

File created: 14-Aug-90 16:11:42 {DSK}<usr>local>lde>loops>src>SYSTEM>LOOPSINSPECT.;3

changes to: (CLASSES InspectorClassIVs)
(VARS LOOPSINSPECTCOMS)

previous date: 13-Aug-90 16:24:53 {DSK}<usr>local>lde>loops>src>SYSTEM>LOOPSINSPECT.;2

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

::
:: Copyright (c) 1983, 1984, 1985, 1986, 1987, 1988, 1990 by Venue & Xerox Corporation. All rights reserved.

(RPAQQ **LOOPSINSPECTCOMS**

[

::: Classes and functions which make instances and classes be inspectable through the ordinary LISP inspector

```
(CLASSES InspectorClassIVs)
(ADDVARS (INSPECTMACROS (instance ObjPropertiesFn ObjInspectFetchFn ObjInspectStoreFn ObjPropCommandFn
                        ObjValueCommandFn ObjTitleCommand ObjTitleFn)
          (class ObjPropertiesFn ObjInspectFetchFn ObjInspectStoreFn ObjPropCommandFn
                ObjValueCommandFn ObjTitleCommand ObjTitleFn)
          (annotatedValue ObjPropertiesFn ObjInspectFetchFn ObjInspectStoreFn ObjPropCommandFn
                ObjValueCommandFn ObjTitleCommand ObjTitleFn)))
```

::: Functions for producing the ClassTree

```
(METHODS Class.InspectFetch Class.InspectPropCommand Class.InspectProperties Class.InspectStore
          Class.InspectTitle Class.InspectTitleMenu Class.InspectValueCommand Class.TitleCommand
          InspectorClassIVs.InspectFetch InspectorClassIVs.InspectProperties InspectorClassIVs.InspectStore
          InspectorClassIVs.InspectTitle InspectorClassIVs.InspectTitleMenu
          InspectorClassIVs.InspectValueCommand InspectorClassIVs.TitleCommand Object.InspectFetch
          Object.InspectPropCommand Object.InspectProperties Object.InspectStore Object.InspectTitle
          Object.InspectTitleMenu Object.InspectValueCommand Object.TitleCommand)
(FNS InspectValProp ObjAddDelete ClassIVAddDelete ObjInspectFetchFn ObjInspectStoreFn ObjPropCommandFn
  ObjPropertiesFn ObjTitleCommand ObjTitleFn ObjValueCommandFn)
```

::: JRB - Patch for Lyric Inspector bug

```
(FNS INSPECTW.CREATE)
(DECLARE%: EVAL@COMPILE DONTCOPY (EXPORT (MACROS ExtractObj))
```

::: Classes and functions which make instances and classes be inspectable through the ordinary LISP inspector

(DEFCLASSES InspectorClassIVs)

```
(DEFCLASS InspectorClassIVs (MetaClass Class doc "Special class used by inspector to keep track of whether it is
displaying all IVs or only local IVs." Edited%:
(* dbg%: "22-JUN-82 11:57"))
```

```
(Supers Object)
(InstanceVariables (class NIL doc "class whose IVs are being examined")
  (iv NIL doc "name of IV whose properties are being examined in class")
  (localFlg NIL doc "T when only local properties are to be seen"))
```

```
(ADDTOVAR INSPECTMACROS (instance ObjPropertiesFn ObjInspectFetchFn ObjInspectStoreFn ObjPropCommandFn
                                ObjValueCommandFn ObjTitleCommand ObjTitleFn)
  (class ObjPropertiesFn ObjInspectFetchFn ObjInspectStoreFn ObjPropCommandFn
        ObjValueCommandFn ObjTitleCommand ObjTitleFn)
  (annotatedValue ObjPropertiesFn ObjInspectFetchFn ObjInspectStoreFn ObjPropCommandFn
        ObjValueCommandFn ObjTitleCommand ObjTitleFn))
```

