

File created: 27-Apr-93 16:48:06 {DSK}<new>loops>library-src>LOOPSMS.;4

changes to: (TEMPLATES _! SEND |ApplyMethod|)

previous date: 6-Feb-93 16:07:01 {DSK}<new>loops>library-src>LOOPSMS.;3

Read Table: XCL

Package: INTERLISP

Format: XCCS

; Copyright (c) 1985, 1986, 1987, 1988, 1990, 1992, 1993 by Venue & Xerox Corporation. All rights reserved.

(RPAQQ **LOOPSMSCOMS**

(

;;; Tailor MasterScope to understand Loops constructs

;;; Load MasterScope, since its a library module.

```
(P (COND ((EQ MAKESYSNAME ' :LYRIC)
  ;; Nuke the FILEDATES properties so we're sure the Lyric compiled Medley versions are loaded.
  (REMPROP 'MASTERSCOPE 'FILEDATES)
  (REMPROP 'MSANALYZE 'FILEDATES)
  (REMPROP 'MSPARSE 'FILEDATES)))
 (FILESLOAD (FROM VALUEOF LOOPSLIBRARYDIRECTORY)
  MASTERSCOPE MSPARSE))
(COMS
```

;;; The verb SEND

```
(P (MSADDRELATION '(SEND SENDS SENDING SENT)
  '(SENDNOTSELF SENDSELF))
 (MSADDMODIFIER 'SEND 'SELF 'SENDSELF)
 (MSADDMODIFIER 'SEND 'NOTSELF 'SENDNOTSELF))
(ADDVARS (DESCRIBELST ("sends msgs to self: " (GETRELATION FN '(SEND SELF)))
  ("sends msgs: " (GETRELATION FN '(SEND NOTSELF)))))
```

;;; Need templates for _Apply

```
(TEMPLATES |DefineClass| |DeleteCIV| |DeleteCV| |DeleteIV| _ _! _IV |_New| |_Process| |_Process!|
  |_Proto| |_Super| SEND |_SuperFringe| |_Super?| |_Try| |\\BindMethodInfo| |AddCIV| |AddCV|
  |AddIV| |ApplyMethod|)
(COMS
```

;;; The verbs IMPLEMENT, SPECIALIZE and OVERRIDE

```
(FNS |LoopsMethodMSGETDEF| |MethodOverrides?|)
(P (MSADDRELATION '(IMPLEMENT IMPLEMENTS IMPLEMENTING IMPLEMENTED))
 (SETSYNONYM '(SPECIALIZES &)
  '(@ (|MethodOverrides?| X)
  IMPLEMENTS 1))
 (SETSYNONYM '(SPECIALIZING &)
  '(@ (|MethodOverrides?| X)
  IMPLEMENTS 1))
 (SETSYNONYM '(OVERRIDES &)
  '(@ (|MethodOverrides?| X T)
  IMPLEMENTS 1))
 (SETSYNONYM '(OVERRIDING &)
  '(@ (|MethodOverrides?| X T)
  IMPLEMENTS 1))))
(COMS
```

;;; The verb GET

```
(P (MSADDRELATION '(GET GETS GETTING GOTTEN)
  '(GETNOTSELF GETSELF))
 (MSADDMODIFIER 'GET 'IV '(GETNOTSELF GETSELF))
 (MSADDMODIFIER 'GET 'SELF 'GETSELF)
 (MSADDMODIFIER 'GET 'NOTSELF 'GETNOTSELF)
 (SETSYNONYM 'GOT 'GOTTEN))
(ADDVARS (DESCRIBELST ("gets IVs of self: " (GETRELATION FN '(GET SELF)))
  ("gets IVs: " (GETRELATION FN '(GET NOTSELF)))))
(TEMPLATES |GetClassIV| |GetIVHere| |GetValue| |GetValueOnly| |GetClassIVHere|)
(P (ADDTEMPLATEWORD 'GETCVSELF)
 (ADDTEMPLATEWORD 'GETCVNOTSELF)
 (MSADDMODIFIER 'GET 'CV '(GETCVSELF GETCVNOTSELF))
 (MSADDMODIFIER 'GET 'CVSELF 'GETCVSELF)
 (MSADDMODIFIER 'GET 'CVNOTSELF 'GETCVNOTSELF))
(TEMPLATES |GetCVHere| |GetClassValue| |GetClassValueOnly|)
(ADDVARS (DESCRIBELST ("gets CVs of self: " (GETRELATION FN '(GET CVSELF)))
  ("gets CVs: " (GETRELATION FN '(GET CVNOTSELF)))))
(COMS
```

;;; The verb PUT

```
(P (MSADDRELATION '(PUT PUTS PUTTING PUT)
  '(PUTNOTSELF PUTSELF))
  (MSADDMODIFIER 'PUT 'IV '(PUTNOTSELF PUTSELF))
  (MSADDMODIFIER 'PUT 'SELF 'PUTSELF)
  (MSADDMODIFIER 'PUT 'NOTSELF 'PUTNOTSELF))
(ADDVARS (DESCRIBELST ("puts IVs of self: " (GETRELATION FN '(PUT SELF)))
  ("puts IVs: " (GETRELATION FN '(PUT NOTSELF)))))
(TEMPLATES |PutClassIV| |PutValue| |PutValueOnly|)
(P (ADDTEMPLATEWORD 'PUTCVSELF)
  (ADDTEMPLATEWORD 'PUTCVNOTSELF)
  (MSADDMODIFIER 'PUT 'CV '(PUTCVSELF PUTCVNOTSELF))
  (MSADDMODIFIER 'PUT 'CVSELF 'PUTCVSELF)
  (MSADDMODIFIER 'PUT 'CVNOTSELF 'PUTCVNOTSELF))
(TEMPLATES |PutClassValue| |PutClassValueOnly| |PutCVHere|)
(ADDVARS (DESCRIBELST ("puts CVs of self: " (GETRELATION FN '(PUT CVSELF)))
  ("puts CVs: " (GETRELATION FN '(PUT CVNOTSELF)))))
(COMS
```

