File created: 21-Apr-86 14:03:45 **{POGO:PARC:XEROX}<LOOPS>TESTER>LTLISPFUNTESTS.;2**

changes to: (INSTANCES LTP30 LTP31 LTP32 LTP38 LTP39 LTP40 LTP8 LTP85 LTP26 LTP27 LTP28 LTP29 LTP33 LTP34 LTP35 LTP36 LTP37 LTP6 LTP7 LTP2 LTP81 LTP82 LTP83 LTP84 LTFDC LTFGetClassValue LTFGetValueOnly LTFPutClassValue LTFPutValueOnly LTFPutClass LTFRenameMethod LTFCalledFns LTFRenameVariable LTFMoveVariable LTFMoveMethod LTFGetLocalState LTFPutLocalState LTFGetObjectRec LTFClass LTFtype? LTFGetObjectNames LTFGetNthValue LTFGetInitialValue LTFSendSuper)

previous date: 18-Mar-85 12:04:27 {POGO:PARC:XEROX}<LOOPS>TESTER>LTLISPFUNTESTS.;1

Read Table: OLD-INTERLISP-FILE

Package: INTERLISP

Format: XCCS


```
(RPAQQ LTLISPFUNTESTSCOMS ((* File created by MITTAL)
                           (* database of test instances for testing Loops Functions)
                           (CLASSES)
                           (METHODS)
                           (FNS)
                           (VARS)
                           (INSTANCES * LTLispFunTestsInstances)))
```

(* * File created by MITTAL)

(* * database of test instances for testing Loops Functions)

```
(DEFCLASSES)

(RPAQQ LTLispFunTestsInstances
       (LTFSendSuper LTFGetInitialValue LTFPutNthValue LTFGetNthValue LTFGetObjectNames LTFtype? LTFClass
              LTFGetObjectRec LTFReplaceActiveValue LTFPutLocalState LTFGetLocalState LTFRenameMethodFunction
              LTFMoveMethod LTFMoveVariable LTFMoveClassVariable LTFRenameVariable LTFCalledFns LTFRenameMethod
              LTFTryMethod LTFDoMethod LTFPutItOnly LTFGetItOnly LTFPutIt LTFGetIt LTFPutMethodOnly
              LTFGetMethodOnly LTFPutClassOnly LTFGetClassOnly LTFPutClass LTFGetClass LTFPutClassValueOnly
              LTFPutValueOnly LTFPushClassValue LTFPutClassValue LTFAddValue LTFPushValue LTFGetClassValueOnly
              LTFGetValueOnly LTFGetClassValue LTFDM LTFDC LTP84 LTP83 LTP82 LTP81 LTP2 LTP7 LTP6 LTP3 LTP37
              LTP36 LTP35 LTP34 LTP33 LTP29 LTP28 LTP27 LTP26 LTP85 LTP8 LTP40 LTP39 LTP38 LTP32 LTP31 LTP30))

(DEFINST LOOPSTestLispFunc (LTFSendSuper "M]SC@=MN")
       (name #. ($A LTFSendSuper NIL RememberName))
       (TestDesc "_Super: Invoked directly")
       (TestExpr (ANDALL (ATEST (EQ (SEND ($ LSS1)
                                          SendSuper1)
                                    6)
                                "_Super fielded by next higher class")
                         (ATEST (EQ (SEND ($ LSS1)
                                          SendSuper2)
                                    22)
                                "_Super fielded by next higher class, which uses another _Super")
                         (ATEST (EQ (SEND ($ LSS1)
                                          SendSuper3)
                                    31)
                                "_Super fielded by the second super class")
                         (ATEST (EQ (SEND ($ LSS1)
                                          SendSuper4)
                                    6)
                                "_Super fielded by the left super class but the right super class also has the
                                method"))
              FailedExpr #.NotSetValue)
       (SubTest #. ($A (LTP84 LTP83)
                       NIL MakeBackLink)
              Tested? #.NotSetValue Failed #.NotSetValue)
       (PreTest #. ($A (LTFSend)
                       NIL MakeBackLink)
              Tested? #.NotSetValue Failed #.NotSetValue)
       (AltTest #.NotSetValue Tested? #.NotSetValue Failed #.NotSetValue)
       (CasesUsed #. ($A (LSS1)
                         NIL MakeBackLink))
       (SyntaxTest #.NotSetValue Tested? #.NotSetValue Failed #.NotSetValue)
       (UsesObj #. ($A (LOOPSTestSuper LOOPSTestClass2 LOOPSTestClass1 LOOPSTestSS)
                       NIL AllowRemove))
       (LispName SENDSUPER))

(DEFINST LOOPSTestLispFunc (LTFGetInitialValue "M]SC@=MG")
       (name #. ($A LTFGetInitialValue NIL RememberName))
       (TestDesc "GetInitialValue: instance has no values")
       (TestExpr (AND (ATEST (EQUAL (GetInitialValue ($ LTCA6)
                                                     (QUOTE GIV1))
                                    (QUOTE GIV1))
                             "GetInitialValue when class has simple value")
                      (ATEST (EQUAL (GetInitialValue ($ LTCA6)
```

```
                                        (QUOTE GIV2))
                                    6)
                                "GetInitialValue when class has active value"))
              FailedExpr #.NotSetValue)
       (SubTest #. ($A (LTP82)
                       NIL MakeBackLink)
              Tested? #.NotSetValue Failed #.NotSetValue)
       (PreTest #. ($A (LTFGetValue)
                       NIL MakeBackLink)
              Tested? #.NotSetValue Failed #.NotSetValue)
       (AltTest #.NotSetValue Tested? #.NotSetValue Failed #.NotSetValue)
       (CasesUsed #. ($A (LTCA6)
                         NIL MakeBackLink))
       (SyntaxTest #.NotSetValue Tested? #.NotSetValue Failed #.NotSetValue)
       (UsesObj #. ($A (LOOPSTestClass1)
                       NIL AllowRemove)))

(DEFINST LOOPSTestLispFunc (LTFPutNthValue "DCR@@EÅ")
       (name #. ($A LTFPutNthValue NIL RememberName))
       (TestDesc "PutNthValue")
       (SetUp (PROGN (PutNthValue ($ LVC2)
                              1 10)
                     (PutNthValue ($ LVC2)
                              3 30)))
       (TestExpr (AND (EQ (GetNthValue ($ LVC2)
                                  1)
                          10)
                      (EQ (GetNthValue ($ LVC2)
                                  3)
                          30)))
       (PreTest #. ($A (LTFGetNthValue)
                       NIL MakeBackLink))
       (CasesUsed #. ($A (LVC2)
                         NIL MakeBackLink))
       (LispName PutNthValue))

(DEFINST LOOPSTestLispFunc (LTFGetNthValue "DCR@@EÄ")
       (name #. ($A LTFGetNthValue NIL RememberName))
       (TestDesc "GetNthValue")
       (TestExpr (AND (EQ (GetNthValue ($ LVC1)
                                  1)
                          1)
                      (EQ (GetNthValue ($ LVC1)
                                  3)
                          3)
                      (ATEST (EQ (SEND ($ LVC1)
                                       Length)
                                  3)
                             "Length of indexedVars is correct"))
              FailedExpr #.NotSetValue)
       (SubTest #.NotSetValue Tested? #.NotSetValue Failed #.NotSetValue)
       (PreTest #.NotSetValue Tested? #.NotSetValue Failed #.NotSetValue)
       (AltTest #.NotSetValue Tested? #.NotSetValue Failed #.NotSetValue)
       (CasesUsed #. ($A (LVC1)
                         NIL MakeBackLink))
       (SyntaxTest #.NotSetValue Tested? #.NotSetValue Failed #.NotSetValue)
       (UsesObj #. ($A NIL NIL AllowRemove))
       (LispName GetNthValue))

(DEFINST LOOPSTestLispFunc (LTFGetObjectNames "NQR@@Av")
       (name #. ($A LTFGetObjectNames NIL RememberName))
       (TestDesc "GetObjectNames")
       (LispName GetObjectNames))

(DEFINST LOOPSTestLispFunc (LTFtype? "O\R@@[`i")
       (name #. ($A LTFtype? NIL RememberName))
       (TestDesc "type?")
