

One of the key components of CLOS in inheritance. The CLOS Browser provides functionality for displaying this structure and for extending it. It also provides functions for displaying and changing the class definitions and method definitions which make up a system written in CLOS.

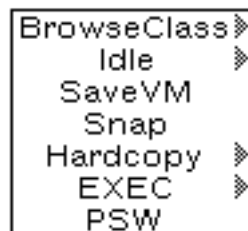
Creating a Browser

A browser can be created in two ways:

- Via a menu option from the Background Menu
- By calling the function CLOS-BROWSER:BROWSE-CLASS on a class

Creating a browser via the Background Menu

When the CLOS-BROWSER module is loaded, an entry is added to the Background Menu, as shown below:



Selecting the menu item BrowseClass brings up a window, with a prompt for the name of the class to use as the root of the browser as shown below.



Type in the name of the class you wish to browse at the flashing cursor, and the class graph will be drawn in the window.

Creating a browser programmatically

Browsers can also be created by calling the function BROWSE-CLASS:

(BROWSE-CLASS &OPTIONAL CLASS-NAME-OR-LIST &KEY :WINDOW-OR-TITLE :GOOD-CLASSES :POSITION) **[Function]**

This function brings up a browser on the class named or the list of classes named. If a window is supplied for the *:WINDOW-OR-TITLE* argument, then the browser is created in that window, else an appropriately sized window is created. The window is positioned at the *:POSITION* argument or, if not supplied, then the position is set via the mouse. If a text string is supplied for the *:WINDOW-OR-TITLE* argument, then that string is used for the window title, else the string "CLOS-browse" is used. If *:GOOD-CLASSES* is supplied, then only those classes in the list are displayed.

Using the Class browser

Instances of CLOS-BROWSER are operated on through a mouse-based interface.

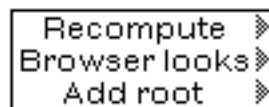
Buttoning on the browser will cause one of the following menus to be popped up:

- One menu appears when the left or middle button is pressed while the mouse is in the title bar. This menu has operations that apply to the browser itself.
- The other menu appears when the middle button is pressed when the mouse is on one of the nodes in the browser.

If the left button is pressed when the mouse is on a node, that node is boxed. This marks the node for some operations.

Options in the title bar menu

The following menu appears when you left- or middle-button in the title bar.



Recompute and it's suboptions

Selecting the Recompute option and dragging the mouse to the right causes the following submenu to appear:



Most of these items change the appearance of the browser, not the contents.

Recompute	Recomputes the browser from the starting objects. It does not recompute the labels for each node if those labels are cached in the Label-Cache slot of the browser.
Recompute Labels	Recompute the browser from the starting objects, including the labels.
Recompute inPlace	Recompute the browser without affecting the scrolled location of the lattice within the window.
Clear caches	Clear the caches of the nodes.

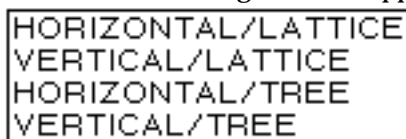
Browser looks and it's suboptions

Selecting the Browser looks menu item and sliding to the right causes the following submenu to appear:



Selecting one of these options changes the looks of the browser.

Shape to hold	Make the window for the browser just large enough to contain the browser.
Change font size	Causes a menu of alternative font sizes to pop up. Selecting one of these causes the browser to be redrawn with the nodes at that font size.
Change format	Causes the following menu to appear:



Horizontal/Lattice Lays out the grapher as an horizontal lattice.

Vertical/Lattice Lays out the grapher as a vertical lattice.

Horizontal/Tree Lays out the grapher as a horizontal tree.

Vertical/Tree lays out the grapher as a vertical tree.

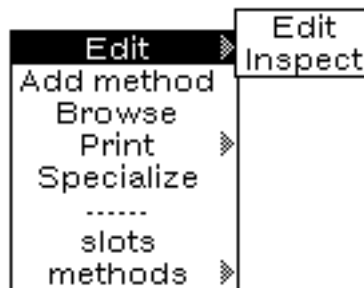
Options in the Middle-button menu

The following menu appears when you middle-button over a node in the graph:



Edit and it's suboptions

Selecting Edit causes an editor on the class definition to be brought up. Sliding the mouse to the right causes the following menu to appear:



Edit Edits the class named by the node

Inspect Inspects the class object named by the node.

Add Method

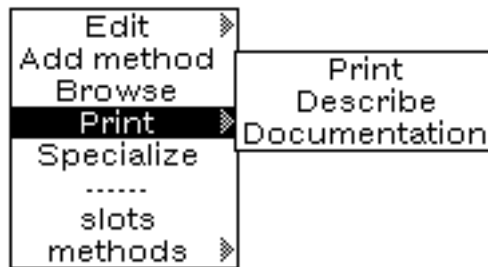
Selecting the Add Method option brings up an editor window with a template for a method to be added to that class. When the editor is done the method is installed for that class and the menu updated.

Browse

Selecting the Browse option causes a browser to be created starting with that class as the root.

Print and it's suboptions

Selecting Print prints out the class definition. Sliding the mouse to the right causes the following menu to appear:



Print Print's the class definition

Describe Describes the class, listing it's metaclass, it's supers classes, it's subclasses, it's CPL, and the number of methods specialized to it.

Documentation Print's the documentation string for the class

Specialize

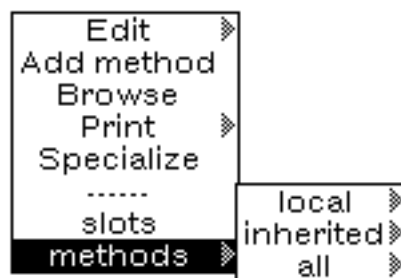
Selecting the Specialize option brings up an editor window with a template for a subclass to be added to that class. When the editor is done the class is installed and the browser updated.

Slots

Selecting the Slots option is the same as selecting the Edit option, it brings up an editor on the class definition.

Methods

The Methods option allows you to edit one of the methods defined for that class. Selecting it and sliding to the right brings up the following sub-menu:



Local Bring up a menu of the local methods, ie methods directly defined for this class

Inherited Bring up a menu of the methods this class inherits from it's superclasses.

All Bring up a menu of all the methods defined for this class, both local and inherited.

Selecting an item with the left button from the resulting menu brings up an editor on that method. If there are multiple methods that apply, a gray triangle appears in the right edge of the menu next to that item. Sliding to the right brings up a menu of method specializers to select the appropriate method.