```
(RPAQQ CMLCOMPILECOMS
       [(COMS (FUNCTIONS CL:DISASSEMBLE)
              (FNS FAKE-COMPILE-FILE INTERLISP-FORMAT-P INTERLISP-NLAMBDA-FUNCTION-P COMPILE-FILE-EXPRESSION
                   COMPILE-FILE-WALK-FUNCTION ARGTYPE.STATE COMPILE.CHECK.ARGTYPE COMPILE.FILE.DEFINEQ
                   COMPILE-FILE-SETF-SYMBOL-FUNCTION COMPILE-FILE-EX/IMPORT COMPILE.FILE.APPLY
                   COMPILE.FILE.RESET COMPILE-IN-CORE)
              (FNS COMPILE-FILE-SCAN-FIRST)
```
                                                                 ; This function is support for AR#11185
```
              (VARS ARGTYPE.VARS)
              (PROP COMPILE-FILE-EXPRESSION DEFINEQ * SETF-SYMBOL-FUNCTION PRETTYCOMPRINT)
              (FUNCTIONS COMPILE-FILE-DECLARE%:))
        [COMS (FNS NEWDEFC)
              (DECLARE%: DONTEVAL@LOAD DOCOPY (P (MOVD 'NEWDEFC 'DEFC]
        (PROP FILETYPE CMLCOMPILE)
        (DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVARS (ADDVARS (NLAMA)
                                                                             (NLAML)
                                                                             (LAMA FAKE-COMPILE-FILE])


(CL:DEFUN CL:DISASSEMBLE (NAME-OR-COMPILED-FUNCTION &KEY LEVEL-P (RADIX 8)
                                                    (OUTPUT *STANDARD-OUTPUT*)
                                                    FIRST-BYTE MARKED-PC)
    (PRINTCODE (if (CCODEP NAME-OR-COMPILED-FUNCTION)
                   then NAME-OR-COMPILED-FUNCTION
                 else (CL:COMPILE NIL (if (CL:SYMBOLP NAME-OR-COMPILED-FUNCTION)
                                          then (CL:SYMBOL-FUNCTION NAME-OR-COMPILED-FUNCTION)
                                        else NAME-OR-COMPILED-FUNCTION)))
               LEVEL-P RADIX OUTPUT FIRST-BYTE MARKED-PC))

(DEFINEQ

(FAKE-COMPILE-FILE
  (CL:LAMBDA
   (FILENAME &KEY LAP REDEFINE OUTPUT-FILE (SAVE-EXPRS T)
        (COMPILER-OUTPUT T)
        (PROCESS-ENTIRE-FILE NIL PEFP))            ; Edited 29-Jun-90 19:19 by nm
   (LET
    (COMPILE.FILE.AFTER VALUE COMPILE.FILE.VALUE (NLAML NLAML)
          (NLAMA NLAMA)
          (LAMS LAMS)
          (LAMA LAMA)
          (DFNFLG NIL))
    (DECLARE (CL:SPECIAL COMPILE.FILE.AFTER COMPILE.FILE.VALUE NLAML NLAMA LAMS LAMA DFNFLG))
    (RESETLST
        (RESETSAVE NIL (LIST 'RESETUNDO)
              (RESETUNDO))
        (RESETSAVE COUTFILE COMPILER-OUTPUT)
        (RESETSAVE STRF REDEFINE)
        (RESETSAVE SVFLG (AND SAVE-EXPRS REDEFINE 'DEFER))
        (RESETSAVE LAPFLG LAP)
        (LET ((*PACKAGE* *INTERLISP-PACKAGE*)
              (*READ-BASE* 10)
              (LOCALVARS SYSLOCALVARS)
              (SPECVARS T)
              STREAM LSTFIL ROOTNAME INTERLISP-FORMAT ENV FORM)
             (DECLARE (CL:SPECIAL *PACKAGE* *READ-BASE* LOCALVARS SPECVARS LSTFIL))
             [RESETSAVE NIL (LIST (FUNCTION CLOSEF?)
                                  (SETQ STREAM (OPENSTREAM FILENAME 'INPUT]
             (CL:MULTIPLE-VALUE-SETQ (ENV FORM)
                   (\PARSE-FILE-HEADER STREAM 'RETURN T))
             (SETQ INTERLISP-FORMAT (AND ENV (NEQ ENV *COMMON-LISP-READ-ENVIRONMENT*)))
             (if (NOT PEFP)
                 then (SETQ PROCESS-ENTIRE-FILE INTERLISP-FORMAT))
             (if LAP
                 then (SETQ LSTFIL COUTFILE))
             (SETQ FILENAME (FULLNAME STREAM))
             (RESETSAVE NIL (LIST (FUNCTION COMPILE.FILE.RESET)
                                  [SETQ OUTPUT-FILE (OPENSTREAM (OR OUTPUT-FILE (PACKFILENAME.STRING
                                                                                 'VERSION NIL 'EXTENSION
```

```
                                                                                    COMPILE.EXT 'BODY FILENAME))
                                                                    'OUTPUT
                                                                    'NEW
                                                                    '((TYPE BINARY]
                                          STREAM
                                          (ROOTFILENAME FILENAME)))
                    (if OUTPUT-FILE