;;; ... USE THE IV/CV/OBJECT/MSG ...

```
(P (MSADDDTYPE 'IV '(PUTNOTSELF PUTSELF GETNOTSELF GETSELF)
  '(IVS))
  (MSADDDTYPE 'CV '(PUTCVNOTSELF PUTCVSELF GETCVNOTSELF GETCVSELF)
  '(CVS))
  (ADDTEMPLATEWORD 'OBJECT)
  (MSADDDTYPE 'OBJECT '(OBJECT)
  '(OBJECTS))
  (MSADDDTYPE 'MSG '(SENDNOTSELF SENDSELF)
  '(MESSAGE SELECTOR MSGS MESSAGES SELECTORS)))
(TEMPLATES $ $!)
(COMS
```

;;; Extend the CHECK MasterScope command to verify certain constraints in Loops

```
(FNS |CheckLoopsFiles|)
(METHODS |Class.MSCheck| |Method.MSCheck|)
(ADDVARS (MSCHECKFNS |CheckLoopsFiles|))
(COMS
```

;;; A hook that lets method objects check their MasterScope data

```
(FNS |LoopsAnalyzeHook|)
(METHODS |Method.CheckMSData|)
(ADDVARS (ANALYZEUSERFNS |LoopsAnalyzeHook|))
(COMS
```

;;; Teach MasterScope how to analyze Classes and Methods

```
(FNS |AnalyzeClass| |AnalyzeMethod|)
(METHODS |Class.MSAnalyze| |Method.MSAnalyze|)
(P (MSADDANALYZE 'CLASSES 'CLASS)
  (MSADDANALYZE 'METHODS 'METHOD 'METHOD-FNS '|LoopsMethodMSGETDEF|))
```

;;; Definer for Masterscope method templates and templates for some system methods

```
(DEFINE-TYPES METHOD-TEMPLATES)
(FUNCTIONS DEF-METHOD-TEMPLATE)
(VARIABLES *LOOPS-METHOD-TEMPLATES*)
(FNS |QuotedArg?| |LoopsSELFp| |_PutGetTemplate| GET-METHOD-TEMPLATE MUNG-CV-TEMPLATE
  NAME-OBJECT-TEMPLATE PUT-METHOD-TEMPLATE)
(METHOD-TEMPLATES |AddCV| |AddIV| |Copy| |CopyCV| |CopyIV| |CreateClass| |DeleteIV| |Get| |New|
  |NewWithValues| |Put| |Rename| |SetName| |UnSetName|))
```

;;; Tailor MasterScope to understand Loops constructs

;;; Load MasterScope, since its a library module.

```
(COND
  ((EQ MAKESYSNAME ':LYRIC)
   ;; Nuke the FILEDATES properties so we're sure the Lyric compiled Medley versions are loaded.
   (REMPROP 'MASTERSCOPE 'FILEDATES)
   (REMPROP 'MSANALYZE 'FILEDATES)
   (REMPROP 'MSPARSE 'FILEDATES)))
(FILESLOAD (FROM VALUEOF LOOPSLIBRARYDIRECTORY)
  MASTERSCOPE MSPARSE)
```

;;; The verb SEND

```
(MSADDRELATION '(SEND SENDS SENDING SENT)
```

```

{MEDLEY}<loops>obsolete>LOOPSMS.;1

  (SENDNOTSELF SENDSELF))

(MSADDMODIFIER 'SEND 'SELF 'SENDESELF)

(MSADDMODIFIER 'SEND 'NOTSELF 'SENDNOTSELF)

(ADDTOTVAR DESCRIBELST ("sends msgs to self: " (GETRELATION FN '(SEND SELF)))
  ("sends msgs: " (GETRELATION FN '(SEND NOTSELF))))

;;; Need templates for _Apply

(SETTEMPLATE '|DefineClass| '(@ EXPR '(|if| (|QuotedArg?| (CAR EXPR))
  |then| '(NIL OBJECT)
  |else| 'EVAL)
  '(|if| (|QuotedArg?| (CADR EXPR))
  |then| '(NIL (|..| OBJECT))
  |else| 'EVAL)
  EVAL . PPE))

(SETTEMPLATE '|DeleteCIV| '(@ EXPR '(EVAL , (|if| (|QuotedArg?| (CADR EXPR))
  |then| '(NIL PUTNOTSELF)
  |else| 'EVAL)
  PROP . PPE)))

(SETTEMPLATE '|DeleteCV| '(@ EXPR '(EVAL , (|if| (|QuotedArg?| (CAR EXPR))
  |then| '(NIL PUTCVNOTSELF)
  |else| 'EVAL)
  PROP)))

(SETTEMPLATE '|DeleteIV| '(@ EXPR '(EVAL , (|if| (|QuotedArg?| (CADR EXPR))
  |then| (|if| (|LoopsSELFp| (CAR EXPR))
  |then| '(NIL PUTSELF)
  |else| '(NIL PUTNOTSELF))
  |else| 'EVAL)
  PROP . PPE)))

(SETTEMPLATE '|_ '(@ EXPR (|_PutGetTemplate|
  EXPR))

(SETTEMPLATE '|_!
  '(BOTH (NIL EVAL)
  (@ '(, (CAR EXPR)
  , (COND
  ((AND (LISTP (CADR EXPR))
  (EQ 'QUOTE (CAR (CADR EXPR))))
  (CADR (CADR EXPR)))
  (T '*UNKNOWN*)))
  ,@ (CDDR EXPR))
  (GETTEMPLATE '|_)))

(SETTEMPLATE '|_IV
  '(@ EXPR (COND
  ((|LoopsSELFp| (CAR EXPR))
  ' (EVAL GETSELF |..| EVAL))
  (T ' (EVAL GETNOTSELF |..| EVAL)))))

(SETTEMPLATE '|_New|
  '(@ EXPR (GETTEMPLATE '|_)))

(SETTEMPLATE '|_Process|
  '(@ EXPR (GETTEMPLATE '|_)))

(SETTEMPLATE '|_Process!|
  '(@ EXPR (GETTEMPLATE '|_!)))

(SETTEMPLATE '|_Proto|
  '(@ EXPR (GETTEMPLATE '|_)))

(SETTEMPLATE '|_Super|
  '(@ EXPR (COND
  ((|LoopsSELFp| (CAR EXPR))
  ' (EVAL SENDSELF |..| EVAL))
  (T ' (EVAL SENDNOTSELF |..| EVAL)))))

(SETTEMPLATE '|SEND '(@ EXPR (GETTEMPLATE '|_!)))

(SETTEMPLATE '|_SuperFringe|
  '(@ EXPR (|_PutGetTemplate|
  EXPR))

(SETTEMPLATE '|_Super?|
  '(@ EXPR (|_PutGetTemplate|
  EXPR))

(SETTEMPLATE '|_Try|
  '(@ EXPR (|_PutGetTemplate|