       [TestExpr (AND (type? class ($ LOOPSTestLispFunc))
                      (type? instance ($ LTFtype?))
                      (NULL (type? instance ($ LOOPSTestLispFunc)))
                      (NULL (type? class ($ LTFtype?)))
                      (NULL (type? instance (QUOTE (a b c]
       (SubTest #.NotSetValue Tested? #.NotSetValue)
       (PreTest #.NotSetValue Tested? #.NotSetValue)
       (AltTest #.NotSetValue Tested? #.NotSetValue)
       (SyntaxTest #.NotSetValue Tested? #.NotSetValue)
       (LispName type?))

(DEFINST LOOPSTestLispFunc (LTFClass "O\R@@[`h")
       (name #. ($A LTFClass NIL RememberName))
       (TestDesc "Class function:")
       [TestExpr (AND (EQ (Class ($ LOOPSTestObject))
                          ($ LOOPSTestMeta))
                      (EQ (Class ($ LTFClass))
                          ($ LOOPSTestLispFunc))
                      (EQ (Class ($ LOOPSTestMeta))
                          ($ MetaClass]
```

```
       (SubTest #.NotSetValue Tested? #.NotSetValue)
       (PreTest #.NotSetValue Tested? #.NotSetValue)
       (AltTest #.NotSetValue Tested? #.NotSetValue)
       (SyntaxTest #.NotSetValue Tested? #.NotSetValue)
       (LispName Class))

(DEFINST LOOPSTestLispFunc (LTFGetObjectRec "O\R@@[!")
       (name #. ($A LTFGetObjectRec NIL RememberName))
       (TestDesc "GetObjectRec")
       [TestExpr (AND (EQ (GetObjectRec (QUOTE LOOPSTestObject))
                          ($ LOOPSTestObject))
                      (EQ (GetObjectRec ($ LOOPSTestObject))
                          ($ LOOPSTestObject))
                      (EQ (GetObjectRec (QUOTE LTFGetObjectRec))
                          ($ LTFGetObjectRec))
                      [NULL (GetObjectRec (QUOTE (a b c]
                      (NULL (GetObjectRec (GENSYM]
       (SubTest #.NotSetValue Tested? #.NotSetValue)
       (PreTest #.NotSetValue Tested? #.NotSetValue)
       (AltTest #.NotSetValue Tested? #.NotSetValue)
       (SyntaxTest #.NotSetValue Tested? #.NotSetValue)
       (LispName GetObjectRec))

(DEFINST LOOPSTestLispFunc (LTFReplaceActiveValue "OKR@@E`l")
       (name #. ($A LTFReplaceActiveValue NIL RememberName)))

(DEFINST LOOPSTestLispFunc (LTFPutLocalState "OKR@@E`k")
       (name #. ($A LTFPutLocalState NIL RememberName))
       (TestDesc "PutLocalState")
       (SetUp (PROGN (ATEST (PutLocalState (GetValueOnly ($ LOOPSTestClass6)
                                                         (QUOTE PLS1))
                                           (QUOTE PLS1)
                                           ($ LOOPSTestClass6)
                                           (QUOTE PLS1))
                            "put into local state of active value")
                     (ATEST (PutLocalState (GetValueOnly ($ LOOPSTestClass6)
                                                         (QUOTE PLS3))
                                           (QUOTE PLS3)
                                           ($ LOOPSTestClass6)
                                           (QUOTE PLS3))
                            "put into local state of nested active value")))
       (TestExpr (ANDALL (ATEST (EQUAL (GetLocalState (GetValueOnly ($ LOOPSTestClass6)
                                                                    (QUOTE PLS1))
                                                      ($ LOOPSTestClass6)
                                                      (QUOTE PLS1))
                                       (QUOTE PLS1))
                                "get local state of active value set by PutLocalState")
                         (ATEST (EQUAL (GetLocalState (GetValueOnly ($ LOOPSTestClass6)
                                                                    (QUOTE PLS3))
                                                      ($ LOOPSTestClass6)
                                                      (QUOTE PLS3))
                                       (QUOTE PLS3))
                                "get local state of nested active value set by PutLocalState")))
       (ResetExp (ResetPutLocalStateVars))
       (PreTest #. ($A (LTFGetLocalState)
                       NIL MakeBackLink))
       (UsesObj #. ($A (LOOPSTestClass6)
                       NIL AllowRemove))
       (LispName PutLocalState))

(DEFINST LOOPSTestLispFunc (LTFGetLocalState "OKR@@E`j")
       (name #. ($A LTFGetLocalState NIL RememberName))
       (TestDesc "GetLocalState")
       (TestExpr (AND (ATEST (EQUAL (GetLocalState (GetValueOnly ($ LOOPSTestClass6)
                                                                 (QUOTE GLS1))
                                                   ($ LOOPSTestClass6)
                                                   (QUOTE GLS1))
                                    (QUOTE LTC6))
                             "GetLocalState of active value")
                      (ATEST (EQUAL (GetLocalState (GetValueOnly ($ LOOPSTestClass6)
                                                                 (QUOTE GLS2))
                                                   ($ LOOPSTestClass6)
                                                   (QUOTE GLS2))
                                    (QUOTE LTC6))
                             "GetLocalState of nested active value"))
              FailedExpr #.NotSetValue)
       (SubTest #. ($A (LTP81)
                       NIL MakeBackLink)
              Tested? #.NotSetValue Failed #.NotSetValue)
       (PreTest #. ($A (LTFGetValueOnly)
                       NIL MakeBackLink)
              Tested? #.NotSetValue Failed #.NotSetValue)
       (AltTest #.NotSetValue Tested? #.NotSetValue Failed #.NotSetValue)
       (SyntaxTest #.NotSetValue Tested? #.NotSetValue Failed #.NotSetValue)
       (UsesObj #. ($A (LOOPSTestCases LOOPSTestClass6)
                       NIL AllowRemove))
       (LispName GetLocalState))
```

```
(DEFINST LOOPSTestLispFunc (LTFRenameMethodFunction "OKR@@E‘i")
       (name #. ($A LTFRenameMethodFunction NIL RememberName)))

(DEFINST LOOPSTestLispFunc (LTFMoveMethod "OKR@@E‘h")