                        then (RESETSAVE LCFIL OUTPUT-FILE)
                             (PRINT-COMPILE-HEADER (LIST STREAM)
                                     '("COMPILE-FILEd")
                                     ENV))
                    (WITH-READER-ENVIRONMENT ENV
                        (PROG ((DEFERRED.EXPRESSIONS NIL)
                               (*PRINT-ARRAY* T)
                               (*PRINT-LEVEL* NIL)
                               (*PRINT-LENGTH* NIL)
                               (FIRSTFORMS NIL)
                               (AFTERS NIL)
                               (SCRATCH.LCOM '{CORE}SCRATCH.LCOM)
                                DUMMYFILE TEMPVAL)
                               (DECLARE (CL:SPECIAL DEFERRED.EXPRESSIONS *PRINT-ARRAY* *PRINT-LEVEL* *PRINT-LENGTH*
                                                    FIRSTFORMS AFTERS DEFERS))
                                                        ; Edited by TT (11-June-90 : for AR#11185) all contents of file
                                                        ; are read, and each forms are compiled.(This reading method is
                                                        ; for supporting "FIRST", "NOTFIRST" tag.)
                               [RESETSAVE NIL (LIST (FUNCTION CLOSEF?)
                                                    (SETQ DUMMYFILE (OPENSTREAM SCRATCH.LCOM 'BOTH 'NEW]
                          LPDUMP
                               [if (EQUAL (CAR FORM)
                                          'RPAQQ)
                                   then                                  ; This is the support method of "COMPILERVARS" (2-July-1990
                                                                         ; TT)
                                        (SETQ TEMPVAL (CADDR FORM))
                                        (if (SETQ TEMPVAL (ASSOC 'DECLARE%: TEMPVAL))
                                            then (if (SETQ TEMPVAL (FMEMB 'COMPILERVARS (FMEMB 'DOEVAL@COMPILE TEMPVAL)))
                                                     then (SETQ DFNFLG T)
                                                          (if [SETQ TEMPVAL (FMEMB 'ADDVARS (SETQ TEMPVAL (CADR TEMPVAL]
                                                              then (CL:DOLIST (ARG (CDR TEMPVAL))
                                                                        (APPLY 'ADDTOVAR ARG)]
                               (COMPILE-FILE-EXPRESSION FORM DUMMYFILE NIL PROCESS-ENTIRE-FILE)
                               (SKIPSEPRCODES STREAM)
                               (if (EOFP STREAM)
                                   then (CLOSEF STREAM)
                                        (for FORM in FIRSTFORMS do (COMPILE-FILE-EXPRESSION FORM OUTPUT-FILE NIL
                                                                        PROCESS-ENTIRE-FILE T))
                                        (COPYBYTES DUMMYFILE OUTPUT-FILE 0 (GETFILEPTR DUMMYFILE))
                                        (CLOSEF? DUMMYFILE)
                                        (DELFILE (FULLNAME DUMMYFILE))
                                        (AND PROCESS-ENTIRE-FILE (for EXP in (REVERSE DEFERRED.EXPRESSIONS)
                                                                      do (APPLY* (CAR EXP)
                                                                                 (CDR EXP)
                                                                                 OUTPUT-FILE)))
                                        (for FORM in AFTERS do (COMPILE-FILE-EXPRESSION FORM OUTPUT-FILE NIL
                                                                    PROCESS-ENTIRE-FILE T))
                                        (RETURN))
                               (SETQ FORM (READ STREAM))
                               (GO LPDUMP))
                           (PRINT NIL OUTPUT-FILE))
                       (SETQ COMPILE.FILE.VALUE (CLOSEF OUTPUT-FILE))))) ; Do these after UNDONLSETQ entered
             (MAPC (REVERSE COMPILE.FILE.AFTER)
                   (FUNCTION EVAL))
             COMPILE.FILE.VALUE)))
```