::: Functions for producing the ClassTree

(\BatchMethodDefs)

```
(METH Class InspectFetch (datum property)
  "Fetch summaries of class for the inspector -- different for Metaclass Supers etc"
  (category (Object)))
```

```
(METH Class InspectPropCommand NIL "No changes allowed on properties" (category (Object)))
```

```
(METH Class InspectProperties (datum)
  "(part of Class Inspector)"
  (category (Object)))
```

```
(METH Class InspectStore (datum property newValue window)
  "Can't change the class from the inspector using the summary"
  (category (Object)))
```

```

(METH Class InspectTitle (datum)
  "Return a string for a title in inspect window"
  (category (Object)))

(METH Class InspectTitleMenu NIL "Menu for commands for the inspector" (category (Object)))

(METH Class InspectValueCommand (datum property value window)
  "What to do when a class value is selected"
  (category (Object)))

(METH Class TitleCommand (datum window)
  "Do commands in title field"
  (category (Object)))

(METH InspectorClassIVs InspectFetch (datum property window)
  "Internal Inspector Fn"
  (category (Object)))

(METH InspectorClassIVs InspectProperties (datum)
  "Compute the list of IVs for the represented class"
  (category (Object)))

(METH InspectorClassIVs InspectStore (datum property newValue window)
  "Put in a new default value for iv in class."
  (category (Object)))

(METH InspectorClassIVs InspectTitle (datum)
  "Compute the title for this instance which is representing the ivs of another class"
  (category (Object)))

(METH InspectorClassIVs InspectTitleMenu NIL "Puts up menu for IV inspector title" (category (Object)))

(METH InspectorClassIVs InspectValueCommand (datum property value window)
  "Puts up menu for IV inspector value selection"
  (category (Object)))

(METH InspectorClassIVs TitleCommand (datum window)
  "Puts up menu of selections for class instance variables for the Inspector"
  (category (Object)))

(METH Object InspectFetch (datum property window)
  "part of Object Inspector"
  (category (Object)))

(METH Object InspectPropCommand (datum property window)
  "Part of object inspector"
  (category (Object)))

(METH Object InspectProperties (datum)
  "List the set of IVs for an instance"
  (category (Object)))

(METH Object InspectStore (datum property newValue window)
  "Store a value in the instance"
  (category (Object)))

(METH Object InspectTitle (datum)
  "Return the title that should be used in an inspect window looking at self"
  (category (Object)))

(METH Object InspectTitleMenu NIL "Put a menu for the Instance Title Command" (category (Object)))

(METH Object InspectValueCommand (datum property value window)
  "part of Object Inspector"
  (category (Object)))

(METH Object TitleCommand (datum window)
  "Puts up menu of selections for instances for the Inspector"
  (category (Object)))

(Method ((Class InspectFetch) self datum property) ; smL 24-Apr-86 12:48
  "Fetch summaries of class for the inspector -- different for Metaclass Supers etc"
  (COND
    ((LISTP datum) ; (* List all values, even inherited)
     (_ (CAR datum)
        ListAttribute! property))
    (T ; (* List only local values)
     (_ datum ListAttribute property))))

(Method ((Class InspectPropCommand) self) ; dgb: 10-JUN-82 16:47
  "No changes allowed on properties"
  NIL)

(Method ((Class InspectProperties) self datum) ; dgb: 27-AUG-82 13:01
  "(part of Class Inspector)"
  '(Metaclass Supers IVs CVs Methods))

```

```
(Method ((Class InspectStore) self datum property newValue window)
; dgb: 11-JUN-82 09:52
"Can't change the class from the inspector using the summary"
(ERROR "Can't Store into class this way"))
```

```
(Method ((Class InspectTitle) self datum)
; dgb: 11-JUN-82 09:53
"Return a string for a title in inspect window"
(CONCAT (COND
  ((LISTP datum)
   "All properties of ")
  (T "Local properties of "))
"Class "
(ClassName (ExtractObj self))))
```

```
(Method ((Class InspectTitleMenu) self)
; smL 8-Apr-87 11:32
"Menu for commands for the inspector"
[MENU (MenuGetOrCreate ClassInspectTitleMenu ' ((Browse 'Browse "Browse the class and its sub-classes"
(SUBITEMS (Browse 'Browse "Browse the class and its
sub-classes")
(BrowseSupers 'BrowseSupers "Browse the class
and its super-classes")))
(Edit 'Edit "Edit the class")
(All 'All "Show all the IVs, including the inherited ones")
(Local 'Local "Show only the local IVs, excluding the
inherited ones")
(Refetch 'Refetch "Recompute the contents of the inspector"])]
```

```
(Method ((Class InspectValueCommand) self datum property value window)
; smL 8-Apr-87 11:34
"What to do when a class value is selected"
(SELECTQ [MENU (MenuGetOrCreate ClassInspectMenu ' ((Inspect 'Inspect "Inspect the value"]
(Inspect (COND
  ((EQ property 'IVs)
   (LET ((civ (_ ($ InspectorClassIVs
New))
  (_@
civ class self)
(INSPECT civ)))
(T (INSPECT value))))
NIL))
```

```
(Method ((Class TitleCommand) self datum window)
; smL 9-Jun-86 19:26
"Do commands in title field"
(SELECTQ (_ self InspectTitleMenu)
(Browse (Browse self))
(BrowseSupers (_New
($ SupersBrowser)
Browse self))
(Edit (_ self Edit)
(INSPECTW.REDISPLAY window))
(All (COND
  ((NLISTP datum)
   (WINDOWPROP window 'DATUM (LIST datum))
   (INSPECTW.REDISPLAY window))))
(Local (COND
  ((LISTP datum)
   (WINDOWPROP window 'DATUM self)
   (INSPECTW.REDISPLAY window))))
(Refetch (INSPECTW.REDISPLAY window)
NIL))
```

```
(Method ((InspectorClassIVs InspectFetch) self datum property window)
; dgb: 13-OCT-83 22:06
"Internal Inspector Fn"
(GetClassIV (@ class)
(OR (@ iv)
property)
(AND (@ iv)
(InspectValProp property))))
```

```
(Method ((InspectorClassIVs InspectProperties) self datum)
; smL 11-Apr-86 15:01
"Compute the list of IVs for the represented class"
[COND
  [(@ iv)
  (* examining the properties of an iv.
NIL is for the value)
(CONS 'Value (COND
  ((@ localFlg)
  (* Only local info)
  (_ (@ class)
ListAttribute
'IVProps
(@ iv)))
(T
  (* Get inherited info)
  (_ (@ class)
ListAttribute!
'IVProps
(@ iv))
```

```

(T (COND
  ((@ localFlg)
   (_ (@ class)
      ListAttribute
      'IVs))
  (T
   (_ (@ class)
      ListAttribute!
      'IVs}))
  (* Get inherited info)

