

File created: 21-Apr-86 14:05:39 {POGO:PARC:XEROX}<LOOPS>TESTER>LTMETHODTESTS.;2

changes to: (INSTANCES LTP25 LTP23 LTP46 LTP59 LTP60 LTP61 LTP62 LTP63 LTP64 LTP21 LTP22 LTP41 LTP42 LTP43
LTP45 LTP66 LTMLListInClass LTMAdd LTMDestroyInClass LTMNew LTMDestroyInObj
)

previous date: 18-Mar-85 12:03:14 {POGO:PARC:XEROX}<LOOPS>TESTER>LTMETHODTESTS.;1

Read Table: OLD-INTERLISP-FILE

Package: INTERLISP

Format: XCCS

(* * Copyright (c) 1985, 1986 by Xerox Corporation. All rights reserved.)

(RPAQQ **LTMETHODTESTSCOMS**

((* File created by MITTAL)
(* database of test instances for testing methods in Class and Object)
(CLASSES)
(METHODS)
(FNS)
(VARS)
(INSTANCES * LTMMethodInstances)))

(* * File created by MITTAL)

(* * database of test instances for testing methods in Class and Object)

(DEFCLASSES)

(RPAQQ **LTMMethodInstances**

(LTMTitleCommandInObj LTMSetNameInObj LTMRenameInObj LTMPutInObj LTMPPV!InObj LTMPP!InObj LTMPPInObj
LTMLList!InObj LTMLListInObj LTMInspectValueCommandInObj LTMInspectTitleMenuInObj
LTMInspectTitleInObj LTMInspectStoreInObj LTMInspectPropertiesInObj LTMInspectPropCommandInObj
LTMInspectFetchInObj LTMHasIVInObj LTMEditInObj LTMDumpFactsInObj LTMDestroy!InObj LTMDestroyInObj
LTMTitleCommandInClass LTMSetNameInClass LTMRenameInClass LTMPutInClass LTMPPV!InClass
LTMPP!InClass LTMPPInClass LTMLList!InClass LTMInspectValueCommandInClass
LTMInspectTitleMenuInClass LTMInspectTitleInClass LTMInspectStoreInClass
LTMInspectPropertiesInClass LTMInspectPropCommandInClass LTMInspectFetchInClass LTMHasIVInClass
LTMEditInClass LTMDumpFactsInClass LTMDestroy!InClass LTMDestroyInClass LTMWhereIs LTMUnderstands
LTMUnSetName LTMTraceIt LTMReturnDefaultValue LTMPutIVProp LTMPrintOn LTMNoObjectForMsg
LTMMessageNotUnderstood LTMInstantiate LTMInstOf! LTMInstOf LTMInspect LTMIVMissing LTMDoMethod
LTMDestroyIV LTMCopyShallow LTMCopyDeep LTMClassName LTMClass LTMBreakIt LTMAt LTMAssocKB LTMAddIV
LTMSubClasses LTMReplaceSupers LTMPPMethod LTMPPM LTMOnFile LTMNewWithValues LTMNewTemp LTMNew
LTMMoveMethod LTMMethodDoc LTMInitialize LTMHasCV LTMFetchMethod LTMEditMethod
LTMDisplaySubClasses LTMDefMethod LTMCopyMethod LTMCommentMethods LTMDestroyInClass LTMAdd
LTMLListInClass LTP66 LTP77 LTP76 LTP45 LTP43 LTP42 LTP41 LTP22 LTP21 LTP64 LTP63 LTP62 LTP61 LTP60
LTP59 LTP46 LTP23 LTP44 LTP25))

(DEFINST LOOPSTestMethod (LTMTitleCommandInObj "ONR@@AAÏ")
(name #. (\$A LTMTitleCommandInObj NIL RememberName)))

(DEFINST LOOPSTestMethod (LTMSetNameInObj "ONR@@AA
")
(name #. (\$A LTMSetNameInObj NIL RememberName)))