(**INTERLISP-FORMAT-P**
```
  [LAMBDA (STREAM)                                              (* bvm%: " 3-Aug-86 14:01")
    (SELCHARQ (PEEKCCODE STREAM)
        (; NIL)
        ((^F "(")
             T)
        NIL])
```

(**INTERLISP-NLAMBDA-FUNCTION-P**
```
  [LAMBDA (X)                                                   (* lmm " 7-May-86 20:12")
    (AND (LITATOM X)
         (FMEMB (ARGTYPE X)
                '(1 3))
         (NOT (CL:SPECIAL-FORM-P X)))
```

(**COMPILE-FILE-EXPRESSION**
```
  [LAMBDA (FORM COMPILED.FILE COMPILE.TIME.TOO DEFER FORCE-OUTPUT-P)
                                                              ; Edited 30-Jun-90 18:31 by nm
    (DECLARE (CL:SPECIAL COMPILED.FILE))
    (AND (LISTP FORM)
         (SELECTQ (CAR FORM)
```

```
            ((DECLARE%: FILECREATED)
                (COMPILE-FILE-SCAN-FIRST FORM COMPILED.FILE NIL T COMPILE.TIME.TOO DEFER FORCE-OUTPUT-P))
            ((DEFMACRO)
                (LET* ((DEFINITION (REMOVE-COMMENTS FORM))
                       (NAME (XCL::%%DEFINER-NAME 'DEFMACRO DEFINITION))
                       (BODY (XCL::%%EXPAND-DEFINER 'DEFMACRO DEFINITION)))
                    (CL:EVAL BODY)
                    (COMPILE-FILE-EXPRESSION BODY COMPILED.FILE COMPILE.TIME.TOO DEFER FORCE-OUTPUT-P)))
            ((PROGN)
                (for X in (CDR FORM) do (COMPILE-FILE-EXPRESSION X COMPILED.FILE COMPILE.TIME.TOO DEFER
                                            FORCE-OUTPUT-P)))
            ((QUOTE)                                         ; ignore top level quoted expression -i
                NIL)
            ((CL:COMPILER-LET)                               ;  top level compiler-let. bind variables and recursively compile
                                                             ; sub-expressions. This is here mainly  for b PCL has top level
                                                             ; compiler-lets
                [LET [(VARS (CL:MAPCAR #'(CL:LAMBDA (X)
                                            (if (CL:CONSP X)
                                                then (CAR X)
                                              else X))
                                (CADR FORM)))
                      (VALS (CL:MAPCAR #'[CL:LAMBDA (X)
                                            (if (CL:CONSP X)
                                                then (CL:EVAL (CADR X]
                                (CADR FORM]
                    (CL:PROGV VARS VALS
                        (CL:MAPC #'(CL:LAMBDA (X)
                                      (COMPILE-FILE-EXPRESSION X COMPILED.FILE COMPILE.TIME.TOO DEFER
                                          FORCE-OUTPUT-P))
                            (CDDR FORM)))])
            ((CL:EVAL-WHEN)
                [LET [[EVAL.SPECIFIED (OR (FMEMB 'EVAL (CADR FORM))
                                          (FMEMB 'CL:EVAL (CADR FORM]
                      [LOAD.SPECIFIED (OR (FMEMB 'LOAD (CADR FORM))
                                          (FMEMB 'CL:LOAD (CADR FORM]
                      (COMPILE.SPECIFIED (OR (FMEMB 'COMPILE (CADR FORM))