```

```

(Method ((InspectorClassIVs InspectStore) self datum property newValue window)
; dgb: 13-OCT-83 22:06

```

```

"Put in a new default value for iv in class."
;; The OR and AND are a trick to compute correct arguments. iv is a varname if it exists, otherwise it is
;; the property

```

```

(PutClassIV (@ class)
  (OR (@ iv)
   property)
  newValue
  (AND (@ iv)
   (InspectValProp property))))

```

```

(Method ((InspectorClassIVs InspectTitle) self datum) ; smL 30-Sep-86 14:50
"Compute the title for this instance which is representing the ivs of another class"

```

```

(CONCAT (COND
  ((@ localFlg)
   "Local ")
  (T "All "))
(COND
  ((@ iv)
   (CONCAT "properties of " (@ iv)))
  (T "IVs")))
" of Class "
(Classname (@ class))))

```

```

(Method ((InspectorClassIVs InspectTitleMenu) self) ; smL 30-Sep-86 14:46

```

```

"Puts up menu for IV inspector title"
[MENU (MenuGetOrCreate ClassIVsInspectTitleMenu '((AllValues 'All "Show all IVs of this class, including
inherited ones")
(LocalValues 'Local "Show only locally defined IVs of this
class")
(Add/Delete 'Add/Delete "Add or delete an IV or property
from the class")
(IVs 'IVs "Show the IVs of the class")
(Refetch 'Refetch "Recompute and redisplay the inspector"))

```

```

(Method ((InspectorClassIVs InspectValueCommand) self datum property value window)
; smL 5-Aug-86 17:42

```

```

"Puts up menu for IV inspector value selection"
(SELECTQ [MENU (create MENU
  CHANGEOFFSETFLG _ T
  ITEMS _ `((Inspect 'Inspect "Inspect value of item")
  ("Save Value" 'SaveIt "(SavedValue)_<selectedItem>")
  ,@ (if (NULL (@ iv))
  then '((Properties 'Props "Inspect Properties of this Variable"]
(Inspect (INSPECT value))
(SaveIt (printout PROMPTWINDOW "(SavedValue)_" (PutSavedValue value)
T))
(Props (change (@ iv)
  property)
  (INSPECTW.REDISPLAY window))
(* Store away the iv to be examined, and redisplay)
NIL))

```

```

(Method ((InspectorClassIVs TitleCommand) self datum window) ; smL 30-Sep-86 14:49

```

```

"Puts up menu of selections for class instance variables for the Inspector"
(SELECTQ (_ self InspectTitleMenu)
  (Class
   (WINDOWPROP window 'DATUM (@ class)))
  (Add/Delete (ClassIVAddDelete self datum window)
  (INSPECTW.REDISPLAY window))
  (All
   (change (@ localFlg)
    NIL)
   (INSPECTW.REDISPLAY window))
  (IVs (change (@ iv)
  NIL)
  (INSPECTW.REDISPLAY window))
  (Local
   (change (@ localFlg)
    T)
   (INSPECTW.REDISPLAY window))
  (Refetch (INSPECTW.REDISPLAY window))
  NIL))
(* Inspect the class of this instance)
(* Reset to inherited mode)
(* Show only those values which are in instance itself)

```

```

(Method ((Object InspectFetch) self datum property window) ; dgb: 26-NOV-82 23:25

