Mango "wakes up" inside the Workbench!

When mango "wakes up" inside the Eclipse workbench, the first thing it must do is determine the targeted method to specify and all if its dependents. This collection of methods is called the "method population". In the screen shots below, mango takes a look at some its own code, specifically the method Hash.f(Kons, float), which redirects an expression to a float value. It turns out the corresponding method population is enormous! But you can navigate the "Mango Explorer" peer view to the familiar "Package Explorer" to start with something easier. The number of dependents (including self) is shown in parentheses after the fully qualified method names. Mutual method recursion is indicated with the red circle icon.

Purple Methods

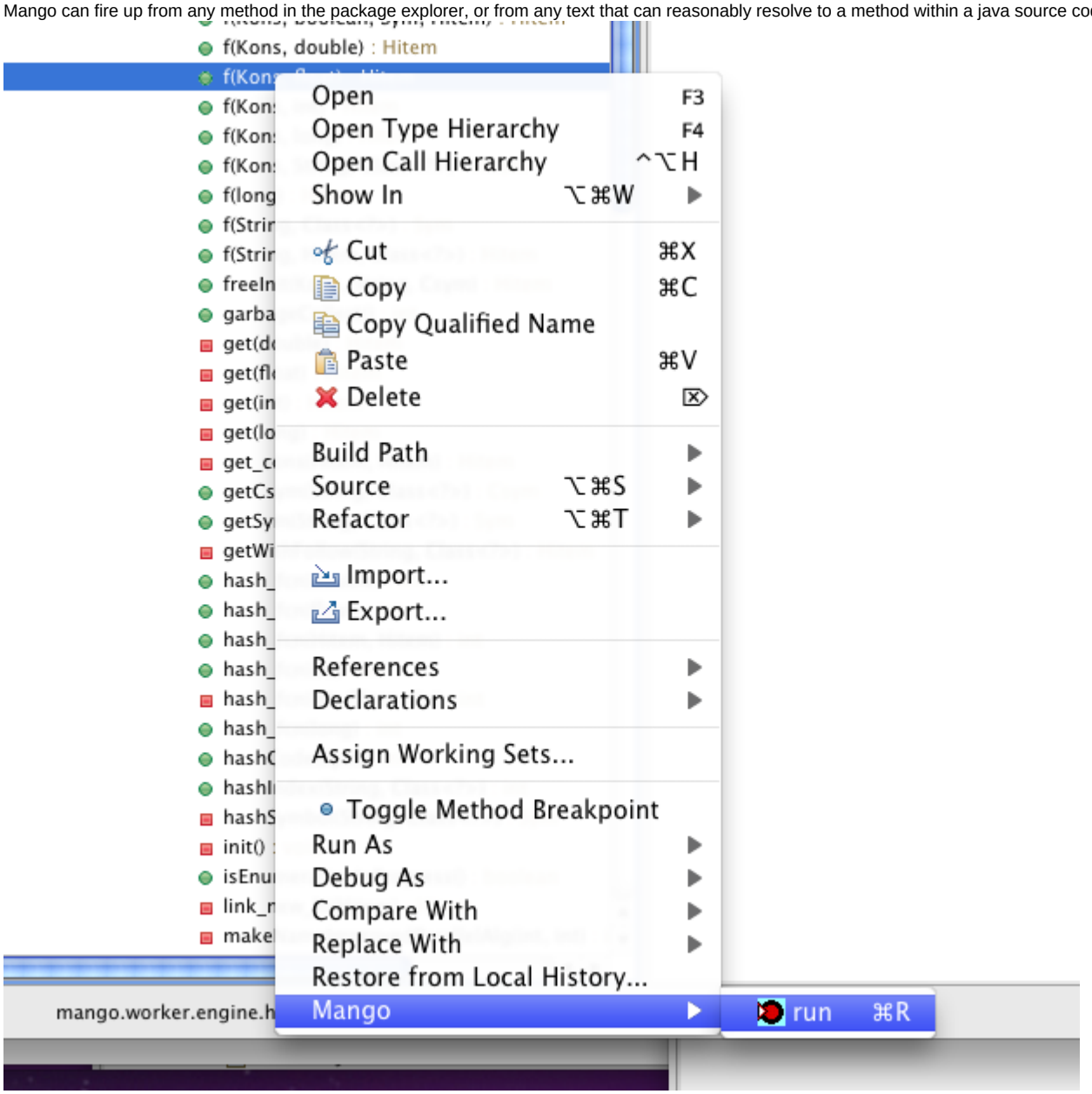
The purple methods indicate source methods which refer to with no source or mango specification. Notice any method descending from the "java" package will automatically be switched into the peer "MangoFormal" class. These methods are in the "System" project, which the user must designate. Of course the plugin supplies the initial System, which can be further populated as required.

Red Methods

The red methods in this particular example are "mango native" method specifications, meaning they wire directly into mango without source code intermediary. There is already a seed library of native methods to support the formal System project, the plugin just isn't aware of it yet.

Real Fun

Once the plugin becomes aware of the previously specified methods and the rule base which drives the rewriter engine, then the real fun will start!

Run command

Mango can fire up from any method in the package explorer, or from any text that can reasonably resolve to a method within a java source code editor.