

File created: 15-Jun-90 18:38:14 {DSK}<usr>local>Ide>lispcore>internal>library>MESATYPES.;2

changes to: (VARS MESATYPESCOMS)

previous date: 5-Oct-84 12:16:00 {DSK}<usr>local>Ide>lispcore>internal>library>MESATYPES.;1

Read Table: INTERLISP

Package: INTERLISP

Format: MCCS

::
:: Copyright (c) 1984, 1990 by Venue & Xerox Corporation. All rights reserved.

(RPAQQ MESATYPESCOMS

(* Defines three new record types%: MESATYPE, MESARECORD, and MESAARRAY. Also provides a number of macros to manipulate objects of these record types. None of this package need be present in the compiled version of a client package.)

(* Public stuff)

(MACROS MESASIZE MESASETQ MESAEQUAL FMESAELT MESAELT MESASETA)

(PROP ARGNAMES MESASIZE MESASETQ MESAEQUAL FMESAELT MESAELT MESASETA)

(* Private stuff)

(INITVARS (\MESATYPES (HASHARRAY 20)))

(P (pushnew CLISPRECORDTYPES 'MESATYPE)

(pushnew CLISPRECORDTYPES 'MESARECORD)

(pushnew CLISPRECORDTYPES 'MESAARRAY)

(MOVD 'RECORD 'MESATYPE)

(MOVD 'RECORD 'MESARECORD)

(MOVD 'RECORD 'MESAARRAY)

(PROP USERRECORDTYPE MESATYPE MESARECORD MESAARRAY)

(FNS MESATYPEFN MESARECORDFN MesaRecordFields MesaRecordSubblock MesaRecordCreateMethod MESAARRAYFN
MesaArrayOffsets MesaArrayFindOffset RemoveLast))

(* Defines three new record types%: MESATYPE, MESARECORD, and MESAARRAY.

Also provides a number of macros to manipulate objects of these record types.

None of this package need be present in the compiled version of a client package.)

(* Public stuff)

(DECLARE%: EVAL@COMPILE

(PUTPROPS MESASIZE MACRO [args (PROG ((recordName (CAR args)) (* Returns the size of record recordName)
(RETURN (EVAL '(INDEXF (fetch (% , recordName THISISTHELASTFIELD) of T))

(PUTPROPS MESASETQ MACRO [args (PROG ((a (CAR args))
(b (CADR args))
(type (CADDR args)))

(* Copies the contents of "b" into "a" Returns a. Should be used to translate statement of the form "a ← b;" when a and b are neither numbers nor pointers. A safer (ie, more correct) way to do this would be to say "(foreach field f in type st f has both a fetch and a replace method do (replace (type f) of a with (fetch (type f) of b)))"

(RETURN '(\BLT % , a % , b (MESASIZE % , type))

(PUTPROPS MESAEQUAL MACRO [args (PROG ((a (CAR args))
(b (CADR args))
(type (CADDR args)))

(* Compares a and b for equality, where a and b are instances of record type.
a and b can be multiple words long.)

(RETURN '(for word from 0 to (SUB1 (MESASIZE % , type)) always (EQ (\GETBASE % , a
word)
(\GETBASE % , b
word))

(PUTPROPS FMESAELT MACRO [args (PROG ((array (CAR args))
(arrayType (CADR args))
(indexes (CDDR args))
indexRangeList indexOffsetList)

(* Returns a pointer to the indicated element of array. Unsafe, because it returns a pointer to the middle of the structure, which would confuse the garbage collector if you held onto the element pointer longer than the array pointer.)

[SETQ indexRangeList (EVAL '(fetch (% , arrayType INDEXLIST) of T]
[SETQ indexOffsetList (EVAL '(fetch (% , arrayType OFFSETLIST) of T]
(RETURN (LIST '\ADDBASE array (CONS 'IPLUS (for index in indexes as indexRange in
indexRangeList
as offset in indexOffsetList
collect '([OPENLAMBDA (index)
(OR (AND (ILEQ % , (CAR
indexRange
)
index)

```

(ILEQ index %,
  (CDR
   indexRange
  )))
(ERROR,
  indexOutOfRange
)
(TIMES %, offset (
  DIFFERENCE
  index %,
  (CAR
   indexRange
  ]
  %, index])

```

```

(PUTPROPS MESAELT MACRO [args (PROG ((arrayType (CADR args))
                                     elementType)

```

(* Returns the selected element of the array. Copies it into a freshly allocated box to avoid returning a pointer to the middle of the structure, which might confuse the garbage collector.)

```

[SETQ elementType (EVAL '(fetch (%, arrayType ELEMENTTYPE) of T]
(RETURN '(MESASETQ (create %, elementType)
              %,
              (CONS 'FMESAELT args)
              %, elementType)

```

```

(PUTPROPS MESASETA MACRO [args (PROG ((eltArgs (RemoveLast args))
                                       (arrayType (CADR args))
                                       (newValue (CAR (LAST args)))
                                       elementType)

```

(* Replaces the index'th element of array, provided that array is a contiguous run of objects of type elementType)

```

[SETQ elementType (EVAL '(fetch (%, arrayType ELEMENTTYPE) of T]
(RETURN '(MESASETQ %, (CONS 'FMESAELT eltArgs)
              %, newValue %, elementType)