```

```

"part of Object Inspector"

```

```

[COND
  ((NLISTP datum)
   (GetValueOnly self property)) (* getting the usual value)
  (T (PROG (varName prop (vp (CADR datum)))
    (COND
      (vp (SETQ varName vp)
        (SETQ prop (InspectValProp property)))
      (T (SETQ varName property)))
    (RETURN (COND
      ((EQ 'LocalValues (CADR datum)) (* These are the values not stored in the class)
       (GetIVHere self varName prop))
      (T (GetValueOnly self varName prop)))))

(Method ((Object InspectPropCommand) self datum property window)
  ;smL 10-Sep-86 19:34

"Part of object inspector"
(PROG (newValue varName prop)
  (AND (type? instance datum)
   (_ datum InstOf 'InspectorClassIVs)
   (RETURN NIL)) ; Can't set class values from inspector

;; Set up varName and property
[COND
  ((OR (NLISTP datum)
   (NULL (CADR datum)))
   (SETQ varName property))
  (T (SETQ varName (CADR datum))
   (SETQ prop (InspectValProp property)) ; Find out what the user wants to do

;; Get a new value, or RETURN NIL if ERROR or ^E
(SELECTQ [COND
  [(AND (LISTP datum)
   (CADR datum)) ; Already inspecting properties
  (if (EQ property 'Value)
    then [MENU (MenuGetOrCreate ObjPropInspectPropValMenu
      '((PutValue 'PutValue "Set value, possibly invoking activeValues"
        (SUBITEMS (PutValue 'PutValue "Set value, possibly invoking
          activeValues")
          (PutValueOnly 'PutValueOnly "Overwrite existing
            value")
          ("Use saved value" 'UnSaveIt "Replace selected value
            by saved value"]
    else (MENU (MenuGetOrCreate ObjPropInspectPropsMenu
      '((PutValue 'PutValue "Set value, possibly invoking activeValues"
        (SUBITEMS (PutValue 'PutValue "Set value, possibly invoking
          activeValues")
          (PutValueOnly 'PutValueOnly "Overwrite existing value")
          ("Use saved value" 'UnSaveIt "Replace selected value by
            saved value"))
      (BreakIt 'BreakIt "Place a BREAK around the property"
        (SUBITEMS ("Break on Put" 'BreakOnPut "Place a BREAK around
          the property that is triggered when the value
            is changed")
          ("Break on access" 'BreakIt "Place a BREAK around the
            property that is triggered when the value is
              accessed"))
      (TraceIt 'TraceIt "Place a TRACE around the property"
        (SUBITEMS ("Trace on Put" 'TraceOnPut "Place a TRACE around
          the property that is triggered when the value
            is changed")
          ("Trace on access" 'TraceIt "Place a TRACE around the
            property that is triggered when the value is
              accessed"))
      (UnBreakIt 'UnBreakIt "Remove the BREAK from the property"]
    (T (MENU (MenuGetOrCreate ObjPropInspectMenu '((PutValue 'PutValue "Set value, possibly
      invoking activeValues"
        (SUBITEMS (PutValue 'PutValue "Set
          value, possibly
            invoking
              activeValues")
          (PutValueOnly 'PutValueOnly
            "Overwrite existing
              value")
          ("Use saved value"
            'UnSaveIt "Replace selected
              value by saved value"))
      (Properties 'Props "Inspect Properties of
        this Variable")
      (BreakIt 'BreakIt "Place a BREAK around the
        value" (SUBITEMS ("Break on Put"
          'BreakOnPut
            "Place a BREAK
              around the value
                that is triggered
                  when the value is
                    changed")
          ("Break on access"

