UNINTERRUPTABLY
      (\PUTBASE DATUM OFFSET (LOGOR (LOGAND 61440 (\GETBASE DATUM OFFSET))
                                    (LOGAND (\HILOC NEWVALUE)
                                             4095)))
      (\PUTBASE DATUM (ADD1 OFFSET)
                      (\LOLOC NEWVALUE))
      NEWVALUE))
    ARGS))
```

(DECLARE%: EVAL@COMPILE

(RPAQQ \SMALLP 1)

(RPAQQ \FIXP 2)

(RPAQQ \FLOATP 3)

(RPAQQ \LITATOM 4)

(RPAQQ \LISTP 5)

(RPAQQ \ARRAYP 6)

(RPAQQ \STACKP 8)

(RPAQQ \VMEMPAGEP 10)

(RPAQQ \STREAM 11)

(RPAQQ \NEW-ATOM 21)

```
(CONSTANTS \SMALLP \FIXP \FLOATP \LITATOM \LISTP \ARRAYP \STACKP \VMEMPAGEP \STREAM \NEW-ATOM)
)
```

(RPAQQ \BUILT-IN-SYSTEM-TYPES

```
((SMALLP)
 (FIXP 2)
 (FLOATP 2)
 (LITATOM)
 (LISTP 4 (0 2))
 (ARRAYP 6 (0))
 (STRINGP 6 (0))
```



```

(STACKP 2 NIL \RECLAIMSTACKP)
(CHARACTER)
(VMEMPAGEP 256 NIL RELEASINGVMEMPAGE)
(STREAM)
(BITMAP)
(COMPILED-CLOSURE 4 (0 2))
(ONED-ARRAY 8 (0))
(TWOD-ARRAY 10 (0))
(GENERAL-ARRAY 10 (0 8))
(BIGNUM)
(RATIO)
(COMPLEX)
(PATHNAME)
(NEW-ATOM 10 (2 4 6))
(FILLER22)
(FILLER23)
(FILLER24)
(FILLER25)
(FILLER26)
(FILLER27)
(FILLER28)
(FILLER29)
(FILLER30))

```

```

;; END EXPORTED DEFINITIONS
;; FOLLOWING DEFINITIONS EXPORTED

```

```
(DECLARE%: EVAL@COMPILE
```

```

[BLOCKRECORD DTD ((NIL BITS 2)
                  (DTDOBSOLETE FLAG)
                  (DTDFINALIZABLE FLAG)
                  (DTDNAME POINTER)
                  (DTDCNT0 WORD)
                  (DTDSIZE WORD)
                  (DTDFREE FULLXPOINTER)
                  (DTDLOCKEDP FLAG)
                  (DTDHUNKP FLAG)
                  (DTDGCTYPE BITS 2)
                  (DTDDDESCRS POINTER)
                  (DDTYPEPECS POINTER)
                  (DTDPTRS POINTER)
                  (DTDOLDCNT FIXP)
                  (DTDNEXTPAGE FIXP)
                  (DDTYPEENTRY WORD)
                  ; The word stored in the type table for objects of this type. Hi bits have numberp tags, ref countable, etc.
                  (DTSUPERTYPE WORD)
                  ; True for type of a redeclared datatype--not allowed to allocate more of these
                  ; True if finalization exists for this type
                  ; Type name -- a symbol
                  ; Incremental box count -- this plus DTDOLDCNT is the true box count
                  ; Length of datum in words
                  ; Pointer to first object on free chain, or NIL. Not used for LISTP
                  ; True if objects of this type must be locked down (not pagefault)
                  ; True if this type is used as an array hunk type
                  ; For hunk datatypes, is analogous to arrayblock's GCTYPE
                  ; List of word offsets inside datum where reference-counted pointers are stored -- used by GC
                  ; 'Box count' -- number of objects of this type ever allocated
                  ; Currently only for LISTP pages -- page number of next page on chain of non-full cons pages
                  )
(AccessFNS DTD ((DTDCNTLOC (\ADDBASE DATUM 4)
                       (DTDCNT (IPLUS (fetch DTDOLDCNT DATUM)
                                       (fetch DTDCNT0 DATUM))
                       (UNINTERRUPTABLY
                        (replace DTDOLDCNT of DATUM with NEWVALUE)
                        (replace DTDCNT0 of DATUM with 0)))
)
)

```