(DEFINST LOOPSTestMethod (LTMRenameInObj "ONR@@AAÏ

```

{MEDLEY}<loops>test>froml.1>LTMETHODTESTS.;1

")
  (name #. ($A LTRenameInObj NIL RememberName))
(DEFINST LOOPSTestMethod (LTMPutInObj "ONR@@AÁÈ")
  (name #. ($A LTMPutInObj NIL RememberName))
(DEFINST LOOPSTestMethod (LTMPPV!InObj "ONR@@AÁ%
")
  (name #. ($A LTMPPV!InObj NIL RememberName))
(DEFINST LOOPSTestMethod (LTMP!InObj "ONR@@AÁ ")
  (name #. ($A LTMPP!InObj NIL RememberName))
(DEFINST LOOPSTestMethod (LTMP!InObj "ONR@@AÁÉ")
  (name #. ($A LTMPP!InObj NIL RememberName))
(DEFINST LOOPSTestMethod (LTMList!InObj "ONR@@AÁ'")
  (name #. ($A LTMList!InObj NIL RememberName))
(DEFINST LOOPSTestMethod (LTMListInObj "ONR@@AÁÁ")
  (name #. ($A LTMListInObj NIL RememberName))
  (TestDesc "List method in OBJECT: IVs")
  (TestExpr (ATEST (MEMBER (QUOTE LIO)
    (SEND ($ LTCC2)
      List
      (QUOTE IVs)))
    "local IV is returned by List"))
  (CasesUsed #. ($A (LTCC2)
    NIL MakeBackLink))
(DEFINST LOOPSTestMethod (LTMInspectValueCommandInObj "ONR@@AÁÁ")
  (name #. ($A LTMInspectValueCommandInObj NIL RememberName))
(DEFINST LOOPSTestMethod (LTMInspectTitleMenuInObj "ONR@@AÁÁ")
  (name #. ($A LTMInspectTitleMenuInObj NIL RememberName))
(DEFINST LOOPSTestMethod (LTMInspectTitleInObj "ONR@@AÁÁ")
  (name #. ($A LTMInspectTitleInObj NIL RememberName))
(DEFINST LOOPSTestMethod (LTMInspectStoreInObj "ONR@@AÁÁ")
  (name #. ($A LTMInspectStoreInObj NIL RememberName))
(DEFINST LOOPSTestMethod (LTMInspectPropertiesInObj "ONR@@AÁÁ")
  (name #. ($A LTMInspectPropertiesInObj NIL RememberName))
(DEFINST LOOPSTestMethod (LTMInspectPropCommandInObj "ONR@@AÁ ")
  (name #. ($A LTMInspectPropCommandInObj NIL RememberName))
(DEFINST LOOPSTestMethod (LTMInspectFetchInObj "ONR@@A ")
  (name #. ($A LTMInspectFetchInObj NIL RememberName))
(DEFINST LOOPSTestMethod (LTMHasIVInObj "ONR@@A~")
  (name #. ($A LTMHasIVInObj NIL RememberName))
(DEFINST LOOPSTestMethod (LTMEditInObj "ONR@@A}")
  (name #. ($A LTMEditInObj NIL RememberName))
(DEFINST LOOPSTestMethod (LTMDumpFactsInObj "ONR@@A|")
  (name #. ($A LTMDumpFactsInObj NIL RememberName))
(DEFINST LOOPSTestMethod (LTMDestroy!InObj "ONR@@A{")
  (name #. ($A LTMDestroy!InObj NIL RememberName))
(DEFINST LOOPSTestMethod (LTMDestroyInObj "ONR@@Az")
  (name #. ($A LTMDestroyInObj NIL RememberName))
  (TestDesc "Destroy method in OBJECT: named object")
  (SetUp (PROGN (SETQ LTV4 (SEND ($ LTDC2)
    New))
    (SEND LTV4 SetName (QUOTE LTDC21))
    (SEND LTV4 Destroy)))
  [TestExpr (AND (NULL (GetObjectRec ($ LTDC21)
    (SubTest #.NotSetValue Tested? #.NotSetValue)
  (PreTest #. ($A (LTMNew LTFGetObjectRec)
    NIL MakeBackLink)
    Tested? #.NotSetValue)
  (AltTest #.NotSetValue Tested? #.NotSetValue)
  (SyntaxTest #.NotSetValue Tested? #.NotSetValue)
  (UsesObj #. ($A (LTDC2)
    NIL AllowRemove)))
(DEFINST LOOPSTestMethod (LTMTtitleCommandInClass "ONR@@Ay")
  (name #. ($A LTMTtitleCommandInClass NIL RememberName))
(DEFINST LOOPSTestMethod (LTMSetNameInClass "ONR@@Ax")
  (name #. ($A LTMSetNameInClass NIL RememberName))
(DEFINST LOOPSTestMethod (LTMRenameInClass "ONR@@Aw")

```

```

{MEDLEY}<loops>test>from1.1>LTMETHODTESTS.;1

    (name #. ($A LTMRenameInClass NIL RememberName)))

(DEFINST LOOPSTestMethod (LTMPutInClass "ONR@@Av")
  (name #. ($A LTMPutInClass NIL RememberName)))

(DEFINST LOOPSTestMethod (LTMPPV!InClass "ONR@@Au")
  (name #. ($A LTMPPV!InClass NIL RememberName)))

(DEFINST LOOPSTestMethod (LTMP!InClass "ONR@@At")
  (name #. ($A LTMP!InClass NIL RememberName)))

(DEFINST LOOPSTestMethod (LTMP!InClass "ONR@@As")
  (name #. ($A LTMP!InClass NIL RememberName)))

(DEFINST LOOPSTestMethod (LTMList!InClass "ONR@@Ar")
  (name #. ($A LTMList!InClass NIL RememberName)))

(DEFINST LOOPSTestMethod (LTMInspectValueCommandInClass "ONR@@Ap")
  (name #. ($A LTMInspectValueCommandInClass NIL RememberName)))

(DEFINST LOOPSTestMethod (LTMInspectTitleMenuInClass "ONR@@Ao")
  (name #. ($A LTMInspectTitleMenuInClass NIL RememberName)))

(DEFINST LOOPSTestMethod (LTMInspectTitleInClass "ONR@@An")
  (name #. ($A LTMInspectTitleInClass NIL RememberName)))

(DEFINST LOOPSTestMethod (LTMInspectStoreInClass "ONR@@Am")
  (name #. ($A LTMInspectStoreInClass NIL RememberName)))

(DEFINST LOOPSTestMethod (LTMInspectPropertiesInClass "ONR@@Al")
  (name #. ($A LTMInspectPropertiesInClass NIL RememberName)))

(DEFINST LOOPSTestMethod (LTMInspectPropCommandInClass "ONR@@Ak")
  (name #. ($A LTMInspectPropCommandInClass NIL RememberName)))

(DEFINST LOOPSTestMethod (LTMInspectFetchInClass "ONR@@Aj")
  (name #. ($A LTMInspectFetchInClass NIL RememberName)))

(DEFINST LOOPSTestMethod (LTMHasIVInClass "ONR@@Ai")
  (name #. ($A LTMHasIVInClass NIL RememberName)))

(DEFINST LOOPSTestMethod (LTMEditInClass "ONR@@Ah")
  (name #. ($A LTMEditInClass NIL RememberName)))

(DEFINST LOOPSTestMethod (LTMDumpFactsInClass "ONR@@Ag")
  (name #. ($A LTMDumpFactsInClass NIL RememberName)))

(DEFINST LOOPSTestMethod (LTMDestroy!InClass "ONR@@Af")
  (name #. ($A LTMDestroy!InClass NIL RememberName)))

(DEFINST LOOPSTestMethod (LTMDestroyInClass "ONR@@Ae")
  (name #. ($A LTMDestroyInClass NIL RememberName))
  (TestDesc "Destroy method in CLASS:")
  (SetUp (SEND ($ LTDClass)
              Destroy))
  [TestExpr (AND (NULL (GetObjectRec (QUOTE LTDClass)
                                     [ResetExp (DC (QUOTE LTDClass)
                                     (QUOTE (LOOPSTestCases)
                                     (SubTest #.NotSetValue Tested? #.NotSetValue)
                                     (PreTest #. ($A (LTFGetObjectRec)
                                     NIL MakeBackLink)
                                     Tested? #.NotSetValue)
                                     (AltTest #.NotSetValue Tested? #.NotSetValue)
                                     (SyntaxTest #.NotSetValue Tested? #.NotSetValue)
                                     (UsesObj #. ($A (LTDClass)
                                     NIL AllowRemove)))]
  (DEFINST LOOPSTestMethod (LTMWhereIs "ONR@@Ad")
  (name #. ($A LTMWhereIs NIL RememberName)))

(DEFINST LOOPSTestMethod (LTMUnderstands "ONR@@Ac")
  (name #. ($A LTMUnderstands NIL RememberName)))

(DEFINST LOOPSTestMethod (LTMUnSetName "ONR@@Ab")
  (name #. ($A LTMUnSetName NIL RememberName)))

(DEFINST LOOPSTestMethod (LTMTraceIt "ONR@@Aa")
  (name #. ($A LTMTraceIt NIL RememberName)))

(DEFINST LOOPSTestMethod (LTMReturnDefaultValue "ONR@@A`")
  (name #. ($A LTMReturnDefaultValue NIL RememberName)))

(DEFINST LOOPSTestMethod (LTMPutIVProp "ONR@@A_")
  (name #. ($A LTMPutIVProp NIL RememberName)))

(DEFINST LOOPSTestMethod (LTMPrintOn "ONR@@A^")
  (name #. ($A LTMPrintOn NIL RememberName)))