```

```

'BreakIt "Place a
BREAK around the value
that is triggered when
the value is
accessed"))
(TraceIt 'TraceIt "Place a TRACE around the
value" (SUBITEMS ("Trace on Put"
'TraceOnPut
"Place a TRACE
around the value
that is triggered
when the value is
changed")
("Trace on access"
'TraceIt "Place a
TRACE around the value
that is triggered when
the value is
accessed"))
(UnBreakIt 'UnBreakIt "Remove the BREAK from
the value")
(PutValue [SETQ newValue (CAR (OR (ERSETQ (PromptEval (CONCAT "Enter the new " property " for "
datum)
(GETPROMPTWINDOW window 3)
T))
(RETURN NIL]
(CLOSEW (GETPROMPTWINDOW window))
(PutValue self varName newValue prop)
(INSPECTW.REDISPLAY window property))
(PutValueOnly [SETQ newValue (CAR (OR (ERSETQ (PromptEval (CONCAT "Enter the new " property " for
datum)
(GETPROMPTWINDOW window 3)
T))
(RETURN NIL]
(CLOSEW (GETPROMPTWINDOW window))
(PutValueOnly self varName newValue prop)
(INSPECTW.REDISPLAY window property))
(UnSaveIt (_ self InspectStore datum property (SavedValue)
window)
(INSPECTW.REDISPLAY window property))
(UnBreakIt (UnBreakIt self (COND
((AND (LISTP datum)
(CADR datum))
(CADR datum))
(T property))
(COND
((AND (LISTP datum)
(CADR datum))
property)
(T NIL))
'IV)
(INSPECTW.REDISPLAY window property))
(TraceOnPut (TraceIt self (COND
((AND (LISTP datum)
(CADR datum))
(CADR datum))
(T property))
(COND
((AND (LISTP datum)
(CADR datum))
property)
(T NIL))
'IV)
(INSPECTW.REDISPLAY window property))
(TraceIt (TraceIt self (COND
((AND (LISTP datum)
(CADR datum))
(CADR datum))
(T property))
(COND
((AND (LISTP datum)
(CADR datum))
property)
(T NIL))
'IV T)
(INSPECTW.REDISPLAY window property))
(BreakOnPut (BreakIt self (COND
((AND (LISTP datum)
(CADR datum))
(CADR datum))
(T property))
(COND
((AND (LISTP datum)
(CADR datum))
property)
(T NIL))
'IV)

```

```

      (INSPECTW.REDISPLAY window property))
(BreakIt (BreakIt self (COND
  ((AND (LISTP datum)
        (CADR datum)
        (CADR datum)
        (T property))
  (COND
    ((AND (LISTP datum)
          (CADR datum)
          property)
     (T NIL))
    'IV T)
(INSPECTW.REDISPLAY window property))
(Props (PROG [goodSize (reg (WINDOWPROP window 'REGION)
  (COND
    [(NLISTP datum)
     (WINDOWPROP window 'DATUM (SETQ datum (LIST self property NIL)
      (T (RPLACA (CDR datum)
                property)))]
  [COND
    ([ILESSP (fetch HEIGHT of reg)
             (SETQ goodSize (IMIN 200 (ITIMES (IPLUS 2 (LENGTH (_ (CAR datum)
                                                                    ListAttribute!
                                                                    'IVProps property)))]
                                                                    13]
             ; Won't contain the items. Reshape window
             (SHAPEW window (create REGION copying reg HEIGHT _ goodSize)
              (INSPECTW.REDISPLAY window)))
  NIL))

```

(Method ((Object InspectProperties) self datum) ; smL 11-Apr-86 15:01

```

"List the set of IVs for an instance"
[COND
  ((OR (NLISTP datum)
        (NULL (CADR datum)))
   (_ self ListAttribute 'IVS))
  (T (CONS 'Value (COND
    ((CADDR datum) (* only local properties)
     (_ self ListAttribute 'IV (CADR datum)))
    (T (* all properties)
     (_ self ListAttribute! 'IV (CADR datum))

```

(Method ((Object InspectStore) self datum property newValue window) ; dgb: 26-NOV-82 23:26

```

"Store a value in the instance"
[COND
  ((NLISTP datum)
   (PutValueOnly self property newValue))
  (T
   (* If (CADR datum) NEQ NIL then it is varName, and property is propName, else property is varName)

   (PROG ((varName (CADR datum)))
    (PutValueOnly self (OR varName property)
     newValue
     (AND varName (InspectValProp property))

```

(Method ((Object InspectTitle) self datum) ; smL 2-May-86 11:53

```

"Return the title that should be used in an inspect window looking at self"
[PROG ((names (GetObjectNames self T)))
  (OR (LISTP datum)
      (SETQ datum (LIST datum NIL NIL)))
  (RETURN (CONCAT (COND
    ((CADDR datum)
     "Local ")
    (T "All "))
    (COND
     ((CADR datum)
      "IVProps of ")
     (T "Values of "))
    (ClassName self)
    " "
    (COND
     (names (LIST "$" (CAR names)))
     (T (LOC self)))
    ". "
    (OR (CADR datum)
        ""])]

```

(Method ((Object InspectTitleMenu) self) ; smL 17-Apr-86 15:05

```

"Put a menu for the Instance Title Command"
[MENU (MenuGetOrCreate InstanceTitleMenu ' ((Class 'Class "Inspect the class of this instance")
  (AllValues 'AllValues "Look at inherited values as well as local values")
  (LocalValues 'LocalValues "Show only values actually in instance")
  (Add/Delete 'Add/Delete "Add or delete from the IVs or props")
  (IVs 'IVs "Show object's instance variables")
  ("Save Value" 'SaveIt " (SavedValue)_<object being inspected>")