                                             (FMEMB 'CL:COMPILE (CADR FORM]
                    (COND
                       [(NOT LOAD.SPECIFIED)
                        (COND
                           ((OR COMPILE.SPECIFIED (AND COMPILE.TIME.TOO EVAL.SPECIFIED))
                            (for INNER-FORM in (CDDR FORM) do (EVAL INNER-FORM]
                       (T (for INNER-FORM in (CDDR FORM) do (COMPILE-FILE-EXPRESSION INNER-FORM COMPILED.FILE
                                                               (OR COMPILE.SPECIFIED (AND COMPILE.TIME.TOO
                                                                                          EVAL.SPECIFIED))
                                                               DEFER FORCE-OUTPUT-P])
            ((CL:IN-PACKAGE)                                 ; This is special because it has to be dumped to the output
                                                             ; BEFORE the package changes
                (PRINT FORM COMPILED.FILE)
                (EVAL FORM))
            ((CL:MAKE-PACKAGE CL:SHADOW CL:SHADOWING-IMPORT EXPORT CL:UNEXPORT CL:USE-PACKAGE CL:UNUSE-PACKAGE
                    IMPORT)                                  ; This is Special also, becouse  the compiling Environment Must
                                                             ; be changed.(see CLtL, 11.7. Package System Functions and
                                                             ; Variables)  edited by TT(10-April-90)
                (PRINT FORM COMPILED.FILE)
                (EVAL FORM))
            ((CL:SETQ)                                       ; Gasly kludge because cl:setq needs to run in the init before
                                                             ; macroexpansion is enabled
                (COMPILE-FILE-EXPRESSION (EXPANDMACRO FORM T)
                    COMPILED.FILE COMPILE.TIME.TOO DEFER FORCE-OUTPUT-P))
            (LET [(PROP (OR (GETPROP (CAR FORM)
                                'COMPILE-FILE-EXPRESSION)
                            (GETPROP (CAR FORM)
                                'COMPILE.FILE.EXPRESSION]
                (if [AND (NOT PROP)
                         (NOT (CL:SPECIAL-FORM-P (CAR FORM)))
                         (NOT (INTERLISP-NLAMBDA-FUNCTION-P (CAR FORM)))
                         (NEQ FORM (SETQ FORM (CL:MACROEXPAND-1 FORM]
                    then (COMPILE-FILE-EXPRESSION FORM COMPILED.FILE COMPILE.TIME.TOO DEFER FORCE-OUTPUT-P)
                  else (if COMPILE.TIME.TOO
                           then (EVAL FORM))
                       (if PROP
                           then (COMPILE.FILE.APPLY PROP FORM DEFER FORCE-OUTPUT-P)
                         elseif [NOT (EQUAL FORM (SETQ FORM (WALK-FORM FORM :WALK-FUNCTION
                                                               (FUNCTION COMPILE-FILE-WALK-FUNCTION]
                           then (COMPILE-FILE-EXPRESSION FORM COMPILED.FILE COMPILE.TIME.TOO DEFER
                                    FORCE-OUTPUT-P)
                         else (COMPILE.FILE.APPLY (FUNCTION PRINT)
                                    FORM DEFER FORCE-OUTPUT-P])
```