```

```

    (EXPR)))
(SETTEMPLATE '|\\BindMethodInfo| '(! NIL NIL EVAL))
(SETTEMPLATE '|AddCIV| '(@ EXPR `(EVAL ,(|if| (|QuotedArg?| (CADR EXPR))
      |then| '(NIL PUTNOTSELF)
      |else| 'EVAL)
    EVAL
    (PROP EVAL) . PPE)))
(SETTEMPLATE '|AddCV| '(@ EXPR `(EVAL ,(|if| (|QuotedArg?| (CAR EXPR))
      |then| '(NIL PUTCVNOTSELF)
      |else| 'EVAL)
    EVAL)))
(SETTEMPLATE '|AddIV| '(@ EXPR `(EVAL ,(|if| (|QuotedArg?| (CADR EXPR))
      |then| (|if| (|LoopsSELFp| (CAR EXPR))
        |then| '(NIL PUTSELF)
        |else| '(NIL PUTNOTSELF)
      |else| 'EVAL)
    EVAL PROP . PPE)))
(SETTEMPLATE '|ApplyMethod| '(BOTH (EVAL EVAL EVAL ,(COND
      ((|QuotedArg?| (CADDR EXPR))
      '(NIL OBJECT))
      (T 'EVAL)) . PPE)
  (@ `,(CAR EXPR)
    ,(CADR EXPR))
  (GETTEMPLATE '|_!|)))

```

;;; The verbs IMPLEMENT, SPECIALIZE and OVERRIDE

(DEFINEQ

(|LoopsMethodMSGETDEF|

```

(LAMBDA (NAME TYPE SOURCE OPTIONS) ; Edited 8-Jun-88 17:01 by JAMES.PA
  (LET* ((METHBODY (CDR (REMOVE-COMMENTS (GETDEF NAME 'METHOD-FNS SOURCE OPTIONS))))
        (CLP?
         (NAME&ARGS (|if| (EQ (CAR METHBODY)
                              ':FUNCTION-TYPE)
                          |then| (|pop| METHBODY)
                              (SETQ CLP? (EQ (|pop| METHBODY)
                                              ':CL))
                          |pop| METHBODY)
                   |else| (|pop| METHBODY))))
    ;; If the source can be found, this will return something of the form:
    ;; (Method ((class-name selector) arg1 arg2...) body1...)
    ;; Masterscope wants to analyze something of the form:
    ;; (LAMBDA (ARGS) BODY...) or (CL:LAMBDA (ARGS) BODY...) depending on whether :FUNCTION-TYPE is specified
    (AND METHBODY `,(|if| CLP?
      |then| 'CL:LAMBDA
      |else| 'LAMBDA)
    ,(CDR NAME&ARGS)
    ,@METHBODY))))

```

(|MethodOverrides?|

```

(LAMBDA (|meth| |noSendSuper?|) (* |smL| " 9-Apr-87 16:33")
  (** |Does| |this| |method| |override| \a |super| |method?|)
  (LET ((|methObj| ($! |meth|)))
    (AND |methObj| (|_ |methObj| |InstOf!| ' |Method|)
      (|for| |class| |in| (|_ ($! (@ |methObj| |className|)
        |ListAttribute|
        '|Supers|)
      |thereis| (* |gee| |is| |there| |any| |super| |method| |at| |all?|)
        (AND (|type?| |class| ($! |class|)
          (|FetchMethod| ($! |class|)
            (@ |methObj| |selector|)))
          (LET ((|specialize?| (OR (TESTRELATION |meth| '(CALL)
            '|_Super|)
              (TESTRELATION |meth| '(CALL)
                '|_SuperFringe|)
              (TESTRELATION |meth| '(CALL)
                '|_Super?|))))
            (* |does| |this| |override| |or| |specialize?|)
          (|if| |noSendSuper?|
            |then| (NOT |specialize?|)
            |else| |specialize?|))))))
  )
(MSADDERELATION '(IMPLEMENT IMPLEMENTS IMPLEMENTING IMPLEMENTED))

```

```
(SETSYNONYM '(SPECIALIZES &)
' (@ (|MethodOverrides?| X)
  IMPLEMENTS 1))

(SETSYNONYM '(SPECIALIZING &)
' (@ (|MethodOverrides?| X)
  IMPLEMENTS 1))

(SETSYNONYM '(OVERRIDES &)
' (@ (|MethodOverrides?| X T)
  IMPLEMENTS 1))

(SETSYNONYM '(OVERRIDING &)
' (@ (|MethodOverrides?| X T)
  IMPLEMENTS 1))
```