```

```
(DEFINST LOOPSTestMethod (LTMNoObjectForMsg "ONR@@A")
  (name #. ($A LTMNoObjectForMsg NIL RememberName)))

(DEFINST LOOPSTestMethod (LTMMessageNotUnderstood "ONR@@A\")
  (name #. ($A LTMMessageNotUnderstood NIL RememberName)))

(DEFINST LOOPSTestMethod (LTMInstantiate "ONR@@A[")
  (name #. ($A LTMInstantiate NIL RememberName)))

[DEFINST LOOPSTestMethod (LTMInstOf! "ONR@@AZ")
  (name #. ($A LTMInstOf! NIL RememberName))
  (TestDesc "InstOf! METHOD")
  (TestExpr (ANDALL (ATEST (SEND ($ LTMInstOf!)
    InstOf!
    (QUOTE LOOPSTestMethod))
    "Instance")
  (ATEST (SEND ($ LTMInstOf!)
    InstOf!
    (QUOTE LOOPSTestObject))
    "Instance once removed from tested class")
  (ATEST (NOT (SEND ($ LTMInstOf!)
    InstOf!
    (QUOTE LOOPSTestSuper)))
    "Instance NOT InstOf of tested class"]

[DEFINST LOOPSTestMethod (LTMInstOf "ONR@@AY")
  (name #. ($A LTMInstOf NIL RememberName))
  (TestDesc "InstOf METHOD")
  (TestExpr (ANDALL (ATEST (SEND ($ LTMInstOf)
    InstOf
    (QUOTE LOOPSTestMethod))
    "check for Instance")
  (ATEST (NOT (SEND ($ LTFSend)
    InstOf
    (QUOTE LOOPSTestMethod)))
    "check for an Instance which is NOT an InstOf the class"]

(DEFINST LOOPSTestMethod (LTMInspect "ONR@@AX")
  (name #. ($A LTMInspect NIL RememberName)))

(DEFINST LOOPSTestMethod (LTMIVMissing "ONR@@AW")
  (name #. ($A LTMIVMissing NIL RememberName)))

(DEFINST LOOPSTestMethod (LTMDoMethod "ONR@@AV")
  (name #. ($A LTMDoMethod NIL RememberName)))

(DEFINST LOOPSTestMethod (LTMDeleteIV "ONR@@AU")
  (name #. ($A LTMDeleteIV NIL RememberName)))

(DEFINST LOOPSTestMethod (LTMCopyShallow "ONR@@AT")
  (name #. ($A LTMCopyShallow NIL RememberName)))

(DEFINST LOOPSTestMethod (LTMCopyDeep "ONR@@AS")
  (name #. ($A LTMCopyDeep NIL RememberName)))

(DEFINST LOOPSTestMethod (LTMClassName "ONR@@AR")
  (name #. ($A LTMClassName NIL RememberName)))

(DEFINST LOOPSTestMethod (LTMClass "ONR@@AQ")
  (name #. ($A LTMClass NIL RememberName)))

(DEFINST LOOPSTestMethod (LTMBreakIt "ONR@@AP")
  (name #. ($A LTMBreakIt NIL RememberName)))

(DEFINST LOOPSTestMethod (LTMAAt "ONR@@AO")
  (name #. ($A LTMAAt NIL RememberName)))

(DEFINST LOOPSTestMethod (LTMAssocKB "ONR@@AN")
  (name #. ($A LTMAssocKB NIL RememberName)))

(DEFINST LOOPSTestMethod (LTMAddIV "ONR@@AM")
  (name #. ($A LTMAddIV NIL RememberName)))

(DEFINST LOOPSTestMethod (LTMSubClasses "ONR@@AL")
  (name #. ($A LTMSubClasses NIL RememberName)))

(DEFINST LOOPSTestMethod (LTMReplaceSupers "ONR@@AK")
  (name #. ($A LTMReplaceSupers NIL RememberName))
  (TestDesc "Replace Supers method")
  [SetUp (SEND ($ LOOPSTestClassRS)
    ReplaceSupers
    (QUOTE (LOOPSTestCases)
  [TestExpr (AND (EQUAL (SEND ($ LOOPSTestClassRS)
    List
    (QUOTE Supers))
    (QUOTE (LOOPSTestCases)))]
```

```

      (EQUAL (GetValue ($ LOOPSTestClassRS)
                (QUOTE GV1))
             (QUOTE LTP10])
[ResetExp (SEND ($ LOOPSTestClassRS)
  ReplaceSupers
  (QUOTE (LOOPSTestClassCF)
[PreTest #. ($A (LTP62 LTFGetValue)
  NIL MakeBackLink))
(UsesObj #. ($A (LOOPSTestCases LOOPSTestClassRS LOOPSTestClassCF)
  NIL AllowRemove)))

(DEFINST LOOPSTestMethod (LTMPPMethod "ONR@@AJ")
  (name #. ($A LTMPPMethod NIL RememberName)))

(DEFINST LOOPSTestMethod (LTMPPM "ONR@@AI")
  (name #. ($A LTMPPM NIL RememberName)))

(DEFINST LOOPSTestMethod (LTMonFile "ONR@@AH")
  (name #. ($A LTMonFile NIL RememberName)))

(DEFINST LOOPSTestMethod (LTMNewWithValues "ONR@@AG")
  (name #. ($A LTMNewWithValues NIL RememberName))
  (TestDesc "NewWithValues: creating an instance")
  [SetUp (SETQ LTV7 (SEND ($ LOOPSTestCases)
    NewWithValues
    (QUOTE ((name LTC0)
             (Var1 5)
             (Test1 6 BackLink 7)
  (TestExpr (AND (type? instance LTV7)
    (EQ (Class LTV7)
        ($ LOOPSTestCases))
    (EQUAL (GetValue LTV7 (QUOTE name))
            (QUOTE LTC0))
    (EQUAL (GetValue LTV7 (QUOTE Var1))
            5)
    (EQUAL (GetValue LTV7 (QUOTE Test1))
            6)
    (EQUAL (GetValue LTV7 (QUOTE Test1))
            (GetValueOnly LTV7 (QUOTE Test1)))
    (EQUAL (GetValue LTV7 (QUOTE Test1))
            (QUOTE BackLink))
    7)))
  (ResetExp (SEND LTV7 Destroy))
  (PreTest #. ($A (LTFPutValueOnly LTFGetValueOnly LTFGetValue LTFClass LTFtype?)
    NIL MakeBackLink))
  (UsesObj #. ($A (LOOPSTestCases)
    NIL AllowRemove)))

(DEFINST LOOPSTestMethod (LTMNewTemp "ONR@@AF")
  (name #. ($A LTMNewTemp NIL RememberName)))

(DEFINST LOOPSTestMethod (LTMNew "ONR@@AE")
  (name #. ($A LTMNew NIL RememberName))
  (TestDesc "New method in METACLASS: Creating a class")
  [SetUp (SEND ($ Class)
    New
    (QUOTE LTNClass)
    (QUOTE (Object)
[TestExpr (AND (type? class ($ LTNClass))
  (EQ (Class ($ LTNClass))
      ($ Class)
[ResetExp (SEND ($ LTNClass)
  Destroy))
(SubTest #. ($A (LTP66)
  NIL MakeBackLink)
  Tested? #.NotSetValue)
[PreTest #. ($A (LTFClass LTFtype?)
  NIL MakeBackLink)
  Tested? #.NotSetValue)
(AltTest #.NotSetValue Tested? #.NotSetValue)
(SyntaxTest #.NotSetValue Tested? #.NotSetValue))

(DEFINST LOOPSTestMethod (LTMMoveMethod "ONR@@AD")
  (name #. ($A LTMMoveMethod NIL RememberName)))

(DEFINST LOOPSTestMethod (LTMMethodDoc "ONR@@AC")
  (name #. ($A LTMMethodDoc NIL RememberName)))

(DEFINST LOOPSTestMethod (LTMInitialize "ONR@@AB")
  (name #. ($A LTMInitialize NIL RememberName)))

(DEFINST LOOPSTestMethod (LTMHasCV "ONR@@AA")
  (name #. ($A LTMHasCV NIL RememberName)))

(DEFINST LOOPSTestMethod (LTMFetchMethod "ONR@@A@")
  (name #. ($A LTMFetchMethod NIL RememberName))
  (TestDesc "FetchMethod in CLASS: local")