(**COMPILE-FILE-WALK-FUNCTION**
```
  [LAMBDA (FORM)                                                (* lmm "26-Jun-86 17:25")
    (if (NLISTP FORM)
        then FORM
      else (CL:VALUES FORM (INTERLISP-NLAMBDA-FUNCTION-P (CAR FORM])
```

(**ARGTYPE.STATE**
```
  [LAMBDA NIL
    (for X in ARGTYPE.VARS do (PRINTOUT T X %, (EVAL (CADR X))
                                              T])
```

(**COMPILE.CHECK.ARGTYPE**   (* lmm "15-Jun-85 16:58")
```
  [LAMBDA (X AT)
    (if (NEQ AT (LET (BLKFLG)
                     (COMP.ARGTYPE X)))
        then                                                          ; Incorrectly on one of the defining lists
            (for ATYPEPAIR in ARGTYPE.VARS
               do (LET [(VAL (FMEMB X (EVALV (CADR ATYPEPAIR]
                        (if (EQ AT (CAR ATYPEPAIR))
                            then (if VAL
                                     then (PRINTOUT COUTFILE "Compiler confused: " X " on " (CADR ATYPEPAIR)
                                                    " but compiler doesn't think its a "
                                                    (CADDR ATYPEPAIR)))
                                 [/SETTOPVAL (CADR ATYPEPAIR)
                                      (CONS X (PROGN (GETTOPVAL (CADR ATYPEPAIR]
                            else (if VAL
                                     then (PRINTOUT COUTFILE "Warning: compiler thought " X " "
                                                    (LIST 'a (OR (CADDR (ASSOC AT ARGTYPE.VARS))
                                                                 "LAMBDA spread")
                                                          'function)
                                                    " was a "
                                                    (CADDR ATYPEPAIR)
                                                    " because it was incorrectly on "
                                                    (CADR ATYPEPAIR)
                                                    T)
                                 (/SETTOPVAL (CADR ATYPEPAIR)
                                      (REMOVE X (PROGN (GETTOPVAL (CADR ATYPEPAIR])
```

(**COMPILE.FILE.DEFINEQ**   (* bvm%: "18-Sep-86 14:35")
```
  [LAMBDA (FORM LCFIL)
    (for DEF in (CDR FORM) unless (FMEMB (CAR DEF)
                                         DONTCOMPILEFNS)
       do (COMPILE.CHECK.ARGTYPE (CAR DEF)
                (ARGTYPE (CADR DEF)))
          (BYTECOMPILE2 (CAR DEF)
                (COMPILE1A (CAR DEF)
                       (CADR DEF)
                       NIL])
```

(**COMPILE-FILE-SETF-SYMBOL-FUNCTION**   (* bvm%: " 8-Sep-86 16:55")
```
  [LAMBDA (FORM LCFIL)
    (if [AND (FMEMB (CAR (LISTP (CL:THIRD FORM)))
                    '(FUNCTION CL:FUNCTION))
             (EQ (CAR (LISTP (CL:SECOND FORM)))
                 'QUOTE)
             (CL:CONSP (CL:SECOND (CL:THIRD FORM]
        then (BYTECOMPILE2 (CADR (CL:SECOND FORM))
                    (CADR (CL:THIRD FORM)))
        else (PRINT (WALK-FORM FORM :WALK-FUNCTION (FUNCTION COMPILE-FILE-WALK-FUNCTION))
                    LCFIL])
```

(**COMPILE-FILE-EX/IMPORT**   (* bvm%: " 3-Aug-86 15:05")
```
  [LAMBDA (FORM LCFIL RDTBL)

        (* * "EXPORT, IMPORT, SHADOW, USE-PACKAGE are all implicitly EVAL@COMPILE, since they have to affect the
        package being used to read what follows")

    (PRINT FORM LCFIL RDTBL)
    (EVAL FORM])
```

(**COMPILE.FILE.APPLY**   ; Edited 29-Jun-90 19:21 by nm
```
  [LAMBDA (PROP FORM DEFER FORCE-OUTPUT-P)
    (if FORCE-OUTPUT-P
        then (PRINT FORM COMPILED.FILE)
        else (if DEFER
                 then (push DEFERRED.EXPRESSIONS (CONS PROP FORM))
                 else (APPLY* PROP FORM COMPILED.FILE])
```