```

)

```

(PUTPROPS MESASIZE ARGUMENTS (recordName))
(PUTPROPS MESASETQ ARGUMENTS (a b type))
(PUTPROPS MESAEQUAL ARGUMENTS (a b type))
(PUTPROPS FMESAELT ARGUMENTS (array arrayType index1 [...] indexn))
(PUTPROPS MESAELT ARGUMENTS (array arrayType index1 [...] indexn))
(PUTPROPS MESASETA ARGUMENTS (array arrayType index1 [...] indexn newValue))

```

(* Private stuff)

```

(RPAQ? \MESATYPES (HASHARRAY 20))
(pushnew CLISPRECORDTYPES 'MESATYPE)
(pushnew CLISPRECORDTYPES 'MESARECORD)
(pushnew CLISPRECORDTYPES 'MESAARRAY)
(MOVD 'RECORD 'MESATYPE)
(MOVD 'RECORD 'MESARECORD)
(MOVD 'RECORD 'MESAARRAY)
(PUTPROPS MESATYPE USERRECORDTYPE MESATYPEFN)
(PUTPROPS MESARECORD USERRECORDTYPE MESARECORDFN)
(PUTPROPS MESAARRAY USERRECORDTYPE MESAARRAYFN)

```

(DEFINEQ

```

MESATYPEFN
[LAMBDA (typeDecl)
  (PROG ((typeName (CADR typeDecl))
        (isType (CADDR typeDecl))
        (rest (CDDDR typeDecl)))
    (RETURN (NCONC (LIST 'MESARECORD typeName (LIST (CONS 'DATA isType)))
                  rest]))

```

(* hts%: "24-Mar-84 19:46")

'NEWVALUE fieldSize)
'NEWVALUE])

(MesaRecordCreateMethod
[LAMBDA (recordName rest) (* edited%: "31-Mar-84 16:31")

(* Returns a create method for the type, if the user has not already done so.)

(if (for thing in rest thereis (EQ 'CREATE (CAR thing)))
then NIL
else (LIST '(CREATE (\ALLOCBLOCK (LRSH (ADD1 (MESASIZE %, recordName))
1]))

(MESAARRAYFN
[LAMBDA (arrayDecl) (* hts%: "18-Apr-84 14:21")

(PROG ((arrayName (CADR arrayDecl))
(indexDeclarations (CADDR arrayDecl))
(elementType (CADDRR arrayDecl))
(rest (CDDDDR arrayDecl))
arrayOffsets)

[SETQ indexDeclarations (for indexDecl in indexDeclarations collect (CONS (EVAL (CAR indexDecl))
(EVAL (CADR indexDecl))

(* Evaluate arraybounds so that they can be expressions rather

than integers.)

(SETQ arrayOffsets (MesaArrayOffsets indexDeclarations elementType))
(RETURN (APPEND '[MESARECORD %, arrayName ((DATA %, (MesaArrayFindOffset indexDeclarations
arrayOffsets elementType)

WORD))

(ACCESSFNS ((INDEXLIST (QUOTE %, indexDeclarations))
(OFFSETLIST (QUOTE %, arrayOffsets))
(ELEMENTTYPE (QUOTE %, elementType)

rest])

(MesaArrayOffsets
[LAMBDA (indexDeclarations elementType) (* hts%: "24-Mar-84 20:15")

(if (NULL indexDeclarations)
then NIL
else (PROG ((restOfOffsets (MesaArrayOffsets (CDR indexDeclarations)
elementType))
(RETURN (CONS (MesaArrayFindOffset (CDR indexDeclarations)
restOfOffsets elementType)
restOfOffsets]))

(MesaArrayFindOffset
[LAMBDA (indexDeclarations arrayOffsets elementType) (* hts%: "18-Apr-84 14:29")

(if indexDeclarations
then (ITIMES (ADD1 (IDIFFERENCE (CDAR indexDeclarations)
(CAAR indexDeclarations)))
(CAR arrayOffsets))
else (EVAL '(MESASIZE %, elementType))

(RemoveLast
[LAMBDA (list) (* hts%: "26-Mar-84 00:04")

(PROG ((newList (COPY list))
length)
(SETQ length (LENGTH newList))
(if (ILEQ length 1)
then (RETURN NIL)
else (RPLACD (FNTH newList (SUB1 length))
(RETURN newList]))

)

FUNCTION INDEX

MesaArrayFindOffset4	MesaRecordCreateMethod4	MesaRecordSubblock3
MESAARRAYFN4	MesaRecordFields3	MESATYPEFN2
MesaArrayOffsets4	MESARECORDFN3	RemoveLast4

PROPERTY INDEX

FMESAELT2	MESAELT2	MESARECORD2	MESASETQ2	MESATYPE2
MESAARRAY2	MESAEQUAL2	MESASETA2	MESASIZE2	

MACRO INDEX

FMESAELT ...1	MESAELT2	MESAEQUAL ..1	MESASETA ...2	MESASETQ ...1	MESASIZE ...1
---------------	---------------	---------------	---------------	---------------	---------------

VARIABLE INDEX

MESATYPES .2
