

File created: 10-May-2025 11:09:24 {DSK}<home>runner>work>medley>medley>loadups>build>I-NEW.;

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

(RPAQQ I-NEWCOMS

```
((ADDVARS (LOCKEDFNS \CLOCK0 \GETINTERNALCLOCK \BOXIDIFFERENCE \BOXIPLUS \BLT \SLOWIQUOTIENT)
  (LOCKEDVARS \RCLKSECOND \RCLKMILLISECOND \MISCSTATS))
(ADDVARS (LOCKEDFNS ERROR RAID \M44ACTIONVMMEMFILE \ACTIONVMMEMFILESUBR \ACTIONVMMEMPAGES \CLEANUPDISKQUEUE
  \CLEARCB \DISKERROR \DOACTIONDISKPAGES \DODISKCOMMAND \EXTENDISFMAP \M44DOEXTENDVMMEMFILE
  \GETDISKCB \INITBFS \INSUREVMMEMFILE \LISPEROR \LOOKUPFMAP \READISKDA \VIRTUALDISKDA
  \CLEARWORDS \TESTPARTITION \EXTENDEDVMMEMINIT \WHICHPART \INITIALIZESWAPDISK
  \SWAPDISKERROR)
  (LOCKEDVARS \DISKREQUESTBLOCK \SWAPREQUESTBLOCK \MAINDISK \SWAPDSK1 \SWAPDSK2 \ISFCHUNKSIZE
    \EMUSCRATCH \EMUDISKBUFFERS \EMUSWAPBUFFERS \EMUDISKBUFEND \MAXSWAPBUFFERS \#DISKBUFFERS
    \InterfacePage \ISFMAP \ISFSCRATCHCAS \ISFSCRATCHDAS \SYSDISK \#SWAPBUFFERS \MAXDISKDas
    \DISKDEBUG \SPAREDISKWRITEBUFFER \#EMUBUFFERS \EMUBUFFERS \LASTVMMEMFILEPAGE \XVmem
    \XVmemFmapBase \XVmemFmapHighBase \XVmemDiskBase))
(FNS I.MAKEINITBFS)
(ADDVARS (LOCKEDFNS FLIPCUSORBAR \SETIOPOINTERS \KEYHANDLER \KEYHANDLER1 \CONTEXTAPPLY \LOCKPAGES
  \DECODETRANSITION \SMASHLINK \INCUSECOUNT LLSH \MAKEFREEBLOCK \DECUSECOUNT \MAKENUMBER
  \ADDBASE \PERIODIC.INTERRUPTFRAME \DOBUFFEREDTRANSITIONS \TIMER.INTERRUPTFRAME
  \CAUSEINTERRUPT \DOMOUSECHORDING \KEYBOARDOFF \TRACKCURSOR \HARDCUSORUP
  \HARDCUSORPOSITION \HARDCUSORDOWN \SOFTCURSORUP \SOFTCURSORUPCURRENT
  \SOFTCURSORPOSITION \SOFTCURSORDOWN \SOFTCURSORPILOTBITBLT)
  (LOCKEDVARS \InterfacePage \CURSORHOTSPOTX \CURSORHOTSPOTY \CURRENTCURSOR \SOFTCURSORWIDTH
    \SOFTCURSORHEIGHT \SOFTCURSORP \SOFTCURSORUPP \SOFTCURSORUPBM \SOFTCURSORDOWNBM
    \SOFTCURSORBBT1 \SOFTCURSORBBT2 \SOFTCURSORBBT3 \SOFTCURSORBBT4 \SOFTCURSORBBT5
    \SOFTCURSORBBT6 \CURSORDESTINATION \CURSORDESTHEIGHT \CURSORDESTWIDTH
    \CURSORDESTRASTERWIDTH \CURSORDESTLINE \CURSORDESTLINEBASE \PENDINGINTERRUPT
    \PERIODIC.INTERRUPT \PERIODIC.INTERRUPT.FREQUENCY \LASTUSERACTION \MOUSECHORDTICKS
    \KEYBOARDDEVENTQUEUE \KEYBUFFERING \SCREENWIDTH \SCREENHEIGHT \TIMER.INTERRUPT.PENDING
    \EM.MOUSEX \EM.MOUSEY \EM.CUSORX \EM.CUSORY \EM.UTILIN \EM.REALUTILIN \EM.KBDAD0
    \EM.KBDAD1 \EM.KBDAD2 \EM.KBDAD3 \EM.DISINTERRUPT \EM.CUSORBITMAP \EM.KBDAD4 \EM.KBDAD5
    \MISCSTATS \RCLKSECOND))
(FNS I.\LOCKFN I.\LOCKVAR I.\LOCKCELL I.\LOCKWORDS I.\LOCKCODE)
(ADDVARS (LOCKEDFNS \FAULTHANDLER \FAULTINIT \DOVE.FAULTINIT \D01.FAULTINIT \DL.FAULTINIT \CHAIN.UP.RPT
  \MAKESPACEFORLOCKEDPAGE \PAGEFAULT \WRITEMAP \LOOKUPPAGE \LOCKEDPAGE \LOADVMMEMPAGE
  \MOVEREALPAGE \INVALIDADDR \INVALIDVP \SELECTREALPAGE \TRANSFERPAGE \SPECIALRP
  \UPDATECHAIN \MARKPAGEVACANT \FLUSHPAGE \CLEARWORDS \FLUSHVM \DNEWPAGE
  \ASSURE.FPTOVP.PAGE \DNEWEPHEMERALPAGE \WRITEDIRTYPAGE1 \COPYSYS0 \COPYSYS0SUBR
  \RELEASEWORKINGSET \DOFLUSHVM \DOLOCKPAGES \DOTEMPLOCKPAGES \TEMPUNLOCKPAGES \MP.ERROR
  RAID \DL.NEWFAULTINIT \DL.MARK.PAGES.UNAVAILABLE \DL.UNMAPPAGES \DL.ASSIGNBUFFERS
  \D01.ASSIGNBUFFERS \DOCOMPRESSVMEM \MOVEVMMEMFILEPAGE \SET.VMEM.FULL.STATE \HINUM \LONUM
  \ATOMCELL SETTOPVAL)
  (LOCKEDVARS \REALPAGETABLE \RPTLAST \PAGEFAULTCOUNTER \UPDATECHAINFREQ \RPOFFSET \RPTSIZE
    \LOCKEDPAGETABLE \EMBUFBASE \EMBUFRP \LASTACCESSEDVMMEMPAGE \MAXSHORTSEEK
    \MAXCLEANPROBES \MINSHORTSEEK \DIRTYPAGECOUNTER \DIRTYPAGEHINT \VMEM.INHIBIT.WRITE
    \VMEM.PURE.LIMIT \VMEM.FULL.STATE \GUARDVMMEMFULL \VMEM.COMPRESS.FLG \KBDSTACKBASE
    \MISCSTACKBASE \DOFAULTINIT \FPTOVP \MACHINEYPE \VMEMACCESSFN \TELERAIDBUFFER
    \EMUDISKBUFFERS \EMUDISKBUFEND \MAXSWAPBUFFERS \EMUBUFFERS \#EMUBUFFERS \#SWAPBUFFERS
    \#DISKBUFFERS \RCLKSECOND \RCLKMILLISECOND \VALSPACE \EMUSWAPBUFFERS \EM.CUSORBITMAP
    \PAGEMAP \PageMapTBL \IOCPAGE \IOPAGE \MISCSTATS \DEFSpace \InterfacePage
    \LASTVMMEMFILEPAGE \DoveIORegion \MaxScreenPage \NEWVMMEMPAGEADDED))
(FNS I.DUMPINITPAGES)
(VARS INITCONSTANTS)
(FNS I.SETUPPAGEMAP I.ADDPME I.MAKEROOMFORPME I.MAPPAGES)
(FNS I.SETUPSTACK I.\SETUPSTACK1 I.\SETUPGUARDBLOCK I.\MAKEFREEBLOCK)
(ADDVARS (LOCKEDFNS \RESETSTACK0 \MAKEFRAME \SETUPSTACK1 \MAKEFREEBLOCK \FAULTHANDLER \KEYHANDLER
  \DUMMYKEYHANDLER \DOTELERAID \DUMMYTELERAID \DOHARDRETURN \DOGC \CAUSEINTERRUPT
  \INTERRUPTFRAME \CODEFORTFRAME \DOSTACKOVERFLOW \UNLOCKPAGES \DOMISCAPPLY)
  (LOCKEDVARS \InterfacePage \DEFSpace \STACKSPACE \KBDSTACKBASE \MISCSTACKBASE \SAVED.USER.CONTEXT
    \RUNNING.PROCESS \NEED.HARDRESET.CLEANUP))
(FNS I.INITGC)
(FNS I.NTYPX I.\ALLOCMDSPAGE I.\MAKEMDSENTRY I.\INITMDSPAGE I.\ASSIGNDATATYPE1 I.\TYPENUMBERFROMNAME
  I.\CREATECELL I.\NEW2PAGE)
(FNS I.CREATEMDSTYPETABLE I.INITDATATYPES I.INITDATATYPENAMES)
(VARS \BUILT-IN-SYSTEM-TYPES)
(FNS I.FSETVAL I.SETPROPLIST I.PUTDEFN I.\BLT)
(FNS I.\MKATOM I.\CREATE.SYMBOL I.\INITATOMPAGE I.\MOVEBYTES I.\STKMIN)
(FNS I.COPYATOM I.INITATOMS)
(FNS I.MAKEINITFIRST I.\COPY I.MAKEINITLAST)
(FNS I.\CONS.UFN I.\MAIKO.CONS.UFN I.\INITCONSPAGE I.\NEXTCONSPAGE)
(FNS I.\GETBASEBYTE I.\PUTBASEBYTE I.CREATEPAGES I.\NEW4PAGE)
(FILES (SYSLOAD FROM VALUEOF DIRECTORIES)
  CMLARRAY-SUPPORT)
(FNS I.ALLOCSTRING I.%%COPY-ONED-ARRAY I.%%COPY-STRING-TO-ARRAY)
(FNS I.\#BLOCKDATACELLS I.\PREFIXALIGNMENT? I.\ALLOCBLOCK I.\MAIKO.ALLOCBLOCK I.\ALLOCBLOCK.NEW
  I.\MAKEFREEARRAYBLOCK I.\MERGEBACKWARD I.\LINKBLOCK I.\ALLOCHUNK)
(FNS I.PREINITARRAYS I.POSTINITARRAYS I.FILEARRAYBASE I.FILEBLOCKTRAILER I.FILECODEBLOCK
  I.FILEPATCHBLOCK)
(FNS I.\SETUP.HUNK.TYPENUMBERS I.\COMPUTE.HUNK.TYPEDECLS I.\TURN.ON.HUNKING I.\SETUP.TYPENUM.TABLE)
(FNS I.DCODERD)
[VAR \OPCODES (I.CODERDTBL (COPYREADTABLE 'ORIG)
(P (SETSYNTAX (CHARCODE ^Y)
```

```

      '[MACRO (LAMBDA (FILE RDTBL)
                (EVALFORMAKEINIT (READ FILE RDTBL)
                I.CODERDTBL)
      (SETSYNTAX (CHARCODE %|))
      '(MACRO ALWAYS READVBAR)
      I.CODERDTBL)
      (READTABLEPROP I.CODERDTBL 'USESILPACKAGE NIL))
      (FNS I.INITUFNTABLE I.\SETUFNENTRY)
      (VARS INITPTRS INITVALUES)
      (DECLARE%: EVAL@COMPILE DONTCOPY (FILES (LOADCOMP)
                MAKEINIT))))

(ADDTTOVAR LOCKEDFNS \CLOCK0 \GETINTERNALCLOCK \BOXIDIFFERENCE \BOXIPLUS \BLT \SLOWIQUOTIENT)

(ADDTTOVAR LOCKEDVARS \RCLKSECOND \RCLKMILLISECOND \MISCSTATS)

(ADDTTOVAR LOCKEDFNS ERROR RAID \M44ACTONVMEMFILE \ACTONVMEMFILESUBR \ACTONVMEMPAGES \CLEANUPDISKQUEUE \CLEARCB
\DISKERROR \DOACTONDISKPAGES \DODISKCOMMAND \EXTENDISFMAP \M44DOEXTENDVMEMFILE
\GETDISKCB \INITBFS \INSUREVMEMFILE \LISPERERROR \LOOKUPFMAP \REALDISKDA \VIRTUALDISKDA
\CLEARWORDS \TESTPARTITION \EXTENDEDVMEMINIT \WHICHPART \INITIALIZESWAPDISK
\SWAPDISKERROR)

(ADDTTOVAR LOCKEDVARS \DISKREQUESTBLOCK \SWAPREQUESTBLOCK \MAINDISK \SWAPDSK1 \SWAPDSK2 \ISFCHUNKSIZE
\EMUSCRATCH \EMUDISKBUFFERS \EMUSWAPBUFFERS \EMUDISKBUFEND \MAXSWAPBUFFERS
\#DISKBUFFERS \InterfacePage \ISFMAP \ISFSCRATCHCAS \ISFSCRATCHDAS \SYSDISK
\#SWAPBUFFERS \MAXDISKDas \DISKDEBUG \SPAREDISKWRITEBUFFER \#EMUBUFFERS \EMUBUFFERS
\LASTVMEMFILEPAGE \Xvmem \XvmemFmapBase \XvmemFmapHighBase \XvmemDiskBase)

(DEFINEQ

(I.MAKEINITBFS
[LAMBDA NIL
      (PROGN
        (I.\LOCKWORDS (SETQ I.MAINDISK ([LAMBDA ($$1)
          (I.PUTBASEPTR $$1 0 NIL)
          $$1]
          (I.\ALLOCBLOCK 18)))
          36)
        (I.PUTBASEPTR I.MAINDISK 26 (I.\COPY 'DSK))
        (I.\LOCKWORDS (SETQ I.SWAPDSK1 ([LAMBDA ($$1)
          (I.PUTBASEPTR $$1 0 NIL)
          $$1]
          (I.\ALLOCBLOCK 18)))
          36)
        (I.\LOCKWORDS (SETQ I.SWAPDSK2 ([LAMBDA ($$1)
          (I.PUTBASEPTR $$1 0 NIL)
          $$1]
          (I.\ALLOCBLOCK 18)))
          36)
        (PROGN
          (I.\LOCKWORDS (SETQ I.SWAPREQUESTBLOCK (I.\ALLOCBLOCK (LRSH (IPLUS (IPLUS 42 60)
1)
1)))
          (+ 42 60))
          (I.\LOCKWORDS (SETQ I.DISKREQUESTBLOCK (I.\ALLOCBLOCK (LRSH (IPLUS (IPLUS 42 60)
1)
1)))
          (+ 42 60)))
          (to 3 bind PREV (FIRSTCB _ (I.\ALLOCBLOCK 3)) first (I.\LOCKWORDS (SETQ PREV FIRSTCB)
6)
          do (I.\LOCKWORDS (SETQ PREV ([LAMBDA ($$1)
            (PROG1 (SETQ $$1 (I.\ALLOCBLOCK 3))
              (I.PUTBASEPTR $$1 0 PREV]
              NIL))
            6)
          finally (I.PUTBASEPTR FIRSTCB 0 PREV)
              (I.PUTBASEPTR I.MAINDISK 14 FIRSTCB))
          (SETQ I.FREEPAGEFID (I.\ALLOCBLOCK 3))
          (for I from 0 to 4 do (I.PUTBASE I.FREEPAGEFID I (UNSIGNED -1 16]))
        )
      )

(ADDTTOVAR LOCKEDFNS FLIPCUSORBAR \SETIOPPOINTERS \KEYHANDLER \KEYHANDLER1 \CONTEXTAPPLY \LOCKPAGES
\DECODETRANSITION \SMASHLINK \INCUSECOUNT LSH \MAKEFREEBLOCK \DECUSECOUNT
\MAKENUMBER \ADDBASE \PERIODIC.INTERRUPTFRAME \DOBUFFEREDTRANSITIONS
\TIMER.INTERRUPTFRAME \CAUSEINTERRUPT \DOMOUSECHORDING \KEYBOARDOFF \TRACKCURSOR
\HARDCUSORUP \HARDCUSORPOSITION \HARDCUSORDOWN \SOFTCURSORUP \SOFTCURSORUPCURRENT
\SOFTCURSORPOSITION \SOFTCURSORDOWN \SOFTCURSORPILOTBITBLT)

(ADDTTOVAR LOCKEDVARS \InterfacePage \CURSORHOTSPOTX \CURSORHOTSPOTY \CURRENTCURSOR \SOFTCURSORWIDTH
\SOFTCURSORHEIGHT \SOFTCURSORP \SOFTCURSORUPP \SOFTCURSORUPBM \SOFTCURSORDOWNBM
\SOFTCURSORBBT1 \SOFTCURSORBBT2 \SOFTCURSORBBT3 \SOFTCURSORBBT4 \SOFTCURSORBBT5
\SOFTCURSORBBT6 \CURSORDESTINATION \CURSORDESTHEIGHT \CURSORDESTWIDTH
\CURSORDESTRASTERWIDTH \CURSORDESTLINE \CURSORDESTLINEBASE \PENDINGINTERRUPT
\PERIODIC.INTERRUPT \PERIODIC.INTERRUPT.FREQUENCY \LASTUSERACTION \MOUSECHORDTICKS
\KEYBOARDEVENTQUEUE \KEYBUFFERING SCREENWIDTH SCREENHEIGHT \TIMER.INTERRUPT.PENDING

```

```

\EM.MOUSEX \EM.MOUSEY \EM.CURSORK \EM.CURSORY \EM.UTILIN \EM.REALUTILIN \EM.KBDAD0
\EM.KBDAD1 \EM.KBDAD2 \EM.KBDAD3 \EM.DISPIINTERUPT \EM.CURSORBITMAP \EM.KBDAD4
\EM.KBDAD5 \MISCSTATS \RCLKSECOND)

```

(DEFINEQ

(I.\LOCKFN

```

[LAMBDA (FN) (*)
  (I.\LOCKCELL (SETQ FN (I.\ATOMCELL (PROGN (I.\COPY FN))
                                           10)))
  (COND
    ((NEQ 0 (LRSH (I.GETBASE FN 0)
                  15))
     (I.\LOCKCODE (I.GETBASEPTR FN 0]))
  )

```

(I.\LOCKVAR

```

[LAMBDA (VAR) (*)
  (I.\LOCKCELL (I.\ATOMCELL (PROGN (I.\COPY VAR)
                                     12))

```

(I.\LOCKCELL

```

[LAMBDA (X NPGS) (*)
  (MKI.LOCKPAGES (I.VAG2 (I.HILOC X)
                          (LOGAND (I.LOLOC X)
                                   65280))
    (OR NPGS 1])

```

(I.\LOCKWORDS

```

[LAMBDA (BASE NWORDS) (*)
  (MKI.LOCKPAGES (I.VAG2 (I.HILOC BASE)
                          (LOGAND (I.LOLOC BASE)
                                   65280))
    (COND
      (NWORDS (LRSH (IPLUS (IPLUS (LOGAND (I.LOLOC BASE)
                                             255)
                                      NWORDS)
                      255)
                8))
      (T 1])

```

(I.\LOCKCODE

```

[LAMBDA (CODEBLOCK) (*)
  (I.\LOCKWORDS CODEBLOCK (LLSH (I.\BLOCKDATA CELLS CODEBLOCK)
                                 1])