```

(Refetch 'Refetch "Refetch current values")
(Edit 'Edit "Edit the object being inspected"])

(Method ((Object InspectValueCommand) self datum property value window)
;sml 5-Aug-86 17:47

"part of Object Inspector"

(PROG (newValue varName prop selection)
(AND (Instance? datum)
(_ datum InstOf 'InspectorClassIVs)
(RETURN NIL))

(* Can't set class values from inspector)

(* * Set up varName and property)

[COND

((OR (NLISTP datum)
(NULL (CADR datum)))
(SETQ varName property))
(T (SETQ varName (CADR datum))
(SETQ prop (InspectValProp property))
(SELECTQ [SETQ selection (COND

[(AND (LISTP datum)
(CADR datum))

(MENU (MenuGetOrCreate InspectOrSaveMenu

'((PutValue 'PutValue "Set value, possibly invoking
activeValues" (SUBITEMS (PutValue 'PutValue "Set
value, possibly
invoking
activeValues")
(PutValueOnly 'PutValueOnly
"Overwrite existing
value")
("Use (SavedValue)"
'UnSaveIt "Replace selected
value by saved value")))

("Save Value" 'SaveIt "Save selected value in
(SavedValue)")

(Inspect 'Inspect "Inspect value of item"]

(T (MENU (MenuGetOrCreate InspectOrPropsMenu

'((PutValue 'PutValue "Set value, possibly invoking
activeValues" (SUBITEMS (PutValue 'PutValue
"Set value,
possibly invoking
activeValues")
(PutValueOnly 'PutValueOnly
"Overwrite existing
value")
("Use saved value"
'UnSaveIt "Replace selected
value by saved value")))

(Properties 'Props "Inspect Properties of this Variable")

(BreakIt 'BreakIt "Place a BREAK around the value"

(SUBITEMS ("Break on Put" 'BreakOnPut "Place a
BREAK around the value that is
triggered when the value is
changed")

("Break on access" 'BreakIt "Place a BREAK
around the value that is triggered
when the value is accessed")))

(TraceIt 'TraceIt "Place a TRACE around the value"

(SUBITEMS ("Trace on Put" 'TraceOnPut "Place a
TRACE around the value that is
triggered when the value is
changed")

("Trace on access" 'TraceIt "Place a TRACE
around the value that is triggered
when the value is accessed")))

(UnBreakIt 'UnBreakIt "Remove the BREAK from the value")

("Save Value" 'SaveIt "Save selected value in
(SavedValue)")
(Inspect 'Inspect "Inspect value of item"]

(UnSaveIt (_ self InspectStore datum property (SavedValue)
window)

(INSPECTW.REPLACE window property (_ self InspectFetch datum property window))

(PutValueOnly [SETQ newValue (CAR (OR (ERSETQ (PromptEval (CONCAT "Enter the new " property " for

" datum)

(GETPROMPTWINDOW window 3)
T))

(RETURN NIL]

(CLOSEW (GETPROMPTWINDOW window))

(PutValueOnly self varName newValue prop)

(INSPECTW.REPLACE window property (_ self InspectFetch datum property window))

(PutValue [SETQ newValue (CAR (OR (ERSETQ (PromptEval (CONCAT "Enter the new " property " for "
datum)

(GETPROMPTWINDOW window 3)
T))

(RETURN NIL]

(CLOSEW (GETPROMPTWINDOW window))


```

(PutValue self varName newValue prop)
(INSPECTW.REPLACE window property (_ self InspectFetch datum property window))
(UnBreakIt (UnBreakIt self (COND
  ((AND (LISTP datum)
        (CADR datum))
   (CADR datum))
  (T property))
  (COND
   ((AND (LISTP datum)
         (CADR datum))
    property)
   (T NIL))
  'IV)
(INSPECTW.REPLACE window property (_ self InspectFetch datum property window))
((TraceIt TraceOnPut)
 (TraceIt self (COND
  ((AND (LISTP datum)
        (CADR datum))
   (CADR datum))
  (T property))
  (COND
   ((AND (LISTP datum)
         (CADR datum))
    property)
   (T NIL))
  'IV
  (EQ 'TraceIt selection))
(INSPECTW.REPLACE window property (_ self InspectFetch datum property window))
((BreakIt BreakOnPut)
 (BreakIt self (COND
  ((AND (LISTP datum)
        (CADR datum))
   (CADR datum))
  (T property))
  (COND
   ((AND (LISTP datum)
         (CADR datum))
    property)
   (T NIL))
  'IV
  (EQ 'BreakIt selection))
(INSPECTW.REPLACE window property (_ self InspectFetch datum property window))
(Props (PROG [goodSize (reg (WINDOWPROP window 'REGION)
  (COND
   [(NLISTP datum)
    (WINDOWPROP window 'DATUM (SETQ datum (LIST self property NIL)
    (T (RPLACA (CDR datum)
              property)))]
  [COND
   [(ILESSP (fetch HEIGHT of reg)
            (SETQ goodSize (IMIN 200 (ITIMES (IPLUS 2 (LENGTH (_ (CAR datum)
                                                                    ListAttribute!
                                                                    'IVProps property)))]
                                                                    13]
            (* Won't contain the items. Reshape window)
    (SHAPEW window (create REGION copying reg HEIGHT _ goodSize]
  (INSPECTW.REDISPLAY window))
(SaveIt (printout PROMPTWINDOW (CHARACTER 7)
  "(SavedValue) _ "
  (PutSavedValue value)
  T))
(Inspect (INSPECT value))
NIL))