```
(DECLARE%: EVAL@COMPILE
```

```
(PUTPROPS \GETDTD MACRO ((typeNum)
                        (ADDBASE \DTDSpaceBase (ITIMES typeNum 18))))
)

```

```
(DEFOPTIMIZER \TYPEMASK.UFN (&REST X)
  (LET [(CE (CONSTANTEXPRESSIONP (CADR X)
                                (if CE
                                  then `( (OPCODES TYPEMASK.N , (CAR CE))
                                          , (CAR X))
                                  else ' IGNOREMACRO)))]
)

```

```
(DECLARE%: EVAL@COMPILE
```

```
(RPAQQ \GUARDSTORAGEFULL 128)
```

```
(RPAQQ \GUARD1STORAGEFULL 64)
```

```
(CONSTANTS \GUARDSTORAGEFULL \GUARD1STORAGEFULL)
)

```

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS \NxtMDSPage \LeastMDSPage \SecondArrayPage \SecondMDSPage \MDSFREELISTPAGE \MaxSysTypeNum
\MaxTypeNumber \STORAGEFULL \INTERRUPTSTATE \PENDINGINTERRUPT)
)

:: END EXPORTED DEFINITIONS

(RPAQQ STORAGEFULLSTATES ((\SFS.NORMAL NIL)
(\SFS.NOTSWITCHABLE 1)
(\SFS.SWITCHABLE 2)
(\SFS.ARRAYSWITCHED 3)
(\SFS.FULLYSWITCHED 4)))

(DECLARE%: EVAL@COMPILE

(RPAQQ \SFS.NORMAL NIL)

(RPAQQ \SFS.NOTSWITCHABLE 1)

(RPAQQ \SFS.SWITCHABLE 2)

(RPAQQ \SFS.ARRAYSWITCHED 3)

(RPAQQ \SFS.FULLYSWITCHED 4)

(CONSTANTS (\SFS.NORMAL NIL)
(\SFS.NOTSWITCHABLE 1)
(\SFS.SWITCHABLE 2)
(\SFS.ARRAYSWITCHED 3)
(\SFS.FULLYSWITCHED 4))
)
)

:: for MAKEINIT

(DEFINEQ

(CREATEMDSTYPETABLE

[LAMBDA NIL

; Edited 8-Feb-91 16:10 by jds

:: called only under MAKEINIT to initialize the main data space type table
:: This isn't the only place data-type entries get initialized in the INIT.
:: --\CREATE.SYMBOL takes care of initing atom pages.
:: -- POSTINITARRAYS does some array-space initing
:: -- \ALLOCBLOCK of course creates new pages & inits their entries
:: -- \ALLOCMDSPAGE ditto

(CREATEPAGES \MDSTypeTable \MDSTTsize NIL T)
[PROG (VP)

:: FIRST SET ALL TO NOREF

(SETQ VP 0)
(FRPTQ (UNFOLD \MDSTTsize WORDSPERPAGE)
(\PUTBASE \MDSTypeTable VP \TT.NOREF)
(add VP 1))

:: NOW SET UP SMALLPS

[for SEGMENT in (LIST \SmallPosHi \SmallNegHi)
do (for PAGE from 0 to (SUB1 PAGESPERSEGMENT) by (FOLDLO \MDSIncrement WORDSPERPAGE)
do (\MAKEMDSENTRY (LOGOR PAGE (UNFOLD SEGMENT PAGESPERSEGMENT))
(LOGOR \TT.NOREF \TT.FIXP \TT.NUMBERP \TT.ATOM \SMALLP)
(for PAGE from 0 to (SUB1 PAGESPERSEGMENT) by (FOLDLO \MDSIncrement WORDSPERPAGE)
do (\MAKEMDSENTRY (LOGOR PAGE (UNFOLD \CHARHI PAGESPERSEGMENT))
(LOGOR \TT.NOREF \CHARACTERP))

(CREATEPAGES \MISCSTATS (FOLDLO \MDSIncrement WORDSPERPAGE)
NIL T)
(\MAKEMDSENTRY (PAGELOC \MISCSTATS)
(LOGOR \TT.NOREF \TT.FIXP \TT.NUMBERP \TT.ATOM \FIXP])

(INITDATATYPES

[LAMBDA NIL

; Edited 9-Feb-91 17:49 by jds

::: Called only under MAKEINIT. Create the initial data type table from the info in the list INITIALDTDCONTENTS, whose elements are in type number
::: order and of the form (name size pointer-fields finalization). Called before it is possible to make new atoms, so the DTDNAME field will not be filled in
::: until INITDATATYPENAMES runs. We have to run this before turning on atoms so that we can create strings and pnames.

(LET [(NSYSTYPES (ALLOCAL (LENGTH INITIALDTDCONTENTS)
(CREATEPAGES \DTDSpaceBase 1 NIL T)

:: First DTD page is locked, probably because CONS microcode touches the listp dtd. Not sure this is essential

(CREATEPAGES (\ADDBASE \DTDSpaceBase WORDSPERPAGE)
(SUB1 (FOLDHI (ADD1 (TIMES (ADD1 NSYSTYPES)
\DTDSize)))