```

```

(ADDTTOVAR LOCKEDFNS \FAULTHANDLER \FAULTINIT \DOVE.FAULTINIT \D01.FAULTINIT \DL.FAULTINIT \CHAIN.UP.RPT
\MAKESPACEFORLOCKEDPAGE \PAGEFAULT \WRITEMAP \LOOKUPPAGE MAP \LOCKEDPAGEP
\LOADVMEMPAGE \MOVEREALPAGE \INVALIDADDR \INVALIDVP \SELECTREALPAGE \TRANSFERPAGE
\SPECIALRP \UPDATECHAIN \MARKPAGEVACANT \FLUSHPAGE \CLEARWORDS \FLUSHVM \DNEWPAGE
\ASSURE.FPTOVP.PAGE \DNEWEPHEMERALPAGE \WRITEDIRTYPAGE1 \COPYSYS0 \COPYSYS0SUBR
\RELEASEWORKINGSET \DOFLUSHVM \DOLOCKPAGES \DOTEMPLOCKPAGES \TEMPUNLOCKPAGES
\MP.ERROR RAID \DL.NEWFAULTINIT \DL.MARK.PAGES.UNAVAILABLE \DL.UNMAPPAGES
\DL.ASSIGNBUFFERS \D01.ASSIGNBUFFERS \DOCOMPRESSVMEM \MOVEVMEMFILEPAGE
\SET.VMEM.FULL.STATE \HINUM \LONUM \ATOMCELL SETTOPVAL)

```

```

(ADDTTOVAR LOCKEDVARS \REALPAGETABLE \RPTLAST \PAGEFAULTCOUNTER \UPDATECHAINFREQ \RPOFFSET \RPTSIZE
\LOCKEDPAGETABLE \EMBUFBASE \EMBUFVP \EMBUFRP \LASTACCESSEDVMEMPAGE \MAXSHORTSEEK
\MAXCLEANPROBES \MINSHORTSEEK \DIRTYPAGECOUNTER \DIRTYPAGEHINT \VMEM.INHIBIT.WRITE
\VMEM.PURE.LIMIT \VMEM.FULL.STATE \GUARDVMEMFULL \VMEM.COMPRESS.FLG \KBDSTACKBASE
\MISCSTACKBASE \DOFAULTINIT \FPTOVP \MACHINETYPE \VMEMACCESSFN \TELERAIDBUFFER
\EMUDISKBUFFERS \EMUDISKBUFEND \MAXSWAPBUFFERS \EMUBUFFERS \#EMUBUFFERS
\#SWAPBUFFERS \#DISKBUFFERS \RCLKSECOND \RCLKMILLISECOND \VALSPACE \EMUSWAPBUFFERS
\EM.CURSORBITMAP \PAGE MAP \PageMapTBL \IOCBPAGE \IOPAGE \MISCSTATS \DEFSpace
\InterfacePage \LASTVMEMFILEPAGE \DoveIORegion \MaxScreenPage \NEWVMEMPAGEADDED)

```

(DEFINEQ

(I.DUMPINITPAGES

```

[LAMBDA (CODEFIRSTPAGE CODENEXTPAGE VERSIONS) (*)
  (I.ADDPME (IPLUS (LLSH (I.HILOC (I.VAG2 20 0))
                       8)
              (LRSH (I.LOLOC (I.VAG2 20 0))
                     8))
    T)
  (for I from CODEFIRSTPAGE to (SUB1 CODENEXTPAGE) do (I.ADDPME I T))
  (I.MAPPAGES 0 (ADD1 131069)
    (FUNCTION I.MAKEROOMFORPME))
  (I.MAPPAGES 0 (ADD1 131069)
    (FUNCTION I.ADDPME))

```

```

(PROGN
  (I.PUTBASE (I.VAG2 20 0)
    19 NEXTPM)
  (I.PUTBASEFIXP (I.VAG2 20 0)
    82
    (SUB1 NEXTVMEM))
  (I.PUTBASEFIXP (I.VAG2 20 0)
    84
    (SUB1 NEXTVMEM))
  (I.PUTBASE (I.VAG2 20 0)
    22
    (I.GETBASE (I.VAG2 5 0)
      0))
  (I.PUTBASE (I.VAG2 20 0)
    23
    (I.GETBASE ([LAMBDA (VPAGE)
      (DECLARE (LOCALVARS VPAGE))
      (I.ADDBASE (I.VAG2 5 0)
        (IPLUS (I.GETBASE (I.VAG2 20 512)
          (LRSH VPAGE 5))
          (LOGAND VPAGE 31]
        (IPLUS (LLSH (I.HILOC (I.VAG2 20 512))
          8)
          (LRSH (I.LOLOC (I.VAG2 20 512))
            8)))
        0))
    [COND
      (VERSIONS (I.PUTBASE (I.VAG2 20 0)
        8
        (CAR VERSIONS))
        (I.PUTBASE (I.VAG2 20 0)
          10
          (CADDR VERSIONS))
        (I.PUTBASE (I.VAG2 20 0)
          9
          (CADR VERSIONS])
        (I.PUTBASE (I.VAG2 20 0)
          15 5603))
      (I.MAPPAGES 0 (ADD1 131069)
        (FUNCTION DUMPVP))
      (PROG ((FILE (OUTPUT)))
        [COND
          ((NOT (RANDACCESSP FILE))
            ; SYSOUT file is sequential; have to get it random access for
            ; this
            (OUTPUT (SETQ FILE (OPENFILE (CLOSEF FILE)
              'BOTH])
              (SETFILEPTR FILE MKI.Page0Byte))
            (DUMPVP (IPLUS (LLSH (I.HILOC (I.VAG2 20 0))
              8)
              (LRSH (I.LOLOC (I.VAG2 20 0))
                8]))
          )
        )
      (RPAQQ INITCONSTANTS
        (
          ;; (LISPNAME VALUE BCPLNAME UCODENAME)

          (CDRCODING 1 T T)
          ; IF CDRCODING=0, CDR CODING IS OFF, OTHERWISE ON

          ;; type numbers -- repeated on LLBASiC too
          (\SMALLP 1 SMALLTYPE SmallType)
          (\FIXP 2 INTEGERTYPE FixpType)
          (\FLOATP 3 FLTPPTYPE FloatpType)
          (\LITATOM 4 ATOMTYPE AtomType)
          (\LISTP 5 LISTTYPE ListType)
          (\ARRAYP 6 ARRAYPTRTYPE ArrayType)
          (\STRINGP 7 STRINGPTRTYPE)
          (\STACKP 8)
          (\CHARACTERP 9)
          (\VMEMPAGEP 10 NIL VMemPagePType)
          (\STREAM 11 NIL STREAMTYPE)

          ;; TYPE TABLE CONSTANTS -----
          (\TT.TYPEMASK 2047 TTTypeMask T)
          (\TT.NOREF 32768 NIL T)
          (\TT.SYMBOLP 16384 NIL T)
          (\TT.FIXP 8192)
          (\TT.NUMBERP 4096)
          (\TT.ATOM 2048)

          ;; page map -----
          (\PMBlockSize 32 PMBLOCKSIZE)
          (\STATSize 8 T)
          (\NumPMTpages 8)

```

```

(\EmptyPMTEntry 65535 T)
(\FirstVmemBlock 2 T)
(\MAXVMPAGE 131069)
(\MAXVMSEGMENT 255)
;; interface page
(\IFPValidKey 5603 T)
;; MDS
(\FirstMDSPage 32766)
(\MaxMDSPage 524285)
(\DefaultSecondMDSPage 65532)
(\MDSIncrement 512)
(\PagesPerMDSUnit 2)
;; arrays
(\ARRAYSPACE (46 0))
(\FirstArraySegment 46)
(\FirstArrayPage 11776)
(\ARRAYSPACE2 (64 0))
(\DefaultSecondArrayPage 32768)
;; stack block constants
(\StackMask 57344 T T)
(\FxtnBlock 49152 T T)
(\GuardBlock 57344 T T)
(\BFBlock 32768 T T)
(\FreeStackBlock 40960 T T)
(\NotStackBlock 0)
(\MinExtraStackWords 32 T T)
;; backspace kludge
(ERASECHARCODE 0 T)
;; GC constants
(\HTICNT 1024 NIL T)
(\HTSTKBIT 512 NIL T)
(\HTCNTMASK 64512 NIL T)
(\HTMAINSIZE 65536 NIL T)
(\HTCOLLSIZE 1048576 NIL T)
(\HTENDFREE 1 NIL T)
(\HTFREEPTR 0 NIL T)
;; pointers and lengths of various data spaces
(\ATOMSPACE (0 0)
  (ATOMspace NIL)
  (atomHiVal NIL))
(\AtomHI 0)
(\CHARHI 7)
(\AtomHashTable (21 0)
  (AHTspace AHTbase))
(\AtomHTpages 256 AHTSIZE)
(\LastAtomPage 255)
(\MaxAtomFrLst 65535)
(\SMALLPOSPSPACE (14 0))
(\SmallPosHi 14 SMALLPOSpace smallpl)
(\SMALLNEGSPACE (15 0))
(\SmallNegHi 15 SMALLNEGspace smallneg)
(\NumSmallPages 512)
;; PNAME SPACEin the old world; used for initial atoms now.
(\PNPSPACE (8 0)
  (PNPspace PNPbase))
(\PNAME.HI 8)
(\OLDATOMSPACE (44 0))
(\ATOM.HI 44)
;; Definitions in old atom world
(\DEFSPACE (10 0)
  (DEFspace DEFbase)
  (DEFspace DEFbase))
(\DEF.HI 10)
(\VALSPACE (12 0)
  (TOPVALspace TOPVALbase)
  (VALspace VALbase))
(\VAL.HI 12)
(\PLISTSPACE (2 0)
  (PLISTspace PLISTbase))
(\PLIST.HI 2)

```

; Was 37776Q pre 16-meg initial

; (FOLDLO \MDSIncrement WORDSPERPAGE)

; Was 40000Q before 16meg initial

; none of the above

; HTCOLL size in words

; overlap character space and the atom hash table space

; NEW ATOM SPACE

; HI PART OF NEW ATOM SPACE

```

(\PAGEMAP (5 0)
  (PAGEMAPspace PAGEMAPbase))
(\NumPageMapPages 256)
(\PageMapTBL (20 512)
  (PMTspace PMTbase))
(\InterfacePage (20 0)
  (INTERFACESpace INTERFACEbase)
  (INTERFACESpace INTERFACEbase))
(\IOPAGE (0 65280))
(\DoveIORegion (0 16384))
(\IOCBPAGE (0 256))
(\FPTOVP (2 0))
(\MDSTypeTable (24 0)
  (MDSTYPESpace MDSTYPEbase)
  (MDSTYPESpace MDSTYPEbase))
(\MDSTTsize 1024 T)
; in Pages
(\MISCSTATS (20 2560)
  (STATSspace MISCSTATSbase))
(\UFNTable (20 3072)
  NIL
  (STATSspace UFNTablebase))
(\UFNTableSize 2)
(\DTDSpaceBase (20 4096)
  (DTDspace DTDbase)
  (DTDspace DTDbase))
(\DTDSIZE 18 T)
(\LISTPDTD (20 4186))
(\EndTypeNumber 2047)
(\LOCKEDPAGETABLE (20 28672))
(\NumLPTPages 16)
(\STACKSPACE (1 0)
  (STACKspace NIL)
  (STACKspace NIL))
(\GuardStackAddr 61440)
(\LastStackAddr 65534)
(\STACKHI 1 T T)
(\HTMAIN (22 0)
  (HTMAINspace HTMAINbase)
  (HTMAINspace HTMAINbase))
(\HTMAINnpages 256 T)
(\HTOVERFLOW (23 0)
  NIL
  (NIL HTOVERFLOWbase))
(\HTBIGCOUNT (23 32768))
(\HTCOLL (28 0)
  NIL
  (HTCOLLspace HTCOLLbase))
(\DISPLAYREGION (18 0))
(\DlBCPLspace 0 T LEmubrHiVal)
(\D0BCPLspace 0 T)
;; Interface Page locations
(\CurrentFXP 0 T T)
(\ResetFXP 1 T T)
(\SubovFXP 2 T T)
(\KbdFXP 3 T T)
(\HardReturnFXP 4 T T)
(\GCFXP 5)
(\FAULTFXP 6 T T)
(\MiscFXP 14 T T)
(\TeleRaidFXP 24 T T)
;; emulator segment locations
(DCB.EM 272)
(DISPIINTERRUPT.EM 273)
(CURSORBITMAP.EM 281)
(KBDAD0.EM 65052)
(KBDAD1.EM 65053)
(KBDAD2.EM 65054)
(KBDAD3.EM 65055)
(UTILIN.EM 65048)
(CURSORSX.EM 278)
(CURSORSY.EM 279)
(MOUSEX.EM 276)
(MOUSEY.EM 277)
(\LispKeyMask 8192 T T)
(\BcplKeyMask 4352 T T)
; Machine types
(\MAIKO 3)
(\DOLPHIN 4)
(\DORADO 5)
(\DANDELION 6)
(\DAYBREAK 8)
;; FOR DLION (AND DAYBREAK)
(\VP.DISPLAY 4608)

```

```

(\NP.DISPLAY 202)
; for Dorado display 1024x808 pixels / (16 pixels/word x 256
; words/page)

(\NP.WIDEDOVEDISPLAY 243)
; Wide Dove display 1152x864 pixels

(\WIDEDOVEDISPLAYWIDTH 1152)
; Includes 4 pages for cursor
(\RP.AFTERDISPLAY 206)
; if big screen

(\RP.AFTERDOVEDISPLAY 243)

(\RP.DISPLAY 0)
(\RP.TEMPDISPLAY 2561)
(\RP.MISCLOCKED 2804)
; (+ \RP.TEMPDISPLAY \NP.WIDEDOVEDISPLAY)

(\RP.STACK 768)
(\VP.STACK 256)
(\RP.MAP 256)
(\NP.MAP 256)
(\RP.IOPAGE 512)
; The DOVE IOCBPAGE can go anywhere, but should be under
; the 1mbyte range

(\RP.DOVEIOCBPAGE 543)
(\RP.DOVEIORGN 544)
(\VP.DOVEIORGN 64)
(\DOVEIORGNSIZE 64)
(\VP.IOPAGE 255)
(\VP.IFPAGE 5120)
(\VP.FPTOVP 512)
(\NP.FPTOVP 4096)
(\RP.FPTOVP 1024)
(\RP.STARTBUFFERS 640)
(\VP.TYPETABLE 6144)
(\NP.TYPETABLE 1024)
(\RP.TYPETABLE 5120)
(\VP.GCTABLE 5632)
(\NP.GCTABLE 256)
(\RP.GCTABLE 6144)
(\VP.GCOVERFLOW 5888)
(\NP.GCOVERFLOW 1)
(\RP.GCOVERFLOW 6400)
(\FP.IFPAGE 2)
(\VP.IOCBS 1)
(\VP.PRIMARYMAP 5122)
(\VP.SECONDARYMAP 1280)
(\VP.LPT 5232)
(\VP.INITSCRATCH 8)
(\VP.RPT 128)
(\VP.BUFFERS 218)
; DLion processor commands

(\DL.PROCESSORBUSY 32768)
(\DL.SETTOD 32769)
(\DL.READTOD 32770)
(\DL.READPID 32771)
(\DL.BUTTON 32772))

```

(DEFINEQ

(I.SETUPPAGEMAP

```

[LAMBDA NIL
  (PROG NIL
    (MKI.KEYPAGE (I.VAG2 5 0)
      NIL T)
    (I.CREATEPAGES (I.VAG2 20 512)
      8 NIL T)
    (for I from 0 to (SUB1 (LLSH 8 8)) do (I.PUTBASE (I.VAG2 20 512)
      I 65535))
    (SETQ NEXTTPM 0)
    (for I from 0 to (SUB1 (LRSH 256 5))
      bind (PAGEMAPKEY _ (LRSH (PROGN (IPLUS (LLSH (I.HILOC (I.VAG2 5 0))
      8)
      (LRSH (I.LOLOC (I.VAG2 5 0))
      8)))
      5))
      do
        (I.PUTBASE (I.VAG2 20 512)
          (IPLUS PAGEMAPKEY I)
          NEXTTPM)
        (SETQ NEXTTPM (IPLUS NEXTTPM 32)))
    (SETQ NEXTVMEM 2)
    (I.CREATEPAGES (I.VAG2 20 28672)
      16 NIL T))

```

(I.ADDPME

```

[LAMBDA (VP NEWPAGEOK)

```



```

(I.SETUPSTACK1 2 0 0 (IDIFFERENCE 768 2)
  0 RESETPC RESETPTR NIL INITFLG))
(I.PUTBASE (I.VAG2 20 0)
  1 0)
(I.PUTBASE (I.VAG2 20 0)
  6 0)
(I.PUTBASE (I.VAG2 20 0)
  2 0)
(I.PUTBASE (I.VAG2 20 0)
  3 0)
(I.SETUPGUARDBLOCK (IDIFFERENCE 768 2)
  2)
(I.PUTBASE (I.VAG2 20 0)
  30
  (I.SETUPGUARDBLOCK 768 (IDIFFERENCE (IDIFFERENCE 9216 768)
    2)))
(I.PUTBASE (I.VAG2 20 0)
  7
  (I.SETUPGUARDBLOCK (IDIFFERENCE 9216 2)
    2))

```

(I.SETUPSTACK1

```

[LAMBDA (STKP ALINK CLINK STKEND NARGS PC DEFPTR ARGS INITFLG ARGSLLENGTH)
  (*)
  (COND
    ([OR INITFLG (IGREATERP (IDIFFERENCE STKEND STKP)
      (IPLUS (PROG1 (I.GETBASE DEFPTR 0)
        )
        (PROG1 4
          )
        )
      )
    ])
    (PROG ((SP STKP))
      (if ARGSLLENGTH
        then (SETQ ARGSLLENGTH (MIN ARGSLLENGTH NARGS))
        (I.BLT (I.VAG2 1 SP)
          ARGS
          (LLSH ARGSLLENGTH 1))
          (SETQ SP (PLUS SP (TIMES ARGSLLENGTH 2)))
          (SETQ ARGS))
        (FRPTQ NARGS [I.PUTBASEPTR (I.VAG2 1 0)
          SP
          (AND ARGS (PROG1 (CAR ARGS)
            (SETQ ARGS (CDR ARGS))))]
          )
          (SETQ SP (PLUS SP 2)))
        (AND (PROG1 (COND
          ((ODDP SP 4)
            (I.PUTBASEPTR (I.VAG2 1 0)
              SP NIL)
              (SETQ SP (PLUS SP 2))
              T))
            (I.PUTBASE (I.VAG2 1 SP)
              0 32768))
          (LOGAND (LRSH ([LAMBDA ($$PUTBITS)
            (DECLARE (LOCALVARS $$PUTBITS))
            (I.PUTBASE $$PUTBITS 0 (LOGOR (LOGAND (I.GETBASE $$PUTBITS 0)
              65279)
              (LLSH (LOGAND 1 1)
                8]
              (I.VAG2 1 SP))
            8)
            1))
            (I.PUTBASE (I.VAG2 1 SP)
              1 STKP)
            (SETQ STKP (IPLUS SP 2))
            (LRSH ([LAMBDA ($$PUTBITS)
              (DECLARE (LOCALVARS $$PUTBITS))
              (I.PUTBASE $$PUTBITS 0 (LOGOR (LOGAND (I.GETBASE $$PUTBITS 0)
                255)
                (LLSH (CONSTANT (CL:READ-FROM-STRING "#B11000001")
                  8]
                (I.VAG2 1 STKP))
              8)
              (LOGAND ([LAMBDA ($$PUTBITS)
                (DECLARE (LOCALVARS $$PUTBITS))
                (I.PUTBASE $$PUTBITS 0 (LOGOR (LOGAND (I.GETBASE $$PUTBITS 0)
                  65280)
                  (LOGAND 0 255]
                  (I.VAG2 1 STKP))
                255)
                (I.PUTBASE (I.VAG2 1 STKP)
                  8 SP)
                (I.PUTBASE (I.VAG2 1 STKP)
                  1
                  (IPLUS ALINK 10 1))
                (I.PUTBASE (I.VAG2 1 STKP)
                  9