;;; The verb GET

```
(MSADDERELATION '(GET GETS GETTING GOTTEN)
' (GETNOTSELF GETSELF))

(MSADDMODIFIER 'GET 'IV ' (GETNOTSELF GETSELF))

(MSADDMODIFIER 'GET 'SELF 'GETSELF)

(MSADDMODIFIER 'GET 'NOTSELF 'GETNOTSELF)

(SETSYNONYM 'GOT 'GOTTEN)

(ADDOVAR DESCRIBELST ("gets IVs of self: " (GETRELATION FN '(GET SELF)))
  ("gets IVs: " (GETRELATION FN '(GET NOTSELF))))

(SETTEMPLATE '|GetClassIV| '(! NIL @ EXPR (COND
  ((NOT (AND (LISTP (CADR EXPR))
    (EQ (CAADR EXPR)
      'QUOTE)))
    '(EVAL EVAL PROP . PPE))
  ((|LoopsSELFP| (CAR EXPR))
    '(EVAL (NIL GETSELF)
      PROP . PPE))
  (T '(EVAL (NIL GETNOTSELF)
    PROP . PPE))))))

(SETTEMPLATE '|GetIVHere| '(@ EXPR (COND
  ((NOT (AND (LISTP (CADR EXPR))
    (EQ (CAADR EXPR)
      'QUOTE)))
    '(EVAL EVAL PROP . PPE))
  ((|LoopsSELFP| (CAR EXPR))
    '(EVAL (NIL GETSELF)
      PROP . PPE))
  (T '(EVAL (NIL GETNOTSELF)
    PROP . PPE))))))

(SETTEMPLATE '|GetValue| '(! NIL @ EXPR (COND
  ((NOT (AND (LISTP (CADR EXPR))
    (EQ (CAADR EXPR)
      'QUOTE)))
    '(EVAL EVAL PROP . PPE))
  ((|LoopsSELFP| (CAR EXPR))
    '(EVAL (NIL GETSELF)
      PROP . PPE))
  (T '(EVAL (NIL GETNOTSELF)
    PROP . PPE))))))

(SETTEMPLATE '|GetValueOnly| '(@ EXPR (COND
  ((NOT (AND (LISTP (CADR EXPR))
    (EQ (CAADR EXPR)
      'QUOTE)))
    '(EVAL EVAL PROP . PPE))
  ((|LoopsSELFP| (CAR EXPR))
    '(EVAL (NIL GETSELF)
      PROP . PPE))
  (T '(EVAL (NIL GETNOTSELF)
    PROP . PPE))))))

(SETTEMPLATE '|GetClassIVHere| '(@ EXPR (COND
  ((NOT (AND (LISTP (CADR EXPR))
    (EQ (CAADR EXPR)
      'QUOTE)))
    '(EVAL EVAL PROP . PPE))
  ((|LoopsSELFP| (CAR EXPR))
    '(EVAL (NIL GETSELF)
      PROP . PPE))
  (T '(EVAL (NIL GETNOTSELF)
    PROP . PPE))))))
```

```

(ADDTEMPLATEWORD 'GETCVSELF)
(ADDTEMPLATEWORD 'GETCVNOTSELF)
(MSADDMODIFIER 'GET 'CV ' (GETCVSELF GETCVNOTSELF))
(MSADDMODIFIER 'GET 'CVSELF 'GETCVSELF)
(MSADDMODIFIER 'GET 'CVNOTSELF 'GETCVNOTSELF)
(SETTEMPLATE '|GetCVHere| '(@ EXPR (COND
  ((NOT (AND (LISTP (CADR EXPR))
             (EQ (CAADR EXPR)
                 'QUOTE)))
   '(EVAL EVAL PROP . PPE))
  ((|LoopsSELFP| (CAR EXPR))
   '(EVAL (NIL GETCVSELF)
          PROP . PPE))
  (T '(EVAL (NIL GETCVNOTSELF)
            PROP . PPE))))))
(SETTEMPLATE '|GetClassValue| '(! NIL @ EXPR (COND
  ((NOT (AND (LISTP (CADR EXPR))
             (EQ (CAADR EXPR)
                 'QUOTE)))
   '(EVAL EVAL PROP . PPE))
  ((|LoopsSELFP| (CAR EXPR))
   '(EVAL (NIL GETCVSELF)
          PROP . PPE))
  (T '(EVAL (NIL GETCVNOTSELF)
            PROP . PPE))))))
(SETTEMPLATE '|GetClassValueOnly| '(@ EXPR (COND
  ((NOT (AND (LISTP (CADR EXPR))
             (EQ (CAADR EXPR)
                 'QUOTE)))
   '(EVAL EVAL PROP . PPE))
  ((|LoopsSELFP| (CAR EXPR))
   '(EVAL (NIL GETCVSELF)
          PROP . PPE))
  T
  '(EVAL (NIL GETCVNOTSELF)
          PROP . PPE))))))
(ADDOVAR DESCRIBELST ("gets CVs of self: " (GETRELATION FN '(GET CVSELF)))
 ("gets CVs: " (GETRELATION FN '(GET CVNOTSELF))))

```

;;; The verb PUT

```

(MSADDERELATION ' (PUT PUTS PUTTING PUT)
 ' (PUTNOTSELF PUTSELF))
(MSADDMODIFIER 'PUT 'IV ' (PUTNOTSELF PUTSELF))
(MSADDMODIFIER 'PUT 'SELF 'PUTSELF)
(MSADDMODIFIER 'PUT 'NOTSELF 'PUTNOTSELF)
(ADDOVAR DESCRIBELST ("puts IVs of self: " (GETRELATION FN '(PUT SELF)))
 ("puts IVs: " (GETRELATION FN '(PUT NOTSELF))))
(SETTEMPLATE '|PutClassIV| '(! NIL @ EXPR (COND
  ((NOT (AND (LISTP (CADR EXPR))
             (EQ (CAADR EXPR)
                 'QUOTE)))
   '(EVAL EVAL EVAL PROP . PPE))
  ((|LoopsSELFP| (CAR EXPR))
   '(EVAL (NIL PUTSELF)
          EVAL PROP . PPE))
  (T '(EVAL (NIL PUTNOTSELF)
            EVAL PROP . PPE))))))
(SETTEMPLATE '|PutValue| '(! NIL @ EXPR (COND
  ((NOT (AND (LISTP (CADR EXPR))
             (EQ (CAADR EXPR)
                 'QUOTE)))
   '(EVAL EVAL EVAL PROP . PPE))
  ((|LoopsSELFP| (CAR EXPR))
   '(EVAL (NIL PUTSELF)
          EVAL PROP . PPE))
  (T '(EVAL (NIL PUTNOTSELF)
            EVAL PROP . PPE))))))
(SETTEMPLATE '|PutValueOnly| '(@ EXPR (COND
  ((NOT (AND (LISTP (CADR EXPR))