       (name #. ($A LTFMoveMethod NIL RememberName))
       (TestDesc "Move Method across two classes")
       (SetUp (MoveMethod (QUOTE LOOPSTestClass1)
                     (QUOTE LOOPSTestClass2)
                     (QUOTE MMTest1)))
       [TestExpr (AND [FMEMB (QUOTE LOOPSTestClass2.MMTest1)
                              (CalledFns (QUOTE (LOOPSTestClass2]
                       (NOT (FMEMB (QUOTE LOOPSTestClass1.MMTest1)
                                   (CalledFns (QUOTE (LOOPSTestClass1]
       (ResetExp (MoveMethod (QUOTE LOOPSTestClass2)
                         (QUOTE LOOPSTestClass1)
                         (QUOTE MMTest1)))
       (SubTest #. ($A (LTP2)
                       NIL MakeBackLink)
              Tested? #.NotSetValue)
       (PreTest #. ($A (LTFCalledFns)
                       NIL MakeBackLink)
              Tested? #.NotSetValue)
       (AltTest #.NotSetValue Tested? #.NotSetValue)
       (CasesUsed #. ($A (LTC3)
                         NIL MakeBackLink))
       (SyntaxTest #.NotSetValue Tested? #.NotSetValue)
       (UsesObj #. ($A (LOOPSTestClass1 LOOPSTestClass2)
                       NIL AllowRemove))
       (LispName MoveMethod))

(DEFINST LOOPSTestLispFunc (LTFMoveVariable "OKR@@EÓ")
       (name #. ($A LTFMoveVariable NIL RememberName))
       (TestDesc "Move IV across classes")
       (SetUp (MoveVariable (QUOTE LOOPSTestClass2)
                     (QUOTE LOOPSTestClass3)
                     (QUOTE MVTest1)))
       [TestExpr (AND (FMEMB (QUOTE MVTest1)
                              (SEND ($ LOOPSTestClass3)
                                    List
                                    (QUOTE IVs)))
                       (NOT (FMEMB (QUOTE MVTest1)
                                   (SEND ($ LOOPSTestClass2)
                                         List
                                         (QUOTE IVs]
       (ResetExp (MoveVariable (QUOTE LOOPSTestClass3)
                         (QUOTE LOOPSTestClass2)
                         (QUOTE MVTest1)))
       (SubTest #. ($A (LTP7 LTP6)
                       NIL MakeBackLink)
              Tested? #.NotSetValue)
       (PreTest #. ($A (LTMListInClass)
                       NIL MakeBackLink)
              Tested? #.NotSetValue)
       (AltTest #.NotSetValue Tested? #.NotSetValue)
       (SyntaxTest #.NotSetValue Tested? #.NotSetValue)
       (UsesObj #. ($A (LOOPSTestClass3 LOOPSTestClass2)
                       NIL AllowRemove))
       (LispName MoveVariable))

(DEFINST LOOPSTestLispFunc (LTFMoveClassVariable "OKR@@EÑ")
       (name #. ($A LTFMoveClassVariable NIL RememberName)))

(DEFINST LOOPSTestLispFunc (LTFRenameVariable "OKR@@EÏ")
       (name #. ($A LTFRenameVariable NIL RememberName))
       (TestDesc "Rename a IV in a class")
       (SetUp (RenameVariable (QUOTE LOOPSTestClass1)
                     (QUOTE RVTest1)
                     (QUOTE RVTest2)))
       [TestExpr (AND (FMEMB (QUOTE RVTest2)
                              (SEND ($ LOOPSTestClass1)
                                    List
                                    (QUOTE IVs)))
                       (NOT (FMEMB (QUOTE RVTest1)
                                   (SEND ($ LOOPSTestClass1)
                                         List
                                         (QUOTE IVs]
       (ResetExp (RenameVariable (QUOTE LOOPSTestClass1)
                         (QUOTE RVTest2)
                         (QUOTE RVTest1)))
       (SubTest #. ($A (LTP3)
                       NIL MakeBackLink)
              Tested? #.NotSetValue)
       (PreTest #. ($A (LTMListInClass)
                       NIL MakeBackLink)
              Tested? #.NotSetValue)
       (AltTest #.NotSetValue Tested? #.NotSetValue)
```

```
       (SyntaxTest #.NotSetValue Tested? #.NotSetValue)
       (UsesObj #. ($A (LOOPSTestClass1)
                       NIL AllowRemove))
       (StandBy (LTP4 LTP5))
       (LispName RenameVariable))

(DEFINST LOOPSTestLispFunc (LTFCalledFns "OGR@@E ")
       (name #. ($A LTFCalledFns NIL RememberName))
       (TestDesc "CalledFns")
       [TestExpr (EQUAL (CalledFns (QUOTE (LOOPSTestClassCF)))
                        (QUOTE (LOOPSTestClassCF.CF1 LOOPSTestClassCF.CF2]
       (SubTest #.NotSetValue Tested? #.NotSetValue)
       (PreTest #.NotSetValue Tested? #.NotSetValue)
       (AltTest #.NotSetValue Tested? #.NotSetValue)
       (SyntaxTest #.NotSetValue Tested? #.NotSetValue)
       (UsesObj #. ($A (LOOPSTestClassCF)
                       NIL AllowRemove))
       (LispName CalledFns))

(DEFINST LOOPSTestLispFunc (LTFRenameMethod "OGR@@E ")
       (name #. ($A LTFRenameMethod NIL RememberName))
       (TestDesc "renaming of existing methods")
       (SetUp (RenameMethod (QUOTE LOOPSTestCases)
                     (QUOTE RMTest1)
                     (QUOTE RMTest2)))
       [TestExpr (AND [FMEMB (QUOTE LOOPSTestCases.RMTest2)
                             (CalledFns (QUOTE (LOOPSTestCases]
                      (NOT (FMEMB (QUOTE LOOPSTestCases.RMTest1)
                                  (CalledFns (QUOTE (LOOPSTestCases]
       (ResetExp (RenameMethod (QUOTE LOOPSTestCases)
                        (QUOTE RMTest2)
                        (QUOTE RMTest1)))
       (SubTest #.NotSetValue Tested? #.NotSetValue)
       (PreTest #. ($A (LTFCalledFns)
                       NIL MakeBackLink)
              Tested? #.NotSetValue)
       (AltTest #.NotSetValue Tested? #.NotSetValue)
       (SyntaxTest #.NotSetValue Tested? #.NotSetValue)
       (UsesObj #. ($A (LOOPSTestCases)
                       NIL AllowRemove))
       (LispName RenameMethod))

(DEFINST LOOPSTestLispFunc (LTFTryMethod "OER@@E G")
       (name #. ($A LTFTryMethod NIL RememberName)))

(DEFINST LOOPSTestLispFunc (LTFDoMethod "OER@@E F")
       (name #. ($A LTFDoMethod NIL RememberName)))

(DEFINST LOOPSTestLispFunc (LTFPutItOnly "OER@@E E")
       (name #. ($A LTFPutItOnly NIL RememberName)))

(DEFINST LOOPSTestLispFunc (LTFGetItOnly "OER@@E D")
       (name #. ($A LTFGetItOnly NIL RememberName)))