```

```

(IPLUS CLINK 10))
(I.PUTBASEPTR (I.VAG2 1 STKP)
 2 DEFPTR)
(I.PUTBASE (I.VAG2 1 STKP)
 5 PC)
(SETQ SP (IPLUS STKP (PROGN 10)))
[COND
 ((NOT INITFLG)
  (RPTQ (LLSH (ADD1 (SIGNED (I.GETBASE DEFPTR 2)
                           16))
          1)
        (PROGN
          (I.PUTBASE (I.VAG2 1 0)
                     SP 65535)
          (SETQ SP (PLUS SP 2])
          [I.PUTBASE (I.VAG2 1 STKP)
                    4
                    (SETQ SP (PLUS SP (PROGN 4]
                    (I.MAKEFREEBLOCK SP (IDIFFERENCE STKEND SP))
                    (RETURN STKP]))
          (*))
  (*))

```

(I.SETUPGUARDBLOCK

```

[LAMBDA (STKP LEN)
  (I.PUTBASE (I.VAG2 1 STKP)
    0 57344)
  (I.PUTBASE (I.VAG2 1 STKP)
    1 LEN)
  STKP])

```

(I.MAKEFREEBLOCK

```

[LAMBDA (STK SIZE)
  (PROGN
    (I.PUTBASE (I.VAG2 1 STK)
      1 SIZE)
    (I.PUTBASE (I.VAG2 1 STK)
      0 40960]))

```

)

```

(ADDTOVAR LOCKEDFNS \RESETSTACK0 \MAKEFRAME \SETUPSTACK1 \MAKEFREEBLOCK \FAULTHANDLER \KEYHANDLER
  \DUMMYKEYHANDLER \DOTELEAID \DUMMYTELEAID \DOHARDRETURN \DOGC \CAUSEINTERRUPT
  \INTERRUPTFRAME \CODEFORTFRAME \DOSTACKOVERFLOW \UNLOCKPAGES \DOMISCAPPLY)

```

```

(ADDTOVAR LOCKEDVARS \InterfacePage \DEFSPACE \STACKSPACE \KBDSTACKBASE \MISCSTACKBASE \SAVED.USER.CONTEXT
  \RUNNING.PROCESS \NEED.HARDRESET.CLEANUP)

```

(DEFINEQ

(I.INITGC

```

[LAMBDA NIL
  (I.CREATEPAGES (I.VAG2 22 0)
    (LRSH (IPLUS 65536 255)
      8)
    T T)
  (I.CREATEPAGES (I.VAG2 23 0)
    1 T T)
  (I.CREATEPAGES (I.VAG2 23 32768)
    1 T)
  (I.CREATEPAGES (I.VAG2 28 0)
    1 NIL T)
  (I.CREATEPAGES (I.ADDBASE (I.VAG2 28 0)
    256)
    (SUB1 (LRSH (IPLUS 1048576 255)
      8))
    T)
  (I.PUTBASEFIXP (I.VAG2 28 0)
    0 0)
  (I.PUTBASEFIXP (I.VAG2 28 0)
    2 2])

```

)

(DEFINEQ

(I.INTYPX

```

[LAMBDA (X)
  (LOGAND (I.GETBASE (I.VAG2 24 0)
    (LRSH (IPLUS (LLSH (I.HILOC X)
                      8)
            (LRSH (I.LOLOC X)
                  8))
    1))
  2047])

```

(I.ALLOCMDSPAGE

```

[LAMBDA (TYP)                                     (*)
  (PROG (VP VPTR)
    BEG [COND
      [(SETQ VP I.MDSFREELISTPAGE)
        (SETQ VPTR ([LAMBDA ($$1)
          (I.VAG2 (LRSH (SETQ $$1 VP)
            8)
          (LLSH (LOGAND $$1 255)
            8]
          NIL))
        (PROG ((NXT (I.GETBASEPTR VPTR 0)))
          (COND
            ((AND NXT (NOT (SMALLP NXT)))
              (\MP.ERROR 26 "MDS Free Page link bad. ^N to continue" (PROG1 I.MDSFREELISTPAGE
                (SETQ I.MDSFREELISTPAGE)
              )
              (GO BEG))
            (T (SETQ I.MDSFREELISTPAGE NXT]
          (T (NIL)
            (SETQ VP I.NxtMDSPage)
            (SETQ I.NxtMDSPage (IDIFFERENCE VP (LRSH 512 8)))
            (SETQ VPTR ([LAMBDA ($$1)
              (I.VAG2 (LRSH (SETQ $$1 VP)
                8)
              (LLSH (LOGAND $$1 255)
                8]
              NIL))
            (MKI.NEWPAGE (I.ADDBASE (MKI.NEWPAGE VPTR)
              256]
          (I.MAKEMDSENTRY VP TYP)
          (RETURN VPTR]))

```

(I.MAKEMDSENTRY

```

[LAMBDA (VP V)                                     (*)
  (I.PUTBASE (I.VAG2 24 0)
    (LRSH VP 1)
    (COND
      ((NIL)
        (LOGOR 32768 V))
      (T V))

```

(I.INITMDSPAGE

```

[LAMBDA (BASE SIZE PREV)                           (*)
  (PROG ((SLOP (IREMAINDER 256 SIZE))
    NPAGES LIMIT)
    (COND
      ((AND (NEQ SLOP 0)
        (ILESSP SLOP (LRSH SIZE 1))
        (ILESSP SIZE 256))
        (SETQ NPAGES (IQUOTIENT 512 256))
        (SETQ LIMIT 256))
      (T (SETQ NPAGES 1)
        (SETQ LIMIT 512)))
    (to NPAGES do (for (DISP _ 0) while (ILEQ (SETQ DISP (PLUS DISP SIZE))
      LIMIT)
      do (I.PUTBASEPTR BASE 0 PREV)
        (SETQ PREV BASE)
        (SETQ BASE (I.ADDBASE BASE SIZE)))
      (SETQ BASE (I.ADDBASE BASE SLOP)))
    (RETURN PREV])

```

(I.ASSIGNDATATYPE1

```

[LAMBDA (NAME DESCRIPTORS SIZE SPECS PTRFIELDS SUPERTYPE) (*)
  (PROG ((I.NTYPX (I.TYPENUMBERFROMNAME NAME))
    (SUPERTYPENUMBER (COND
      (SUPERTYPE (OR (I.TYPENUMBERFROMNAME SUPERTYPE)
        (ERROR SUPERTYPE ":INCLUDED datatype but not currently declared"))
      )
      (T 0)))
    DTD REDECLARED NEWTYPENUM NEWDTD)
    [COND
      (I.NTYPX
        (SETQ DTD (I.ADDBASE (I.VAG2 20 4096)
          (ITIMES I.NTYPX 18)))
        (COND
          ((AND (EQUAL PTRFIELDS (I.GETBASEPTR DTD 10))
            (EQUAL SIZE (I.GETBASE DTD 3)))
            (I.PUTBASEPTR DTD 6 DESCRIPTORS)

```

```
(I.PUTBASE DTD 17 SUPERTYPENUMBER)
(RETURN I.NTYPX))
((EQ (I.GETBASE DTD 3)
    0) (*)
)
([OR (EQ CROSSCOMPILING T)
      (AND CROSSCOMPILING (NEQ 'Y (ASKUSER 30 (SELECTQ CROSSCOMPILING
          (Y 'Y)
          'N)
          (LIST (COND
              (SIZE "OK TO REDECLARE DATATYPE ")
              (T "OK to deallocate DATATYPE "))
              NAME]
          )
          (*)))
      (RETURN I.NTYPX))
      ((IGREATERP I.NTYPX I.MaxSysTypeNum)
        (SETQ REDECLARED T))
      (T
        (ERROR "ILLEGAL DATA TYPE" NAME])
      *)
(COND
  ((NOT SIZE)
   )
  (T
   (COND
     ((AND (EQ I.MaxTypeNumber 2047)
            (OR (NULL I.NTYPX)
                REDECLARED))
      (LISPERROR "DATA TYPES FULL" NAME))))
  (PROGN
   [COND
    ((OR (NULL I.NTYPX)
         REDECLARED)
       (*))
    (SETQ NEWTYPENUM (SETQ I.MaxTypeNumber (PLUS I.MaxTypeNumber 1)))
    (SETQ NEWDTD (I.ADDBASE (I.VAG2 20 4096)
                             (ITIMES NEWTYPENUM 18)))
    (COND
      ((IGEQ (IPLUS (LOGAND (I.LOLOC NEWDTD)
                           255)
               18)
             256)
        (MKI.NEWPAGE (I.ADDBASE NEWDTD 18)
                      T)))
    (COND
      [REDECLARED
       (LET ([NEWTYPEENTRY (LOGOR NEWTYPENUM (LOGAND (I.GETBASE DTD 16)
                                                       (LOGNOT 2047)]
              FOUNDSOME)
           [\MAPMDS I.NTYPX (FUNCTION (LAMBDA (PAGE)
                                     (I.MAKEMDSENTRY PAGE NEWTYPEENTRY)
                                     (SETQ FOUNDSOME T]))
       (COND
        ((NOT FOUNDSOME)
         (SETQ I.MaxTypeNumber (PLUS I.MaxTypeNumber -1)))
        (T (I.PUTBASEPTR DTD 6 NIL)
            (I.PUTBASEPTR DTD 8 NIL)
            (I.BLT NEWDTD DTD 18)
            (PROGN (I.GETBASEPTR NEWDTD 10)
                   (NEQ (LOGAND (LRSH (I.PUTBASE NEWDTD 0 (LOGOR (LOGAND (I.GETBASE NEWDTD 0)
                                                                       57343)
                                                                     (LLSH (LOGAND (COND
                                                                           (T 1)
                                                                           (T 0))
                                                                      1)
                                                                    13))))
                     1)
                    13)
                  1)
                 0)
            (I.PUTBASE NEWDTD 16 NEWTYPEENTRY)
            [I.PUTBASEPTR NEWDTD 0 (NEW-SYMBOL-CODE (PACK* "Obsolete-" NAME)
                                                      (I.ATOMNUMBER (PACK* "Obsolete-" NAME)
                                                                  ))
            (I.PUTBASEPTR DTD 4 NIL)
            ]
          (T
           (SETQ I.NTYPX NEWTYPENUM)
           (I.PUTBASEPTR (SETQ DTD NEWDTD)
                         0
                        (NEW-SYMBOL-CODE NAME (I.ATOMNUMBER NAME]
      (COND
        ((NEQ SIZE 0)
         (I.PUTBASE DTD 3 SIZE)
         (I.PUTBASEPTR DTD 6 (I.COPY DESCRIPTORS))
         (I.PUTBASEPTR DTD 8 (I.COPY SPECS))
         (I.PUTBASEPTR DTD 10 PTRFIELDS)
         (I.PUTBASE DTD 17 SUPERTYPENUMBER)
         (I.PUTBASE DTD 16 I.NTYPX)
         ))
        )
```

(RETURN (CL:VALUES I.NTYPX REDECLARED])

(I.TYPENUMBERFROMNAME

```

[LAMBDA (TYPE)
  (AND TYPE (BIND (INDEX _ (NEW-SYMBOL-CODE TYPE (I.ATOMNUMBER TYPE))) for I from 1 to I.MaxTypeNumber
    do (COND
      ((EQ INDEX (I.GETBASEPTR (I.ADDBASE (I.VAG2 20 4096)
        (ITIMES I 18))
        0))
      (RETURN I))
    )
  )

```

(I.CREATECELL

```

[LAMBDA (TYP)
  (COND
    ((AND (NEQ 1 0)
      (EQ TYP 5))
      (HELP "CREATECELL \LISTP"))
    (LET ((DTD (I.ADDBASE (I.VAG2 20 4096)
      (ITIMES TYP 18)))
      NEWCELL)
      (while (EQ (I.GETBASE DTD 3)
        0)
        do (ERROR "Attempt to CREATE a type not declared yet" (\TYPENAMEFROMNUMBER TYP)))
      (PROGN (COND
        ((SETQ NEWCELL (I.GETBASEPTR DTD 4))
          (I.PUTBASEPTR DTD 4 (I.GETBASEPTR NEWCELL 0))
          (LET [(CNT (SUB1 (I.GETBASE DTD 3)
            (I.PUTBASE NEWCELL CNT 0)
            (I.BLT NEWCELL (I.ADDBASE NEWCELL 1)
              CNT))
              (PROGN NEWCELL)
              NEWCELL)
            (T
              (I.PUTBASEPTR DTD 4 (I.INITMDSPAGE (I.ALLOCMDSPAGE (I.GETBASE DTD 16))
                (I.GETBASE DTD 3)
                (I.GETBASEPTR DTD 4)))
              (I.CREATECELL TYP))
            )
          )
        )
      )
    )
  )

```

(I.NEW2PAGE

```

[LAMBDA (BASE)
  (MKI.NEWPAGE (I.ADDBASE (MKI.NEWPAGE BASE)
    256])

```

)

(DEFINEQ

(I.CREATEMDSTYPETABLE

```

[LAMBDA NIL
  (I.CREATEPAGES (I.VAG2 24 0)
    1024 NIL T)
  [PROG (VP)
    (SETQ VP 0)
    (FRPTQ (LLSH 1024 8)
      (I.PUTBASE (I.VAG2 24 0)
        VP 32768)
      (SETQ VP (PLUS VP 1)))
    [for SEGMENT in (LIST 14 15) do (for PAGE from 0 to (SUB1 256) by (LRSH 512 8)
      do (I.MAKEMDSENTRY (LOGOR PAGE (LLSH SEGMENT 8))
        (LOGOR 32768 8192 4096 2048 1]
      (for PAGE from 0 to (SUB1 256) by (LRSH 512 8) do (I.MAKEMDSENTRY (LOGOR PAGE (LLSH 7 8))
        (LOGOR 32768 9]
    (I.CREATEPAGES (I.VAG2 20 2560)
      (LRSH 512 8)
      NIL T)
    (I.MAKEMDSENTRY (IPLUS (LLSH (I.HILOC (I.VAG2 20 2560))
      8)
      (LRSH (I.LOLOC (I.VAG2 20 2560))
        8))
      (LOGOR 32768 8192 4096 2048 2])

```

(I.INITDATATYPES

```

[LAMBDA NIL
  )

```

```

(LET ((NSYSTYPES (LENGTH INITIALDSTDCONTENTS)))
  (I.CREATEPAGES (I.VAG2 20 4096)
    1 NIL T)
  (I.CREATEPAGES (I.ADDBASE (I.VAG2 20 4096)
    256)
    (SUB1 (LRSH (IPLUS (ADD1 (TIMES (ADD1 NSYSTYPES)
      18))
    255)
    8)))
  [for D in INITIALDSTDCONTENTS bind DTD as TYPENO from 1
  do
    (SETQ DTD (I.ADDBASE (I.VAG2 20 4096)
      (ITIMES TYPENO 18)))
    [I.PUTBASE DTD 16 (LOGOR TYPENO (COND
      ((FMEMB (CAR D)
        ' (SMALLP FIXP FLOATP))
        4096)
      (T 0))
      (COND
        ((FMEMB (CAR D)
          ' (SMALLP FIXP FLOATP LITATOM NEW-ATOM))
          2048)
        (T 0))
      (COND
        ((FMEMB (CAR D)
          ' (SMALLP FIXP))
          8192)
        (T 0))
      (COND
        ((EQ (CAR D)
          'NEW-ATOM)
          32768)
        (T 0))
      (COND
        ((FMEMB (CAR D)
          ' (LITATOM NEW-ATOM))
          (CONSTANT 16384))
        (T 0))
      (COND
        ((NOT (CADR D))
          32768)
        (T 0])
      (COND
        ((EQ (CAR D)
          'NEW-ATOM)
          (I.PUTBASE DTD 17 4)))
      (COND
        ((AND (CAR D)
          (CADR D))
          (I.PUTBASE DTD 3 (CADR D))
        (NEQ 1 0)
        (SETQ I.LISTPDTD (I.ADDBASE (I.VAG2 20 4096)
          (ITIMES 5 18]
        (SETQ I.MaxSysTypeNum (SETQ I.MaxTypeNumber NSYSTYPES))
        NIL])

```

(I.INITDATATYPENAMES

```

[LAMBDA NIL
  (SETQ I.FINALIZATION.FUNCTIONS (I.ALLOCBLOCK (ADD1 2047)
    T))
  [for D in INITIALDSTDCONTENTS as I.NTYPX from 1
  do (LET ((DTD (I.ADDBASE (I.VAG2 20 4096)
    (ITIMES I.NTYPX 18)))
    (FINAL (CADDR D)))
    (I.PUTBASEPTR DTD 0 (I.ATOMNUMBER (CAR D)))
    (I.PUTBASEPTR DTD 10 (I.COPY (CADDR D)))
    (if FINAL
      then
        (NEQ (LOGAND (LRSH (I.PUTBASE DTD 0 (LOGOR (LOGAND (I.GETBASE DTD 0)
          61439)
          (LLSH (LOGAND (COND
            (T 1)
            (T 0))
          1)
          12)))
          1)
          0)
        (I.PUTBASEPTR I.FINALIZATION.FUNCTIONS (LLSH I.NTYPX 1)
          (I.COPY FINAL))
  (PROGN

```

```

      (NEQ (LOGAND (LRSH ([LAMBDA ($$PUTBITS)
        (DECLARE (LOCALVARS $$PUTBITS))
        (I.PUTBASE $$PUTBITS 0 (LOGOR (LOGAND (I.GETBASE $$PUTBITS 0)
          61439)
          (LLSH (LOGAND (COND
            (T 1)
            (T 0))
            1)
          12]
          (I.ADDBASE (I.VAG2 20 4096)
            (ITIMES 0 18)))
          12)
          1)
        0)
      (I.PUTBASEPTR I.FINALIZATION.FUNCTIONS 0 (I\COPY '\RECLAIMARRAYBLOCK])
    )

```

(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))
   (TWO-D-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)))

```

(DEFINEQ

(I.FSETVAL

```

  [LAMBDA (ATM VAL)
    (I.PUTBASEPTR (I.\ATOMCELL ATM 12)
      0 VAL])

```

(I.SETPROPLIST

```

  [LAMBDA (ATM LST)
    (I.PUTBASEPTR (I.\ATOMCELL ATM (CONSTANT 2))
      0 LST])

```

(I.PUTDEFN

```

  [LAMBDA (FN CA SIZE)
    (PROG ((DCELL (I.\ATOMCELL FN 10))
      [BLOCKINFO (PROGN
        (I.FILECODEBLOCK (LRSH (IPLUS SIZE 3)
          2)
          (IPLUS (LOGOR (LLSH (\BYTELT CA 12)
            8)
            (\BYTELT CA (ADD1 12))))
          (PROGN 8]
        (BASE (I.FILEARRAYBASE)))
        (I.PUTBASEPTR DCELL 0 BASE)
        (LOGAND (LRSH (I.PUTBASE DCELL 0 (LOGOR (LOGAND (I.GETBASE DCELL 0)
          53247)
          (LLSH (LOGAND (LOGAND (LRSH (\BYTELT CA 8)
            4)
            3)
            12)))
          3)
        12)
      3)

```