```

```

(EQ (CAADR EXPR)
'QUOTE)))
'(EVAL EVAL EVAL PROP . PPE))
(|LoopsSELFP| (CAR EXPR))
'(EVAL (NIL PUTSELF)
EVAL PROP . PPE))
(T '(EVAL (NIL PUTNOTSELF)
EVAL PROP . PPE))))

```

(ADDTEMPLATEWORD 'PUTCVSELF)

(ADDTEMPLATEWORD 'PUTCVNOTSELF)

(MSADDMODIFIER 'PUT 'CV '(PUTCVSELF PUTCVNOTSELF))

(MSADDMODIFIER 'PUT 'CVSELF 'PUTCVSELF)

(MSADDMODIFIER 'PUT 'CVNOTSELF 'PUTCVNOTSELF)

```

(SETTEMPLATE '|PutClassValue| '(! NIL @ EXPR (COND
((NOT (AND (LISTP (CADR EXPR))
(EQ (CAADR EXPR)
'QUOTE)))
'(EVAL EVAL EVAL PROP . PPE))
(|LoopsSELFP| (CAR EXPR))
'(EVAL (NIL PUTCVSELF)
EVAL PROP . PPE))
(T '(EVAL (NIL PUTCVNOTSELF)
EVAL PROP . PPE))))

```

```

(SETTEMPLATE '|PutClassValueOnly| '@ EXPR (COND
((NOT (AND (LISTP (CADR EXPR))
(EQ (CAADR EXPR)
'QUOTE)))
'(EVAL EVAL EVAL PROP . PPE))
(|LoopsSELFP| (CAR EXPR))
'(EVAL (NIL PUTCVSELF)
EVAL PROP . PPE))
(T '(EVAL (NIL PUTCVNOTSELF)
EVAL PROP . PPE))))

```

```

(SETTEMPLATE '|PutCVHere| '@ EXPR (COND
((NOT (AND (LISTP (CADR EXPR))
(EQ (CAADR EXPR)
'QUOTE)))
'(EVAL EVAL EVAL PROP . PPE))
(|LoopsSELFP| (CAR EXPR))
'(EVAL (NIL PUTCVSELF)
EVAL PROP . PPE))
(T '(EVAL (NIL PUTCVNOTSELF)
EVAL PROP . PPE))))

```

```

(ADDTOVAR DESCRIBELST ("puts CVs of self: " (GETRELATION FN '(PUT CVSELF)))
("puts CVs: " (GETRELATION FN '(PUT CVNOTSELF))))

```

;;; ... USE THE IV/CV/OBJECT/MSG ...

```

(MSADDDTYPE 'IV '(PUTNOTSELF PUTSELF GETNOTSELF GETSELF)
'(IVS))

```

```

(MSADDDTYPE 'CV '(PUTCVNOTSELF PUTCVSELF GETCVNOTSELF GETCVSELF)
'(CVS))

```

(ADDTEMPLATEWORD 'OBJECT)

```

(MSADDDTYPE 'OBJECT '(OBJECT)
'(OBJECTS))

```

```

(MSADDDTYPE 'MSG '(SENDNOTSELF SENDSELF)
'(MESSAGE SELECTOR MSGS MESSAGES SELECTORS))

```

(SETTEMPLATE '\$ '(OBJECT . PPE))

(SETTEMPLATE '\$! '(EVAL . PPE))

;;; Extend the CHECK MasterScope command to verify certain constraints in Loops

(DEFINEQ

(|CheckLoopsFiles|

(LAMBDA (|files|)

(* |smL| "14-Feb-86 16:45")

(* *|Verify| |that| |all| |methods| |defined| |on| |the| |files| |are| |somehow| |kosher|)

(|for| |file| |in| |files| |do| (|for| |method| |in| (FILECOMSLST |file| 'METHODS)

```

      |do| ( _ ($! |method|)
            |MSCheck| |file| NIL))
    (|for| |class| |in| (FILECOMSLST |file| 'CLASSES) |do| ( _ ($! |class|)
      |MSCheck| |file| NIL))))))
)

```

(|\\BatchMethodDefs|)

```

(METH |Class| |MSCheck| (|onFile| |outputFile|)
  "Use MasterScope to check the consistency of the class"
  (|category| (|Masterscope| |Class|)))

```

```

(METH |Method| |MSCheck| (|onFile| |outputFile|)
  "Check that the method is kosher"
  (|category| (|Masterscope| |Method|)))

```

```

(|Method| ((|Class| |MSCheck|)
  |self| |onFile| |outputFile|) ; smL 11-Apr-86 14:45

```

;; Use MasterScope to check the consistency of the class

```

(LET ((|comments| NIL)
      (|className| ( _ |self| |ClassName|))
      (|localIVs| ( _ |self| |ListAttribute| 'IV\s))
      (|localCVs| ( _ |self| |ListAttribute| 'CV\s)))
  (|for| |iv| |in| (LDIFFERENCE |localIVs| (MASTERSCOPE `(WHAT IV IN ',|localIVs| IS USED BY ANY)))
    |do| (|push| |comments| (CONCAT "IV " |iv| " defined but not used by any")))
  (|for| |cv| |in| (LDIFFERENCE (LDIFFERENCE |localCVs| (MASTERSCOPE `(WHAT CV IN ',|localCVs| IS USED BY
    ANY)))
    (MASTERSCOPE `(WHAT IV IN ',|localCVs| IS USED BY ANY)))
    |do| (|push| |comments| (CONCAT "CV " |cv| " defined but not used by any")))
  (|if| |comments|
    |then| (|printout| |outputFile| T .FONT BOLDFONT |className| .FONT DEFAULTFONT ": "
      (|for| |comment| |in| |comments| |do| (|printout| |outputFile| .TAB 15 |comment| T))
      |className|
    |else| NIL)))