(DEFINST LOOPSTestLispFunc (LTFPutIt "OER@@E C")
       (name #. ($A LTFPutIt NIL RememberName)))

(DEFINST LOOPSTestLispFunc (LTFGetIt "OER@@E B")
       (name #. ($A LTFGetIt NIL RememberName)))

(DEFINST LOOPSTestLispFunc (LTFPutMethodOnly "OER@@E A")
       (name #. ($A LTFPutMethodOnly NIL RememberName)))

(DEFINST LOOPSTestLispFunc (LTFGetMethodOnly "OER@@E @")
       (name #. ($A LTFGetMethodOnly NIL RememberName)))

(DEFINST LOOPSTestLispFunc (LTFPutClassOnly "OER@@E =")
       (name #. ($A LTFPutClassOnly NIL RememberName)))

(DEFINST LOOPSTestLispFunc (LTFGetClassOnly "OER@@E <")
       (name #. ($A LTFGetClassOnly NIL RememberName)))

(DEFINST LOOPSTestLispFunc (LTFPutClass "OER@@E ;")
       (name #. ($A LTFPutClass NIL RememberName))
       (TestDesc "PutClass")
       (LispName PutClass))

(DEFINST LOOPSTestLispFunc (LTFGetClass "OER@@E :")
       (name #. ($A LTFGetClass NIL RememberName)))

(DEFINST LOOPSTestLispFunc (LTFPutClassValueOnly "OER@@E 9")
       (name #. ($A LTFPutClassValueOnly NIL RememberName)))

(DEFINST LOOPSTestLispFunc (LTFPutValueOnly "OER@@E 8")
       (name #. ($A LTFPutValueOnly NIL RememberName))
       (TestDesc "PutValueOnly")
       (SetUp (PROGN (ATEST (PutValueOnly ($ LTC9)
```

```
                                    (QUOTE Var1)
                                    7)
                             "put ordinary value")
                      (ATEST (PutValueOnly ($ LTC9)
                                    (QUOTE Test1)
                                    (create activeValue localState_5
                                           getFn_NIL
                                           putFn_
                                           (QUOTE MakeBackLink)))
                             "put active value")))
       (TestExpr (ANDALL (ATEST (EQ (GetValueOnly ($ LTC9)
                                          (QUOTE Var1))
                                    7)
                                "get the ordinary value put by PutValueOnly")
                         (ATEST (EQACTVAL (GetValueOnly ($ LTC9)
                                                 (QUOTE Test1))
                                          (QUOTE (5 NIL MakeBackLink)))
                                "get the active value put by PutValueOnly")))
       (ResetExp (PROGN (PutValueOnly ($ LTC9)
                                (QUOTE Var1)
                                NotSetValue)
                        (PutValueOnly ($ LTC9)
                                (QUOTE Test1)
                                NotSetValue)))
       (PreTest #. ($A (LTFGetValueOnly)
                       NIL MakeBackLink))
       (CasesUsed #. ($A (LTC9)
                         NIL MakeBackLink))
       (LispName PutValueOnly))

(DEFINST LOOPSTestLispFunc (LTFPushClassValue "OER@@E 7")
       (name #. ($A LTFPushClassValue NIL RememberName)))

(DEFINST LOOPSTestLispFunc (LTFPutClassValue "OER@@E 6")
       (name #. ($A LTFPutClassValue NIL RememberName))
       (TestDesc "PutClassValue: in a class")
       (SetUp (PutClassValue ($ LOOPSTestClass3)
                      (QUOTE PCV1)
                      (QUOTE PCV)))
       (TestExpr (EQUAL (GetClassValue ($ LOOPSTestClass3)
                                (QUOTE PCV1))
                        (QUOTE PCV)))
       (ResetExp (PutClassValue ($ LOOPSTestClass3)
                        (QUOTE PCV1)
                        NIL))
       (SubTest #. ($A (LTP37 LTP36 LTP35 LTP34 LTP33)
                       NIL MakeBackLink)
              Tested? #.NotSetValue)
       (PreTest #. ($A (LTFGetClassValue)
                       NIL MakeBackLink)
              Tested? #.NotSetValue)
       (AltTest #.NotSetValue Tested? #.NotSetValue)
       (SyntaxTest #.NotSetValue Tested? #.NotSetValue)
       (UsesObj #. ($A (LOOPSTestClass3)
                       NIL AllowRemove))
       (LispName PutClassValue))

(DEFINST LOOPSTestLispFunc (LTFAddValue "OER@@E 5")
       (name #. ($A LTFAddValue NIL RememberName)))

(DEFINST LOOPSTestLispFunc (LTFPushValue "OER@@E 3")
       (name #. ($A LTFPushValue NIL RememberName)))

(DEFINST LOOPSTestLispFunc (LTFGetClassValueOnly "OER@@E 1")
       (name #. ($A LTFGetClassValueOnly NIL RememberName)))

(DEFINST LOOPSTestLispFunc (LTFGetValueOnly "OER@@E 0")
       (name #. ($A LTFGetValueOnly NIL RememberName))
       (TestDesc "GetValueOnly")
       (TestExpr (ANDALL (ATEST (EQUAL (GetValueOnly ($ LTC8)
                                              (QUOTE Var1))
                                       5)
                                "getting ordinary value")
                         (ATEST (EQACTVAL (GetValueOnly ($ LTC8)
                                                 (QUOTE Test1))
                                          (QUOTE (NIL NIL MakeBackLink)))
                                "getting active value"))
              FailedExpr #.NotSetValue)
       (SubTest #.NotSetValue Tested? #.NotSetValue Failed #.NotSetValue)
       (PreTest #.NotSetValue Tested? #.NotSetValue Failed #.NotSetValue)
       (AltTest #.NotSetValue Tested? #.NotSetValue Failed #.NotSetValue)
       (CasesUsed #. ($A (LTC8)
                         NIL MakeBackLink))
       (SyntaxTest #.NotSetValue Tested? #.NotSetValue Failed #.NotSetValue)
       (LispName GetValueOnly))

(DEFINST LOOPSTestLispFunc (LTFGetClassValue "OER@@E /")
```

```
       (name #. ($A LTFGetClassValue NIL RememberName))
       (TestDesc "GetClassValue: for a class")
       (TestExpr (EQUAL (GetClassValue ($ LOOPSTestCases)
                                (QUOTE GCV1))
                        (QUOTE GCV)))
       (SubTest #. ($A (LTP29 LTP28 LTP27 LTP26)
                       NIL MakeBackLink)
              Tested? #.NotSetValue)
       (PreTest #.NotSetValue Tested? #.NotSetValue)
       (AltTest #.NotSetValue Tested? #.NotSetValue)
       (SyntaxTest #.NotSetValue Tested? #.NotSetValue)
       (UsesObj #. ($A (LOOPSTestCases)
                       NIL AllowRemove))
       (LispName GetClassValue))

(DEFINST LOOPSTestLispFunc (LTFDM "OER@@E .")