```

        WORDSPERPAGE))
;; Create the rest of the pages we will need for initial dtd. They need not be locked. (ADD1 NSYSTYPES) is because nonexistent type zero
;; occupies table space
;; (ADD1 (TIMES ...)) is because you've got to create the next page for DTD's if you allocate the last one on a page. This arose when I
;; increased the # of system types, and we wound up with NSYSTYPES = 63. Result: Illegal addr in the INIT when it tried to allocate the next
;; DTD. --JDS
[for D in (LOCAL INITIALDTDCONTENTS) bind DTD as TYPENO from 1
do
  ;; Run thru the initial data type decls (the gut-level system datatypes), and declare them in the INIT.DLINIT.
  (SETQ DTD (\GETDTD TYPENO)) ; Create a Data-Type-Descriptor for the new type
  [replace DTDTYPEENTRY of DTD with (LOGOR TYPENO (COND
    ([ALLOCAL (FMEMB (CAR D)
      ' (SMALLP FIXP FLOATP)
      \TT.NUMBERP)
      (T 0))
    (COND
      ([ALLOCAL (FMEMB (CAR D)
        ' (SMALLP FIXP FLOATP LITATOM NEW-ATOM)
        \TT.ATOM)
        (T 0))
      (COND
        ([ALLOCAL (FMEMB (CAR D)
          ' (SMALLP FIXP]
          \TT.FIXP)
          (T 0))
        (COND
          ((ALLOCAL (EQ (CAR D)
            'NEW-ATOM))
            ; Add NewAtom Entry '90/07/18 ON
            \TT.NOREF)
            (T 0))
          (COND
            ([ALLOCAL (FMEMB (CAR D)
              ' (LITATOM NEW-ATOM)
              ; FOR TYPE TESTING BY TYPEMASK.
              (CONSTANT \TT.SYMBOLP))
              (T 0))
            (COND
              ((ALLOCAL (NOT (CADR D)))
              ; no size, no ref. For those types that are really declared later on,
              ; \ASSIGNDATATYPE1 will fix DTDTYPEENTRY to be correct
              \TT.NOREF)
              (T 0) ; Set up the type-mask field with the appropriate meta-type bits

          (COND
            ((EQ (CAR D)
              'NEW-ATOM)
              ;; For NEW-ATOM, mark it a subtype of LITATOM.
              (replace DTDSUPERTYPE of DTD with \LITATOM)))
          (COND
            ((ALLOCAL (AND (CAR D)
              (CADR D)))
              ; Set the data type's size
              (replace DTDSIZE of DTD with (LOCAL (CADR D)