```

```

(|Method| ((|Method| |MSCheck|)
  |self| |onFile| |outputFile|) ; smL 11-Apr-86 15:06

```

;; Check that the method is kosher

```

(LET ((|comments| NIL)
      (|methodName| (@ |method|))
      (|argList| (ARGLIST (@ |method|)))
      (|methodClass| ($! (@ |className|))))
  (MASTERSCOPE `(ANALYZE ',|methodName|)) (* |check| |to| |make| |sure| |that| |the| |first| |arg| |is| |self|)
  (|if| (NEQ (CAR |argList|)
    '|self|)
    |then| (|push| |comments| "first arg is not self") (* |check| |that| |the| |first| |arg| |doesn't| |get| |smashed|)
  (|if| (MEMB (CAR |argList|)
    (GETRELATION |methodName| '(SMASH)))
    |then| (|push| |comments| (|if| (EQ (CAR |argList|)
      '|self|)
      |then| "smashes self"
      |else| (CONCAT "smashes " (CAR |argList|)
        " [its first arg]"))
        (* |look| |for| |msgs| |sent| |to| |self| |that| |are| |not| |defined| |in|
          |the| |class|)
  (LET ((|funnyMsgs| (LDIFFERENCE (GETRELATION |methodName| '(SEND SELF))
    ( _ |methodClass| |ListAttribute| ' |Methods| NIL T)))
    (|if| (NULL |funnyMsgs|)
      |then| NIL
      |elseif| (EQLLENGTH |funnyMsgs| 1)
      |then| (|push| |comments| (CONCAT "sends " (CAR |funnyMsgs|)
        " to self, but the msg is not defined in the class"))
      |else| (|push| |comments| (CONCAT "sends " |funnyMsgs| " to self, but the msgs are not defined
        in the class"))))

```

(* |look| |for| |ivs| |of| |self| |used| |but| |not| |defined| |in| |the| |class| -
NOTE |that| |this| |includes| CVs |as| IVs |because| |of| |the| |new| |:allocation| |stuff|)

```

(LET ((|funnyIVs| (LDIFFERENCE (LDIFFERENCE (UNION (GETRELATION |methodName| ' ((GET SELF)))
  (GETRELATION |methodName| ' ((PUT SELF))))
  ( _ |methodClass| |ListAttribute| 'IV\s NIL T))
  ( _ |methodClass| |ListAttribute| 'CV\s NIL T)))
  (|if| (NULL |funnyIVs|)
    |then| NIL
    |elseif| (EQLLENGTH |funnyIVs| 1)
    |then| (|push| |comments| (CONCAT " uses the IV " (CAR |funnyIVs|)
      " that is not defined in the class"))
    |else| (|push| |comments| (CONCAT " uses the IVs " |funnyIVs| " that are not defined in the
      class"))))
(LET ((|funnyCVs| (LDIFFERENCE (UNION (GETRELATION |methodName| ' ((GET CVSELF)))
  (GETRELATION |methodName| ' ((PUT CVSELF))))
  ( _ |methodClass| |ListAttribute| 'CV\s NIL T)))

```



```

(|if| (NULL |funnyCVs|)
  |then| NIL
  |elseif| (EQLLENGTH |funnyCVs| 1)
  |then| (|push| |comments| (CONCAT " uses the CV " (CAR |funnyCVs|)
    " that is not defined in the class"))
  |else| (|push| |comments| (CONCAT " uses the CVs " |funnyCVs| " that are not defined in the
    class"))))
(|if| |comments|
  |then| (|printout| |outputFile| T .FONT BOLDFONT |methodName| .FONT DEFAULTFONT ": ")
  (|for| |comment| |in| (REVERSE |comments|) |do| (|printout| |outputFile| .TAB 15 |comment| T))
  |else| NIL)))

```

(|\UnbatchMethodDefs|)

(ADDTOVAR **MSCHECKFNS** |CheckLoopsFiles|)

;;; A hook that lets method objects check their MasterScope data

(DEFINEQ

(|LoopsAnalyzeHook|

(LAMBDA (|fnName| |def| |data|) (* |smL:| "11-Nov-85 12:26")

(* * |When| |the| |method| |is| |analyzed| |by| |MasterScope| |try| |to| |send| \a |message| |to| |the| |method| |object| |so| |it|
|can| |check| |the| |data|)

(LET ((|methObj| (\$! |fnName|)))
(|if| (AND |methObj| (_ |methObj| |InstOf!| ' |Method|))
 |then|

(* |The| |methObj| |should| |return| |the|
(|possibly| |altered|) |data|)

(_ |methObj| |CheckMSData| |data|)
|else| |data|))))

)

(|\BatchMethodDefs|)

(METH |Method| |CheckMSData| (|alist|)

"The method function is being analyzed. Check the resulting data and return the (possibly modified) data"
(|category| (|Masterscope| |Method|)))

(|Method| ((|Method| |CheckMSData|)
|self| |alist|)

; smL 10-Feb-86 12:44

;;; The method function is being analyzed. Check the resulting data and return the (possibly modified) data

```

(LET ((|selector| (@ |selector|))
  (|implements| (ASSOC 'IMPLEMENT |alist|)))
  (|if| (NULL |implements|)
    |then| (NCONC1 |alist| (LIST 'IMPLEMENT |selector|))
    |elseif| (NOT (MEMB |selector| (CDR |implements|)))
    |then| (NCONC1 |implements| |selector|)))
|alist|)

```

(|\UnbatchMethodDefs|)

(ADDTOVAR **ANALYZEUSERFNS** |LoopsAnalyzeHook|)

;;; Teach MasterScope how to analyze Classes and Methods

(DEFINEQ

(|AnalyzeClass|

(LAMBDA (|className| |reanalyze?|) (* |smL:| "12-Nov-85 13:13")

(* * |Analyze| |the| |class| |for| |MasterScope|)

```

(LET ((|class| ($! |className|)))
  (|if| (AND |class| (_ |class| |InstOf!| ' |Class|))
    |then| (_ |class| |MSAnalyze| |reanalyze?|))))