       (name #. ($A LTFDM NIL RememberName)))

(DEFINST LOOPSTestLispFunc (LTFDC "OER@@E -")
       (name #. ($A LTFDC NIL RememberName))
       (TestDesc "DC(Define Class)")
       (LispName DC))

(DEFINST LOOPSTestPrimitive (LTP84 "M]SC@=MZ")
       (name #. ($A LTP84 NIL RememberName))
       (TestDesc "Cascaded _Super: Invoked directly")
       (SetUp (ATEST (SEND ($ LSS1)
                           SendSuper5
                           (QUOTE ssVar1))
                     "Invoke the cascaded _Super scheme"))
       (TestExpr (EQ (GetValue ($ LSS1)
                               (QUOTE ssVar1))
                     30)
              FailedExpr #.NotSetValue)
       (ResetExp (PutValue ($ LSS1)
                           (QUOTE ssVar1)
                           0))
       (SubTest #.NotSetValue Tested? #.NotSetValue Failed #.NotSetValue)
       (PreTest #. ($A NIL BuildPreTest MakeBackLink)
              Tested? #.NotSetValue Failed #.NotSetValue)
       (AltTest #.NotSetValue Tested? #.NotSetValue Failed #.NotSetValue)
       (CasesUsed #. ($A (LSS1)
                         NIL MakeBackLink))
       (SyntaxTest #.NotSetValue Tested? #.NotSetValue Failed #.NotSetValue)
       (UsesObj #. ($A (LOOPSTestCases LOOPSTestClass2 LOOPSTestClass1 LOOPSTestSS)
                       NIL AllowRemove))
       (SubTestOf #. ($A (LTFSendSuper)
                         NIL MakeBackLink)))

(DEFINST LOOPSTestPrimitive (LTP83 "M]SC@=MO")
       (name #. ($A LTP83 NIL RememberName))
       (TestDesc "_Super: invoked from the super of a class")
       (TestExpr (ANDALL (ATEST (EQ (SEND ($ LSS2)
                                          SendSuper1)
                                    6)
                                "_Super fielded by next higher class")
                         (ATEST (EQ (SEND ($ LSS2)
                                          SendSuper2)
                                    22)
                                "_Super fielded by next higher class, which uses another _Super")
                         (ATEST (EQ (SEND ($ LSS2)
                                          SendSuper3)
                                    31)
                                "_Super fielded by the second super class")
                         (ATEST (EQ (SEND ($ LSS2)
                                          SendSuper4)
                                    6)
                                "_Super fielded by the left super class but the right super class also has the
                                method"))
              FailedExpr #.NotSetValue)
       (SubTest #. ($A (LTP85)
                       NIL MakeBackLink)
              Tested? #.NotSetValue Failed #.NotSetValue)
       (PreTest #.NotSetValue Tested? #.NotSetValue Failed #.NotSetValue)
       (AltTest #.NotSetValue Tested? #.NotSetValue Failed #.NotSetValue)
       (CasesUsed #. ($A (LSS2)
                         NIL MakeBackLink))
       (SyntaxTest #.NotSetValue Tested? #.NotSetValue Failed #.NotSetValue)
       (UsesObj #. ($A (LOOPSTestSuper LOOPSTestClass2 LOOPSTestClass1 LOOPSTestSS)
                       NIL AllowRemove))
       (SubTestOf #. ($A (LTFSendSuper)
                         NIL MakeBackLink)))

(DEFINST LOOPSTestPrimitive (LTP82 "M]SC@=MJ")
       (name #. ($A LTP82 NIL RememberName))
       (TestDesc "GetInitialValue: instance has values different from initial ones")
       (TestExpr (AND (ATEST (EQUAL (GetInitialValue ($ LTCA7)
```

```
                                          (QUOTE GIV1))
                                   (QUOTE GIV1))
                            "GetInitialValue: simple value in class")
                     (ATEST (EQUAL (GetInitialValue ($ LTCA7)
                                          (QUOTE GIV2))
                                   6)
                            "GetInitialValue: active value in class"))
              FailedExpr #.NotSetValue)
       (SubTest #.NotSetValue Tested? #.NotSetValue Failed #.NotSetValue)
       (PreTest #.NotSetValue Tested? #.NotSetValue Failed #.NotSetValue)
       (AltTest #.NotSetValue Tested? #.NotSetValue Failed #.NotSetValue)
       (CasesUsed #. ($A (LTCA7)
                         NIL MakeBackLink))
       (SyntaxTest #.NotSetValue Tested? #.NotSetValue Failed #.NotSetValue)
       (UsesObj #. ($A (LOOPSTestClass1)
                       NIL AllowRemove))
       (SubTestOf #. ($A (LTFGetInitialValue)
                         NIL MakeBackLink)))

(DEFINST LOOPSTestPrimitive (LTP81 "M]SC@=ME")
       (name #. ($A LTP81 NIL RememberName))
       (TestDesc "GetLocalState: NotSetValue")
       (TestExpr (ATEST (EQUAL (GetLocalState (GetValueOnly ($ LTCF3)
                                                            (QUOTE GLS3))
                                              ($ LTCF3)
                                              (QUOTE GLS3))
                               (QUOTE GLS3))
                        "NotSetValue as local state inherits from class")
              FailedExpr #.NotSetValue)
       (SubTest #.NotSetValue Tested? #.NotSetValue Failed #.NotSetValue)
       (PreTest #.NotSetValue Tested? #.NotSetValue Failed #.NotSetValue)
       (AltTest #.NotSetValue Tested? #.NotSetValue Failed #.NotSetValue)
       (CasesUsed #. ($A (LTCF3)
                         NIL MakeBackLink))
       (SyntaxTest #.NotSetValue Tested? #.NotSetValue Failed #.NotSetValue)
       (UsesObj #. ($A (LOOPSTestClass6)
                       NIL AllowRemove))
       (SubTestOf #. ($A (LTFGetLocalState)
                         NIL MakeBackLink)))

(DEFINST LOOPSTestPrimitive (LTP2 "OLR@@EÁ")
       (name #. ($A LTP2 NIL RememberName))
       (TestDesc "MoveMethod up the class hierarchy")
       (SetUp (MoveMethod (QUOTE LOOPSTestClass1)
                      (QUOTE LOOPSTestSuper)
                      (QUOTE MMTest2)))
       [TestExpr (AND [FMEMB (QUOTE LOOPSTestSuper.MMTest2)
                             (CalledFns (QUOTE (LOOPSTestSuper]
                      (NOT (FMEMB (QUOTE LOOPSTestClass1.MMTest2)
                                  (CalledFns (QUOTE (LOOPSTestClass1]
       (ResetExp (MoveMethod (QUOTE LOOPSTestSuper)
                         (QUOTE LOOPSTestClass1)
                         (QUOTE MMTest2)))
       (SubTest #.NotSetValue Tested? #.NotSetValue)
       (PreTest #. ($A (LTFCalledFns)
                       BuildPreTest MakeBackLink)
              Tested? #.NotSetValue)
       (AltTest #.NotSetValue Tested? #.NotSetValue)
       (SyntaxTest #.NotSetValue Tested? #.NotSetValue)
       (UsesObj #. ($A (LOOPSTestSuper LOOPSTestClass1)
                       NIL AllowRemove))
       (SubTestOf #. ($A (LTFMoveMethod)
                         NIL MakeBackLink)))

(DEFINST LOOPSTestPrimitive (LTP7 "OMR@@A`|")
       (name #. ($A LTP7 NIL RememberName))
       (TestDesc "MoveVariable up the hierarchy")
       (SetUp (MoveVariable (QUOTE LOOPSTestClass2)