```

```

{MEDLEY}<loops>test>from1.1>LTMETHODTESTS.;1

  (TestExpr (EQUAL (SEND ($ LOOPSTestClassL1)
                        FetchMethod
                        (QUOTE M1))
                (QUOTE LOOPSTestClassL1.M1)))
  (SubTest #. ($A (LTP77 LTP76)
                 NIL MakeBackLink))
  (UsesObj #. ($A (LOOPSTestCases LOOPSTestClassL1)
                 NIL AllowRemove))

(DEFINST LOOPSTestMethod (LTMEditMethod "ONR@@A?")
  (name #. ($A LTMEditMethod NIL RememberName)))

(DEFINST LOOPSTestMethod (LTMDisplaySubClasses "ONR@@A>")
  (name #. ($A LTMDisplaySubClasses NIL RememberName)))

(DEFINST LOOPSTestMethod (LTMDefMethod "ONR@@A=")
  (name #. ($A LTMDefMethod NIL RememberName)))

(DEFINST LOOPSTestMethod (LTMCopyMethod "ONR@@A<")
  (name #. ($A LTMCopyMethod NIL RememberName)))

(DEFINST LOOPSTestMethod (LTMCommentMethods "ONR@@A;")
  (name #. ($A LTMCommentMethods NIL RememberName)))

(DEFINST LOOPSTestMethod (LTMDeleteInClass "ONR@@A ")
  (name #. ($A LTMDeleteInClass NIL RememberName))
  (TestDesc "Delete method in CLASS: IV")
  [SetUp (PROGN (SEND ($ LOOPSTestClass5)
                    Delete
                    (QUOTE IV)
                    (QUOTE DIV1))
              (SETQ LTV1 (SEND ($ LOOPSTestClass5)
                              List
                              (QUOTE IVs))
                          (TestExpr (NOT (FMEMB (QUOTE DIV1)
                                             LTV1)))
                          (ResetExp (SEND ($ LOOPSTestClass5)
                                           Add
                                           (QUOTE IV)
                                           (QUOTE DIV1)
                                           NIL))
                          (SubTest #. ($A (LTP45 LTP43 LTP42 LTP41)
                                           NIL MakeBackLink)
                                   Tested? #.NotSetValue)
                          (PreTest #. ($A (LTMListInClass)
                                           NIL MakeBackLink)
                                   Tested? #.NotSetValue)
                          (AltTest #.NotSetValue Tested? #.NotSetValue)
                          (SyntaxTest #.NotSetValue Tested? #.NotSetValue)
                          (UsesObj #. ($A (LOOPSTestClass5)
                                           NIL AllowRemove))
                          (StandBy (LTP44)))
              (DEFINST LOOPSTestMethod (LTMAdd "ONR@@A:")
                (name #. ($A LTMAdd NIL RememberName))
                (TestDesc "Add method: adding IV")
                (SetUp (SEND ($ LOOPSTestClass4)
                            Add
                            (QUOTE IV)
                            (QUOTE AIV1)
                            (QUOTE LTMAdd)))
                (TestExpr (EQUAL (GetValue ($ LOOPSTestClass4)
                                           (QUOTE AIV1))
                                (QUOTE LTMAdd)))
                (ResetExp (SEND ($ LOOPSTestClass4)
                                Delete
                                (QUOTE IV)
                                (QUOTE AIV1)))
                (SubTest #. ($A (LTP22 LTP21)
                                NIL MakeBackLink)
                        Tested? #.NotSetValue)
                (PreTest #. ($A (LTFGetValue)
                                NIL MakeBackLink)
                        Tested? #.NotSetValue)
                (AltTest #.NotSetValue Tested? #.NotSetValue)
                (SyntaxTest #.NotSetValue Tested? #.NotSetValue)
                (UsesObj #. ($A (LOOPSTestClass4)
                                NIL AllowRemove))
                (StandBy #. ($A (LTP25)
                                NIL AllowRemove)))
              (DEFINST LOOPSTestMethod (LTMListInClass "ONR@@Aq")
                (name #. ($A LTMListInClass NIL RememberName))
                (TestDesc "List method in CLASS:IVs")
                [TestExpr (EQUAL (SEND ($ LOOPSTestClassL1)
                                      List
                                      (QUOTE IVs))