```

(|AnalyzeMethod|

(LAMBDA (|methodName| |reanalyze?|) (* |smL:| "12-Nov-85 13:13")

(* * |Analyze| |the| |Loops| |method| |for| |MasterScope|)

```

(LET ((|methodObj| ($! |methodName|)))
  (|if| (AND |methodObj| (_ |methodObj| |InstOf!| ' |Method|))
    |then| (_ |methodObj| |MSAnalyze| |reanalyze?|))))

```

)

(|\BatchMethodDefs|)

```
(METH |Class| |MSAnalyze| (|reanalyze?|)
  "Analyze the class for MasterScope"
  (|category| (|Masterscope| |Class|)))

(METH |Method| |MSAnalyze| (|reanalyze?|)
  "Analyze the method for MasterScope"
  (|category| (|Masterscope| |Method|)))
```

```
(|Method| ((|Class| |MSAnalyze|)
  |self| |reanalyze?|) ; smL: 12-Nov-85 13:10
```

;;; Analyze the class for MasterScope

NIL)

```
(|Method| ((|Method| |MSAnalyze|)
  |self| |reanalyze?|) ; smL 5-Dec-85 10:30
```

;;; Analyze the method for MasterScope

NIL)

```
(|\UnbatchMethodDefs|)
```

```
(MSADDANALYZE 'CLASSES 'CLASS)
```

```
(MSADDANALYZE 'METHODS 'METHOD 'METHOD-FNS ' |LoopsMethodMSGETDEF|)
```

;;; Definer for Masterscope method templates and templates for some system methods

```
(DEF-DEFINE-TYPE METHOD-TEMPLATES "Functions to generate Masterscope templates for LOOPS methods")
```

```
(DEFDEFINER (DEF-METHOD-TEMPLATE (:PROTOTYPE
  (LAMBDA (METHOD)
    ;; The mondo-weird quoting below is to keep the back-quoted structure from expanding too
    ;; early.
    `(DEF-METHOD-TEMPLATE ,METHOD (|self| |message| "other-arguments")
      (LET
        ((|self?| (|LoopsSELFp| |self|)))
        , (LIST
          'BQUOTE
          ' (EVAL
            , (|if| |self?|
              |then| 'SENDSELF
              |else| 'SENDNOTSELF
              "rest of template")))))
      (:UNDEFINER (LAMBDA (METHOD)
        (REMHASH METHOD *LOOPS-METHOD-TEMPLATES*)))
      METHOD-TEMPLATES (METHOD &REST REST)
      `(CL:SETF (CL:GETHASH ',METHOD *LOOPS-METHOD-TEMPLATES*)
        , (|if| (AND (NULL (CDR REST))
          (CL:SYMBOLP (CAR REST)))
          |then| `', (CAR REST)
          |else| `'#' (LAMBDA ,@REST))))
```

```
(CL:DEFVAR *LOOPS-METHOD-TEMPLATES* (CL:MAKE-HASH-TABLE))
```

```
(DEFINEQ
```

```
(|QuotedArg?|
  (LAMBDA (THING) ; Edited 2-Dec-87 13:13 by jrb:
    (AND (LISTP THING)
      (EQ (CAR THING)
        'QUOTE))))
```

```
(|LoopsSELFp|
  (LAMBDA (INSTANCE-NAME) (* |edited:| " 6-Oct-87 13:06")
    (OR (EQ INSTANCE-NAME 'SELF)
      (EQ INSTANCE-NAME '|self|))))
```

```
(|_PutGetTemplate|
  (LAMBDA (EX) ; Edited 11-Apr-88 15:08 by jrb:
```

;;; If the method is something special like Get, Put, or a naming method, return an appropriately munged template.

```
(LET ((|template-function| (GET-METHOD-TEMPLATE (CADR EX))))
  (|if| |template-function|
    |then| (APPLY |template-function| EX)
    |else| `(EVAL , (|if| (|LoopsSELFp| (CAR EX))
      |then| 'SENDSELF
```

```
      |else| 'SENDNOTSELF)
    |..| EVAL)))))
```

(GET-METHOD-TEMPLATE

```
(LAMBDA (MESSAGE)
  (CL:GETHASH MESSAGE *LOOPS-METHOD-TEMPLATES*))
```

(MUNG-CV-TEMPLATE

```
(LAMBDA (|self| |message| |varName| |newValue|)
  (LET ((|self?| (|LoopsSELFp| |self|)))
    `(EVAL , (|if| |self?|
      |then| 'SENDSELF
      |else| 'SENDNOTSELF)
      , (|if| (|QuotedArg?| |varName|)
        |then| ' (NIL PUTCVNOTSELF)
        |else| 'EVAL)
      EVAL))))
```

(NAME-OBJECT-TEMPLATE

```
(LAMBDA (|self| |message| |name|)
  (LET ((|self?| (|LoopsSELFp| |self|)))
    `(EVAL , (|if| |self?|
      |then| 'SENDSELF
      |else| 'SENDNOTSELF)
      , (|if| (|QuotedArg?| |name|)
        |then| ' (NIL OBJECT)
        |else| 'EVAL) . PPE))))
```

(PUT-METHOD-TEMPLATE

```
(LAMBDA (|self| |message| |varName| |newValue| |propName|)
  (LET ((|self?| (|LoopsSELFp| |self|)))
    `(EVAL , (|if| |self?|
      |then| 'SENDSELF
      |else| 'SENDNOTSELF)
      , (|if| (|QuotedArg?| |varName|)
        |then| (|if| |self?|
          |then| ' (NIL PUTSELF)
          |else| ' (NIL PUTNOTSELF))
        |else| 'EVAL)
      RESULT PROP . PPE))))
```