[COND
  ((NEQ CDRCODING 0)
  (SETQ.NOREF \LISTPDTD (\GETDTD \LISTP]
  (SETQ \MaxSysTypeNum (SETQ \MaxTypeNumber NSYSTYPES))
  NIL])

```

(INITDATATYPENAMES

[LAMBDA NIL

; Edited 2-Apr-91 02:17 by sybalsky

;;; Called in MAKEINIT after it is ok to create arrays and new atoms. Here we finish initializing the data type tables -- fill in type names and the list of pointers. Also set finalization for built-in types.

;; Because this is running in the INIT, everything really HAS to be atom numbers, so leave the \ATOMPNAMINDEX call alone in tjhis function.

```

(SETQ \FINALIZATION.FUNCTIONS (\ALLOCBLOCK (ADD1 \EndTypeNumber)
  T))

```

```

[for D in (LOCAL INITIALDTDCONTENTS) as NTYPX from 1
do (LET [(DTD (\GETDTD NTYPX))

```

```

(FINAL (LOCAL (CADDR D) ; d = (name size ptrs finalization)
[replace DTDNAME of DTD with (\ATOMPNAMINDEX (LOCAL (CAR D)

```

```

; Smash the name from our world into his
[replace DTDPTRS of DTD with (COPY (LOCAL (CADDR D) ; And the list of pointer offsets
(if FINAL

```

```

then
(replace DTDFINALIZABLE of DTD with T) ; Set finalization for this type

```

```

(\PUTBASEPTR \FINALIZATION.FUNCTIONS (UNFOLD NTYPX WORDSPERCELL)
(COPY FINAL)

```

```

(PROGN ; Do finalization for array blocks (type 0) specially to avoid
; incompatible change to BUILT-IN-SYSTEM-TYPES

```