```

      (NEQ (LOGAND (LRSH (I.PUTBASE DCELL 0 (LOGOR (LOGAND (I.GETBASE DCELL 0)
                                                           49151)
                                                           (LLSH (LOGAND (COND
                                                                ((EQ (LOGOR (LLSH (\BYTELT CA 12)
                                                                8)
                                                                (\BYTELT CA (ADD1 12)))
                                                                0)
                                                                1)
                                                                (T 0))
                                                                1)
                                                                14)))
                                                                1)
                                                                14)
                                                                0)
      (NEQ (LRSH (I.PUTBASE DCELL 0 (LOGOR (LOGAND (I.GETBASE DCELL 0)
                                                           32767)
                                                           (LLSH (COND
                                                                (T 1)
                                                                (T 0))
                                                                15)))
                                                                15)
                                                                0)
      (NEQ (LOGAND (LRSH (I.PUTBASE DCELL 4 (LOGOR (LOGAND (I.GETBASE DCELL 4)
                                                           65527)
                                                           (LLSH (LOGAND (COND
                                                                (NIL 1)
                                                                (T 0))
                                                                1)
                                                                3)))
                                                                3)
                                                                1)
                                                                0)
      [COND
        ((FMEMB FN LOCKEDFNS)
         (I.\LOCKCELL DCELL 1)
         (I.\LOCKCELL BASE (LRSH (IPLUS (IPLUS (LOGAND (I.LOLOC BASE)
                                                           255)
                                                           (LRSH (IPLUS SIZE 1)
                                                           1))
                                                           255)
                                                           8]
        ((EQ FN (FUNCTION \RESETSTACK))
         (SETQ RESETPTR (I.\FILEARRAYBASE))
         (SETQ RESETPC (LOGOR (LLSH (\BYTELT CA 6)
                                     8)
                               (\BYTELT CA (ADD1 6]
        (AOUT CA 0 SIZE OUTX 'CODE)
        (BOUTZEROS (IDIFFERENCE (SUB1 4)
                                (IMOD (SUB1 SIZE)
                                4)))
        (I.\FILEBLOCKTRAILER BLOCKINFO])

```

(I.\BLT

```

[LAMBDA (DBASE SBASE NWORDS)
  (PROG [(NN (CONSTANT (EXPT 2 14]
  (RETURN (COND
    ((GREATERP NWORDS NN)
     (I.\BLT (I.\ADDBASE DBASE NN)
              (I.\ADDBASE SBASE NN)
              (DIFFERENCE NWORDS NN))
     (I.\BLT DBASE SBASE NN)
    (T (for I from (SUB1 NWORDS) by -1 to 0 do (I.PUTBASE DBASE I (I.GETBASE SBASE I))
        DBASE]))

```

)

(DEFINEQ

(I.\MKATOM

```

[LAMBDA (BASE OFFST LEN FATP NONNUMERICP)
  (PROG [(FATCHARSEENP (AND FATP (NOT (NULL (for I from OFFST to (SUB1 (IPLUS OFFST LEN))
                                                suchthat (IGREATERP (I.GETBASE BASE I)
                                                255]
        HASH HASHENT ATM# PNBASE FIRSTCHAR FIRSTBYTE REPROBE)
        (COND
          ((EQ LEN 0)
           (SETQ HASH 0)
           (SETQ FIRSTBYTE 255)
           (GO LP)))
          [SETQ FIRSTCHAR (COND
            (FATP (I.GETBASE BASE OFFST))
            (T (I.\GETBASEBYTE BASE OFFST]
        (COND
          [(AND (EQ LEN 1)

```


(I.\CREATE.SYMBOL

```
[LAMBDA (BASE OFFSET LEN FATP FATCHARSEENP)
```

$$\begin{pmatrix} \star \\ \star \\ \star \\ \star \\ \star \end{pmatrix}$$

```
(LET ([PNBASE (I.\ALLOCBLOCK (COND  
    (FATCHARSEENP (LRSH (IPLUS (ADD1 LEN)  
        1))  
    (T (LRSH (IPLUS (ADD1 LEN)  
        3) 2]) (*)  
PB CPP ATM)  
[COND  
    ((IGEQ (SETQ ATM I.AtomFrLst)  
        12287) (*  
    (IGEQ (SETQ ATM I.AtomFrLst)  
        65535) (*  
        *) (*  
        *)  
    (SETQ ATM (I.\CREATECELL 21))  
    (I.PUTBASEPTR (I.\ATOMCELL ATM 12)  
        0  
        'NOBIND)) (*)  
    ((EVENP ATM 256) (*  
        *) (*  
        *) (*)  
    (LET [(PN (ITIMES 10 (LRSH ATM 8]  
        (COND  
            ((NEW-SYMBOL-CODE NIL (IGEQ PN (IDIFFERENCE 255 1))) (*)  
            (NIL)))  
        (I.\MAKEMDSENTRY (LRSH ATM 8)  
            (LOGOR 32768 16384 2048 4)) (*)  
        (I.\INITATOMPAGE PN) (*)  
        ]  
[[LAMBDA (DATUMA0066)  
    (DECLARE (LOCALVARS DATUMA0066))  
    (PROGN (I.PUTBASE DATUMA0066 0 (LOGOR (LOGAND 61440 (I.GETBASE DATUMA0066 0))  
        (LOGAND (I.HILOC PNBASE)  
            4095)))  
    (I.PUTBASE DATUMA0066 (ADD1 0)  
        (I.LOLOC PNBASE))  
    PNBASE]  
(COND  
    [(AND (FIXP ATM)  
        (ILESSP ATM 65535))  
    (I.ADDBASE (I.VAG2 44 0)  
        (IPLUS 0 (ITIMES 10 ATM)]  
    (T (I.ADDBASE ATM 0] (*)  
(COND  
    (FATCHARSEENP (I.\BLT (I.ADDBASE PNBASE 1)  
        (I.ADDBASE BASE OFFSET)  
        LEN))  
    [FATP (for I from OFFSET as J from 1 to LEN do (I.\PUTBASEBYTE PNBASE J (I.GETBASE BASE I])  
    (T (I.MOVEBYTES BASE OFFSET PNBASE 1 LEN)))  
(LRSH (I.PUTBASE PNBASE 0 (LOGOR (LOGAND (I.GETBASE PNBASE 0)  
        255)  
        (LLSH LEN 8)))  
    8)  
(COND  
    ((NOT T) (*)  
    (PROGN PNBASE)))  
(SETQ I.AtomFrLst (ADD1 I.AtomFrLst)) (*)  
(AND (FIXP ATM)  
    (SETQ ATM (I.ADDBASE (I.VAG2 0 0)  
        ATM)))  
(COND  
    (FATCHARSEENP (NEQ (LOGAND (LRSH ([LAMBDA ($$PUTBITS)  
        (DECLARE (LOCALVARS $$PUTBITS))  
        (I.PUTBASE $$PUTBITS 3 (LOGOR (LOGAND (I.GETBASE $$PUTBITS 3)  
            57343)  
            (LLSH (LOGAND (COND  
                (T 1)  
                (T 0))  
            1)  
            13]  
        (I.\ATOMCELL ATM (CONSTANT 2))))  
    13)  
    1)  
    0)))  
ATM]))  
  
I.\INITATOMPAGE  
[LAMBDA (PN) (*)  
(COND  
    [NIL (PROG ((OFFSET (LLSH PN 8))  
        VALBASE)  
        ) (*)
```

```
(*)
(I.NEW4PAGE (I.ADDBASE (I.ADDBASE (I.VAG2 8 0)
                                OFFSET)
                                OFFSET))
(I.NEW4PAGE (I.ADDBASE (I.ADDBASE (I.VAG2 10 0)
                                OFFSET)
                                OFFSET))
(I.NEW4PAGE (I.ADDBASE (I.ADDBASE (I.VAG2 2 0)
                                OFFSET)
                                OFFSET))
(I.NEW4PAGE (SETQ VALBASE (I.ADDBASE (I.ADDBASE (I.VAG2 12 0)
                                                OFFSET)
                                                OFFSET)))
(FRPTQ (ITIMES 128 4)
        (I.PUTBASEPTR VALBASE 0 (I.COPY 'NOBIND))
        (SETQ VALBASE (I.ADDBASE VALBASE 2]
(T
(LET ((OFFSET (LLSH PN 8))
      (ATM (LLSH (IQUOTIENT PN 10)
                  8))
      VALBASE)
(for I from 0 to 9 as OFF from OFFSET by 256 do (MKI.NEWPAGE (I.ADDBASE (I.VAG2 44 0)
                                                                    OFF))))
(*)
(for I from 0 to 255 as OFF from OFFSET by 10 do (I.PUTBASEPTR (I.VAG2 44 0)
                        (IPLUS OFF 2)
                        'NOBIND]))
```

(I.\MOVEBYTES

[illegible]

(I.\STKMIN

```

[ LAMBDA (CODE CODEISBLOCK PRINT)
  (DECLARE (LOCALVARS))

  (PROGN

    ;; can be run renamed but will work on local space.

    (if (NOT \OPSTACKEFFECT)
      then (SETQ \OPSTACKEFFECT (\ALLOCBLOCK (FOLDHI 256 BYTESPERCELL)))
            (SETQ \OPLength (\ALLOCBLOCK (FOLDHI 256 BYTESPERCELL)))
            [for I from 0 to 255
              do (\PUTBASEBYTE \OPSTACKEFFECT I
                             (- 2 (LET ((OP (\FINDOP I))
                                         LEVADJ)
                                   (SELECTQ (fetch (OPCODE OPCODENAME)
                                                    OP)
                                     ((FN0 FN1 FN2 FN3 FN4 FNX SWAP NOP APPLYFN RETURN)
                                      2)
                                     ((UNBIND DUNBIND UNWIND POP.N)
                                      -1)
                                     ((BIND SUBRCALL MISCN)
                                      1)
                                     (OR (NUMBERP (if (LISTP (SETQ LEVADJ (fetch (OPCODE LEVADJ)
                                                                                      OP)))
                                                    then (SETQ LEVADJ (CAR LEVADJ))
                                                    else LEVADJ))
                                      0)
                                   LEVADJ))
              ]
            ]
    )
  )
)

```

```

                                (SELECTQ LEVADJ
                                ( (CJUMPN CJUMP)
                                ; these only check if they jump
                                -1)
                                ( (JUMP)
                                2)
                                (PROGN 2]
                                (for I from 0 to 255 do (\PUTBASEBYTE \OPLength I (ADD1 (OR (CADDR (\FINDOP I))
                                -1])
[IF (NOT CODEISBLOCK)
  THEN (SETQ CODE (OR (\GET-COMPILED-CODE-BASE CODE)
                      (fetch (ARRAYP BASE)
                             CODE]
(LLSH (PROG (MAX OP STKE (PC (fetch (FNHEADER STARTPC)
                                   CODE)))
      (DEPTH (IPLUS (IMAX (fetch (FNHEADER NA) of CODE)
                     0)
              8
              (UNFOLD (ADD1 (fetch (FNHEADER PV) of CODE))
                      CELLSPERQUAD)
              4)))
      (SETQ MAX (PLUS DEPTH 8))
;; this PROG computes the depth in cells. The lsh around converts it to D-machine words.
;; the initial maximum is the actual size of the frame, plus 4 extra cells for space to store info in case of an overflow. The default
;; maximum is 8 more than that. By walking the code, it finds if there are any other runs that would increase it beyond that. At jumps or
;; "Maiko check" opcodes, the depth is reset to 0.
LP (if (EQ 0 (SETQ OP (\GETBASEBYTE CODE PC)))
      then ;; end of the function
        (RETURN MAX))
;; the following is for debugging
(AND PRINT (CL:FORMAT T "~%~3o: ~3o d<~3d> mx<~3d>" PC OP DEPTH MAX))
(SELECTQ (SETQ STKE (- 2 (\GETBASEBYTE \OPSTACKEFFECT OP)))
(2 ;; special code indicating that this opcode checks the stack level
  (AND PRINT (PRIN1 "*"))
  (SETQ DEPTH 0))
  (add DEPTH STKE))
(if (GREATERP DEPTH MAX)
    then (SETQ MAX DEPTH))
(CL:INCF PC (\GETBASEBYTE \OPLength OP))
(GO LP))
1])
)
(DEFINEQ
(I.COPYATOM
 [LAMBDA (X)
  (LET ((PKG (CL:SYMBOL-PACKAGE X)))
    (if (NEQ PKG *INTERLISP-PACKAGE*)
        then ;; Kludge time. We don't yet have the machinery to create packages in the init.sysout, so anything that isn't an Interlisp symbol
              ;; has to be turned into a flat-space symbol with appropriate prefix
              (if (EQ PKG *KEYWORD-PACKAGE*)
                  then ;; keywords eval to self, so also set top val
                    (MKI.DSET X X)
                    (SETQ X (CONCAT ":" X))
                  elseif (EQ PKG *LISP-PACKAGE*)
                  then ;; Symbol lives in CL and not available in IL, so add prefix
                    (SETQ X (CONCAT "CL:" X))
                  elseif (NULL PKG)
                  then ;; This is an uninterned symbol, so add #: prefix.
                    (SETQ X (CONCAT "#:" X))
                  elseif (CL:STRING= (CL:PACKAGE-NAME PKG)
                             "SYSTEM")
                  then ;; SYSTEM = SI package. All internal for now.
                    (SETQ X (CONCAT "SI:" X))
                  ELSEIF (CL:STRING= (CL:PACKAGE-NAME PKG)
                             "CONDITIONS")
                  THEN ;; Make it internal. The xcl-package stuff will export the right ones when it starts up.
                    (SETQ X (CONCAT "CONDITIONS:" X))
                  ELSEIF (CL:STRING= (CL:PACKAGE-NAME PKG)
                             "XEROX-COMMON-LISP")
                  THEN ;; Make it internal. The xcl-package stuff will export the right ones when it starts up.
                    (SETQ X (CONCAT "XCL:" X))
                  ELSEIF

```

```

"COMPILER")
THEN ;; Make it internal. The compiler-package stuff will export the right ones when it starts up.
      (SETQ X (CONCAT "COMPILER::" X))
ELSEIF (CL:STRING= (CL:PACKAGE-NAME PKG)
        "FASL")
THEN ;; Make it internal. The fasl-package stuff will export the right ones when it starts up.
      (SETQ X (CONCAT "FASL::" X))
else (HELP "Can only translate symbols in IL, CL, XCL, CONDITIONS, SI, COMPILER, FASL and
          keywords" X]
(LET [(N (NCHARS X))
      [BASE (COND
              ((NEQ 0 (LOGAND (LRSH (I.GETBASE I.SCRATCHSTRING 2)
                                     14)
                              1))
               (%ARRAY-BASE I.SCRATCHSTRING))
              (T (I.GETBASEPTR I.SCRATCHSTRING 0)
                 (OFFST (COND
                         ((NEQ 0 (LOGAND (LRSH (I.GETBASE I.SCRATCHSTRING 2)
                                                 14)
                                         1))
                          (%ARRAY-OFFSET I.SCRATCHSTRING))
                         (T (I.GETBASE I.SCRATCHSTRING 3)
                            (*))
                        (FOR I FROM 1 TO N DO (I.PUTBASEBYTE BASE (IPLUS OFFST I -1)
                                                             (NTHCHARCODE X I)))
                        (I.ATOMNUMBER (I.MKATOM BASE OFFST N))

```

(I.INITATOMS

```

[LAMBDA NIL
  (I.CREATEPAGES (I.VAG2 21 0)
    256)
  (SETQ I.SCRATCHSTRING (I.ALLOCSTRING 255))
  (I.COPYATOM NIL)
  (I.COPYATOM 'NOBIND)
  (for c from 0 to 255 when (OR (ILESSP C (CHARCODE 0))
                                (IGREATERP C (CHARCODE 9))))
    do (I.COPYATOM (CHARACTER C))
  (SETQ I.OneCharAtomBase (I.ADDBASE (I.VAG2 0 0)
    2))
  (I.COPYATOM (FUNCTION \EVALFORM))
  (I.COPYATOM (FUNCTION \GC.HANDLEOVERFLOW))
  (I.COPYATOM (FUNCTION \DTEST.UFN))
  (I.COPYATOM (FUNCTION \OVERFLOWMAKENUMBER))
  (I.COPYATOM (FUNCTION \MAKENUMBER))
  (I.COPYATOM (FUNCTION \SETGLOBAL.UFN))
  (I.COPYATOM (FUNCTION \SETFVAR.UFN))
  (I.COPYATOM (FUNCTION \GCMAPTABLE))
  (I.COPYATOM (FUNCTION \INTERPRETER))
  (OR (EQ (I.ATOMNUMBER (FUNCTION \INTERPRETER))
    256)
    (HELP (FUNCTION \INTERPRETER)
      " not atom 400Q"))
)
(DEFINEQ

```

(I.MAKEINITFIRST

```

[LAMBDA NIL
  (I.CREATEMDSTYPETABLE)
  (I.SETUP.HUNK.TYPENUMBERS)
  (I.INITDATATYPES)
  (I.PREINITARRAYS)
  (I.TURN.ON.HUNKING)
  (I.INITATOMS)
  (I.INITDATATYPENAMES)
  (I.INITUFNTABLE)
  (I.INITGC)
  (MKI.NEWPAGE (I.VAG2 20 0)
    NIL T])

```

(I.COPY

```

[LAMBDA (X)
  (SELECTQ (TYPENAME X)
    ((LITATOM NEW-ATOM)
      (MKI.ATOM X))
    (LISTP (PROG [(R (REVERSE X))
                  (V (I.COPY (CDR (LAST X))
                             LP (COND
                               ((LISTP R)

```

```
(I.MAKEINITLAST
[ LAMBDA (VERSIONS)                                     (*)
  (I.SETUPSTACK T)
  (I.MAKEINITBFS)
  (PROGN                                                  (*
    [SELECTQ (SYSTEMTYPE)
      ((D ALTO)
        [MAPHASH MKI.PLHA (FUNCTION (LAMBDA (P A)
          (I.SETPROPLIST A (I.COPY P]
        [MAPHASH MKI.TVHA (FUNCTION (LAMBDA (V A)
          (I.FSETVAL A (I.COPY (CDR V])
        (PROG (AL GAG)
          (PROGN [MINFS (IMAX (MINFS)
            (ITIMES 2 (ARRAYSIZE (CAR MKI.PLHA)))
            (ARRAYSIZE (CAR MKI.TVHA])
          (RECLAIM)
          (SETQ GAG (GCGAG "[***** GARBAGE COLLECTION - ERROR *****]"))
          [MAPHASH MKI.PLHA (FUNCTION (LAMBDA (P A)
            (push AL (CONS A P]
          (SETQ GAG (GCGAG GAG))
          [MAPC AL (FUNCTION (LAMBDA (X)
            (I.SETPROPLIST (CAR X)
              (I.COPY (CDR X]
          (PROGN (SETQ AL)
            (RECLAIM)
            (SETQ GAG (GCGAG GAG))
            [MAPHASH MKI.TVHA (FUNCTION (LAMBDA (V A)
              (push AL (RPLACA V A]
            (GCGAG GAG))
            [MAPC AL (FUNCTION (LAMBDA (X)
              (I.FSETVAL (CAR X)
                (I.COPY (CDR X]
              )
              )
            (PROG ((AFL (I.FILEARRAYBASE)))
              (BOUTZEROS (IDIFFERENCE (TIMES 2 512)
                (LLSH (IMOD (I.LOLOC AFL)
                  (TIMES 2 256))
                1)))
              (SETQ MKI.CODELASTPAGE ([LAMBDA (PTR)
                (DECLARE (LOCALVARS PTR))
                (IPLUS (LLSH (I.HILOK PTR)
                  8)
                  (LRSH (I.LOLOC PTR)
                    8]
                (I.FILEARRAYBASE)))
              (I.POSTINITARRAYS AFL (IPLUS 11776 MKI.CODESTARTOFFSET)
                MKI.CODELASTPAGE))
```