)

```
(DEF-METHOD-TEMPLATE |AddCV| MUNG-CV-TEMPLATE)
```

```
(DEF-METHOD-TEMPLATE |AddIV| PUT-METHOD-TEMPLATE)
```

```
(DEF-METHOD-TEMPLATE |Copy| NAME-OBJECT-TEMPLATE)
```

```
(DEF-METHOD-TEMPLATE |CopyCV| MUNG-CV-TEMPLATE)
```

```
(DEF-METHOD-TEMPLATE |CopyIV| (|self| |message| |ivName| |toClass|)
  (LET ((|self?| (|LoopsSELFp| |self|)))
    `(EVAL , (|if| |self?|
      |then| 'SENDSELF
      |else| 'SENDNOTSELF)
      , (|if| (|QuotedArg?| |ivName|)
        |then| ' (NIL PUTNOTSELF)
        |else| 'EVAL)
      , (|if| (|QuotedArg?| |toClass|)
        |then| ' (NIL OBJECT)
        |else| 'EVAL))))
```

```
(DEF-METHOD-TEMPLATE |CreateClass| (|self| |message| |name| |supers|)
  (LET ((|self?| (|LoopsSELFp| |self|)))
    `(EVAL , (|if| |self?|
      |then| 'SENDSELF
      |else| 'SENDNOTSELF)
      , (|if| (|QuotedArg?| |name|)
        |then| ' (NIL OBJECT)
        |else| 'EVAL)
      , (|if| (|QuotedArg?| |supers|)
        |then| ' (NIL (|..| OBJECT))
        |else| 'EVAL) . PPE))))
```

```
(DEF-METHOD-TEMPLATE |DeleteIV| (|self| |message| |varName| |propName|)
  (LET ((|self?| (|LoopsSELFp| |self|)))
    `(EVAL , (|if| |self?|
      |then| 'SENDSELF
      |else| 'SENDNOTSELF)
      , (|if| (|QuotedArg?| |varName|)
        |then| (|if| |self?|
          |then| ' (NIL PUTSELF)
```

```

      |else| ' (NIL PUTNOTSELF))
    |else| 'EVAL)
  PROP . PPE)))

```

```

(DEF-METHOD-TEMPLATE |Get| (|self| |message| |varName| |propName|)
  (LET ((|self?| (|LoopsSELFp| |self|)))
    `(EVAL , (|if| |self?|
      |then| 'SENDSELF
      |else| 'SENDNOTSELF)
      , (|if| (|QuotedArg?| |varName|)
        |then| (|if| |self?|
          |then| ' (NIL GETSELF)
          |else| ' (NIL GETNOTSELF))
        |else| 'EVAL)
      PROP . PPE)))

```

```

(DEF-METHOD-TEMPLATE |New| NAME-OBJECT-TEMPLATE)

```

```

(DEF-METHOD-TEMPLATE |NewWithValues| (|self| |message| |valDescriptionList|)
  (LET ((|self?| (|LoopsSELFp| |self|)))
    `(EVAL , (|if| |self?|
      |then| 'SENDSELF
      |else| 'SENDNOTSELF)
      , (|if| (|QuotedArg?| |valDescriptionList|)
        |then| ' (NIL |..| (PUTNOTSELF NIL PROP NIL))
        |else| 'EVAL) . PPE)))

```

```

(DEF-METHOD-TEMPLATE |Put| PUT-METHOD-TEMPLATE)

```

```

(DEF-METHOD-TEMPLATE |Rename| NAME-OBJECT-TEMPLATE)

```

```

(DEF-METHOD-TEMPLATE |SetName| NAME-OBJECT-TEMPLATE)

```

```

(DEF-METHOD-TEMPLATE |UnSetName| NAME-OBJECT-TEMPLATE)

```

```

(PUTPROPS LOOPSMS COPYRIGHT ("Venue & Xerox Corporation" 1985 1986 1987 1988 1990 1992 1993))

```

FUNCTION INDEX

| | | | |
|-----------------------------|--------------------------|-----------------------------|--------------------------|
| AnalyzeClass 9 | LoopsAnalyzeHook 9 | MUNG-CV-TEMPLATE11 | _PutGetTemplate 10 |
| AnalyzeMethod 9 | LoopsMethodMSGDEF ...4 | NAME-OBJECT-TEMPLATE ...11 | |
| CheckLoopsFiles 7 | LoopsSELP 10 | PUT-METHOD-TEMPLATE11 | |
| GET-METHOD-TEMPLATE11 | MethodOverrides? 4 | QuotedArg? 10 | |

TEMPLATE INDEX

| | | | |
|---------------------|---------------------------|---------------------------|----------------------|
| \$7 | GetClassIV 5 | PutClassValueOnly 7 | _Process 3 |
| \$!7 | GetClassIVHere 5 | PutCVHere 7 | _Process! 3 |
| AddCIV 4 | GetClassValue 6 | PutValue 6 | _Proto 3 |
| AddCV 4 | GetClassValueOnly 6 | PutValueOnly 6 | _Super 3 |
| AddIV 4 | GetCVHere 6 | SEND3 | _Super? 3 |
| ApplyMethod 4 | GetIVHere 5 | \\BindMethodInfo 4 | _SuperFringe 3 |
| DefineClass 3 | GetValue 5 | _3 | _Try 3 |
| DeleteCIV 3 | GetValueOnly 5 | !3 | |
| DeleteCV 3 | PutClassIV 6 | _IV3 | |
| DeleteIV 3 | PutClassValue 7 | _New 3 | |

METHOD-TEMPLATE INDEX

| | | | | |
|----------------|----------------------|-------------------|---------------------|--------------------|
| AddCV 11 | CopyCV 11 | DeleteIV 11 | NewWithValues ..12 | SetName 12 |
| AddIV 11 | CopyIV 11 | Get 12 | Put 12 | UnSetName 12 |
| Copy 11 | CreateClass 11 | New 12 | Rename 12 | |

VARIABLE INDEX

| | |
|----------------------------------|--------------------------|
| *LOOPS-METHOD-TEMPLATES*10 | DESCRIBELST3,5,6,7 |
| ANALYZEUSERFNS9 | MSCHECKFNS9 |

DEFINER INDEX

| |
|-----------------------------|
| DEF-METHOD-TEMPLATE10 |
|-----------------------------|

DEFINE-TYPE INDEX

| |
|--------------------------|
| METHOD-TEMPLATES10 |
|--------------------------|