

Proposal for Loops manual contents.

This ordering is based on the Interlisp Reference manual. Numbers in *////*s indicate the corresponding parts in the (new) IRM. Other comments are enclosed in *{}*'s. I envision a manual in a single binder, with large tabs marking the IRM volume separation, and smaller tabs for each chapter.

[[Volume 1 -- Language]]

TOC *[[TOC]]*

Introduction *[[1]]*

Classes and Instances

MessageSending

The Golden Braid

Classes *[[2]]*

{Defines the msg protocol for Classes}

Creating instances

New **msg**

NewWithValues **msg**

_New **msg** sending form

Destroying instances

DestroyInstance **msg**

Destroying a class

Destroy **msg**

Destroy! **msg**

Changing a class

Add **msg**

Delete **msg**

Put **msg**

Rename **msg**

ReplaceSupers **msg**

SetName **msg**

Accessing parts of a class

ListAttribute **msg**

ListAttribute! **msg**

Enumerating parts of classes

AllInstances **msg**

AllInstances! **msg**

SubClasses **msg**

Dealing with Methods

SpecializeMethod **msg**

CopyMethod **msg**

DefMethod **msg**

MoveMethod **msg**

MethodDoc **msg**

Misc

NewClass **msg**

Subclass **msg**

Prototype **msg**

See also `_Proto`

Instances *[[3]]*

{Defines the protocol for objects}

Getting the class of an instance

Function `Class`

Class **msg**

ClassName **msg**

Classes for lisp entities -- `LispClassTable`

Accessing variables

`GetValue`, `PutValue`, `GetIVHere`

`GetClassValue`, `PutClassValue`, `GetClassIVHere`

`@`, `@*`, `_@`

problems with embedded "."s in IV names

`GetIVValue` **msg** if not an instance

`NoValueFound`

CHANGETRAN

`IVMissing` **msg**

`CVMissing` **msg**

Creating instances

`NewInstance` **method**

Naming instances

`SetName` **msg**

`UnSetName` **msg**

`Rename` **msg**

`ErrorOnNameConflict`

Changing an instance

`AddIV` **msg**

`DeleteIV` **msg**

Destroying instances

`Destroy` **message**

Copying Instances

`CopyShallow` **message**

`CopyDeep` **message**

MetaClasses *[[4]]*

{Defines the msg protocol for MetaClasses}

`DestroyClass` **msg**

- MetaClass
- AbstractClass
- AnnotatedValues *[[5]]*
 - NotSetValue
- Methods *[[6]]*
 - Method objects
 - Categories
 - Method functions
 - LAMBDA word Method
 - &OPTIONAL arguments
 - Advising inherited methods
 - _Super
 - _Super?
 - _SuperFringe
 - DoMethod
 - ApplyMethod
 - Functions `ClassNameOfMethodOwner, SelectorOfMethodBeingCompiled, ArgsOfMethodBeingCompiled`
 - Defining new methods
 - DefaultComment
 - SubclassResponsibility
- Conditionals and Iterative Statements *[[9]]*
 - Date type predicates
 - Object?
 - Class?
 - Instance?
 - AnnotatedValue?
 - Other predicates
 - HasIV, HasIV!, HasCV
 - Understands
 - InstOf, InstOf!
 - Iterative statements
 - in-supers-of
- Message sending forms *[[10]]*
 - List all message sending forms (even if described elsewhere)
 - _
 - _!
 - _IV
 - _Try
 - _Proto
 - _Super
 - _Super?
 - _SuperFringe
 - _New
 - _AV
 - Looking up methods

FetchMethod
FetchMethodOrHelp

Miscellaneous *[[12]]*

System version information

Pattern match function MatchDescr

FEATURES

DELASSOC

[[Volume 2 -- Environment]]

Browsing and Exec Level Commands *[[13]]*

UNDOable versions of Loops fns (like PutValue) if typed at top level?