```

[MAPC (APPEND INITVALUES INITPTRS)
  (FUNCTION (LAMBDA (X)
    (I.ATOMNUMBER (CAR X)
    [for X in INITVALUES as A in MKI.VALUES do (SETQ A (EVALV A))
      (I.FSETVAL (CAR X)
        (COND
          ((OR (EQ A T)
              (EQ A NIL)
              (AND (FIXP A)
                  (IGEQ A -65536)
                  (ILEQ A 65535)))
            (I.\COPY A))
          (T (SHOULDNT)
            (EVALV A)))
      (for X in INITPTRS as A in MKI.PTRS do (I.FSETVAL (CAR X)
        (EVALV A)))
      (for X in LOCKEDVARS do
        (IF (GETHASH X MKI.ATOMARRAY)
          THEN (I.\LOCKVAR X)
          ELSE (printout T "***Note: Locked var " X " does not exist, proceeding anyway." T))
        )
      (I.SETUPPAGEMAP)
      (I.DUMPINITPAGES (IPLUS 11776 MKI.CODESTARTOFFSET
        MKI.CODELASTPAGE VERSIONS])
    )
  )
(DEFINEQ
  (I.\CONS.UFN
    [LAMBDA (X Y)
      [COND
        ((ZEROP 1)
          (HELP)
          (PROG ((CELL (I.\CREATECELL 5)))
            (I.PUTBASEPTR CELL 0 X)
            (I.PUTBASEPTR CELL 2 Y)
            (RETURN CELL)
          )
        (PROGN
          (PROGN X)
          (PROGN Y)
          (PROGN 1)
          (PROG (CNS.PAGE CELL)
            [SETQ CNS.PAGE
              (COND
                ((NOT Y)
                  [COND
                    ((AND (SETQ CNS.PAGE ((LAMBDA ($$1)
                      (I.VAG2 (LRSH (SETQ $$1 (I.GETBASEFIXP I.LISTPDTD 14))
                        8)
                      (LLSH (LOGAND $$1 255)
                        8]
                    NIL))
                    (IGREATERP (LRSH (I.GETBASE CNS.PAGE 4)
                      8)
                      0)))
                    (T (SETQ CNS.PAGE (I.\NEXTCONSPAGE)
                      (PROG [(MK.NEWCELL (I.ADDBASE CNS.PAGE (LOGAND (I.GETBASE CNS.PAGE 4)
                        255]
                        255)
                      (LOGAND (I.PUTBASE CNS.PAGE 4 (LOGOR (LOGAND (I.GETBASE CNS.PAGE 4)
                        65280)
                        (LOGAND (LRSH (I.GETBASE .MK.NEWCELL 0)
                          8)
                          255)))
                      255)
                      (LRSH (I.PUTBASE CNS.PAGE 4 (LOGOR (LOGAND (I.GETBASE CNS.PAGE 4)
                        255]
                        (LLSH (PLUS (PROGN (LRSH (I.GETBASE CNS.PAGE 4)
                          8))
                          -1)
                        8)))
                      (I.PUTBASEPTR .MK.NEWCELL 0 X)
                      (LRSH (I.PUTBASE .MK.NEWCELL 0 (LOGOR (LOGAND (I.GETBASE .MK.NEWCELL 0)
                        4095)
                        (LLSH 8 12))))
                      12)
                    (RETURN .MK.NEWCELL)))
                  )
                (EQ (I.INTYPX Y)
                  5)
                (IGREATERP (LRSH (I.GETBASE (SETQ CNS.PAGE (I.VAG2 (I.HILOC Y)
                  (LOGAND (I.LOLOC Y)
                    65280)))
                  4)
                  8)
                  0)
              )
            ]
          )
        )
      ]
    )
  )

```

```

(SETQ CELL
(LET ((CDROFFSET (LOGAND (I.LOLOC Y)
255))
(OFFSET (LOGAND (I.GETBASE CNS.PAGE 4)
255))
CELL PRIOR)
(while (NEQ OFFSET 0)
DO (COND
((AND (ILEQ OFFSET CDROFFSET)
(IGEQ OFFSET (IDIFFERENCE CDROFFSET 14))))
(*)
(COND
(PRIOR
(*)
(LRSH ([LAMBDA ($$PUTBITS)
(DECLARE (LOCALVARS $$PUTBITS))
(I.PUTBASE $$PUTBITS 0)
(LOGOR (LOGAND (I.GETBASE $$PUTBITS 0)
255)
(LLSH (LRSH (I.GETBASE (SETQ CELL
(I.ADDBASE CNS.PAGE
OFFSET))
0)
8)
(I.ADDBASE CNS.PAGE PRIOR))
8))
(T
(*)
(LOGAND (I.PUTBASE CNS.PAGE 4
(LOGOR (LOGAND (I.GETBASE CNS.PAGE 4)
65280)
(LOGAND (LRSH (I.GETBASE (SETQ CELL
(I.ADDBASE CNS.PAGE
OFFSET))
0)
8)
255)))
(LRSH (I.PUTBASE CNS.PAGE 4
(LOGOR (LOGAND (I.GETBASE CNS.PAGE 4)
255)
(LLSH (PLUS (PROGN (LRSH (I.GETBASE CNS.PAGE 4)
8))
-1)
8)))
(I.PUTBASEPTR CELL 0 X)
(LRSH (I.PUTBASE CELL 0 (LOGOR (LOGAND (I.GETBASE CELL 0)
4095)
(LLSH (LOGOR 8 (LRSH (IDIFFERENCE CDROFFSET
OFFSET)
1))
12)))
(RETURN CELL)))
(SETQ PRIOR OFFSET)
(SETQ OFFSET (LRSH (I.GETBASE (I.ADDBASE CNS.PAGE OFFSET)
0)
8]
(*)
(*))
CELL)
(T
(LET
((PG (I.GETBASEFIXP I.LISTPDTD 14))
CELL CPG)
[while (IGREATERP PG 0)
DO
(COND
([SETQ CELL
([LAMBDA (PGA0067)
(DECLARE (LOCALVARS PGA0067))
(LET
((OFFSET (LOGAND (I.GETBASE PGA0067 4)
255))
CELL PRIOR PRIORPRIOR)
(AND
(IGEQ (LRSH (I.GETBASE PGA0067 4)
8)
2)
(while (NEQ OFFSET 0)
DO (COND
((AND PRIOR (ILEQ OFFSET PRIOR)
(IGEQ OFFSET (IDIFFERENCE PRIOR 14))))
(*)
(COND
(PRIORPRIOR
(*)
(LRSH ([LAMBDA ($$PUTBITS)
(DECLARE (LOCALVARS $$PUTBITS))

```



```

(I.PUTBASE $$PUTBITS 0
  (LOGOR (LOGAND (I.GETBASE $$PUTBITS 0)
    255)
    (LLSH (LRSH (I.GETBASE (SETQ CELL
      (I.ADDBASE PGA0067
        OFFSET))
      0)
      8)
      8]
      (I.ADDBASE PGA0067 PRIORPRIOR))
    8))
  (T
    (*)
    (LOGAND (I.PUTBASE PGA0067 4
      (LOGOR (LOGAND (I.GETBASE PGA0067 4)
        65280)
        (LOGAND (LRSH (I.GETBASE (SETQ CELL
          (I.ADDBASE
            PGA0067
            OFFSET))
          0)
          8)
          255)))
      (LRSH (I.PUTBASE PGA0067 4
        (LOGOR (LOGAND (I.GETBASE PGA0067 4)
          255)
          (LLSH (PLUS (PROGN (LRSH (I.GETBASE PGA0067 4)
            8))
            -2)
            8)))
        8)
        (I.PUTBASEPTR (I.ADDBASE PGA0067 PRIOR)
          0 Y)
        (I.PUTBASEPTR CELL 0 X)
        (LRSH (I.PUTBASE CELL 0 (LOGOR (LOGAND (I.GETBASE CELL 0)
          4095)
          (LLSH (LRSH (IDIFFERENCE PRIOR OFFSET)
            1)
            12)))
          12)
          (RETURN CELL)))
        (SETQ PRIORPRIOR PRIOR)
        (SETQ PRIOR OFFSET)
        (SETQ OFFSET (LRSH (I.GETBASE (I.ADDBASE PGA0067 OFFSET)
          0)
          8]
          (SETQ CPG ([LAMBDA ($$1)
            (I.VAG2 (LRSH (SETQ $$1 PG)
              8)
              (LLSH (LOGAND $$1 255)
                8]
                NIL]
                (RETURN CELL))
            (T (SETQ PG (I.GETBASEFIXP CPG 6]
              (OR
                CELL
                ([LAMBDA (PGA0068)
                  (DECLARE (LOCALVARS PGA0068))
                  (LET
                    ((OFFSET (LOGAND (I.GETBASE PGA0068 4)
                      255))
                     CELL PRIOR PRIORPRIOR)
                    (AND
                      (IGEQ (LRSH (I.GETBASE PGA0068 4)
                        8)
                        2)
                      (WHILE (NEQ OFFSET 0)
                        DO (COND
                          ((AND PRIOR (ILEQ OFFSET PRIOR)
                            (IGEQ OFFSET (IDIFFERENCE PRIOR 14)))
                            (*))
                          (COND
                            (PRIORPRIOR
                              (*))
                              (LRSH ([LAMBDA ($$PUTBITS)
                                (DECLARE (LOCALVARS $$PUTBITS))
                                (I.PUTBASE $$PUTBITS 0
                                  (LOGOR (LOGAND (I.GETBASE $$PUTBITS 0)
                                    255)
                                    (LLSH (LRSH (I.GETBASE (SETQ CELL
                                      (I.ADDBASE PGA0068
                                        OFFSET))
                                      0)
                                      8)
                                      8]
                                      (I.ADDBASE PGA0068 PRIORPRIOR))
                                    8))
                                  (T
                                    (*))

```

```

      (LOGAND (I.PUTBASE PGA0068 4
                (LOGOR (LOGAND (I.GETBASE PGA0068 4)
                              65280)
                        (LOGAND (LRSH (I.GETBASE (SETQ CELL
                                                (I.ADDBASE PGA0068
                                                  OFFSET))
                                      0)
                              8)
                        255)))
      (LRSH (I.PUTBASE PGA0068 4 (LOGOR (LOGAND (I.GETBASE PGA0068 4)
                                                255)
      (LLSH (PLUS (PROGN (LRSH (I.GETBASE PGA0068
                                4)
                                8))
            -2)
            8)))
      (I.PUTBASEPTR (I.ADDBASE PGA0068 PRIOR)
                    0 Y)
      (I.PUTBASEPTR CELL 0 X)
      (LRSH (I.PUTBASE CELL 0 (LOGOR (LOGAND (I.GETBASE CELL 0)
                                                4095)
      (LLSH (LRSH (IDIFFERENCE PRIOR OFFSET)
                  1)
            12)))
      12)
      (RETURN CELL)))
      (SETQ PRIORPRIOR PRIOR)
      (SETQ PRIOR OFFSET)
      (SETQ OFFSET (LRSH (I.GETBASE (I.ADDBASE PGA0068 OFFSET)
                                0)
                        8]
      (I.NEXTCONSPAGE]
      (PROGN CNS.PAGE)
      (RETURN CNS.PAGE])

```

(I.\MAIKO.CONS.UFN

```
[LAMBDA (X Y)
(COND
((ZEROP 1)
(HELP)
(PROG ((CELL (I.CREATECELL 5)))
(I.PUTBASEPTR CELL 0 X)
(I.PUTBASEPTR CELL 2 Y)
(RETURN CELL]
(PROGN
(PROGN X)
(PROGN Y)
(PROGN 1)
(PROG (CNS.PAGE)
[SETQ CNS.PAGE
(COND
[(AND (EQ (INTYPX Y)
5)
(IGREATERP (LRSH (I.GETBASE (SETQ CNS.PAGE (I.VAG2 (I.HILOLOC Y)
(LOGAND (I.LOLOC Y)
65280)))
8)
0))
([LAMBDA (PAGE A D)
(DECLARE (LOCALVARS PAGE A D))
(PROG [(MK.NEWCELL (I.ADDBASE PAGE (LOGAND (I.GETBASE PAGE 4)
255])
(LOGAND (I.PUTBASE PAGE 4 (LOGOR (LOGAND (I.GETBASE PAGE 4)
65280)
(LOGAND (LRSH (I.GETBASE .MK.NEWCELL 0)
8)
255))))
(LRSH (I.PUTBASE PAGE 4 (LOGOR (LOGAND (I.GETBASE PAGE 4)
255)
(LLSH (PLUS (PROGN (LRSH (I.GETBASE PAGE 4)
8))
-1)
8))))
(I.PUTBASEPTR .MK.NEWCELL 0 A)
(LRSH (I.PUTBASE .MK.NEWCELL 0 (LOGOR (LOGAND (I.GETBASE .MK.NEWCELL 0)
4095)
(LLSH D 12)))
12)
(RETURN .MK.NEWCELL]
```

```

CNS.PAGE X (IPLUS 8 (LRSH (LOGAND (I.LOLOC Y)
                                255)
                                1])
(T
  ([LAMBDA (PAGE A D)
    (DECLARE (LOCALVARS PAGE A D))
    (PROG [(MK.NEWCELL (I.ADDBASE PAGE (LOGAND (I.GETBASE PAGE 4)
                                                255)
                                                (*))
      (LOGAND (I.PUTBASE PAGE 4 (LOGOR (LOGAND (I.GETBASE PAGE 4)
                                                65280)
      (LOGAND (LRSH (I.GETBASE .MK.NEWCELL 0)
                    8)
                    255)))
      (*))
      (LRSH (I.PUTBASE PAGE 4 (LOGOR (LOGAND (I.GETBASE PAGE 4)
                                                255)
      (LLSH (PLUS (PROGN (LRSH (I.GETBASE PAGE 4)
                              8))
              -1)
              8)))
      8)
      (I.PUTBASEPTR .MK.NEWCELL 0 A)
      (LRSH (I.PUTBASE .MK.NEWCELL 0 (LOGOR (LOGAND (I.GETBASE .MK.NEWCELL 0)
                                                4095)
      (LLSH D 12)))
      12)
      (RETURN .MK.NEWCELL])
    (SETQ CNS.PAGE (I.NEXTCONSPAGE))
  )
  X
  (COND
    ((NULL Y)
     8)
    (T
     (IPLUS
      0
      (LRSH
       (LOGAND
        [I.LOLOC
         (PROGN
          (PROGN (PROGN (PROG [(MK.NEWCELL (I.ADDBASE CNS.PAGE (LOGAND (I.GETBASE CNS.PAGE 4)
                                                                      255)
                                                                      (*))
            (LOGAND (I.PUTBASE CNS.PAGE 4
              (LOGOR (LOGAND (I.GETBASE CNS.PAGE 4)
                          65280)
              (LOGAND (LRSH (I.GETBASE .MK.NEWCELL 0)
                            8)
                            255)))
              255)
              (*))
              (LRSH (I.PUTBASE CNS.PAGE 4
                (LOGOR (LOGAND (I.GETBASE CNS.PAGE 4)
                            255)
                (LLSH (PLUS (PROGN (LRSH (I.GETBASE CNS.PAGE 4)
                                      )
                              8))
                      -1)
                      8)))
              8)
              (I.PUTBASEPTR .MK.NEWCELL 0 Y)
              (LRSH (I.PUTBASE .MK.NEWCELL 0 (LOGOR (LOGAND (I.GETBASE
                                                                .MK.NEWCELL 0)
                                                                4095)
              (LLSH 0 12)))
              12)
              (RETURN .MK.NEWCELL])
            255)
            1]
            (PROGN CNS.PAGE)
            (RETURN CNS.PAGE]))
  )

```

(I.INITCONSPAGE

```

[LAMBDA (BASE LINK)
  (COND
    ((ZEROP 1)
     (HELP))
    (T
     (PROG ((J 254)
            CELL)
      (LOGAND (I.PUTBASE BASE 4 (LOGOR (LOGAND (I.GETBASE BASE 4)
                                                65280)
      (LOGAND (LOGXOR J 6)
              255)))
      255)

```

```

LP (COND
  ((IGREATERP J 8)
   (SETQ CELL (I.ADDBASE BASE (LOGXOR J 6)))
   (I.PUTBASEPTR CELL 0 NIL)
   (SETQ J (IDIFFERENCE J 2))
   (LRSH (I.PUTBASE CELL 0 (LOGOR (LOGAND (I.GETBASE CELL 0)
                                           255)
                                   (LLSH (LOGXOR J 6)
                                           8))))
   8)
  (GO LP)))
(LRSH (I.PUTBASE BASE 4 (LOGOR (LOGAND (I.GETBASE BASE 4)
                                       255)
                               (LLSH 124 8))))
8)
(I.PUTBASEFIXP BASE 6 (IPLUS (LLSH (I.HILOC LINK)
                                   8)
                             (LRSH (I.LOLOC LINK)
                                   8))))
(RETURN BASE])

```

(I.NEXTCONSPAGE

```

[LAMBDA NIL (*
              *)
  (PROG ((N (I.GETBASEFIXP I.LISTPDTD 14))
         PG)
    (SETQ PG (I.ALLOCMDSPAGE (I.GETBASE I.LISTPDTD 16)))
    (I.INITCONSPAGE PG (I.INITCONSPAGE (I.ADDBASE PG 256)
                                       ([LAMBDA ($$1)
                                         (I.VAG2 (LRSH (SETQ $$1 N)
                                                         8)
                                         (LLSH (LOGAND $$1 255)
                                                         8]
                                         NIL))))
    (I.PUTBASEFIXP I.LISTPDTD 14 (IPLUS (LLSH (I.HILOC PG)
                                                8)
                                         (LRSH (I.LOLOC PG)
                                                8))))
    (RETURN PG])
)

```

(DEFINEQ

(I.GETBASEBYTE

```

[LAMBDA (PTR N) (*
                  *)
  (COND
    ((EVENP N)
     (LRSH (PROGN (I.GETBASE PTR (LRSH N 1)))
            8))
    (T (LOGAND (PROGN (I.GETBASE PTR (LRSH N 1)))
               255)))
)

```

(I.PUTBASEBYTE

```

[LAMBDA (PTR DISP BYTE) (*
                           *)
  (SETQ BYTE (PROG1 BYTE))
  [I.PUTBASE PTR (LRSH (SETQ DISP (\DTEST DISP 'SMALLP))
                      1)
  (COND
    [(EVENP DISP 2)
     ([LAMBDA ($$1)
       (IPLUS (LLSH BYTE 8)
              (LOGAND $$1 255]
       (I.GETBASE PTR (LRSH DISP 1]
       (T ([LAMBDA ($$1)
            (IPLUS (LLSH (LRSH $$1 8)
                        8)
                  BYTE]
            (I.GETBASE PTR (LRSH DISP 1]
    BYTE])
)

```

(I.CREATEPAGES

```

[LAMBDA (VA N BLANKFLG LOCKFLG) (*
                                  *)
  (for I from 0 to (SUB1 N) do (MKI.NEWPAGE (I.ADDBASE VA (LLSH I 8))
                                             NIL LOCKFLG BLANKFLG))
  VA])

```

(I.NEW4PAGE

```

[LAMBDA (PTR) (*
                *)

```

```

(MKI.NEWPAGE (I.ADDBASE (MKI.NEWPAGE (I.ADDBASE (MKI.NEWPAGE (I.ADDBASE (MKI.NEWPAGE PTR)
256))
256))
256])

)

(FILESLD (SYSLOAD FROM VALUEOF DIRECTORIES)
CMLARRAY-SUPPORT)