(**COMPILE.FILE.RESET**   (* bvm%: " 9-Sep-86 15:16")
                                                                                    (* Cleans up after brecompile and bcompl have finished
                                                                                    operating,)
```
  [LAMBDA (COMPILED.FILE SOURCEFILE ROOTNAME)

    (if (AND COMPILED.FILE (OPENP COMPILED.FILE))
        then (CLOSE-AND-MAYBE-DELETE COMPILED.FILE))
    (if SOURCEFILE
        then (CLOSEF? SOURCEFILE])
```

```
      (if (NULL RESETSTATE)
          then                                                              (* Finished successfully.)
              (/SETATOMVAL 'NOTCOMPILEDFILES (REMOVE ROOTNAME NOTCOMPILEDFILES))
                                                              (* Removes FILES from NOTCOMPILEDFILES.)])
```

(**COMPILE-IN-CORE**
```
  [LAMBDA (fn-name fn-expr fn-type NOSAVE)
    (DECLARE (SPECVARS LCFIL LAPFLG STRF SVFLG LSTFIL SPECVARS LOCALVARS DONT-TRANSFER-PUTD))
                                                            ; Edited 24-Sep-2023 14:11 by rmk
                                                            (* lmm " 2-Jun-86 22:04")
```

;; in-core compiling for functions and forms, without the interview. if X is a list, we assume that we are being called merely to display the lap and
;; machine code. the form is compiled as the definition of FOO but the compiled CODE is thrown away. --- if X is a litatom, then saving, redefining,
;; and printing is controlled by the flags.

```
          (* in-core compiling for functions and forms, without the interview.
          if X is a list, we assume that we are being called merely to display the lap and machine code.
          the form is compiled as the definition of FOO but the compiled :CODE is thrown away.
          -
          if X is a litatom, then saving, redefining, and printing is controlled by the flags.)

    (LET ((NOREDEFINE NIL)
          (PRINTLAP NIL)
          (DONT-TRANSFER-PUTD T))

        ;; RMK:  Is it really worth saving NULLFILE from one invocation to the next?

        (RESETVARS [(NLAMA NLAMA)
                    (NLAML NLAML)
                    (LAMS LAMS)
                    (LAMA LAMA)
                    (NOFIXFNSLST NOFIXFNSLST)
                    (NOFIXVARSLST NOFIXVARSLST)
                    (COUTFILE (COND
                                ((AND (BOUNDP 'NULLFILE)
                                      (STREAMP NULLFILE)
                                      (OPENP NULLFILE))
                                 NULLFILE)
                                (T (SETQ NULLFILE (OPENSTREAM '{NULL} 'OUTPUT]
                   (RETURN (RESETLST                          ; RESETLST to provide reset context for macros under
                                                              ; COMPILE1 as generated e.g.  by DECL.

                              [PROG ((LCFIL)
                                     [LAPFLG (AND PRINTLAP (COND
                                                            (BYTECOMPFLG T)
                                                            (T 2]
                                     (STRF (NOT NOREDEFINE))
                                     (SVFLG (if (EQ fn-type 'SELECTOR)
                                                then 'SELECTOR
                                                else (NOT NOSAVE)))
                                     (LSTFIL T)
                                     (SPECVARS SYSSPECVARS)
                                     (LOCALVARS T))
                                    (RETURN (PROGN (SETQ fn-expr (COMPILE1A fn-name fn-expr T))
                                                   (PROG ((FREEVARS FREEVARS))
                                                         (RETURN (BYTECOMPILE2 fn-name fn-expr]])
)

(DEFINEQ
```

