

File created: 16-May-90 13:35:04 {DSK}<usr>local>lde>lispcore>sources>CMLMVS.;2

changes to: (VARS CMLMVSCOMS)

previous date: 16-Dec-86 15:45:21 {DSK}<usr>local>lde>lispcore>sources>CMLMVS.;1

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

::
:: Copyright (c) 1985, 1986, 1990 by Venue & Xerox Corporation. All rights reserved.

(RPAQQ **CMLMVSCOMS**

```
[
; Interpreter and compiler support for multiple values. See
; LLMVS for runtime support
(FNS CL:MULTIPLE-VALUE-CALL RETVALUES)
(PROP DMACRO CL:MULTIPLE-VALUE-CALL)
(FUNCTIONS CL:MULTIPLE-VALUE-BIND CL:MULTIPLE-VALUE-LIST CL:MULTIPLE-VALUE-PROG1 CL:MULTIPLE-VALUE-SETQ)
[VAR (NEW-ADVISETEMPLATE ' (ADV-PROG (!VALUE !OTHER-VALUES)
(CL:MULTIPLE-VALUE-SETQ (!VALUE . !OTHER-VALUES)
(ADV-PROG NIL (ADV-RETURN DEF)))
(ADV-RETURN (CL:VALUES-LIST (CONS !VALUE !OTHER-VALUES)

(PROP FILETYPE CMLMVS)
(DECLARE%: DONTVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVERS (ADDVARS (NLAMA CL:MULTIPLE-VALUE-CALL)
(NLAML)
(LAMA))
```

:: Interpreter and compiler support for multiple values. See LLMVS for runtime support

(DEFINEQ

(**CL:MULTIPLE-VALUE-CALL**

```
[NLAMBDA FORMS
(DECLARE (LOCALVARS . T)) ; Edited 16-Dec-86 15:35 by bvm:
;; for interpreted calls only. Note that CL:APPLY will compile ok here, because this is in return context, so UNBIND doesn't get in the way.
(CL:APPLY (\EVAL (CAR FORMS))
(for X in (CDR FORMS) join (CL:MULTIPLE-VALUE-LIST (\EVAL X]))
```

(**RETVALUES**

```
[LAMBDA (POS VALUES FLG) (* bvm%: "10-Nov-86 18:13")
(LET ((P (\STACKARGPTR POS)))
(COND
((fetch (FX INVALIDP) of (SETQ P (fetch (FX CLINK) of P)))
(LISPERROR "ILLEGAL RETURN" VALUES)))
(\SMASHRETURN NIL P)
(AND FLG (RELSTK POS))
(CL:VALUES-LIST VALUES))
)
```

(PUTPROPS **CL:MULTIPLE-VALUE-CALL DMACRO**

```
(DEFMACRO (FN &BODY BODY)
;; How to compile special form MULTIPLE-VALUE-CALL --- for benefit of macro writers, handle some degenerate cases and let the
;; rest turn into an APPLY. This is not an OPTIMIZER because pavcompiler intercepts it for its own use.
[COND
[[AND (LISTP FN)
(MEMB (CAR FN)
' (FUNCTION CL:FUNCTION))
(MEMB (CADR FN)
' (LIST CL:VALUES]
(if (NULL (CDR BODY))
then ; only one source of values. Either sole arg is the result itself, or
; a list of its values is
(CONS (if (EQ (CADR FN)
'LIST)
then 'MVLIST
else 'PROGN)
BODY)
else ; Produce a list consisting of all args spread. This is either the
; result itself, or to be spread as values
' (, (if (EQ (CADR FN)
'LIST)
then 'PROGN
else 'CL:VALUES-LIST)
(NCONC ,@(for F in BODY collect `(\MVLIST ,F)
(T \ (APPLY ,FN (NCONC ,@(for F in BODY collect `(\MVLIST ,F)))
```

(DEFMACRO **CL:MULTIPLE-VALUE-BIND** (VARS VALUES-FORM &REST FORMS)
\ (DESTRUCTURING-BIND ,VARS (CL:MULTIPLE-VALUE-LIST ,VALUES-FORM)

,@FORMS))

(DEFMACRO **CL:MULTIPLE-VALUE-LIST** (FORM)
 `(CL:MULTIPLE-VALUE-CALL (FUNCTION LIST)
 ,FORM))

(DEFMACRO **CL:MULTIPLE-VALUE-PROG1** (FORM . OTHER-FORMS)
 `(CL:VALUES-LIST (PROG1 (CL:MULTIPLE-VALUE-LIST ,FORM)
 ,@OTHER-FORMS)))

(DEFMACRO **CL:MULTIPLE-VALUE-SETQ** (VARIABLES FORM)
 [LET ((LIST (GENSYM)))
 `(LET [(,LIST (CL:MULTIPLE-VALUE-LIST ,FORM)
 (DESTRUCTURING-SETQ ,VARIABLES ,LIST)
 (CAR ,LIST)])

(RPAQQ **NEW-ADVISETEMPLATE** [ADV-PROG (!VALUE !OTHER-VALUES)
 (CL:MULTIPLE-VALUE-SETQ (!VALUE . !OTHER-VALUES)
 (ADV-PROG NIL (ADV-RETURN DEF)))
 (ADV-RETURN (CL:VALUES-LIST (CONS !VALUE !OTHER-VALUES]))

(PUTPROPS **CMLMVS FILETYPE** CL:COMPILE-FILE)

(DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVERS

(ADDTOVAR **NLAMA** CL:MULTIPLE-VALUE-CALL)

(ADDTOVAR **NLAML**)

(ADDTOVAR **LAMA**)

)

(PUTPROPS **CMLMVS COPYRIGHT** ("Venue & Xerox Corporation" 1985 1986 1990))

FUNCTION INDEX

CL:MULTIPLE-VALUE-CALL1 RETVALUES1

MACRO INDEX

CL:MULTIPLE-VALUE-BIND1 CL:MULTIPLE-VALUE-LIST2 CL:MULTIPLE-VALUE-SETQ2
CL:MULTIPLE-VALUE-CALL1 CL:MULTIPLE-VALUE-PROG12

PROPERTY INDEX

CMLMVS2

VARIABLE INDEX

NEW-ADVISETEMPLATE2