```

```

(Method ((Object TitleCommand) self datum window) ;sml 5-Aug-86 17:44
 "Puts up menu of selections for instances for the Inspector"
 (SELECTQ (_ self InspectTitleMenu)
  (Refetch (INSPECTW.REDISPLAY window)
  (Class (* Inspect the class of this instance)
    (INSPECT (Class self)))
  (Add/Delete (if (ObjAddDelete self datum window)
    then (INSPECTW.REDISPLAY window))
  (AllValues (* Reset to default mode)
    (COND
     ((LISTP datum) (* Remove Local Flag from list)
      (RPLACA (CDDR datum)
              NIL)))
    (INSPECTW.REDISPLAY window))
  (LocalValues (* Show only those values which are in instance itself)
    [COND
     [(NLISTP datum)
      (WINDOWPROP window 'DATUM (SETQ datum (LIST self NIL 'LocalValues]
      (T (* Put in Local values flag in list)
        (RPLACA (CDDR datum)
                'LocalValues]
        (INSPECTW.REDISPLAY window))
  (SaveIt (printout PROMPTWINDOW "(SavedValue) _ " (PutSavedValue (ExtractObj datum))

```

```

    T))
  (IVs
    [COND
      ((LISTP datum)
        (RPLACA (CDR datum)
          (INSPECTW.REDISPLAY window))
      (Edit (_ (ExtractObj datum)
        Edit)
        (INSPECTW.REDISPLAY window))
      NIL)
    datum)
  (\UnbatchMethodDefs)
  (DEFINEQ

```

(InspectValProp

```

[LAMBDA (property)
  (COND
    ((EQ property 'Value)
      NIL)
    (T property]))
(* dgb%: "26-NOV-82 23:27")
(* Used to convert Value to NIL for access in the inspector)

```

(ObjAddDelete

```

[LAMBDA (self datum window) ; Edited 23-Nov-87 11:41 by jrb:

```

(* Used in the inspector. The inspect datum is either an object, or a list (obj varName locOrFree) If varName is NIL then examining all IVs, else the properties of the named var. locOrFree is NIL (showing all values) or LocalValues, meaning show only values set in object itself)

```

(LET [(varName (CADR (LISTP datum)
  (SELECTQ [MENU (MenuGetOrCreate AddDeleteMenu ' (Add Delete)
    (Add (LET* ((pWindow (GETPROMPTWINDOW window 2))
      (name (PromptRead "Type name to be added: " pWindow)))
      (COND
        ((NOT (LITATOM name))
          (printout pWindow name " not an atom" T)
          NIL)
        ((NULL name)
          (printout pWindow "Nothing added" T))
        (varName (PutValueOnly self varName NotSetValue name)
          T)
        (T (_ self AddIV name NotSetValue)
          T))))
      (Delete [LET ([names (if varName
        then (_ self ListAttribute! 'IVPROPS varName)
        else (_ self ListAttribute! 'IVS]
        name)
      (if names
        then (SETQ name (MENU (create MENU
          TITLE _ "Delete which one?"
          ITEMS _ names)))
        (COND
          ((NULL name)
            NIL)
          (varName (_ self DeleteIV varName name)
            T)
          (T (_ self DeleteIV name)
            T))
        else (LET ((pWindow (GETPROMPTWINDOW window 2)))
          (CLEARW pWindow)
          (printout pWindow "Nothing to delete!"))
      NIL])

```

(ClassIVAddDelete

```

[LAMBDA (self datum window) (* smL "30-Sep-86 15:42")