(DEFINEQ

(I.ALLOCSTRING
[LAMBDA (N INITCHAR OLD FATFLG)
  (SETQ N (FIX N))
  (COND
    ((OR (ILESSP N 0)
      (IGREATERP N 65535))
      (LISPERROR "ILLEGAL ARG" N)))
  [COND
    ((NULL INITCHAR)
      (SETQ INITCHAR 0))
    [(PROGN
      (AND (SMALLP INITCHAR)
        (IGEQL INITCHAR 0))
      (T (SETQ INITCHAR (CHCON1 INITCHAR))
        (FATP (OR FATFLG (IGREATERP INITCHAR 255)))
        STRINGBASE)
      [SETQ STRINGBASE (I.ALLOCBLOCK (COND
        (FATP (LRSH (IPLUS N 1)
          1))
        (T (LRSH (IPLUS N 3)
          2))
        (COND
          ((STRINGP OLD)
            (PROGN ([LAMBDA ($$1)
              (PROG1 (SETQ $$1
                (PROGN (NEQ (LRSH (I.PUTBASE OLD 2 (LOGOR (LOGAND (I.GETBASE OLD 2)
32767)
(LLSH (COND
(NIL 1)
(T 0))
15)))
15)
0)
(PROGN (NEQ (LOGAND (LRSH (I.PUTBASE OLD 2 (LOGOR (LOGAND (I.GETBASE OLD 2)
49151)
(LLSH (LOGAND (COND
(NIL 1)
(T 0))
1)
14)))
14)
1)
0)
(I.PUTBASEPTR OLD 0 NIL)
NIL)
[LAMBDA (STRING NV)
  (DECLARE (LOCALVARS STRING NV))
  (LET [(%NEW-TYPE-NUMBER (SELECTC NV
    (0 %%THIN-CHAR-TYPENUMBER)
    (1 %%FAT-CHAR-TYPENUMBER)
    (SHOULDNT "Unknown typ value")
    (COND
      ((NEQ 0 (LOGAND (LRSH (I.GETBASE STRING 2)
14)
1))
      (%%SET-ARRAY-TYPE-NUMBER STRING %%NEW-TYPE-NUMBER))
      (T (LOGAND (I.PUTBASE STRING 2 (LOGOR (LOGAND (I.GETBASE STRING 2)
65280)
(LOGAND %%NEW-TYPE-NUMBER 255
)))
255]
OLD
(PROGN (COND
  (FATP 1)
  (T 0))
  ([LAMBDA (STRING NV)
    (DECLARE (LOCALVARS STRING NV))
    (I.PUTBASEFIXP STRING 4 NV)
    (I.PUTBASEFIXP STRING 6 NV)
    [COND
      ((%GENERAL-ARRAY-P STRING)
        (I.PUTBASEPTR STRING 8 (LIST NV)
        NV]
        OLD

```

```

      (PROGN N))
    [PROGN (COND
      ((NOT (EQ 0 0))
        (NEQ (LOGAND (LRSH (I.PUTBASE OLD 2
          (LOGOR (LOGAND (I.GETBASE OLD 2)
            64511)
            (LLSH (LOGAND (COND
              (T 1)
              (T 0))
            1)
          10)))
        1)
        10)
      1)
      0)))
    (COND
      ((NEQ 0 (LOGAND (LRSH (I.GETBASE OLD 2)
        14)
        1))
        (%SET-ARRAY-OFFSET OLD 0))
        (T (I.PUTBASE OLD 3 0]
      (PROGN (NEQ (LOGAND (LRSH (I.PUTBASE OLD 2 (LOGOR (LOGAND (I.GETBASE OLD 2)
        63487)
        (LLSH (LOGAND (COND
          (NIL 1)
          (T 0))
        1)
        11)))
      1)
      11)
      0)
      1)
      (NEQ (LOGAND (LRSH (I.PUTBASE OLD 2 (LOGOR (LOGAND (I.GETBASE OLD 2)
        64511)
        (LLSH (LOGAND (COND
          (NIL 1)
          (T 0))
        1)
        10)))
      1)
      10)
      0)
      1)
      (NEQ (LOGAND (LRSH (I.PUTBASE OLD 2 (LOGOR (LOGAND (I.GETBASE OLD 2)
        65023)
        (LLSH (LOGAND (COND
          (NIL 1)
          (T 0))
        1)
        9)))
      1)
      9)
      0)
      1)
      (NEQ (LOGAND (LRSH (I.PUTBASE OLD 2 (LOGOR (LOGAND (I.GETBASE OLD 2)
        65279)
        (LLSH (LOGAND (COND
          (NIL 1)
          (T 0))
        1)
        8)))
      1)
      8)
      0)
      1)
      0))
      OLD))
    (PROGN (NEQ (LOGAND (LRSH (I.PUTBASE $$1 2 (LOGOR (LOGAND (I.GETBASE $$1 2)
      49151)
      (LLSH (LOGAND (COND
        (NIL 1)
        (T 0))
      1)
      14)))
      1)
      14)
      0)
      1)
      (I.PUTBASEPTR $$1 0 STRINGBASE)
      STRINGBASE)))
    (T (SETQ OLD
      ([LAMBDA ($$1 $$2)
        (PROG1 [SETQ $$2
          ([LAMBDA (DATUMA0079)
            (DECLARE (LOCALVARS DATUMA0079))
            (PROG1 DATUMA0079 (I.PUTBASEFIXP DATUMA0079 6 $$1]
            ([LAMBDA (DATUMA0078 NEWVALUEA0077)
              (DECLARE (LOCALVARS DATUMA0078 NEWVALUEA0077))
              (PROG1 DATUMA0078 (I.PUTBASEFIXP DATUMA0078 4 NEWVALUEA0077]
              [[LAMBDA (DATUMA0076)
                (DECLARE (LOCALVARS DATUMA0076))
                (PROG1 DATUMA0076 (I.PUTBASE DATUMA0076 3 0]
                ([LAMBDA (DATUMA0075 NEWVALUEA0074)

```

```

(DECLARE (LOCALVARS DATUMA0075 NEWVALUEA0074))
(PROG1 DATUMA0075
  (LOGAND (I.PUTBASE DATUMA0075 2 (LOGOR (LOGAND (I.GETBASE DATUMA0075 2)
                                                65280)
                                            (LOGAND NEWVALUEA0074 255)))
    255)))
([LAMBDA (DATUMA0073 NEWVALUEA0072)
  (DECLARE (LOCALVARS DATUMA0073 NEWVALUEA0072))
  (PROG1 DATUMA0073
    (NEQ (LOGAND (LRSH (I.PUTBASE DATUMA0073 2 (LOGOR (LOGAND (I.GETBASE
                                                                DATUMA0073 2)
                                                                64511)
                                                            (LLSH (LOGAND (COND
                                                                (NEWVALUEA0072
                                                                1)
                                                                (T 0))
                                                                1)
                                                                10)))
                                                                10)
                                                                1)
                                                                0)))
    ([LAMBDA (DATUMA0071)
      (DECLARE (LOCALVARS DATUMA0071))
      (PROG1 DATUMA0071
        (NEQ (LOGAND (LRSH (I.PUTBASE DATUMA0071 2
                                                                (LOGOR (LOGAND (I.GETBASE DATUMA0071 2)
                                                                61439)
                                                            (LLSH (LOGAND (COND
                                                                (T 1)
                                                                (T 0))
                                                                1)
                                                                12)))
                                                                12)
                                                                1)
                                                                0)))
        ([LAMBDA (DATUMA0070)
          (DECLARE (LOCALVARS DATUMA0070))
          (PROG1 DATUMA0070
            (NEQ (LRSH (I.PUTBASE DATUMA0070 2 (LOGOR (LOGAND (I.GETBASE DATUMA0070 2)
                                                                32767)
                                                            (LLSH (COND
                                                                (NIL 1)
                                                                (T 0))
                                                                15)))
                                                                15)
                                                                0)))
            ([LAMBDA (DATUMA0069)
              (DECLARE (LOCALVARS DATUMA0069))
              (PROG1 DATUMA0069 (I.PUTBASEPTR DATUMA0069 0 NIL)
                (I.CREATECELL 14]
                (NOT (EQ 0 0)))
                (COND
                  ((EQ (COND
                        (FATP 1)
                        (T 0))
                        1)
                    %%FAT-CHAR-TYPENUMBER)
                  (T %%THIN-CHAR-TYPENUMBER]
                (SETQ $$1 N]
                (PROGN (NEQ (LOGAND (LRSH (I.PUTBASE $$2 2 (LOGOR (LOGAND (I.GETBASE $$2 2)
                                                                49151)
                                                            (LLSH (LOGAND (COND
                                                                (NIL 1)
                                                                (T 0))
                                                                1)
                                                                14)))
                                                                14)
                                                                1)
                                                                0)
                    (I.PUTBASEPTR $$2 0 STRINGBASE)
                    STRINGBASE))
                NIL NIL]
                (COND
                  ((NEQ 0 INITCHAR)
                    (COND
                      (FATP (for I from 0 to (SUB1 N) do (I.PUTBASE STRINGBASE I INITCHAR)))
                      (T (for I from 0 to (SUB1 N) do (I.PUTBASEBYTE STRINGBASE I INITCHAR]
                    OLD])
                  (*
                    (*
                    (LAMBDA (LOCAL-ARRAY)
                      (PROG ((SIZE (ffetch (ONED-ARRAY TOTAL-SIZE) of LOCAL-ARRAY))
                            (BASE (ffetch (ONED-ARRAY BASE) of LOCAL-ARRAY))
                            (OFFSET (ffetch (ONED-ARRAY OFFSET) of LOCAL-ARRAY))
                            (TYPENUMBER (ffetch (ONED-ARRAY TYPE-NUMBER) of LOCAL-ARRAY))
                    (*

```

(I.%%COPY-ONED-ARRAY

```

(LAMBDA (LOCAL-ARRAY)
  (PROG ((SIZE (ffetch (ONED-ARRAY TOTAL-SIZE) of LOCAL-ARRAY))
        (BASE (ffetch (ONED-ARRAY BASE) of LOCAL-ARRAY))
        (OFFSET (ffetch (ONED-ARRAY OFFSET) of LOCAL-ARRAY))
        (TYPENUMBER (ffetch (ONED-ARRAY TYPE-NUMBER) of LOCAL-ARRAY))
  (*

```

```

NCELLS REMOTE-ARRAY REMOTE-BASE)
(if (NEQ OFFSET 0)
  then (ERROR "Can't copy an array with non-zero offset"))
(if (EQ (%%TYPENUMBER-TO-GC-TYPE TYPENUMBER)
  1)
  then (ERROR "Can't copy pointer arrays"))
(SETQ NCELLS (LRSH (IPLUS (ITIMES (IPLUS SIZE OFFSET)
  (%%TYPENUMBER-TO-BITS-PER-ELEMENT TYPENUMBER))
  31)
  5))
[SETQ REMOTE-ARRAY ([LAMBDA (DATUMA0089)
  (DECLARE (LOCALVARS DATUMA0089))
  (PROG1 DATUMA0089 (I.PUTBASEFIXP DATUMA0089 6 SIZE)
    ([LAMBDA (DATUMA0088 NEWVALUEA0087)
      (DECLARE (LOCALVARS DATUMA0088 NEWVALUEA0087))
      (PROG1 DATUMA0088 (I.PUTBASEFIXP DATUMA0088 4 NEWVALUEA0087)
        [[LAMBDA (DATUMA0086)
          (DECLARE (LOCALVARS DATUMA0086))
          (PROG1 DATUMA0086
            (LOGAND (I.PUTBASE DATUMA0086 2 (LOGOR (LOGAND (I.GETBASE DATUMA0086 2)
              65280)
              (LOGAND TYPENUMBER 255))))
            255))]]
          ([LAMBDA (DATUMA0085 NEWVALUEA0084)
            (DECLARE (LOCALVARS DATUMA0085 NEWVALUEA0084))
            (PROG1 DATUMA0085
              (NEQ (LOGAND (LRSH (I.PUTBASE DATUMA0085 2
                (LOGOR (LOGAND (I.GETBASE DATUMA0085 2)
                  65023)
                  (LLSH (LOGAND (COND
                    (NEWVALUEA0084 1)
                    (T 0)))
                    1)
                    9))))
                1)
                9))
              1)
              0))]]
            ([LAMBDA (DATUMA0083 NEWVALUEA0082)
              (DECLARE (LOCALVARS DATUMA0083 NEWVALUEA0082))
              (PROG1 DATUMA0083
                (NEQ (LOGAND (LRSH (I.PUTBASE DATUMA0083 2
                  (LOGOR (LOGAND (I.GETBASE DATUMA0083 2)
                    61439)
                    (LLSH (LOGAND (COND
                      (NEWVALUEA0082 1)
                      (T 0)))
                      1)
                      12))))
                  1)
                  12)
                0))]]
              ([LAMBDA (DATUMA0081 NEWVALUEA0080)
                (DECLARE (LOCALVARS DATUMA0081 NEWVALUEA0080))
                (PROG1 DATUMA0081 (I.PUTBASEPTR DATUMA0081 0 NEWVALUEA0080)
                  (I.CREATECELL 14)
                  (I.ALLOCBLOCK NCELLS))
                (%%CHAR-TYPE-P TYPENUMBER))
                (NEQ 0 (LOGAND (LRSH (I.GETBASE LOCAL-ARRAY 2)
                  9)
                  1)
                  (I.GETBASEFIXP LOCAL-ARRAY 4]
                (SETQ REMOTE-BASE (I.GETBASEPTR REMOTE-ARRAY 0))
                (for I from 0 to (SUB1 (LLSH NCELLS 1)) do (I.PUTBASE REMOTE-BASE I (\GETBASE BASE I)))
                (RETURN REMOTE-ARRAY])

```

(I.%%COPY-STRING-TO-ARRAY

```

[LAMBDA (LOCAL-STRING)
  (PROG ((SIZE (NCHARS LOCAL-STRING))
    REMOTE-BASE REMOTE-ARRAY)
    (SETQ REMOTE-BASE (I.ALLOCBLOCK (LRSH (IPLUS (ITIMES SIZE 8)
      31)
      5)))
    [SETQ REMOTE-ARRAY ([LAMBDA (DATUMA0094)
      (DECLARE (LOCALVARS DATUMA0094))
      (PROG1 DATUMA0094 (I.PUTBASEFIXP DATUMA0094 6 SIZE)
        ([LAMBDA (DATUMA0093)
          (DECLARE (LOCALVARS DATUMA0093))
          (PROG1 DATUMA0093 (I.PUTBASEFIXP DATUMA0093 4 SIZE)
            ([LAMBDA (DATUMA0092)
              (DECLARE (LOCALVARS DATUMA0092))
              (PROG1 DATUMA0092
                (LOGAND (I.PUTBASE DATUMA0092 2 (LOGOR (LOGAND (I.GETBASE DATUMA0092 2)
                  65280)
                  (LOGAND 67 255))))
                255))]]

```



```

([LAMBDA (DATUMA0091)
  (DECLARE (LOCALVARS DATUMA0091))
  (PROG1 DATUMA0091
    (NEQ (LOGAND (LRSH (I.PUTBASE DATUMA0091 2
      (LOGOR (LOGAND (I.GETBASE DATUMA0091 2)
        61439)
      (LLSH (LOGAND (COND
        (T 1)
        (T 0))
      1)
    12)))
    12)
    1)
  0)))
([LAMBDA (DATUMA0090)
  (DECLARE (LOCALVARS DATUMA0090))
  (PROG1 DATUMA0090 (I.PUTBASEPTR DATUMA0090 0 REMOTE-BASE]
  (I.CREATECELL 14]
[for I from 0 to (SUB1 SIZE) do (I.PUTBASEBYTE REMOTE-BASE I (NTHCHARCODE LOCAL-STRING (ADD1 I]
(RETURN REMOTE-ARRAY])

```

)

(DEFINEQ

(I.#BLOCKDATA CELLS

```

[LAMBDA (DATAWORD)
  (PROG ((TYPENO (I.INTYPX DATAWORD)))
    (RETURN (COND
      [(EQ 0 TYPENO)
        (COND
          ((AND (EQ 0 (I.INTYPX DATAWORD))
            (IGEQ (I.HILOC DATAWORD)
              46))
            (IDIFFERENCE (I.GETBASE (I.ADDBASE DATAWORD (IMINUS 2))
              1)
              2))
          (T (\ILLEGAL.ARG DATAWORD]
        (T (OR (AND (OR I.HUNKING? (NEQ 0 (LOGAND (LRSH (I.GETBASE (I.ADDBASE (I.VAG2 20 4096)
          (ITIMES TYPENO 18))
          6)
          14)
          1)))
            (LRSH (I.GETBASE (I.ADDBASE (I.VAG2 20 4096)
              (ITIMES TYPENO 18))
              3)
              1))
            (\ILLEGAL.ARG DATAWORD]))

```

(I.PREFIXALIGNMENT?

```

[LAMBDA (ARLEN INITONPAGE ALIGN GCTYPE BASE)
  (PROG ((DAT (LRSH [I.LOLOC (PROGN (PROGN (I.ADDBASE BASE 2]
    1))
    (ADJUSTMENT 0)
    FUDGE)
  LP [COND
    ((AND ALIGN (NEQ (SETQ FUDGE (IREMAINDER DAT ALIGN))
      0))
      [SETQ ADJUSTMENT (PLUS ADJUSTMENT (SETQ FUDGE (IDIFFERENCE ALIGN FUDGE]
      (SETQ DAT (PLUS DAT FUDGE]
    (COND
      ([AND INITONPAGE (NEQ (LOGAND DAT (CONSTANT (LOGXOR (SUB1 128)
        -1)))
        (LOGAND (IPLUS DAT INITONPAGE -1)
          (CONSTANT (LOGXOR (SUB1 128)
            -1]
          (SETQ ADJUSTMENT (PLUS ADJUSTMENT (SETQ FUDGE (IDIFFERENCE 128 (IMOD DAT 128]
          (SETQ DAT (PLUS DAT FUDGE))
        ))
      (COND
        ([AND (EQ GCTYPE 2)
          (IGREATERP (IDIFFERENCE ARLEN 2)
            (SETQ FUDGE (IDIFFERENCE 32768 (SETQ DAT (IMOD DAT 32768]
            (SETQ ADJUSTMENT (PLUS ADJUSTMENT FUDGE))
            (SETQ DAT (PLUS DAT FUDGE))
          ))
        (RETURN ADJUSTMENT]))

```

(I.ALLOCBLOCK

```

[LAMBDA (NCELLS GCTYPE INITONPAGE ALIGN)
  (DECLARE (GLOBALVARS I.ArrayFrLst))

```

```
(COND
  ((ILESSP NCELLS 2)
    (COND
      ((ILESSP NCELLS 0)
        (\ILLEGAL.ARG NCELLS)))
      (SETQ NCELLS 2))
    ((IGREATERP NCELLS 65533)
      (\LISPERROR NCELLS "ARRAY STORAGE BLOCK TOO LARGE"))) (*)
  (SELECTQ GCTYPE
    (NIL (SETQ GCTYPE 0))
    (T (SETQ GCTYPE 1))
    NIL) *)
  (COND
    ((AND INITONPAGE (OR (ILESSP INITONPAGE 0)
      (IGREATERP INITONPAGE 128)))
      (\ILLEGAL.ARG INITONPAGE)))
    (COND
      ((NULL ALIGN))
      ((OR (ILESSP ALIGN 0)
        (IGREATERP ALIGN 128))
        (\ILLEGAL.ARG ALIGN))
      ((ILEQ ALIGN 1)
        (SETQ ALIGN))
      ((AND INITONPAGE (PROGN
        NIL))
        (*))
      (ERROR "INITONPAGE and ALIGN too high")))
    (OR (AND I.HUNKING? (ILEQ NCELLS 64)
      (I.ALLOCHUNK NCELLS GCTYPE INITONPAGE ALIGN))
      (PROG ((ARLEN (IPLUS NCELLS 2))
        ABLOCK)
        RETRY
        (PROGN
          (SETQ ABLOCK (OR (NIL ARLEN GCTYPE INITONPAGE ALIGN)
            (I.ALLOCBLOCK.NEW ARLEN GCTYPE INITONPAGE ALIGN)
            (PROGN (FRPTQ 10 (RECLAIM))
              (*))
              (NIL ARLEN GCTYPE INITONPAGE ALIGN))
            (GO FULL)))
          (NEQ (LOGAND (I.PUTBASE ABLOCK 0 (LOGOR (LOGAND (I.GETBASE ABLOCK 0)
            65534)
            (LOGAND (COND
              (T 1)
              (T 0))
            1)))
            0)
            1)
            (NEQ (LOGAND ([LAMBDA ($$PUTBITS)
              (DECLARE (LOCALVARS $$PUTBITS))
              (I.PUTBASE $$PUTBITS 0 (LOGOR (LOGAND (I.GETBASE $$PUTBITS 0)
                65534)
                (LOGAND (COND
                  (T 1)
                  (T 0))
                1]
              (I.LAMBDA (BASE N)
                (DECLARE (LOCALVARS BASE N))
                (I.ADDBASE (I.ADDBASE BASE N)
                  N)
                ABLOCK
                (IDIFFERENCE (I.GETBASE ABLOCK 1)
                  1)))
            1)
            0)
            (LOGAND (LRSH (I.PUTBASE ABLOCK 0 (LOGOR (LOGAND (I.GETBASE ABLOCK 0)
              65529)
              (LLSH (LOGAND GCTYPE 3)
                1)))
            1)
            3)
            (NIL ABLOCK NIL)
            (PROGN NCELLS)
            (SETQ ABLOCK (I.ADDBASE ABLOCK 2))
            (PROGN ABLOCK)
            (RETURN ABLOCK))
          FULL
          (LISPERROR "ARRAYS FULL" NIL T)
          (GO RETRY]))
          (*))
          (*))
          (*))
  (I.MAIKO.ALLOCBLOCK
    [LAMBDA (NCELLS GCTYPE INITONPAGE ALIGN)
      (*))
      (*))
      (*))
  (DECLARE (GLOBALVARS I.ArrayFrLst))
  (COND
    ((ILESSP NCELLS 2)
      (COND
```