```

```

      (QUOTE (L1 L2 L3]
(SubTest #. ($A (LTP64 LTP63 LTP62 LTP61 LTP60 LTP59)
  NIL MakeBackLink)
  Tested? #.NotSetValue)
(PreTest #.NotSetValue Tested? #.NotSetValue)
(AltTest #.NotSetValue Tested? #.NotSetValue)
(SyntaxTest #.NotSetValue Tested? #.NotSetValue)
(UsesObj #. ($A (LOOPSTestClassL1)
  NIL AllowRemove)))

```

```

(DEFINST LOOPSTestPrimitive (LTP66 "O\R@[`j")
  (name #. ($A LTP66 NIL RememberName))
  (TestDesc "New method: creating an instance")
  (SetUp (SETQ LTV5 (SEND ($ LTNClass)
    New)))
  [TestExpr (AND (type? instance LTV5)
    (EQ (Class LTV5)
      ($ LTNClass)
    (SubTest #.NotSetValue Tested? #.NotSetValue)
    (PreTest #.NotSetValue Tested? #.NotSetValue)
    (AltTest #.NotSetValue Tested? #.NotSetValue)
    (SyntaxTest #.NotSetValue Tested? #.NotSetValue)
    (UsesObj #. ($A (LTNClass)
      NIL AllowRemove))
    (SubTestOf #. ($A (LTMNew)
      NIL MakeBackLink)))

```

```

(DEFINST LOOPSTestPrimitive (LTP77 "NPR@@A8")
  (name #. ($A LTP77 NIL RememberName))
  (TestDesc "FetchMethod in CLASS: inherited")
  (TestExpr (EQUAL (SEND ($ LOOPSTestClassL1)
    FetchMethod
    (QUOTE GM4))
    (QUOTE LOOPSTestCases.GM4)))
  (SubTestOf #. ($A (LTMFetchMethod)
    NIL MakeBackLink)))

```

```

(DEFINST LOOPSTestPrimitive (LTP76 "O]R@@EÄ")
  (name #. ($A LTP76 NIL RememberName))
  (TestDesc "FetchMethod in CLASS: local over inherited")
  (TestExpr (EQUAL (SEND ($ LOOPSTestClassL1)
    FetchMethod
    (QUOTE M2))
    (QUOTE LOOPSTestClassL1.M2)))
  (UsesObj #. ($A (LOOPSTestClassL1 LOOPSTestCases)
    NIL AllowRemove))
  (SubTestOf #. ($A (LTMFetchMethod)
    NIL MakeBackLink)))