                      (QUOTE LOOPSTestCases)
                      (QUOTE MVTest3)))
       [TestExpr (AND (FMEMB (QUOTE MVTest3)
                             (SEND ($ LOOPSTestCases)
                                   List
                                   (QUOTE IVs)))
                      (NOT (FMEMB (QUOTE MVTest3)
                                  (SEND ($ LOOPSTestClass2)
                                        List
                                        (QUOTE IVs]
       (ResetExp (MoveVariable (QUOTE LOOPSTestCases)
                         (QUOTE LOOPSTestClass2)
                         (QUOTE MVTest3)))
       (SubTest #. ($A (LTP8)
                       NIL MakeBackLink)
              Tested? #.NotSetValue)
       (PreTest #. ($A (LTMListInClass)
                       BuildPreTest MakeBackLink)
              Tested? #.NotSetValue)
```

```
       (AltTest #.NotSetValue Tested? #.NotSetValue)
       (SyntaxTest #.NotSetValue Tested? #.NotSetValue)
       (UsesObj #. ($A (LOOPSTestCases LOOPSTestClass2)
                       NIL AllowRemove))
       (SubTestOf #. ($A (LTFMoveVariable)
                         NIL MakeBackLink)))

(DEFINST LOOPSTestPrimitive (LTP6 "OMR@@A‘{")
       (name #. ($A LTP6 NIL RememberName))
       (TestDesc "if MoveVariable moves the props too")
       (SetUp (MoveVariable (QUOTE LOOPSTestClass2)
                     (QUOTE LOOPSTestClass3)
                     (QUOTE MVTest2)))
       [TestExpr (AND (EQUAL (QUOTE (Prop1 Prop2 doc))
                             (SEND ($ LOOPSTestClass3)
                                   List
                                   (QUOTE IVProps)
                                   (QUOTE MVTest2)))
                      (NOT (FMEMB (QUOTE MVTest2)
                                  (SEND ($ LOOPSTestClass2)
                                        List
                                        (QUOTE IVs]
       (ResetExp (MoveVariable (QUOTE LOOPSTestClass3)
                         (QUOTE LOOPSTestClass2)
                         (QUOTE MVTest2)))
       (SubTest #.NotSetValue Tested? #.NotSetValue)
       (PreTest #. ($A (LTMListInClass)
                       BuildPreTest MakeBackLink)
              Tested? #.NotSetValue)
       (AltTest #.NotSetValue Tested? #.NotSetValue)
       (SyntaxTest #.NotSetValue Tested? #.NotSetValue)
       (UsesObj #. ($A (LOOPSTestClass3 LOOPSTestClass2)
                       NIL AllowRemove))
       (SubTestOf #. ($A (LTFMoveVariable)
                         NIL MakeBackLink)))

(DEFINST LOOPSTestPrimitive (LTP3 "OMR@@A‘x")
       (name #. ($A LTP3 NIL RememberName))
       (TestDesc "Rename a CV in a class")
       (SetUp (RenameVariable (QUOTE LOOPSTestClass1)
                     (QUOTE RVTest3)
                     (QUOTE RVTest4)
                     T))
       [TestExpr (AND (FMEMB (QUOTE RVTest4)
                             (SEND ($ LOOPSTestClass1)
                                   List
                                   (QUOTE CVs)))
                      (NOT (FMEMB (QUOTE RVTest3)
                                  (SEND ($ LOOPSTestClass1)
                                        List
                                        (QUOTE CVs]
       (ResetExp (RenameVariable (QUOTE LOOPSTestClass1)
                         (QUOTE RVTest4)
                         (QUOTE RVTest3)
                         T))
       (PreTest #. ($A (LTMListInClass)
                       BuildPreTest MakeBackLink))
       (UsesObj #. ($A (LOOPSTestClass1)
                       NIL AllowRemove))
       (SubTestOf #. ($A (LTFRenameVariable)
                         NIL MakeBackLink)))

(DEFINST LOOPSTestPrimitive (LTP37 "OTR@@AÌg")
       (name #. ($A LTP37 NIL RememberName))
       (TestDesc "PutClassValue: in a inherited prop, but CV is local")
       (SetUp (PutClassValue ($ LOOPSTestClass2)
                     (QUOTE PCV3)
                     (QUOTE LTP37)
                     (QUOTE Prop1)))
       [TestExpr (AND (EQUAL (GetClassValue ($ LOOPSTestClass2)
                                    (QUOTE PCV3)
                                    (QUOTE Prop1))
                             (QUOTE LTP37))
                      (EQUAL (GetClassValue ($ LOOPSTestCases)
                                    (QUOTE PCV3)
                                    (QUOTE Prop1))
                             (QUOTE LTC]
       [ResetExp (PROGN (PutClassValue ($ LOOPSTestCases)
                                (QUOTE PCV3)
                                (QUOTE LTC)
                                (QUOTE Prop1))
                        (PutClassValue ($ LOOPSTestClass2)
                                (QUOTE PCV3)
                                NotSetValue
                                (QUOTE Prop1]
       (SubTest #.NotSetValue Tested? #.NotSetValue)
       (PreTest #.NotSetValue Tested? #.NotSetValue)
```

```
       (AltTest #.NotSetValue Tested? #.NotSetValue)
       (SyntaxTest #.NotSetValue Tested? #.NotSetValue)
       (UsesObj #. ($A (LOOPSTestCases LOOPSTestClass2)
                       NIL AllowRemove))
       (SubTestOf #. ($A (LTFPutClassValue)
                         NIL MakeBackLink)))

(DEFINST LOOPSTestPrimitive (LTP36 "OTR@@AÌf")
       (name #. ($A LTP36 NIL RememberName))
       (TestDesc "PutClassValue: inherited CV in class")
       (SetUp (PutClassValue ($ LOOPSTestClass3)
                     (QUOTE PCV3)
                     (QUOTE LTP36)))
       [TestExpr (AND (EQUAL (GetClassValue ($ LOOPSTestClass3)
                                    (QUOTE PCV3))
                             (QUOTE LTP36))
                      (EQUAL (GetClassValue ($ LOOPSTestCases)
                                    (QUOTE PCV3))
                             (QUOTE LTP36]
       (ResetExp (PutClassValue ($ LOOPSTestCases)
                         (QUOTE PCV3)
                         (QUOTE LTC)))
       (SubTest #. ($A (LTP40)
                       NIL MakeBackLink)
              Tested? #.NotSetValue)
       (PreTest #.NotSetValue Tested? #.NotSetValue)
       (AltTest #.NotSetValue Tested? #.NotSetValue)
       (SyntaxTest #.NotSetValue Tested? #.NotSetValue)
       (UsesObj #. ($A (LOOPSTestClass3 LOOPSTestCases)
                       NIL AllowRemove))
       (SubTestOf #. ($A (LTFPutClassValue)
                         NIL MakeBackLink)))

(DEFINST LOOPSTestPrimitive (LTP35 "OTR@@AÌe")
       (name #. ($A LTP35 NIL RememberName))
       (TestDesc "PutClassValue: local over inherited")
       (SetUp (PutClassValue ($ LOOPSTestClass3)
                     (QUOTE PCV2)
                     (QUOTE LTP35)))
       (TestExpr (EQUAL (GetClassValue ($ LOOPSTestClass3)
                               (QUOTE PCV2))
                        (QUOTE LTP35)))
       (ResetExp (PutClassValue ($ LOOPSTestClass3)
                         (QUOTE PCV2)
                         (QUOTE LTC3)))
       (SubTest #. ($A (LTP39)