```

((ILESSP NCELLS 0)
 (\ILLEGAL.ARG NCELLS)))
(SETQ NCELLS 2))
((IGREATERP NCELLS 65533)
 (\LISPERROR NCELLS "ARRAY STORAGE BLOCK TOO LARGE")))) (*)
(SELECTQ GCTYPE
 (NIL (SETQ GCTYPE 0))
 (T (SETQ GCTYPE 1))
 NIL)
(*)
(*)
(*)

(COND
 ((NULL ALIGN))
 ((OR (ILESSP ALIGN 0)
      (IGREATERP ALIGN 128))
  (\ILLEGAL.ARG ALIGN))
 ((ILEQ ALIGN 1)
  (SETQ ALIGN))
 (AND INITONPAGE (PROGN
                   NIL))
 (ERROR "INITONPAGE and ALIGN too high")))) (*)
(OR (AND I.HUNKING? (ILEQ NCELLS 64)
      (I.ALLOCHUNK NCELLS GCTYPE NIL ALIGN))
  (PROG ((ARLEN (IPLUS NCELLS 2))
        ABLOCK)
    RETRY
    (PROGN
      (SETQ ABLOCK (OR (NIL ARLEN GCTYPE NIL ALIGN)
                       (I.ALLOCBLOCK.NEW ARLEN GCTYPE NIL ALIGN)
                       (PROGN (FRPTQ 10 (RECLAIM))
                              (NIL ARLEN GCTYPE INITONPAGE ALIGN))
                              (GO FULL))))
      (NEQ (LOGAND (I.PUTBASE ABLOCK 0 (LOGOR (LOGAND (I.GETBASE ABLOCK 0)
                                                       65534)
              (LOGAND (COND
                        (T 1)
                        (T 0))
              1)))
            1)
        0)
      (NEQ (LOGAND ([LAMBDA ($$PUTBITS)
                     (DECLARE (LOCALVARS $$PUTBITS))
                     (I.PUTBASE $$PUTBITS 0 (LOGOR (LOGAND (I.GETBASE $$PUTBITS 0)
                                                            65534)
              (LOGAND (COND
                        (T 1)
                        (T 0))
              1]
                     (LAMBDA (BASE N)
                       (DECLARE (LOCALVARS BASE N))
                       (I.ADDBASE (I.ADDBASE BASE N)
                                   N)
                       ABLOCK
                       (IDIFFERENCE (I.GETBASE ABLOCK 1)
                                   1)))
                     1)
            0)
      (LOGAND (LRSH (I.PUTBASE ABLOCK 0 (LOGOR (LOGAND (I.GETBASE ABLOCK 0)
                                                       65529)
              (LLSH (LOGAND GCTYPE 3)
              1)))
            1)
        3)
      (NIL ABLOCK NIL)
      (PROGN NCELLS)
      (SETQ ABLOCK (I.ADDBASE ABLOCK 2))
      (PROG1 (PROGN ABLOCK)
        (PROGN 1))
      (RETURN ABLOCK))
    FULL
    (LISPERROR "ARRAYS FULL" NIL T)
    (GO RETRY))
  (*))

(I.ALLOCBLOCK.NEW
 [LAMBDA (ARLEN GCTYPE INITONPAGE ALIGN)
  (DECLARE (GLOBALVARS I.ArrayFrLst I.NxtArrayPage))
  (PROG (FINALWORD FINALPAGE NEXTFREEBLOCK PREFIXLEN)
    RETRY
    [COND
     ((AND (OR INITONPAGE ALIGN)
          (NEQ 0 (SETQ PREFIXLEN (I.PREFIXALIGNMENT? ARLEN INITONPAGE ALIGN GCTYPE I.ArrayFrLst)
              (*))
          (COND
           ((SETQ PREFIXLEN (I.ALLOCBLOCK.NEW PREFIXLEN))
            (I.MERGEBACKWARD PREFIXLEN)
            (*))

```

```

)
(T (RETURN]
(SETQ FINALWORD (I.ADDBASE (I.ADDBASE I.ArrayFrLst ARLEN)
                          (SUB1 ARLEN)))
                          (*))
(SETQ NEXTFREEBLOCK (I.ADDBASE FINALWORD 1))
[COND
  ((IGREATERP (SETQ FINALPAGE (IPLUS (LLSH (I.HILOC FINALWORD)
                                             8)
                                         (LRSH (I.LOLOC FINALWORD)
                                                  8)))
              (IDIFFERENCE I.NxtMDSPage 128)))
  (*))
  (SELECTQ (NIL (ADD1 (IDIFFERENCE FINALPAGE I.NxtArrayPage))))
  (T
   (*))
  (0
   (*))
  (GO RETRY))
  (RETURN NIL]
  (*))
(until (IGREATERP I.NxtArrayPage FINALPAGE) do (I.MAKEMDSENTRY I.NxtArrayPage 0)
      (I.NEW2PAGE ([LAMBDA ($$1)
                    (I.VAG2 (LRSH (SETQ $$1 I.NxtArrayPage)
                                   8)
                           (LLSH (LOGAND $$1 255)
                                   8]
                           NIL)))
      (SETQ I.NxtArrayPage (IPLUS I.NxtArrayPage 2))))
(RETURN (PROG1 (I.MAKEFREEARRAYBLOCK I.ArrayFrLst ARLEN)
              (SETQ I.ArrayFrLst NEXTFREEBLOCK]))

```



```

        (PROGN HUNK))
      (T
        (SETQ STRADDLERS (I.CONSUMER HUNK STRADDLERS)
          (*))
        (COND
          ((IGREATERP (SETQ FAILCNT (PLUS FAILCNT 1))
            16)
            (*))
          (GO LOSE))
          ((EQ FAILCNT 8)
            (*))
          (SETQ NCELLS (ADD1 HUNKSIZE))
          (AND STRADDLERS (SETQ \HUNKREJECTS (NCONC STRADDLERS \HUNKREJECTS)))
          (GO BEG)))
        (GO LP)))
    LOSE
    (AND STRADDLERS (SETQ \HUNKREJECTS (NCONC STRADDLERS \HUNKREJECTS)))
    (RETURN))
  )

```

(DEFINEQ

(I.PREINITARRAYS

```

[LAMBDA NIL
  (*))
  (DECLARE (GLOBALVARS I.ArrayFrLst I.ArrayFrLst2 I.NxtArrayPage))
  (SETQ I.ArrayFrLst (I.VAG2 46 0))
  (SETQ I.ArrayFrLst2 (I.VAG2 64 0))
  (SETQ I.NxtArrayPage (IPLUS (LLSH (I.HILO I.ArrayFrLst)
    8)
    (LRSH (I.LOLOC I.ArrayFrLst)
    8)))

```

(I.POSTINITARRAYS

```

[LAMBDA (AFTERCODEPTR CODESTARTPAGE CODENEXTPAGE)
  (*))
  (SETQ I.FREEBLOCKBUCKETS (I.ALLOCBLOCK (ADD1 30)))
  (PROG [(EXTRACELLS (IDIFFERENCE (LLSH CODESTARTPAGE 7)
    (IPLUS (LLSH (I.HILO I.ArrayFrLst)
      15)
      (LRSH (I.LOLOC I.ArrayFrLst)
      1)))
    (*))
    ((IGREATERP EXTRACELLS 65535)
      (printout T T T "POSTINITARRAYS: You pre-allocated too much string space." T 19
        "MKI.CODESTARTOFFSET on MAKEINIT should be reduced by about " (IDIFFERENCE (LRSH EXTRACELLS
          7)
          10)
        ". " T)
        (HELP))
      ((IGEQU EXTRACELLS 4)
        (printout T T T "POSTINITARRAYS: There were " (LRSH EXTRACELLS 7)
          " allocated but unused array pages." T T))
      (T (printout T T "POSTINITARRAYS: String space overflowed into code-arrays" T 19 "You should add
        at least " (ADD1 (LRSH (IMINUS EXTRACELLS)
          7))
        " to MKI.CODESTARTOFFSET on MAKEINIT." T)
        (HELP)))
      (I.LINKBLOCK (I.ALLOCBLOCK.NEW EXTRACELLS))
      (SETQ I.ArrayFrLst AFTERCODEPTR)
      (SETQ I.NxtArrayPage CODENEXTPAGE)
      (for VP from (IPLUS (LLSH (I.HILO (I.VAG2 46 0))
        8)
        (LRSH (I.LOLOC (I.VAG2 46 0))
        8))
        to (IPLUS (LLSH (I.HILO I.NxtArrayPage)
          8)
          (LRSH (I.LOLOC I.NxtArrayPage)
          8))
        by (LRSH 512 8) do (I.MAKEMDSENTRY VP 0))

```

(I.FILEARRAYBASE

```

[LAMBDA NIL
  (*))
  (I.ADDBASE (I.VAG2 46 0)
    (IPLUS (LLSH MKI.CODESTARTOFFSET 8)
      (LRSH (IDIFFERENCE (GETFILEPTR (OUTPUT))
        MKI.FirstDataByte)
        1)))

```

(I.FILEBLOCKTRAILER

```

[LAMBDA (BLOCKINFO)
  (*))
  (BOUT16 OUTX 43689)
  (BOUT16 OUTX BLOCKINFO))

```

(I.FILECODEBLOCK

```

[LAMBDA (NCELLS INITONPAGE)
  (PROG (PREFIXLEN (ARLEN (IPLUS NCELLS 2)))
    (COND
      ([NEQ 0 (SETQ PREFIXLEN (I.\PREFIXALIGNMENT? ARLEN INITONPAGE 2 2 (I.FILEARRAYBASE]
        (I.FILEPATCHBLOCK PREFIXLEN)))
      (BOUT16 OUTX 43693)
      (BOUT16 OUTX ARLEN)
      (RETURN ARLEN]))
  )

```

(I.FILEPATCHBLOCK

```

[LAMBDA (ARLEN)
  (BOUT16 OUTX 43688)
  (BOUT16 OUTX ARLEN)
  (COND
    ((IGREATERP ARLEN 1)
      (BOUTZEROS (LLSH (IDIFFERENCE ARLEN 2)
        2))
      (BOUT16 OUTX 43688)
      (BOUT16 OUTX ARLEN)))
  NIL))

```

)

(DEFINEQ

(I.\SETUP.HUNK.TYPENUMBERS

```

[LAMBDA NIL
  (SETQ INITIALDTDCONTENTS
    (APPEND \BUILT-IN-SYSTEM-TYPES
      (I.\COMPUTE.HUNK.TYPEDECLS ' (2 4 5 6 7 8 10 12 16 24 32 42 64)
        1
        '\PTRHUNK)
      (I.\COMPUTE.HUNK.TYPEDECLS ' (1 2 3 4 5 6 7 8 9 10 12 14 16 20 24 28 32 40 48 64)
        0
        '\UNBOXEDHUNK)
      (I.\COMPUTE.HUNK.TYPEDECLS ' (12 16 20 24 28 32 36 42 50 64)
        2
        '\CODEHUNK]))

```

(I.\COMPUTE.HUNK.TYPEDECLS

```

[LAMBDA (SIZELST GCTYPE PREFIX)
  (for HUNKSIZE in SIZELST BIND (FINAL _ (AND (EQ GCTYPE CODEBLOCK.GCT)
    '\RECLAIMCODEBLOCK))
    until (> HUNKSIZE \MAX.CELLSPERHUNK) collect (LIST (PACK* PREFIX HUNKSIZE)
      (UNFOLD HUNKSIZE WORDSPERCELL)
      (COND
        ((EQ GCTYPE PTRBLOCK.GCT)
          ; Compute DTDPTRS list, i.e., which fields are pointers (all of
          ; them)
          (for I from 0 by 2
            to (SUB1 (UNFOLD HUNKSIZE WORDSPERCELL))
            collect I)))
      FINAL]))

```

(I.TURN.ON.HUNKING

```

[LAMBDA NIL
  (SETQ I.UNBOXEDHUNK.TYPENUM.TABLE
    (I.\SETUP.TYPENUM.TABLE ' (1 2 3 4 5 6 7 8 9 10 12 14 16 20 24 28 32 40 48 64)
      0
      '\UNBOXEDHUNK))
  (SETQ I.CODEHUNK.TYPENUM.TABLE (I.\SETUP.TYPENUM.TABLE ' (12 16 20 24 28 32 36 42 50 64)
    2
    '\CODEHUNK))
  (SETQ I.PTRHUNK.TYPENUM.TABLE
    (I.\SETUP.TYPENUM.TABLE ' (2 4 5 6 7 8 10 12 16 24 32 42 64)
      1
      '\PTRHUNK))
  (SETQ I.HUNKING? T))

```

(I.\SETUP.TYPENUM.TABLE

```

[LAMBDA (SIZELST GCTYPE PREFIX)
  (for I from 0 to 64 bind (HUNKSIZE _ -1)
    (SIZEL _ SIZELST)
    (TABLE _ (I.\ALLOCBLOCK (LRSH (IPLUS (IPLUS 4 64)

```



```

[COND
  (LOCALARGS (SETQ STARTLOCALS STARTPC)
    [SETQ LOCALSIZE (LOGAND [IPLUS (ADD1 (LLSH (LRSH (FLENGTH LOCALARGS)
1)
1))
(CONSTANT (SUB1 (IQUOTIENT 4 2]
(CONSTANT (LOGXOR (SUB1 (IQUOTIENT 4 2))
-1]
(*)
(SETQ LOCALSIZE (LLSH LOCALSIZE 1))
(SETQ STARTPC (PLUS STARTPC (LLSH LOCALSIZE 1]
[SETQ REALSIZE (LOGAND (IPLUS (IPLUS STARTPC CODELEN)
(CONSTANT (SUB1 8)))
(CONSTANT (LOGXOR (SUB1 8)
-1]
[SETQ CA (SCRATCHARRAY REALSIZE (LOGAND (IPLUS (ADD1 (LRSH (IPLUS STARTPC 3)
2))
(CONSTANT (SUB1 2)))
(CONSTANT (LOGXOR (SUB1 2)
-1]
(AIN CA STARTPC CODELEN INSTREAM)
(for X on NAMETABLE by (CDDDR X) as NT1 from (IPLUS (SUB1 (LLSH (CONSTANT (I.WORDSPERNAMEENTRY)
1))
(LLSH (PROGN 8)
1))
by (LLSH (CONSTANT (I.WORDSPERNAMEENTRY)
1)
bind (NTBYTESIZE _ (LLSH NTSIZE 1))
do (I.FIXUPSYM CA NT1 (CADDR X)
-1)
(I.SETSTKNTOFFSET CA (IPLUS NT1 NTBYTESIZE)
(SELECTQ (CAR X)
(P (CONSTANT (LLSH 2 14)))
(F (OR FVAROFFSET (SETQ FVAROFFSET (LLSH (LRSH NT1 2)
1)))
(CONSTANT (LLSH 3 14)))
(I (CONSTANT (LLSH 0 14)))
(SHOULDNT))
(CADR X))
)
[COND
  (LOCALARGS
    (for X on LOCALARGS by (CDDR X) as NT from (IPLUS (SUB1 (LLSH (CONSTANT (
I.WORDSPERNAMEENTRY
))
1))
STARTLOCALS)
by (CONSTANT (LLSH (CONSTANT (I.WORDSPERNAMEENTRY)
1))
do (I.FIXUPSYM CA NT (CADR X)
-1)
(I.SETSTKNTOFFSET CA (IPLUS NT LOCALSIZE)
(CONSTANT (LLSH 0 14))
(CAR X))
]
[PROGN
  ([LAMBDA (DEFA0096 VALUEA0095)
    (DECLARE (LOCALVARS DEFA0096 VALUEA0095))
    (\BYTESETA DEFA0096 2 (LRSH VALUEA0095 8))
    (\BYTESETA DEFA0096 (ADD1 2)
      (IMOD VALUEA0095 (CONSTANT (LLSH 1 8]
    CA
    (UNSIGNED (PROGN (COND
      ((EQ ARGTYPE 2)
      -1)
      (T NARGS)))
    16))
  ([LAMBDA (DEFA0098 VALUEA0097)
    (DECLARE (LOCALVARS DEFA0098 VALUEA0097))
    (\BYTESETA DEFA0098 4 (LRSH VALUEA0097 8))
    (\BYTESETA DEFA0098 (ADD1 4)
      (IMOD VALUEA0097 (CONSTANT (LLSH 1 8]
    CA
    (UNSIGNED (PROGN (SUB1 (LRSH (IPLUS (IPLUS NLOCALS NFREEVARS)
1)
1)))
    16))
  [PROGN (\BYTESETA CA 6 (LRSH STARTPC 8))
    (\BYTESETA CA (ADD1 6)
      (IMOD STARTPC (CONSTANT (LLSH 1 8]
    (\BYTESETA CA 8 (LOGOR (LOGAND (\BYTELTA CA 8)
207)
(LLSH (LOGAND ARGTYPE 3)
4)))
    (I.FIXUPPTR CA 11 (I.COPY FRAMEFRAME))
  [PROGN (\BYTESETA CA 12 (LRSH NTSIZE 8))

```

```

        (\BYTESETA CA (ADD1 12)
          (IMOD NTSIZE (CONSTANT (LLSH 1 8]
(\BYTESETA CA 14 NLOCALS)
(\BYTESETA CA 15 (PROGN (OR FVAROFFSET 0)))
([LAMBDA (DEFA0100 VALUEA0099)
  (DECLARE (LOCALVARS DEFA0100 VALUEA0099))
  (\BYTESETA DEFA0100 0 (LRSH VALUEA0099 8))
  (\BYTESETA DEFA0100 (ADD1 0)
    (IMOD VALUEA0099 (CONSTANT (LLSH 1 8]
    CA
    (PROGN (I.\STKMIN CA] (*)
(for X on (READ) by (CDDR X) do (I.FIXUPSYM CA (IPLUS (CAR X)
  STARTPC)
  (CADR X)
  -1))
(for X on (READ) by (CDDR X) do (I.FIXUPSYM CA (IPLUS (CAR X)
  STARTPC)
  (CADR X)
  -1))
[for X on (READ) by (CDDR X) do (I.FIXUPPTR CA (IPLUS (CAR X)
  STARTPC)
  (I.\COPY (CADR X]
(I.PUTDEFN FN CA (IPLUS STARTPC CODELEN])

```

)

(RPAQQ \OPCODES