```

```

(DEFINST LOOPSTestPrimitive (LTP45 "OVR@@AÉ ")
  (name #. ($A LTP45 NIL RememberName))
  (TestDesc "Delete in CLASS: Meta")
  (SubTest #.NotSetValue Tested? #.NotSetValue)
  (PreTest #.NotSetValue Tested? #.NotSetValue)
  (AltTest #.NotSetValue Tested? #.NotSetValue)
  (SyntaxTest #.NotSetValue Tested? #.NotSetValue)
  (SubTestOf #. ($A (LTMDeleteInClass)
    NIL MakeBackLink)))

```

```

(DEFINST LOOPSTestPrimitive (LTP43 "OVR@@A`~")
  (name #. ($A LTP43 NIL RememberName))
  (TestDesc "Delete in CLASS: Method")
  [SetUp (PROGN (SEND ($ LOOPSTestClass5)
    Delete
    (QUOTE Method)
    (QUOTE DM1))
    (SETQ LTV1 (SEND ($ LOOPSTestClass5)
      List
      (QUOTE Methods]
    (TestExpr (AND (NOT (FMEMB (QUOTE DM1)
      LTV1))
      (FMEMB (QUOTE DM3)
        LTV1)))
    (ResetExp (SEND ($ LOOPSTestClass5)
      Add
      (QUOTE Method)
      (QUOTE DM1)
      (QUOTE LOOPSTestClass5.DM1)))
    (SubTest #. ($A (LTP47)
      NIL MakeBackLink)
      Tested? #.NotSetValue)
    (PreTest #.NotSetValue Tested? #.NotSetValue)
    (AltTest #.NotSetValue Tested? #.NotSetValue)
    (SyntaxTest #.NotSetValue Tested? #.NotSetValue)
    (UsesObj #. ($A (LOOPSTestClass5)
      NIL AllowRemove))
    (SubTestOf #. ($A (LTMDeleteInClass)

```

```

NIL MakeBackLink)))

(DEFINST LOOPSTestPrimitive (LTP42 "OVR@@"))
(name #. ($A LTP42 NIL RememberName))
(TestDesc "Delete in CLASS: IVProp")
[SetUp (PROGN (SEND ($ LOOPSTestClass5)
  Delete
  (QUOTE IVProp)
  (QUOTE DIV2)
  (QUOTE Prop1))
  (SETQ LTV1 (SEND ($ LOOPSTestClass5)
    List
    (QUOTE IVProps)
    (QUOTE DIV2]
  (TestExpr (AND (NOT (FMEMB (QUOTE Prop1)
    LTV1))
    (FMEMB (QUOTE doc)
    LTV1)))
  (ResetExp (SEND ($ LOOPSTestClass5)
    Add
    (QUOTE IVProp)
    (QUOTE DIV2)
    (QUOTE DIV2)
    (QUOTE Prop1)))
  (SubTest #.NotSetValue Tested? #.NotSetValue)
  (PreTest #.NotSetValue Tested? #.NotSetValue)
  (AltTest #.NotSetValue Tested? #.NotSetValue)
  (SyntaxTest #.NotSetValue Tested? #.NotSetValue)
  (UsesObj #. ($A (LOOPSTestClass5)
    NIL AllowRemove))
  (SubTestOf #. ($A (LTMDeleteInClass)
    NIL MakeBackLink)))

(DEFINST LOOPSTestPrimitive (LTP41 "OVR@@"))
(name #. ($A LTP41 NIL RememberName))
(TestDesc "Delete in Class: CV")
(Setup (SEND ($ LOOPSTestClass5)
  Delete
  (QUOTE CV)
  (QUOTE DCV1)))
[TestExpr (NOT (FMEMB (QUOTE DCV1)
  (SEND ($ LOOPSTestClass5)
    List
    (QUOTE CVs]
  (ResetExp (SEND ($ LOOPSTestClass5)
    Add
    (QUOTE CV)
    (QUOTE DCV1)
    NIL))
  (SubTest #. ($A (LTP46)
    NIL MakeBackLink)
    Tested? #.NotSetValue)
  (PreTest #.NotSetValue Tested? #.NotSetValue)
  (AltTest #.NotSetValue Tested? #.NotSetValue)
  (SyntaxTest #.NotSetValue Tested? #.NotSetValue)
  (UsesObj #. ($A (LOOPSTestClass5)
    NIL AllowRemove))
  (SubTestOf #. ($A (LTMDeleteInClass)
    NIL MakeBackLink)))

(DEFINST LOOPSTestPrimitive (LTP22 "OOR@@"))
(name #. ($A LTP22 NIL RememberName))
(TestDesc "Add Method: IVProp")
(Setup (SEND ($ LOOPSTestClass4)
  Add
  (QUOTE IVProp)
  (QUOTE AIV1)
  (QUOTE LTP22)
  (QUOTE Prop1)))
[TestExpr (EQUAL (GetValue ($ LOOPSTestClass4)
  (QUOTE AIV1)
  (QUOTE Prop1))
  (QUOTE LTP22)))
(ResetExp (SEND ($ LOOPSTestClass4)
  Delete
  (QUOTE IVProp)
  (QUOTE AIV1)
  (QUOTE Prop1)))
(SubTest #.NotSetValue Tested? #.NotSetValue)
(PreTest #. ($A (LTP9)
  BuildPreTest MakeBackLink)
  Tested? #.NotSetValue)
(AltTest #.NotSetValue Tested? #.NotSetValue)
(SyntaxTest #.NotSetValue Tested? #.NotSetValue)
(UsesObj #. ($A (LOOPSTestClass4)
  NIL AllowRemove))
(SubTestOf #. ($A (LTMAdd)

