Subject: GUI and non-GUI objects in Catalyst
Posted by Jean-Paul Davis on Wed, 02 Dec 2009 23:19:13 GMT
View Forum Message <> Reply to Message

I'm thinking about writing my first Catalyst application, and I'm looking for advice from others more experienced with Catalyst on the best design approach to handle the following. I will have a complicated object (call it "AnObject" for the sake of argument) containing data (including other objects) and methods (including read/write). I want this object to have a GUI method as an optional way to interact with the data in the object, but I do not want to create a GUI unless it's needed; i.e., creating an instance of AnObject should not by itself automatically create a GUI.

The examples in Catalyst are full "applications" in the sense that the application itself is a class derived from a top-level base widget class. I could use this application template and simply not call the GUI method in order to avoid creating/mapping a GUI, but it seems inelegant to use the TLB-widget class as the basis for a non-GUI object class. I could make the GUI a separate object that contains an AnObject, but that introduces a level of referencing between the interaction (GUI) object and the data being manipulated. Or I could reverse this so AnObject contains the GUI object if needed, although I haven't though through the data referencing implications. Would one of these approaches be the most appropriate, or are there other, better approaches?

Jean-Paul