LASTWORD **set by** DefineMethods

LASTCLASS **set by** DefineClass

Loops Icon

Left button menu

Middle button menu

Right button menu

Background menu command?

Browsers, in general

Interaction with GRAPHZOOM

MaxLatticeHeight

MaxLatticeWidth

Shift selecting from the background of a LatticeBrowser will COPYINSERT the graph.

Class Browsers

PPDefault

UpdateClassBrowsers?

File Browsers

Other Browsers

Top level commands

DefineClass

LASTWORD

USERWORDS

DefineMethod

Errors and Breaks *[[14]]*

Function HELPCHECK

LoopsDebugFlg

LoopsDebugFlg

ErrorOnNameConflict

IVMissing

IVValueMissing

MessageNotUnderstood

Breaking and Tracing *[[15]]*

Breaking and Tracing IV access

Breaking and Tracing methods

Gauges

Editing *[[16]]*

EDITDEF

When do changes take effect?

Copying an AV inside of DEdit really adds a pointer to the same AV!

Edit **msg**

MakeEditSource **msg**

InstallEditSource **msg**

File Package *[[17]]*

Loading Loops files

Effect of LDFLG = PROP, ALLPROP, ect

UNDOing

Selective loading of classes, etc from a file.

Filepackage coms

CLASSES

METHODS

INSTANCES

THESE-INSTANCES

Noticing changes

ObjectModified **message**

Moving and adding

MoveToFile **msg**

MoveToFile! **msg**

Saving instances

MakeFileSource **message**

DontSave **IV property**

SaveInstance **message**

SaveInstance? **message**

Compiling *[[18]]*

Macros for GetValue, PutValue

Referring to Loops objects in compiled code

Masterscope *[[19]]*

DWIM *[[20]]*

No method, tries to correct the spelling

No method, but a fn=> use the fn. How about other way?

Performance Issues *[[22]]*

Garbage Collection

Space

Speed

Caching

ClearAllCaches

Evals all forms on the global var ClearAllCaches

Method lookup caches

local cache
 global cache hit rate of 97.5% in our tests

IV Lookup caches

local cache
 global cache hit rate of 98% in our tests

Processes *[[23]]*

`_Process, _Process!`

[[Volume 3 -- Input/Output]]

Reading and Printing *[[25]]*

The # read macro character

How Classes print

`#.($ className)` and the function `$`

`Class.FileOut`

`#.($C className)` and the function `$C`

How Instances print

`Object.PrintOn`

`Object.PP`

`PPObj`

`ObjectAlwaysPPFlag`

`ObjectDontPPFlag`

Interaction with `SYSPRETTYFLG`

How AnnotatedValues print

`DefaultActiveValueClassName`

UIDs

When assigned

Functions `HasUID?`, `UIDP`, `UID`, `GetObjFromUID`, `MapObjectUID`

User Input/Output Packages *[[26]]*

Inspector

Extensions to `?=`

method `doc` in `PROMPTWINDOW`

select selector from menu

doesn't work in the `DEdit` menu

Loops windows *[[28]]*

The default menu available on Loops windows is compatible with Lisp. It even includes a "Hardcopy" item that works correctly with LatticeBrowsers.

Methods `Window.SetRegion` and `Window.SetOuterRegion`

Interaction with ATTACHEDWINDOWS

Index *[[Index]]*

[[Loops Library Packages Manual]]

Virtual Copies

Gauges

[[Release Notes]]

Installation

Variables `LOOPSDIRECTORY`, `OptionalLispuserFiles`, **and** `LoadLoopsForms`.

Maybe even `LOOPUSERSDIRECTORIES`?