```



```

        NIL MakeBackLink)))
(DEFINST LOOPSTestPrimitive (LTP21 "OOR@@@AÔ")
  (name #. ($A LTP21 NIL RememberName))
  (TestDesc "Add method: CV")
  (SetUp (SEND ($ LOOPSTestClass4)
    Add
    (QUOTE CV)
    (QUOTE ACV1)
    (QUOTE LTP21)))
  (TestExpr (EQUAL (GetClassValue ($ LOOPSTestClass4)
    (QUOTE ACV1))
    (QUOTE LTP21)))
  (ResetExp (SEND ($ LOOPSTestClass4)
    Delete
    (QUOTE CV)
    (QUOTE ACV1)))
  (SubTest #. ($A (LTP23)
    NIL MakeBackLink)
    Tested? #.NotSetValue)
  (PreTest #. ($A (LTFGetClassValue)
    BuildPreTest MakeBackLink)
    Tested? #.NotSetValue)
  (AltTest #.NotSetValue Tested? #.NotSetValue)
  (SyntaxTest #.NotSetValue Tested? #.NotSetValue)
  (UsesObj #. ($A (LOOPSTestClass4)
    NIL AllowRemove))
  (SubTestOf #. ($A (LTMAdd)
    NIL MakeBackLink)))
(DEFINST LOOPSTestPrimitive (LTP64 "O\R@@@[")
  (name #. ($A LTP64 NIL RememberName))
  (TestDesc "List method in CLASS: CVs")
  [TestExpr (EQUAL (SEND ($ LOOPSTestClassL1)
    List
    (QUOTE CVs))
    (QUOTE (LC1 LC2 LC3))
  (SubTest #.NotSetValue Tested? #.NotSetValue)
  (PreTest #.NotSetValue Tested? #.NotSetValue)
  (AltTest #.NotSetValue Tested? #.NotSetValue)
  (SyntaxTest #.NotSetValue Tested? #.NotSetValue)
  (UsesObj #. ($A (LOOPSTestClassL1)
    NIL AllowRemove))
  (SubTestOf #. ($A (LTMListInClass)
    NIL MakeBackLink)))
(DEFINST LOOPSTestPrimitive (LTP63 "O\R@@@[Ž")
  (name #. ($A LTP63 NIL RememberName))
  (TestDesc "List in CLASS: Methods")
  [TestExpr (EQUAL (SEND ($ LOOPSTestClassL1)
    List
    (QUOTE Methods))
    (QUOTE (M1 M2 M3))
  (SubTest #. ($A (LTP65)
    NIL MakeBackLink)
    Tested? #.NotSetValue)
  (PreTest #.NotSetValue Tested? #.NotSetValue)
  (AltTest #.NotSetValue Tested? #.NotSetValue)
  (SyntaxTest #.NotSetValue Tested? #.NotSetValue)
  (UsesObj #. ($A (LOOPSTestClassL1)
    NIL AllowRemove))
  (SubTestOf #. ($A (LTMListInClass)
    NIL MakeBackLink)))
(DEFINST LOOPSTestPrimitive (LTP62 "O\R@@@[Ÿ")
  (name #. ($A LTP62 NIL RememberName))
  (TestDesc "List in Class: Supers")
  [TestExpr (EQUAL (SEND ($ LOOPSTestClassL1)
    List
    (QUOTE Supers))
    (QUOTE (LOOPSTestCases))
  (SubTest #.NotSetValue Tested? #.NotSetValue)
  (PreTest #.NotSetValue Tested? #.NotSetValue)
  (AltTest #.NotSetValue Tested? #.NotSetValue)
  (SyntaxTest #.NotSetValue Tested? #.NotSetValue)
  (UsesObj #. ($A (LOOPSTestClassL1)
    NIL AllowRemove))
  (SubTestOf #. ($A (LTMListInClass)
    NIL MakeBackLink)))
(DEFINST LOOPSTestPrimitive (LTP61 "O\R@@@[Û")
  (name #. ($A LTP61 NIL RememberName))
  (TestDesc "List in CLASS: Functions")
  [TestExpr (EQUAL (SEND ($ LOOPSTestClassL1)
    List
    (QUOTE Functions))
    (QUOTE (LOOPSTestClassL1.M1 LOOPSTestClassL1.M2 LOOPSTestClassL1.M3])