(**COMPILE-FILE-SCAN-FIRST**
```
  [LAMBDA (FORM COMPILED.FILE FIRSTFLG DOCOPY EVAL@COMPILE DEFER FORCE-OUTPUT-P)
                                                            ; Edited 30-Jun-90 18:32 by nm
                                                            ; Edited 26-Apr-90 by tt
                                                            ; This is enhancement for Fake Compiler's interpretation of file
                                                            ; package coms
    (PROG ((DFNFLG DFNFLG)
           (FIRST FIRSTFLG)
           (DOCOPY DOCOPY)
           (EVAL@COMPILE EVAL@COMPILE)
           NOTFIRST)
        (if (LISTP FORM)
            then (SELECTQ (CAR FORM)
                    ((DECLARE%:)
                       (CL:DO ((TAIL (CDR FORM)
                                     (CDR TAIL)))
                              ((CL:ENDP TAIL))
                           [if (CL:SYMBOLP (CAR TAIL))
                               then (CASE (CAR TAIL)
                                       ((DOCOPY COPY) (SETQ DOCOPY T))
                                       ((DONTCOPY) (SETQ DOCOPY NIL))
                                       ((COPYWHEN) [SETQ DOCOPY (EVAL (CAR (SETQ TAIL (CDR TAIL])
                                       ((EVAL@LOAD DOEVAL@LOAD DONTEVAL@LOAD) NIL)
                                       ((EVAL@LOADWHEN) (CL:POP TAIL))
                                       ((EVAL@COMPILE DOEVAL@COMPILE) (SETQ EVAL@COMPILE T))
                                       ((DONTEVAL@COMPILE) (SETQ EVAL@COMPILE NIL))
                                       ((EVAL@COMPILEWHEN) [SETQ EVAL@COMPILE (EVAL (CAR (SETQ TAIL
                                                                                             (CDR TAIL])
```

```
                                                  ((FIRST)
                                                      (SETQ FIRST T)
                                                      (SETQ NOTFIRST NIL))
                                                                        ; for First
                                                  ((NOTFIRST)
                                                      (SETQ NOTFIRST T)
                                                      (SETQ FIRST NIL)) ; for Not First
                                                  ((COMPILERVARS) (SETQ DFNFLG T))
                                                                        ; for Compilervars
                                                  (CL:OTHERWISE (CL:FORMAT COUTFILE "Warning: Ignoring unrecognized
                                                                      DECLARE: tag: ~S~%%" (CAR TAIL))))
                            else (COND
                                     ((EQ 'DECLARE%: (CAR (CAR TAIL)))
                                      (COMPILE-FILE-SCAN-FIRST (CAR TAIL)
                                           COMPILED.FILE FIRST DOCOPY EVAL@COMPILE DEFER))
                                     (T (CL:WHEN EVAL@COMPILE
                                           (EVAL (CAR TAIL)))
                                        (CL:WHEN DOCOPY
                                           (CL:IF FIRST
                                               (SETQ FIRSTFORMS (NCONC1 FIRSTFORMS (CAR TAIL)))
                                               (CL:IF NOTFIRST
                                                   (SETQ AFTERS (NCONC1 AFTERS (CAR TAIL)))
                                                   (COMPILE-FILE-EXPRESSION (CAR TAIL)
                                                       COMPILED.FILE EVAL@COMPILE DEFER FORCE-OUTPUT-P))
)))])))
                            ((FILECREATED)
                                (if FORCE-OUTPUT-P
                                    then (PRINT FORM COMPILED.FILE)
                                  else (SETQ FIRSTFORMS (NCONC1 FIRSTFORMS FORM))))
                        NIL])

)
```