                       NIL MakeBackLink)
              Tested? #.NotSetValue)
       (PreTest #.NotSetValue Tested? #.NotSetValue)
       (AltTest #.NotSetValue Tested? #.NotSetValue)
       (SyntaxTest #.NotSetValue Tested? #.NotSetValue)
       (UsesObj #. ($A (LOOPSTestClass3 LOOPSTestCases)
                       NIL AllowRemove))
       (SubTestOf #. ($A (LTFPutClassValue)
                         NIL MakeBackLink)))

(DEFINST LOOPSTestPrimitive (LTP34 "OTR@@AÌd")
       (name #. ($A LTP34 NIL RememberName))
       (TestDesc "PutClassValue: in an instance")
       (SetUp (PutClassValue ($ LTCC1)
                     (QUOTE PCV1)
                     (QUOTE LTP34)))
       (TestExpr (EQUAL (GetClassValue ($ LTCC1)
                               (QUOTE PCV1))
                        (QUOTE LTP34)))
       (ResetExp (PutClassValue ($ LTCC1)
                         (QUOTE PCV1)
                         NIL))
       (SubTest #. ($A (LTP38)
                       NIL MakeBackLink)
              Tested? #.NotSetValue)
       (PreTest #.NotSetValue Tested? #.NotSetValue)
       (AltTest #.NotSetValue Tested? #.NotSetValue)
       (CasesUsed #. ($A (LTCC1)
                         NIL MakeBackLink))
       (SyntaxTest #.NotSetValue Tested? #.NotSetValue)
       (UsesObj #. ($A (LOOPSTestClass3)
                       NIL AllowRemove))
       (SubTestOf #. ($A (LTFPutClassValue)
                         NIL MakeBackLink)))

(DEFINST LOOPSTestPrimitive (LTP33 "OTR@@AÌc")
       (name #. ($A LTP33 NIL RememberName))
       (TestDesc "PutClassValue: local prop")
       (SetUp (PutClassValue ($ LOOPSTestClass3)
                     (QUOTE PCV1)
                     (QUOTE LTP33)
```

```
                    (QUOTE Prop1)))
       (TestExpr (EQUAL (GetClassValue ($ LOOPSTestClass3)
                               (QUOTE PCV1)
                               (QUOTE Prop1))
                        (QUOTE LTP33)))
       (ResetExp (PutClassValue ($ LOOPSTestClass3)
                        (QUOTE PCV1)
                        NIL
                        (QUOTE Prop1)))
       (SubTest #.NotSetValue Tested? #.NotSetValue)
       (PreTest #.NotSetValue Tested? #.NotSetValue)
       (AltTest #.NotSetValue Tested? #.NotSetValue)
       (SyntaxTest #.NotSetValue Tested? #.NotSetValue)
       (UsesObj #. ($A (LOOPSTestClass3)
                       NIL AllowRemove))
       (SubTestOf #. ($A (LTFPutClassValue)
                         NIL MakeBackLink)))

(DEFINST LOOPSTestPrimitive (LTP29 "OTR@@E-")
       (name #. ($A LTP29 NIL RememberName))
       (TestDesc "GetClassValue: local over inherited")
       (TestExpr (EQUAL (GetClassValue ($ LOOPSTestClass1)
                               (QUOTE GCV2))
                        (QUOTE LTP29)))
       (SubTest #. ($A (LTP32)
                       NIL MakeBackLink)
              Tested? #.NotSetValue)
       (PreTest #.NotSetValue Tested? #.NotSetValue)
       (AltTest #.NotSetValue Tested? #.NotSetValue)
       (SyntaxTest #.NotSetValue Tested? #.NotSetValue)
       (UsesObj #. ($A (LOOPSTestCases LOOPSTestClass1)
                       NIL AllowRemove))
       (SubTestOf #. ($A (LTFGetClassValue)
                         NIL MakeBackLink)))

(DEFINST LOOPSTestPrimitive (LTP28 "OTR@@EÕ")
       (name #. ($A LTP28 NIL RememberName))
       (TestDesc "GetClassValue: inherit in class")
       (TestExpr (EQUAL (GetClassValue ($ LOOPSTestClass1)
                               (QUOTE GCV1))
                        (QUOTE GCV)))
       (SubTest #. ($A (LTP31)
                       NIL MakeBackLink)
              Tested? #.NotSetValue)
       (PreTest #.NotSetValue Tested? #.NotSetValue)
       (AltTest #.NotSetValue Tested? #.NotSetValue)
       (SyntaxTest #.NotSetValue Tested? #.NotSetValue)
       (UsesObj #. ($A (LOOPSTestClass1 LOOPSTestCases)
                       NIL AllowRemove))
       (SubTestOf #. ($A (LTFGetClassValue)
                         NIL MakeBackLink)))

(DEFINST LOOPSTestPrimitive (LTP27 "OTR@@EÒ")
       (name #. ($A LTP27 NIL RememberName))
       (TestDesc "GetClassValue: prop from a class")
       (TestExpr (EQUAL (GetClassValue ($ LOOPSTestCases)
                               (QUOTE GCV1)
                               (QUOTE Prop1))
                        (QUOTE GCV1)))
       (SubTest #.NotSetValue Tested? #.NotSetValue)
       (PreTest #.NotSetValue Tested? #.NotSetValue)
       (AltTest #.NotSetValue Tested? #.NotSetValue)
       (SyntaxTest #.NotSetValue Tested? #.NotSetValue)
       (UsesObj #. ($A (LOOPSTestCases)
                       NIL AllowRemove))
       (SubTestOf #. ($A (LTFGetClassValue)
                         NIL MakeBackLink)))

(DEFINST LOOPSTestPrimitive (LTP26 "OTR@@E—")
       (name #. ($A LTP26 NIL RememberName))
       (TestDesc "GetClassValue: for an instance")
       (TestExpr (EQUAL (GetClassValue ($ LTC7)
                               (QUOTE GCV1))
                        (QUOTE GCV)))
       (SubTest #. ($A (LTP30)
                       NIL MakeBackLink)
              Tested? #.NotSetValue)
       (PreTest #.NotSetValue Tested? #.NotSetValue)
       (AltTest #.NotSetValue Tested? #.NotSetValue)
       (CasesUsed #. ($A (LTC7)
                         NIL MakeBackLink))
       (SyntaxTest #.NotSetValue Tested? #.NotSetValue)
       (UsesObj #. ($A (LOOPSTestCases)
                       NIL AllowRemove))
       (SubTestOf #. ($A (LTFGetClassValue)
                         NIL MakeBackLink)))
```

```
(DEFINST LOOPSTestPrimitive (LTP85 "M]SC@=M[")
       (name #. ($A LTP85 NIL RememberName))
       (TestDesc "Cascaded _Super: invoked indirectly")
       (SetUp (ATEST (SEND ($ LSS2)
                           SendSuper5
                           (QUOTE ssVar1))
                     "Invoke the cascaded _Super scheme"))
       (TestExpr (EQ (GetValue ($ LSS2)
                               (QUOTE ssVar1))
                     30)
             FailedExpr #.NotSetValue)
       (ResetExp (PutValue ($ LSS2)
                           (QUOTE ssVar1)
                           0))
       (SubTest #.NotSetValue Tested? #.NotSetValue Failed #.NotSetValue)
       (PreTest #. ($A NIL BuildPreTest MakeBackLink)
             Tested? #.NotSetValue Failed #.NotSetValue)
       (AltTest #.NotSetValue Tested? #.NotSetValue Failed #.NotSetValue)
       (CasesUsed #. ($A (LSS2)
                         NIL MakeBackLink))
       (SyntaxTest #.NotSetValue Tested? #.NotSetValue Failed #.NotSetValue)
       (UsesObj #. ($A (LOOPSTestCases LOOPSTestClass2 LOOPSTestClass1 LOOPSTestSS LOOPSTestSS2)
                       NIL AllowRemove))
       (SubTestOf #. ($A (LTP83)
                         NIL MakeBackLink)))

(DEFINST LOOPSTestPrimitive (LTP8 "ONR@@AÌ")
       (name #. ($A LTP8 NIL RememberName))
       (TestDesc "if Move Variable moves props up the hierarchy")
       (SetUp (MoveVariable (QUOTE LOOPSTestClass2)