```

```
(SubTest #.NotSetValue Tested? #.NotSetValue)
(PreTest #.NotSetValue Tested? #.NotSetValue)
(AltTest #.NotSetValue Tested? #.NotSetValue)
(SyntaxTest #.NotSetValue Tested? #.NotSetValue)
(UsesObj #. ($A (LOOPSTestClassL1)
                NIL AllowRemove))
(SubTestOf #. ($A (LTMListInClass)
                  NIL MakeBackLink))
```

```
(DEFINST LOOPSTestPrimitive (LTP60 "O\R@[Û]")
  (name #. ($A LTP60 NIL RememberName))
  (TestDesc "List in CLASS: IVProps")
  [TestExpr (AND (EQUAL (SEND ($ LOOPSTestClassL1)
                              List
                              (QUOTE IVProps)
                              (QUOTE L2))
                      (QUOTE (Prop1 Prop2)))
                (NULL (SEND ($ LOOPSTestClassL1)
                              List
                              (QUOTE IVProps)
                              (QUOTE L3))
                      (QUOTE (Prop1 Prop2))))
            (SubTest #.NotSetValue Tested? #.NotSetValue)
            (PreTest #.NotSetValue Tested? #.NotSetValue)
            (AltTest #.NotSetValue Tested? #.NotSetValue)
            (SyntaxTest #.NotSetValue Tested? #.NotSetValue)
            (UsesObj #. ($A (LOOPSTestClassL1)
                            NIL AllowRemove))
            (SubTestOf #. ($A (LTMListInClass)
                              NIL MakeBackLink))
```

```
(DEFINST LOOPSTestPrimitive (LTP59 "O\R@[Û]")
  (name #. ($A LTP59 NIL RememberName))
  (TestDesc "List in CLASS: CVProps")
  [TestExpr (AND (EQUAL (SEND ($ LOOPSTestClassL1)
                              List
                              (QUOTE CVProps)
                              (QUOTE LC1))
                      (QUOTE (Prop1 Prop2)))
                (NULL (SEND ($ LOOPSTestClassL1)
                              List
                              (QUOTE CVProps)
                              (QUOTE LC3))
                      (QUOTE (Prop1 Prop2))))
            (SubTest #.NotSetValue Tested? #.NotSetValue)
            (PreTest #.NotSetValue Tested? #.NotSetValue)
            (AltTest #.NotSetValue Tested? #.NotSetValue)
            (SyntaxTest #.NotSetValue Tested? #.NotSetValue)
            (UsesObj #. ($A (LOOPSTestClassL1)
                            NIL AllowRemove))
            (SubTestOf #. ($A (LTMListInClass)
                              NIL MakeBackLink))
```

```
(DEFINST LOOPSTestPrimitive (LTP46 "OVR@AÉÁ")
  (name #. ($A LTP46 NIL RememberName))
  (TestDesc "Delete in CLASS: CVProp")
  [SetUp (PROGN (SEND ($ LOOPSTestClass5)
                     Delete
                     (QUOTE CVProp)
                     (QUOTE DCV2)
                     (QUOTE Prop1))
              (SETQ LTV2 (SEND ($ LOOPSTestClass5)
                               List
                               (QUOTE CVProps)
                               (QUOTE DCV2))
              (TestExpr (AND (NOT (FMEMB (QUOTE Prop1)
                                         LTV2))
                            (FMEMB (QUOTE doc)
                                    LTV2)))
            (ResetExp (SEND ($ LOOPSTestClass5)
                            Add
                            (QUOTE CVProp)
                            (QUOTE DCV2)
                            (QUOTE DCV2)
                            (QUOTE Prop1)))
            (SubTest #.NotSetValue Tested? #.NotSetValue)
            (PreTest #.NotSetValue Tested? #.NotSetValue)
            (AltTest #.NotSetValue Tested? #.NotSetValue)
            (SyntaxTest #.NotSetValue Tested? #.NotSetValue)
            (UsesObj #. ($A (LOOPSTestClass5)
                            NIL AllowRemove))
            (SubTestOf #. ($A (LTP41)
                              NIL MakeBackLink))
```

```
(DEFINST LOOPSTestPrimitive (LTP23 "OOR@AÀ")
  (name #. ($A LTP23 NIL RememberName))
  (TestDesc "Add Method: CVProp")
  [SetUp (SEND ($ LOOPSTestClass4)
              Add
```

```

      (QUOTE CVProp)
      (QUOTE ACV1)
      (QUOTE LTP23)
      (QUOTE Prop1))
(TestExpr (EQUAL (GetClassValue ($ LOOPSTestClass4)
      (QUOTE ACV1)
      (QUOTE Prop1))
      (QUOTE LTP23)))
(ResetExp (SEND ($ LOOPSTestClass4)
  Delete
  (QUOTE CVProp)
  (QUOTE ACV1)
  (QUOTE Prop1)))
(SubTest #.NotSetValue Tested? #.NotSetValue)
(PreTest #.NotSetValue Tested? #.NotSetValue)
(AltTest #.NotSetValue Tested? #.NotSetValue)
(SyntaxTest #.NotSetValue Tested? #.NotSetValue)
(UsesObj #. ($A (LOOPSTestClass4)
  NIL AllowRemove))
(SubTestOf #. ($A (LTP21)
  NIL MakeBackLink)))

```

```

(DEFINST LOOPSTestPrimitive (LTP44 "OVR@@A` ")
  (name #. ($A LTP44 NIL RememberName))
  (TestDesc "Delete in CLASS: Supers")
  [SetUp (PROGN (SEND ($ LOOPSTestClass5)
    Delete
    (QUOTE Super)
    (QUOTE LOOPSTestSuper))
    (SETQ LTV1 (SEND ($ LOOPSTestClass5)
      List
      (QUOTE Supers]
  (TestExpr (AND (NOT (FMEMB (QUOTE LOOPSTestSuper)
    LTV1))
    (FMEMB (QUOTE NamedObject)
    LTV1)))
  (ResetExp (SEND ($ LOOPSTestClass5)
    Add
    (QUOTE Super)
    (QUOTE LOOPSTestSuper)))
  (UsesObj #. ($A (LOOPSTestClass5)
    NIL AllowRemove))
  (SubTestOf #. ($A NIL NIL MakeBackLink)))

```

```

(DEFINST LOOPSTestPrimitive (LTP25 "OOR@@AIP")
  (name #. ($A LTP25 NIL RememberName))
  (TestDesc "Add Method: IVProp to a inherited IV")
  (SetUp (SEND ($ LOOPSTestClass4)
    Add
    (QUOTE IVProp)
    (QUOTE AIV2)
    (QUOTE LTP25)
    (QUOTE Prop1)))
  [TestExpr (AND (EQUAL (GetValue ($ LOOPSTestClass4)
    (QUOTE AIV2)
    (QUOTE Prop1))
    (QUOTE LTP25))
    (EQUAL (GetValue ($ LOOPSTestSuper)
    (QUOTE AIV2)
    (QUOTE Prop1))
    (QUOTE Orig]
  (ResetExp (SEND ($ LOOPSTestClass4)
    Delete
    (QUOTE IVProp)
    (QUOTE AIV2)
    (QUOTE Prop1)))
  (SubTest #.NotSetValue Tested? #.NotSetValue)
  (PreTest #.NotSetValue Tested? #.NotSetValue)
  (AltTest #.NotSetValue Tested? #.NotSetValue)
  (SyntaxTest #.NotSetValue Tested? #.NotSetValue)
  (UsesObj #. ($A (LOOPSTestSuper LOOPSTestClass4)
    NIL AllowRemove))
  (SubTestOf #. ($A NIL NIL MakeBackLink)))

```

VARIABLE INDEX

LMethodInstances1
