Status: Rejected
Priority: Normal
Assignee:
Target version:

Description
It is proposed that Module#used be invoked when a module participates in refinement, much like how the Module#included and Module#extended hooks work at present. It may be the case that #used is not the greatest name for this method, and that #refined would be more fitting; it's not semantically correct, but neither is #extended.

In any event, the absence of such a hook seems to be an oversight, as one might surely wish to defer some action until such time as its effects have been explicitly requested by the refiner.

I've provided below what seems to be the simplest way to add this functionality, but it's quite likely I've missed some nuance regarding how refinements operate internally. Any insight into how the approach could be improved would be much appreciated.

diff --git a/eval.c b/eval.c
index f598525..bafafd7 100644
--- a/eval.c
+++ b/eval.c
@@ -1179,8 +1179,12 @@ using_module_recursive(const rb_cref_t *cref, VALUE klass)
 void
 rb_using_module(const rb_cref_t *cref, VALUE module)
 {  
     ID id_used;
+    CONST_ID(id_used, "used");
+    Check_Type(module, T_MODULE);
+    using_module_recursive(cref, module);
+    rb_funcall(module, id_used, 1, cref->klass);
+    rb_clear_method_cache_by_class(rb_cObject);
 }