```

(* Function called from the inspector to add or delete an instance variable from a class)

```

(LET [name (varName (AND (_ self InstOf 'InspectorClassIVs)
  (@ iv)
  (SELECTQ [MENU (MenuGetOrCreate AddDeleteMenu ' (Add Delete)
    (Add [LET* [(pWindow (GETPROMPTWINDOW window 2))
      (name (MKATOM (PromptRead "Type name to be added: " pWindow]
      (COND
        (varName (PutValueOnly (@ class)
          varName NIL name))
        (T (_ (@ class)
          Add
          ' IV name NIL])
      (Delete [LET* [(names (if varName
        then (_ (@ class)
          ListAttribute
          ' IV varName)

```



```

; allow PROPERTIES to be a function
(APPLY* PROPERTIES DATUM]
(SETQ VALUEMARGIN (\INSPECTW.VALUE.MARGIN
(COND
(PROPPRINTFN (for PROP in PROPERTIESLST
collect (APPLY* PROPPRINTFN PROP
DATUM)))
(T PROPERTIESLST))
IWFONT))
(SETQ MAXVALUEWIDTH
(COND
(PROPERTIESLST (IMIN (IMAX (bind X for PROP in PROPERTIESLST
largest (STRINGWIDTH (APPLY* FETCHFN
DATUM
PROP)
IWFONT T)
finally (RETURN $$EXTREME))
16)
MaxInspectorPropertyValueWidth))
(T
30)))
; no fields to inspect
(SETQ WINDOW (DECODE.WINDOW.ARG WHERE (IPLUS VALUEMARGIN
MAXVALUEWIDTH)
(IMIN MaxInspectorWindowHeight
(ITIMES (COND
(PROPERTIESLST (LENGTH
PROPERTIESLST
))
(T 1))
(FONTHEIGHT IWFONT)))
(\INSPECT.COMPUTE.TITLE TITLE DATUM)))
(DSPFONT IWFONT WINDOW)
(DSPRIGHTMARGIN 50000 WINDOW)
; for now, can't handle multiple PROPCOMMANDFN output. Put
; right margin way out.
(WINDOWPROP WINDOW 'DATUM DATUM)
; initialize the properties of the window.
(WINDOWPROP WINDOW 'STOREFN STOREFN)
(WINDOWPROP WINDOW 'FETCHFN FETCHFN)
(WINDOWPROP WINDOW 'PROPCOMMANDFN PROPCOMMANDFN)
(WINDOWPROP WINDOW 'VALUECOMMANDFN VALUECOMMANDFN)
(WINDOWPROP WINDOW 'INSPECTWTITLE TITLE)
(WINDOWPROP WINDOW 'TITLECOMMANDFN TITLECOMMANDFN)
(WINDOWPROP WINDOW 'SELECTIONFN SELECTIONFN)
(WINDOWPROP WINDOW 'PROPERTIES PROPERTIES)
(WINDOWPROP WINDOW 'PROPPRINTFN PROPPRINTFN)
(WINDOWPROP WINDOW 'BUTTONEVENTFN (FUNCTION
\ITEM.WINDOW.BUTTON.HANDLER))
(WINDOWPROP WINDOW 'COPYBUTTONEVENTFN (FUNCTION
\ITEM.WINDOW.COPY.HANDLER))
(WINDOWPROP WINDOW 'REPAINTFN (FUNCTION INSPECTW.REPAINTFN))
(WINDOWPROP WINDOW 'SCROLLFN (FUNCTION SCROLLBYREPAINTFN))
;; when we create the window, record the read print environment so that the window methods can
;; use the same one, rather than inheriting form the mouse process.
(WINDOWPROP WINDOW 'PROFILE PROFILE)
(RETURN (INSPECTW.REDISPLAY WINDOW NIL VALUEMARGIN])

```

```

)
(DECLARE%: EVAL@COMPILE DONTCOPY
;; FOLLOWING DEFINITIONS EXPORTED

```

```

(DECLARE%: EVAL@COMPILE
(PUTPROPS ExtractObj MACRO (OPENLAMBDA (datum)
(if (type? annotatedValue datum)
then (fetch annotatedValue of datum)
elseif (LISTP datum)
then (CAR datum)
else datum)))
)
)

```

```

;; END EXPORTED DEFINITIONS
(PUTPROPS LOOPSINSPECT COPYRIGHT ("Venue & Xerox Corporation" 1983 1984 1985 1986 1987 1988 1990))

```

FUNCTION INDEX

ClassIVAddDelete	10	ObjAddDelete	10	ObjPropCommandFn	11	ObjTitleFn	11
InspectValProp	10	ObjInspectFetchFn	11	ObjPropertiesFn	11	ObjValueCommandFn	11
INSPECTW.CREATE	11	ObjInspectStoreFn	11	ObjTitleCommand	11		

MACRO INDEX

ExtractObj	12
------------------	----

VARIABLE INDEX

INSPECTMACROS	1
---------------------	---