```

(replace DTDFINALIZABLE of (\GETDTD 0) with T)
(\PUTBASEPTR \FINALIZATION.FUNCTIONS 0 (COPY '\RECLAIMARRAYBLOCK])

```

```

)
(DECLARE%: DONTCOPY
(ADDTOVAR INITVALUES (\NxtMDSPage \FirstMDSPage)
              (\LeastMDSPage \FirstMDSPage)
              (\SecondMDSPage \DefaultSecondMDSPage)
              (\SecondArrayPage \DefaultSecondArrayPage)
              (\MDSFREELISTPAGE)
              (\MaxSysTypeNum 0)
              (\MaxTypeNumber))
(ADDTOVAR INITPTRS (\FINALIZATION.FUNCTIONS))
(ADDTOVAR NEWCOMS (FNS NTYPX \ALLOCMDSPAGE \MAKEMDSENTRY \INITMDSPAGE \ASSIGNDATATYPE1 \TYPENUMBERFROMNAME
              \CREATECELL \NEW2PAGE)
              (FNS CREATEMDSTYPETABLE INITDATATYPES INITDATATYPENAMES)
              (VARS \BUILT-IN-SYSTEM-TYPES))
(ADDTOVAR RDCOMS (FNS NTYPX TYPENAME \TYPENAMEFROMNUMBER))
(ADDTOVAR RDVALS (\MaxTypeNumber))
(ADDTOVAR RD.SUBFNS (\ARRAYTYPENAME LAMBDA (X)
              'ARRAYP))
(ADDTOVAR EXPANDMACROFNS \GETDTD PUTBASEPTRX REPLACEFIELD FETCHFIELD \GETBITS \PUTBITS \TESTBITS GETBASEBITS
              PUTBASEBITS FFETCHFIELD FREPLACEFIELD FREPLACEFIELDVAL REPLACEFIELDVAL NCREATE
)
(ADDTOVAR MKI.SUBFNS (\GCDISABLED . NIL)
              (CREATECELL . I.\CREATECELL)
              (\CHECKFORSTORAGEFULL . NIL))
(ADDTOVAR DONTCOMPILEFNS CREATEMDSTYPETABLE INITDATATYPES INITDATATYPENAMES)
)
(DECLARE%: DOEVAL@COMPILE DONTCOPY
(LOCALVARS . T)
)
(PUTPROPS LLDATATYPE FILETYPE CL:COMPILE-FILE)
(DECLARE%: EVAL@COMPILE DONTCOPY
(FILESLOAD (LOADCOMP)
              DTDECLARE)
)
(PUTPROPS LLDATATYPE COPYRIGHT ("VENUE, Oakland, CA" 1982 1983 1984 1985 1986 1987 1988 1989 1990 1991 1992
              1993 1994 1995 1999))

```

FUNCTION INDEX

/REPLACEFIELD	11	NCREATE2	11	\ASSIGNDATATYPE1	5	\SET.STORAGE.STATE	4
BOXCOUNT	9	NTYPX	2	\BLOCKDATAP	12	\SETTYPMASK	4
CONSCOUNT	9	PUTBASEPTRX	11	\CHECKFORSTORAGEFULL	3	\STLNP	15
CREATECELL	7	REPLACEFIELD	8	\CREATECELL	7	\STMDS.APPROX	15
CREATMDSTYPETABLE	18	REPLACEFIELDVAL	11	\DOSTORAGEFULLINTERRUPT	4	\STMDSTYPE	15
DATATYPEP	12	STORAGE	13	\DTEST	9	\STORAGE.HUNKTYPE	16
DATATYPES	12	STORAGE.LEFT	14	\DTEST.UFN	9	\STORAGE.TYPE	14
FETCHFIELD	8	TYPENAME	12	\INITMDSPAGE	5	\TYPECHECK	9
GETDESCRIPTORS	11	TYPENAMEP	12	\INSTANCE-P	10	\TYPECHECK.UFN	10
GETFIELDSPECS	11	USERDATATYPES	12	\INSTANCEP.UFN	10	\TYPMASK.UFN	2
GETSUPERTYPE	11	\ADVANCE.STORAGE.STATE	4	\MAKEMDSENTRY	5	\TYPENAMEFROMNUMBER	12
INITDATATYPENAMES	19	\ALLOCMDSPAGE	2	\MAPMDS	3	\TYPENUMBERFROMNAME	7
INITDATATYPES	18	\ALLOCPAGEBLOCK	2	\NEW2PAGE	5	\TYPEP.UFN	2
NCREATE	11	\ALLOCVIRTUALPAGEBLOCK	3	\RESOLVE.TYPENUMBER	7		

VARIABLE INDEX

ASSIGNDATATYPE.ASKUSERWAIT	7	INITVALUES	20	STORAGE.ARRAYSIZES	16
CROSSCOMPILING	7	MKI.SUBFNS	20	STORAGEFULLSTATES	18
DONTCOMPILEFNS	20	RD.SUBFNS	20	\BUILT-IN-SYSTEM-TYPES	16
EXPANDMACROFNS	20	RDCOMS	20	\STORAGEFULL	7
INEWCOMS	20	RDVALS	20	\STORAGEFULLSTATE	7
INITPTRS	20	SP	7		

CONSTANT INDEX

\ARRAYP	16	\LISTP	16	\SFS.NORMAL	18	\STREAM	16
\FIXP	16	\LITATOM	16	\SFS.NOTSWITCHABLE	18	\VMEMPAGEP	16
\FLOATP	16	\NEW-ATOM	16	\SFS.SWITCHABLE	18		
\GUARD1STORAGEFULL	17	\SFS.ARRAYSWITCHED	18	\SMALLP	16		
\GUARDSTORAGEFULL	17	\SFS.FULLYSWITCHED	18	\STACKP	16		

OPTIMIZER INDEX

PUTBASEPTRX	16	TYPENAMEP	12	\INSTANCE-P	13	\TYPMASK.UFN	17
-------------------	----	-----------------	----	-------------------	----	--------------------	----

RECORD INDEX

DTD	17	HUNKSTAT	16
-----------	----	----------------	----

PROPERTY INDEX

LLDATATYPE	20
------------------	----

MACRO INDEX

\GETDTD	17
---------------	----
