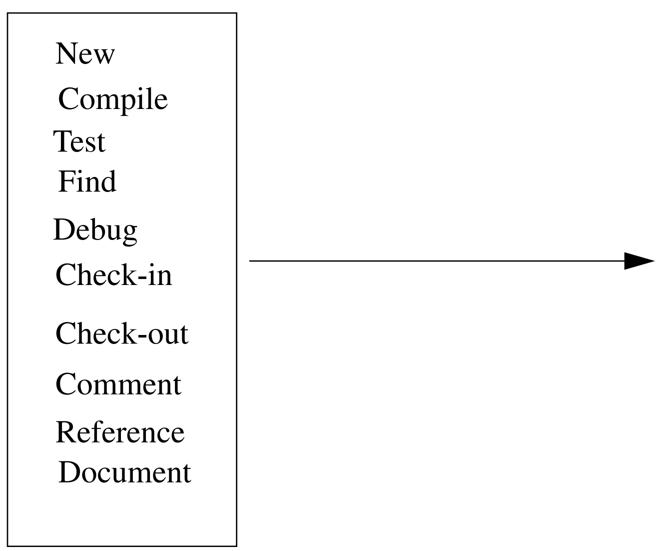
Browsers

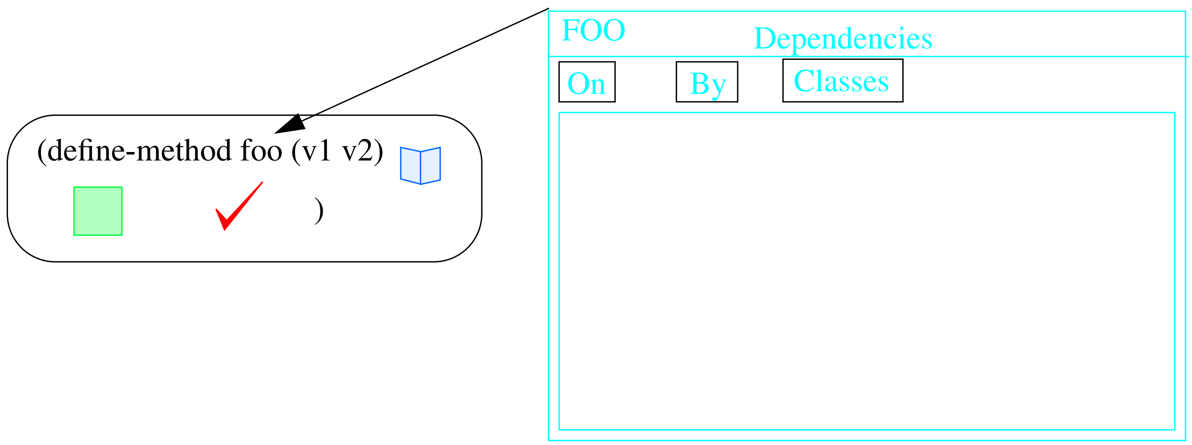
Browsers allow the condensed survey of larger data sets to reveal set groupings or connections when available, or just a complete listing. Groupings an sets imply graphical support, listings can use common widgets. The main challenge here is to get the sorting of elements right. The graphical layout of large graphs to support the visualization will be the singlemost important element.

Toolbar

Code is abbreviated in shaped icons which can be expanded into editable text. There is a tool bar that contains options and tools for aiding changes. These include opening and saving files compilation of single units and larger sets, high efficency compilation (the BIG RED button), inclusion of other code and module specification including exports and inports. When the tool is selected the toolbar will reformat to support that functionality, replaced with things like compilation flags and an animated window to show what is being done in the background. Many buttons expand into further functionality, such as running a test or entire program unit with profiling invoked changes the Tests Run window to visualize status.

Most toolbar and editting buttons have keyboard accessible equivalents for selection and movement. Mouse actions have a consistent abstraction across sub-interfaces in the environment, for example the left button is used to select the pointed to item for activation or actions.

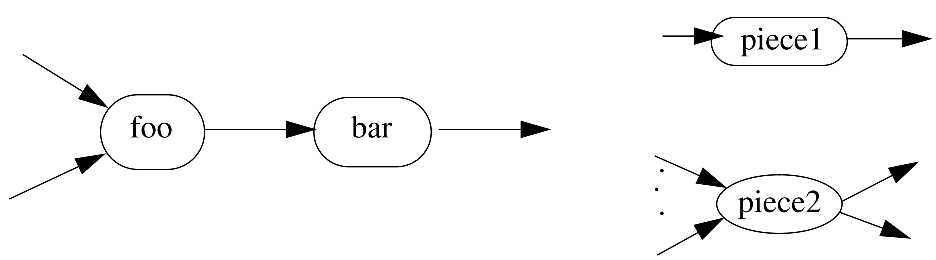




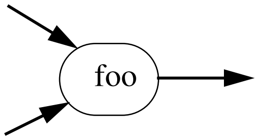
Debugging provides cues in the program elements, including the setting and activation of breakpoints and the integrated ability to inspect the status of a given variable.

Memory profiling and usage animations to illustrate consing and GC activity may be useful as part of the profiling functionality.

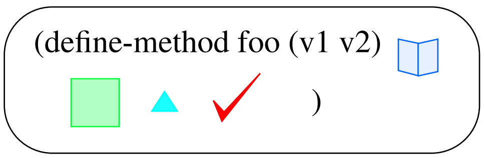
Code editing support includes a structure advisor, a pure text edit mode, and a plumbing mode for graphically linking incompletely specified component “black boxes” from the method and class level upwards.



Gwydion Environment Annotation (a cartoon).



Graphical object, a method. Click on it to expand into code, or to illustrate the associated icon properties. Two inputs and one output of unillustrated type. Compiler will indicate if joined types are incompatable by voiding plumbing between units. Each arrowhead can be clicked upon to reveal its type in a pop-up window.



Partially expanded code object containing a name label and hypercode icons for contained information. Clicking on icons expands each into either an embedded or supplemental object.