;; This function is support for AR#11185

```
(RPAQQ ARGTYPE.VARS ((1 NLAML "NLAMBDA spread")
                     (2 LAMA "LAMBDA nospread")
                     (0 LAMS "LAMBDA spread")
                     (3 NLAMA "NLAMBDA no-spread")))

(PUTPROPS DEFINEQ COMPILE-FILE-EXPRESSION COMPILE.FILE.DEFINEQ)

(PUTPROPS * COMPILE-FILE-EXPRESSION NILL)

(PUTPROPS SETF-SYMBOL-FUNCTION COMPILE-FILE-EXPRESSION COMPILE-FILE-SETF-SYMBOL-FUNCTION)

(PUTPROPS PRETTYCOMPRINT COMPILE-FILE-EXPRESSION NILL)


(CL:DEFUN COMPILE-FILE-DECLARE%: (FORM COMPILED.FILE EVAL@COMPILE DOCOPY DEFER)
    (CL:DO ((TAIL (CDR FORM)
                  (CDR TAIL)))
           ((CL:ENDP TAIL))
        (CL:IF (CL:SYMBOLP (CAR TAIL))
            (CASE (CAR TAIL)
                ((EVAL@LOAD DOEVAL@LOAD DONTEVAL@LOAD) NIL)
                ((EVAL@LOADWHEN) (CL:POP TAIL))
                ((EVAL@COMPILE DOEVAL@COMPILE) (SETQ EVAL@COMPILE T))
                ((DONTEVAL@COMPILE) (SETQ EVAL@COMPILE NIL))
                ((EVAL@COMPILEWHEN) [SETQ EVAL@COMPILE (EVAL (CAR (SETQ TAIL (CDR TAIL]))
                ((COPY DOCOPY) (SETQ DOCOPY T))
                ((DONTCOPY) (SETQ DOCOPY NIL))
                ((COPYWHEN) [SETQ DOCOPY (EVAL (CAR (SETQ TAIL (CDR TAIL]))
                ((FIRST) )
                ((NOTFIRST COMPILERVARS) )
                (CL:OTHERWISE (CL:FORMAT COUTFILE "Warning: Ignoring unrecognized DECLARE: tag: ~S~%%"
                                 (CAR TAIL))))
            [COND
                ((EQ 'DECLARE%: (CAR (CAR TAIL)))
                 (COMPILE-FILE-DECLARE%: (CAR TAIL)
                      COMPILED.FILE EVAL@COMPILE DOCOPY DEFER))
                (T (CL:WHEN EVAL@COMPILE
                      (EVAL (CAR TAIL)))
                   (CL:WHEN DOCOPY
                      (COMPILE-FILE-EXPRESSION (CAR TAIL)
                           COMPILED.FILE EVAL@COMPILE DEFER))])))

(DEFINEQ

(NEWDEFC
  [LAMBDA (NM DF)                                          (* bvm%: "30-Sep-86 23:12")
    [COND
        ((EQ SVFLG 'DEFER)
         (push COMPILE.FILE.AFTER (LIST (FUNCTION NEWDEFC)
                                        (KWOTE NM)
                                        (KWOTE DF)
                                        T)))
```

```
        ((OR (NULL DFNFLG)
             (EQ DFNFLG T))
         [COND
            ((GETD NM)
             (VIRGINFN NM T)
             (COND
                ((NULL DFNFLG)
                 (CL:FORMAT *ERROR-OUTPUT* "~&(~S redefined)~%%" NM)
                 (SAVEDEF NM]
          (/PUTD NM DF T))
        (T  ;; Save on CODE prop.  Be nice and change it from archaic CCODEP object to modern compiled code object.

            (/PUTPROP NM 'CODE (if (ARRAYP DF)
                                   then (create COMPILED-CLOSURE
                                                FNHEADER _ (fetch (ARRAYP BASE) of DF))
                                   else DF]
     DF])

)

(DECLARE%: DONTEVAL@LOAD DOCOPY

(MOVD 'NEWDEFC 'DEFC)
)

(PUTPROPS CMLCOMPILE FILETYPE CL:COMPILE-FILE)

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

(ADDTOVAR NLAMA )

(ADDTOVAR NLAML )

(ADDTOVAR LAMA FAKE-COMPILE-FILE)
)

(PUTPROPS CMLCOMPILE COPYRIGHT ("Venue & Xerox Corporation" 1985 1986 1987 1990))
```

FUNCTION INDEX

PROPERTY INDEX

VARIABLE INDEX