The function `LOADLOOPS`

The file `LOOPSSITE`

Just load the file `LOOPS`

Initial screen setup

Customer Support

Differences from old Loops

KnowledgeBases no longer a part of Loops

Functions `ReadLeafObj`, `AllGlobalNames`, `RememberName`, **and** `GlobalName` **no longer exist**

Variables `WritingSummaryFlg`, `WritingLayerFlg`, `LeafInstanceFlg`, `FirstEnvFlg`, `OpenKBFiles`, `DefaultKBName`, `CurrentEnvironment`, `CurrentNameTable`, **and** `CurrentUIDTable` **no longer exist**

Macro `Modified` **no longer exists**

Rules no longer a part of Loops

Rules are not distributed. If they are needed, they can be loaded by first recompiling them and then loading the file `LOOPSRULES-ROOT`. This replaces a few functions and methods in the standard system and then load the rules files.

Code cleaned up

Variables `VarNameIndexes`, `PrintStatusWindow`, **and** `TTY` **no longer exist**

Functions `DC`, `DE`, `UE`, `EM`, `EI`, `EC`, **and** `FILE` **no longer exist**

Macro `@@` **no longer exists.** Change `(@@ foo)` to `(@ ::foo)`, or even `(@ foo)`

Usermacros `PU`, `UE`, **and** `EU` **no longer exist**

The argument that controls updating in the window methods

`SetOuterRegion`, `SetRegion`, `Shape`, **and** `Shape1` **have all been changed to** `noUpdateFlg`. They used to be a mixture of `updateFlg` **and** `dontUpdateFlg`.

The interpretation of the `left`, `bottom`, `width`, **and** `height` **IVs in the class** `Window` **is now the same as Lisp windows:** `left` **and** `bottom` **refer to the lower left corner of the outside of the window,** `width` **and** `height` **refer to the outside dimensions of the window (including title).**

The file `LoopsMixin` **has been deleted.** The classes `DatedObject`, `IndirectObj`, `Node`, `Perspective`, `NamedObject`, `GlobalNamedObject`, `TextItem`, `VarLength`, `StrucMeta`, `ListMetaClass`, `TempClass`, **and** `Template` **no longer exist.**

Old messages `List` and `List!` are now called `ListAttribute` and `ListAttribute!`. This change was required for future migration to `CommonLoops`. The old methods for `List` and `List!` are still available in the `LOOPSBACKWARDS` file.

The functions `DebugLoops`, `LOOPSDIR`, `i/d`, and `TESTLOOPS` have been removed from the system.

The functions `CheckDestroyedObjects` and `RemoveClassDef` have been removed from the system.

I/O

Support for reading in old style read macro (like `$(localState getFn putFn)` or `#$Mumble`) is available in the file `LOOPSBACKWARDS`.

UIDs are no longer strings. They are CONSES of session-id's and uid numbers.

Active Values

Most of the old functions for active values have been moved to `LOOPSBACKWARDS`. The functions `GetLocalState`, `PutLocalState`, `GetLocalStateOnly`, `PutLocalStateOnly`, `ReplaceActiveValue`, `MakeActiveValue`, and `DefAVP` are still around, and seem to work. The exception is `MakeActiveValue` — it now always `EMBEDS`.

Functions `GetActiveValueGetFn`, `GetActiveValuePutFn`, `GetActiveValueLocalState`, `PtcieauGtn`, `alveGetFn`, etc no longer exist. Replacements are available in the file `LOOPSBACKWARDS`.

`NotSetValue` is no longer ?—it is now an active value. This may require changes to any code that directly referred to ?.

Virtual Copies

The messages `MakeCopy99`, `MakeCopyActiveValue99`, `MakeCopyList99`, and `MakeCopyObject99` in `VirtualCopies` have been renamed `\Internal/MakeCopy`, `\Internal/MakeCopyActiveValue`, etc

Method properties no longer exist

Function `PushNewValue` no longer defined. Instead, use `CHANGETRAN`.

Can't load new Loops files into old Loops.

Conversion of old Loops code to new Loops

Old code will need to be recompiled.

File `LOOPSBACKWARDS`

Function `ConvertLoopsFiles`

Future Directions