                     (QUOTE LOOPSTestCases)
                     (QUOTE MVTest4)))
       [TestExpr (AND (EQUAL (QUOTE (Prop1 Prop2 doc))
                             (SEND ($ LOOPSTestCases)
                                   List
                                   (QUOTE IVProps)
                                   (QUOTE MVTest4)))
                      (NOT (FMEMB (QUOTE MVTest4)
                                  (SEND ($ LOOPSTestClass2)
                                        List
                                        (QUOTE IVs]
       (ResetExp (MoveVariable (QUOTE LOOPSTestCases)
                         (QUOTE LOOPSTestClass2)
                         (QUOTE MVTest4)))
       (SubTest #.NotSetValue Tested? #.NotSetValue)
       (PreTest #. ($A (LTMListInClass)
                       BuildPreTest MakeBackLink)
             Tested? #.NotSetValue)
       (AltTest #.NotSetValue Tested? #.NotSetValue)
       (SyntaxTest #.NotSetValue Tested? #.NotSetValue)
       (UsesObj #. ($A (LOOPSTestClass2 LOOPSTestCases)
                       NIL AllowRemove))
       (SubTestOf #. ($A (LTP7)
                         NIL MakeBackLink)))

(DEFINST LOOPSTestPrimitive (LTP40 "OTR@@AÌk")
       (name #. ($A LTP40 NIL RememberName))
       (TestDesc "PutClassValue:in a prop inherited in a class")
       (SetUp (PutClassValue ($ LOOPSTestClass3)
                     (QUOTE PCV3)
                     (QUOTE LTP40)
                     (QUOTE Prop1)))
       [TestExpr (AND (EQUAL (GetClassValue ($ LOOPSTestClass3)
                                     (QUOTE PCV3)
                                     (QUOTE Prop1))
                             (QUOTE LTP40))
                      (EQUAL (GetClassValue ($ LOOPSTestCases)
                                     (QUOTE PCV3)
                                     (QUOTE Prop1))
                             (QUOTE LTP40]
       (ResetExp (PutClassValue ($ LOOPSTestCases)
                         (QUOTE PCV3)
                         (QUOTE LTC)
                         (QUOTE Prop1)))
       (SubTest #.NotSetValue Tested? #.NotSetValue)
       (PreTest #.NotSetValue Tested? #.NotSetValue)
       (AltTest #.NotSetValue Tested? #.NotSetValue)
       (SyntaxTest #.NotSetValue Tested? #.NotSetValue)
       (UsesObj #. ($A (LOOPSTestCases LOOPSTestClass3)
                       NIL AllowRemove))
       (SubTestOf #. ($A (LTP36)
                         NIL MakeBackLink)))

(DEFINST LOOPSTestPrimitive (LTP39 "OTR@@AÌj")
       (name #. ($A LTP39 NIL RememberName))
       (TestDesc "PutClassValue: local prop over inherited")
       (SetUp (PutClassValue ($ LOOPSTestClass3)
```

```
                    (QUOTE PCV2)
                    (QUOTE LTP39)
                    (QUOTE Prop1)))
       (TestExpr (EQUAL (GetClassValue ($ LOOPSTestClass3)
                               (QUOTE PCV2)
                               (QUOTE Prop1))
                         (QUOTE LTP39)))
       (ResetExp (PutClassValue ($ LOOPSTestClass3)
                         (QUOTE PCV2)
                         (QUOTE LTC3)
                         (QUOTE Prop1)))
       (SubTest #.NotSetValue Tested? #.NotSetValue)
       (PreTest #.NotSetValue Tested? #.NotSetValue)
       (AltTest #.NotSetValue Tested? #.NotSetValue)
       (SyntaxTest #.NotSetValue Tested? #.NotSetValue)
       (UsesObj #. ($A (LOOPSTestClass3 LOOPSTestCases)
                       NIL AllowRemove))
       (SubTestOf #. ($A (LTP35)
                         NIL MakeBackLink)))

(DEFINST LOOPSTestPrimitive (LTP38 "OTR@@AÌi")
       (name #. ($A LTP38 NIL RememberName))
       (TestDesc "PutClassValue: prop via an instance")
       (SetUp (PutClassValue ($ LTCC1)
                    (QUOTE PCV1)
                    (QUOTE LTP38)
                    (QUOTE Prop1)))
       (TestExpr (EQUAL (GetClassValue ($ LTCC1)
                               (QUOTE PCV1)
                               (QUOTE Prop1))
                         (QUOTE LTP38)))
       (ResetExp (PutClassValue ($ LTCC1)
                         (QUOTE PCV1)
                        NIL
                         (QUOTE Prop1)))
       (SubTest #.NotSetValue Tested? #.NotSetValue)
       (PreTest #.NotSetValue Tested? #.NotSetValue)
       (AltTest #.NotSetValue Tested? #.NotSetValue)
       (CasesUsed #. ($A (LTCC1)
                         NIL MakeBackLink))
       (SyntaxTest #.NotSetValue Tested? #.NotSetValue)
       (UsesObj #. ($A (LOOPSTestClass3)
                       NIL AllowRemove))
       (SubTestOf #. ($A (LTP34)
                         NIL MakeBackLink)))

(DEFINST LOOPSTestPrimitive (LTP32 "OTR@@EÙ")
       (name #. ($A LTP32 NIL RememberName))
       (TestDesc "GetClassValue: local prop over inherited")
       (TestExpr (EQUAL (GetClassValue ($ LOOPSTestClass1)
                               (QUOTE GCV2)
                               (QUOTE Prop1))
                         (QUOTE LTP29)))
       (SubTest #.NotSetValue Tested? #.NotSetValue)
       (PreTest #.NotSetValue Tested? #.NotSetValue)
       (AltTest #.NotSetValue Tested? #.NotSetValue)
       (SyntaxTest #.NotSetValue Tested? #.NotSetValue)
       (UsesObj #. ($A (LOOPSTestCases LOOPSTestClass1)
                       NIL AllowRemove))
       (SubTestOf #. ($A (LTP29)
                         NIL MakeBackLink)))

(DEFINST LOOPSTestPrimitive (LTP31 "OTR@@EÜ")
       (name #. ($A LTP31 NIL RememberName))
       (TestDesc "GetClassValue: inherit prop in class")
       (TestExpr (EQUAL (GetClassValue ($ LOOPSTestClass1)
                               (QUOTE GCV1)
                               (QUOTE Prop1))
                         (QUOTE GCV1)))
       (SubTest #.NotSetValue Tested? #.NotSetValue)
       (PreTest #.NotSetValue Tested? #.NotSetValue)
       (AltTest #.NotSetValue Tested? #.NotSetValue)
       (SyntaxTest #.NotSetValue Tested? #.NotSetValue)
       (UsesObj #. ($A (LOOPSTestClass1 LOOPSTestCases)
                       NIL AllowRemove))
       (SubTestOf #. ($A (LTP28)
                         NIL MakeBackLink)))

(DEFINST LOOPSTestPrimitive (LTP30 "OTR@@EÛ")
       (name #. ($A LTP30 NIL RememberName))
       (TestDesc "GetClassValue: prop in an instance")
       (TestExpr (EQUAL (GetClassValue ($ LTC7)
                               (QUOTE GCV1)
                               (QUOTE Prop1))
                         (QUOTE GCV1)))
       (SubTest #.NotSetValue Tested? #.NotSetValue)
       (PreTest #.NotSetValue Tested? #.NotSetValue)
```

```
       (AltTest #.NotSetValue Tested? #.NotSetValue)
       (CasesUsed #. ($A (LTC7)
                         NIL MakeBackLink))
       (SyntaxTest #.NotSetValue Tested? #.NotSetValue)
       (UsesObj #. ($A (LOOPSTestCases)
                       NIL AllowRemove))
       (SubTestOf #. ($A (LTP26)
                         NIL MakeBackLink)))

(PUTPROPS LTLISPFUNTESTS COPYRIGHT ("Xerox Corporation" 1985 1986))
```

VARIABLE INDEX