```

((0 -X- 0)
(1 CAR 0 T 0 \CAR.UFN)
(2 CDR 0 T 0 \CDR.UFN)
(3 LISTP 0 T 0 LISTP)
(4 NTYPX 0 T 0 NTYPX)
(5 TYPEP 1 TYPEP 0 \TYPEP.UFN)
(6 DTEST 4 ATOM 0 \DTEST.UFN)
(7 UNWIND 2 T (UNWIND 1)
  \UNWIND.UFN)
(8 FN0 4 FN 1)
(9 FN1 4 FN 0)
(10 FN2 4 FN -1)
(11 FN3 4 FN -2)
(12 FN4 4 FN -3)
(13 FNX 5 FNX FNX)
(14 APPLYFN 0 T -1)
(15 CHECKAPPLY* 0 T 0 \CHECKAPPLY* (4K 12K))
(16 RETURN 0 T (JUMP 1)
  \HARDRETURN)
(17 BIND 2)
(18 UNBIND 0)
(19 DUNBIND 0)
(20 RPLPTR.N 1 T -1 \RPLPTR.UFN (4K))
(21 GCREF 1 T 0 \HTFIND)
(22 ASSOC 0 T -1 ASSOC (4K DORADO))
(23 GVAR_ 4 ATOM 0 \SETGLOBALVAL.UFN)
(24 RPLACA 0 T -1 \RPLACA.UFN 4K)
(25 RPLACD 0 T -1 \RPLACD.UFN 4K)
(26 CONS 0 T -1 \CONS.UFN)
(27 CMLASSOC 0 T -1 CL::%%SIMPLE-ASSOC (4K DORADO))
(28 FMEMB 0 T -1 FMEMB (4K DORADO))
(29 CMLMEMBER 0 T -1 CL::%%SIMPLE-MEMBER (4K DORADO))
(30 FINDKEY 1 T 0 \FINDKEY.UFN)
(31 CREATECELL 0 T 0 \CREATECELL 4K)
(32 BIN 0 T 0 \BIN 4K)
(33 BOUT 0 T -1 \BOUT (4K DORADO))
(34 POPDISP 0 T 0 \POPDISP.UFN (4K DORADO))
(35 RESTLIST 1 T -1 \RESTLIST.UFN)
(36 MISCN 2 T 1 \MISCN.UFN (DORADO DLION DBREAK))
(37 unused)
(38 RPLCONS 0 T -1 \RPLCONS (4K DORADO))
(39 LISTGET 0 T -1 LISTGET (4K DORADO))
(40 unused)
(41 unused)
(42 unused)
(43 unused)
(44 EVAL 0 T 0 \EVAL)
(45 ENVCALL 0 T (JUMP 0)
  \ENVCALL.UFN)
(46 TYPECHECK 0 T 0 \TYPECHECK.UFN)
(47 STKSCAN 0 T 0 \STKSCAN)
(48 BUSBLT 1 (WORDSOUT BYTESOUT BYTESOUTSWAPPED NYBBLESOUT WORDSIN BYTESIN BYTESINSWAPPED
  NYBBLESINSWAPPED)
  -3 \BUSBLT.UFN (4K DORADO))
(49 MISC8 1 (IBLT1 IBLT2)
  -7 \MISC8.UFN (4K DORADO))
(50 UBFLOAT3 1 (POLY MATRIX.3X3 MATRIX.4X4 MATRIX.133 MATRIX.331 MATRIX.144 MATRIX.441 UBASET1)
  (-2 1)
  \UNBOXFLOAT3
  (4K DORADO))

```

```

(51 TYPEMASK.N 1 T 0 \TYPEMASK.UFN)
(52 RDPROLOGPTR 0 T 0 RAID (4K DORADO))
(53 RDPROLOGTAG 0 T 0 RAID (4K DORADO))
(54 WRTPTR&TAG 0 T -2 RAID (4K DORADO))
(55 WRTPTR&0TAG 0 T -1 RAID (4K DORADO))
(56 MISC7 1 (PSEUDOCOLOR \FASTBITMAPBIT)
-6 \MISC7.UFN (4K DORADO))
(57 DOVEMISC 1 (READIW WRITEIO WRTIMER BYTESWAP LOCKMEM NOTIFYIOP SETWP)
(0 -1 0 0 0 -3 0 0))
(58 EQL 0 T -1 EQL)
(59 DRAWLINE 0 T -8 \DRAWLINE.UFN (4K DORADO))
(60 STORE.N 1 T 0 \STORE.N.UFN)
(61 COPY.N 1 T 1 \COPY.N.UFN)
(62 RAID 0 T 0 RAID T)
(63 \RETURN 0 T 0 \RETURN)
((64 70)
IVAR 0 IVAR 1)
(71 IVARX 1 IVAR 1)
((72 78)
PVAR 0 PVAR 1)
(79 PVARX 1 PVAR 1)
((80 86)
FVAR 0 FVAR 1)
(87 FVARX 1 FVAR 1)
((88 94)
PVAR_ 0 PVAR 0)
(95 PVARX_ 1 PVAR 0)
(96 GVAR 4 ATOM 1)
(97 ARG0 0 T 0 \ARGO T)
(98 IVARX_ 1 IVAR 0)
(99 FVARX_ 1 FVAR 0)
(100 COPY 0 T 1)
(101 MYARGCOUNT 0 T 1 \MYARGCOUNT T)
(102 MYALINK 0 T 1)
(103 ACONST 4 ATOM 1)
(104 %'NIL 0 T 1)
(105 %'T 0 T 1)
(106 %'0 0 T 1)
(107 %'1 0 T 1)
(108 SIC 1 SIC 1)
(109 SNIC 1 SNIC 1)
(110 SICX 2 SICX 1)
(111 GCONST 4 GCONST 1)
(112 unused)
(113 READFLAGS 0 T 0 \READFLAGS)
(114 READRP 0 T 0 \READRP)
(115 WRITEMAP 0 T -2 \WRITEMAP DORADO)
(116 READPRINTERPORT 0 T 1 \READPRINTERPORT.UFN 4K)
(117 WRITEPRINTERPORT 0 T 0 \WRITEPRINTERPORT.UFN 4K)
(118 PILOTBITBLT 0 T -1 \PILOTBITBLT)
(119 RCLK 0 T 0 \RCLKSUBR)
(120 MISC1 1 (error INPUT OUTPUT error error error error error error RWMUFMAN)
0 \MISC1.UFN)
(121 MISC2 1 (?0 ?1 ?2 ?3 ?4 ?5 ?6 ?7 ?10)
-1 \MISC2.UFN)
(122 RECLAIMCELL 0 T 0 \GCRECLAIMCELL DORADO)
(123 GCSCAN1 0 T 0 \GCSCAN1)
(124 GCSCAN2 0 T 0 \GCSCAN2)
(125 SUBRCALL 2 SUBRCALL)
(126 CONTEXTSWITCH 0 T 0 \CONTEXTSWITCH)
(127 RETCALL 4 FN (JUMP 1)
\RETCALL)
((128 143)
JUMP 0 JUMP JUMP NIL)
((144 159)
FJUMP 0 JUMP CJUMP NIL)
((160 175)
TJUMP 0 JUMP CJUMP NIL)
(176 JUMPX 1 JUMPX JUMP)
(177 JUMPXX 2 JUMPXX JUMP)
(178 FJUMPX 1 JUMPX CJUMP)
(179 TJUMPX 1 JUMPX CJUMP)
(180 NFJUMPX 1 JUMPX NCJUMP)
(181 NTJUMPX 1 JUMPX NCJUMP)
(182 AREF1 0 T -1 %%AREF1 (4K DORADO))
(183 ASET1 0 T -2 %%ASET1 (4K DORADO))
((184 190)
PVAR_^ 0 PVAR -1 NIL)
(191 POP 0 T -1)
(192 POP.N 1 T (POP.N 1)
\POP.N.UFN)
(193 ATOMCELL.N 1 T 0 \ATOMCELL)
(194 GETBASEBYTE 0 T -1 \GETBASEBYTE)
(195 INSTANCEP 4 ATOM 0 \INSTANCEP.UFN NIL)
(196 BLT 0 T -2 \BLT)
(197 MISC10 1 T -9 \MISC10.UFN (4K DORADO))
(198 P-MISC2 1 (GET-NEXT-RUN)

```

```

-1 \P-MISC2.UFN)
(199 PUTBASEBYTE 0 T -2 \PUTBASEBYTE)
(200 GETBASE.N 1 T 0)
(201 GETBASEPTR.N 1 T 0)
(202 GETBITS.N.FD 2 T 0)
(203 unused)
(204 CMLEQUAL 0 T -1 CL:EQUAL (4K 12K DORADO))
(205 PUTBASE.N 1 T -1 \PUTBASE.UFN)
(206 PUTBASEPTR.N 1 T -1 \PUTBASEPTR.UFN)
(207 PUTBITS.N.FD 2 T -1 \PUTBITS.UFN)
(208 ADDBASE 0 T -1 \ADDBASE)
(209 VAG2 0 T -1 \VAG2)
(210 HILOC 0 T 0)
(211 LOLOC 0 T 0)
(212 PLUS2 0 T -1 \SLOWPLUS2 *)
(213 DIFFERENCE 0 T -1 \SLOWDIFFERENCE *)
(214 TIMES2 0 T -1 \SLOWTIMES2 *)
(215 QUOTIENT 0 T -1 \SLOWQUOTIENT *)
(216 IPLUS2 0 T -1 \SLOWIPLUS2)
(217 IDIFFERENCE 0 T -1 \SLOWIDIFFERENCE)
(218 ITIMES2 0 T -1 \SLOWITIMES2)
(219 IQOTIENT 0 T -1 \SLOWIQOTIENT)
(220 IREMAINDER 0 T -1 IREMAINDER)
(221 IPLUS.N 1 T 0 \SLOWIPLUS2 (4K 12K))
(222 IDIFFERENCE.N 1 T 0 \SLOWIDIFFERENCE (4K 12K))
(223 BASE-< 0 T -1 \BASE-<.UFN (4K 12K DORADO))
(224 LLSH1 0 T 0 \SLOWLLSH1)
(225 LLSH8 0 T 0 \SLOWLLSH8)
(226 LRSH1 0 T 0 \SLOWLRSH1)
(227 LRSH8 0 T 0 \SLOWLRSH8)
(228 LOGOR2 0 T -1 \SLOWLOGOR2)
(229 LOGAND2 0 T -1 \SLOWLOGAND2)
(230 LOGXOR2 0 T -1 \SLOWLOGXOR2)
(231 LSH 0 T -1 LSH T)
(232 FPLUS2 0 T -1 \SLOWFPLUS2 4K)
(233 FDIFFERENCE 0 T -1 \SLOWFDIFFERENCE 4K)
(234 FTIMES2 0 T -1 \SLOWFTIMES2 4K)
(235 FQUOTIENT 0 T -1 \SLOWFQUOTIENT 4K)
(236 UBFLOAT2 1 (UFADD UFSUB UFISUB UFMULT UFDIV UFGREAT UFMAX UFMIN UFREM UBAREF1)
(-1 1)
\UNBOXFLOAT2
(4K DORADO))
(237 UBFLOAT1 1 (BOX UNBOX UFABS UFNEGATE UFIX)
(0 1)
\UNBOXFLOAT1
(4K DORADO))
(238 AREF2 0 T -2 %%AREF2 (4K DORADO))
(239 ASET2 0 T -3 %%ASET2 (4K DORADO))
(240 EQ 0 T -1)
(241 IGREATERP 0 T -1 \SLOWIGREATERP)
(242 FGREATERP 0 T -1 \SLOWFGREATERP)
(243 GREATERP 0 T -1 GREATERP)
(244 EQUAL 0 T -1 EQUAL)
(245 MAKENUMBER 0 T -1 \MAKENUMBER 4K)
(246 BOXIPLUS 0 T -1 \BOXIPLUS 4K)
(247 BOXIDIFFERENCE 0 T -1 \BOXIDIFFERENCE 4K)
(248 FLOATBLT 1 (FLOATWRAP FLOATUNWRAP FLOAT FIX FPLUS FDIFFERENCE FDIFFERENCE FPLUSABS ABSDIFFERENCE
ABSFPLUS FTIMES)
-3 \FLOATBLT (4K DORADO))
(249 FFTSTEP 0 T -1 \FFTSTEP (4K DORADO))
(250 MISC3 1 (EXPONENT MAGNITUDE FLOAT COMP BLKFMAX BLKFMIN BLKFABSMAX BLKFABSMIN FLOATTOBYTE ARRAYREAD
LINES-EQUAL-P)
-2 \MISC3.UFN (4K DORADO))
(251 MISC4 1 (ARRAY.TIMES ARRAY.PERM ARRAY.PLUS ARRAY.DIFFERENCE ARRAY.MAGIC 3MATCH BMBIT ARRAYWRITE)
-3 \MISC4.UFN)
(252 UPCTRACE 0 T 0 NIL (4K 12K))
(253 SWAP 0 T 0)
(254 NOP 0 T 0)
(255 = 0 T -1 CL::%= (4K DORADO)))

```

```
(RPAQ I.CODERDTBL (COPYREADTABLE 'ORIG))
```

```
(SETSYNTAX (CHARCODE ^Y)
'[MACRO (LAMBDA (FILE RDTBL)
(EVALFORMAKEINIT (READ FILE RDTBL)
I.CODERDTBL))
```

```
(SETSYNTAX (CHARCODE %|)
'(MACRO ALWAYS READVBAR)
I.CODERDTBL)
```

```
(READTABLEPROP I.CODERDTBL 'USESILPACKAGE NIL)
```

```
(DEFINEQ
```

```
(I.INITUFNTABLE
```

```
[LAMBDA NIL
```

```
(*)
```

```

(I.CREATEPAGES (I.VAG2 20 3072)
  2 NIL T)
(for I from 0 to 255 do (I.SETUFNENTRY I '\UNKNOWN.UFN 0 0))
(for X in \OPCODES when (CADDR (CDDR X)) do (I.SETUFNENTRY (PROG ((OP (CAR X)))
  (RETURN (if (LISTP OP)
    then (CAR OP)
    else OP)))
  (CADDR (CDDR X))
  [COND
    [(LISTP (CADDR (CDDR X)))
      (CADR (CADDR (CDDR X))
        (T (IDIFFERENCE (IPLUS 1 (COND
          ((EQ (CADDR X)
            0)
          (T 1))))
          (CADDR (CDDR X))
        (SELECTQ (CADDR X)
          (0 0)
          (1 1)
          (2 2)
          (3 (*
            3)
          (4 4)
          (5 5)
          (SHOULDNT]))
          (LAMBDA (INDEX FN NARGS NEXTRA)
            (SETQ INDEX (I.ADDBASE (I.ADDBASE (I.VAG2 20 3072)
              INDEX)
              INDEX))
            (I.PUTBASE INDEX 0 (I.ATOMNUMBER FN))
            (LRSH (I.PUTBASE INDEX 1 (LOGOR (LOGAND (I.GETBASE INDEX 1)
              255)
              (LLSH NEXTRA 8))))
            8)
            (LOGAND (I.PUTBASE INDEX 1 (LOGOR (LOGAND (I.GETBASE INDEX 1)
              65280)
              (LOGAND NARGS 255))))
            255]))
          )
(RPAQQ INITPTRS
  ((\MAINDISK)
  (\SWAPDSK1)
  (\SWAPDSK2)
  (\SWAPREQUESTBLOCK)
  (\DISKREQUESTBLOCK)
  (\FREEPAGEFID)
  (\FINALIZATION.FUNCTIONS)
  (\OneCharAtomBase NIL)
  (\SCRATCHSTRING)
  (\LISTPDTD)
  (\FREEBLOCKBUCKETS)
  (\ArrayFrLst)
  (\ArrayFrLst2)
  (\UNBOXEDHUNK.TYPENUM.TABLE)
  (\CODEHUNK.TYPENUM.TABLE)
  (\PTRHUNK.TYPENUM.TABLE)))
(RPAQQ INITVALUES ((\NxtMDSPage \FirstMDSPage)
  (\LeastMDSPage \FirstMDSPage)
  (\SecondMDSPage \DefaultSecondMDSPage)
  (\SecondArrayPage \DefaultSecondArrayPage)
  (\MDSFREELISTPAGE)
  (\MaxSysTypeNum 0)
  (\MaxTypeNumber)
  (\AtomFrLst 0)
  (\NxtArrayPage)
  (\HUNKING?)))
(DECLARE%: EVAL@COMPILE DONTCOPY
(FILELOAD (LOADCOMP)
  MAKEINIT)
)

```

FUNCTION INDEX

I.%%COPY-ONED-ARRAY	31	I.POSTINITARRAYS	38	I.\LOCKFN	3
I.%%COPY-STRING-TO-ARRAY	32	I.PREINITARRAYS	38	I.\LOCKVAR	3
I.ADDPME	7	I.PUTDEFN	15	I.\LOCKWORDS	3
I.ALLOCSTRING	29	I.SETPROPLIST	15	I.\MAIKO.ALLOCBLOCK	34
I.COPYATOM	20	I.SETUPPAGEMAP	7	I.\MAIKO.CONSUMFN	26
I.CREATEMDSTYPETABLE	13	I.SETUPSTACK	8	I.\MAKEFREEARRAYBLOCK	36
I.CREATEPAGES	28	I.\#BLOCKDATACELLS	33	I.\MAKEFREEBLOCK	10
I.DCODERD	40	I.\ALOCBLOCK	33	I.\MAKEMDSENTRY	11
I.DUMPINITPAGES	3	I.\ALOCBLOCK.NEW	35	I.\MERGEBACKWARD	36
I.FILEARRAYBASE	38	I.\ALLOCHUNK	37	I.\MKATOM	16
I.FILEBLOCKTRAILER	38	I.\ALLOCMDSPAGE	11	I.\MOVEBYTES	19
I.FILECODEBLOCK	39	I.\ASSIGNDATATYPE1	11	I.\NEW2PAGE	13
I.FILEPATCHBLOCK	39	I.\BLT	16	I.\NEW4PAGE	28
I.FSETVAL	15	I.\COMPUTE.HUNK.TYPEDECLS	39	I.\NEXTCONSPAGE	28
I.INITATOMS	21	I.\CONSUMFN	23	I.\PREFIXALIGNMENT?	33
I.INITDATATYPENAMES	14	I.\COPY	21	I.\PUTBASEBYTE	28
I.INITDATATYPES	13	I.\CREATE.SYMBOL	17	I.\SETUFNENTRY	45
I.INITGC	10	I.\CREATECELL	13	I.\SETUP.HUNK.TYPENUMBERS	39
I.INITUFNTABLE	44	I.\GETBASEBYTE	28	I.\SETUP.TYPENUM.TABLE	39
I.MAKEINITBFS	2	I.\INITATOMPAGE	18	I.\SETUPGUARDBLOCK	10
I.MAKEINITFIRST	21	I.\INITCONSPAGE	27	I.\SETUPSTACK1	9
I.MAKEINITLAST	22	I.\INITMDSPAGE	11	I.\STKMIN	19
I.MAKEROOMFORPME	8	I.\LINKBLOCK	36	I.\TURN.ON.HUNKING	39
I.MAPPAGES	8	I.\LOCKCELL	3	I.\TYPENUMBERFROMNAME	13
I.NTPX	10	I.\LOCKCODE	3		

VARIABLE INDEX

I.CODERDTBL	44	INITPTRS	45	LOCKEDFNS	2,3,10	\BUILT-IN-SYSTEM-TYPES ..	15
INITCONSTANTS	4	INITVALUES	45	LOCKEDVARS	2,3,10	\OPCODES	42
