

hi guys

in my struggle with a 'scriptable' layout for OSC interfaces i came along the need / idea to make the controls more open to manipulation in terms of their properties (not the hardwired things like callbacks).

so one could imagine a uniform interface like

```
void Ctrl::SetProperty(const String& name, const Value& v);  
void Ctrl::GetProperty(const String& name, Value& v);
```

now this would enable / map things like

```
c.SetProperty("background", Black);  
c.GetProperty("background", v);  
  
c.SetProperty("min", 123);  
c.GetProperty("max", v);  
  
c.SetProperty("xalign", "left");  
c.SetProperty("xpos", 123);  
c.SetProperty("yalign", "center");  
c.SetProperty("ypos", 45);  
  
c.SetProperty("pattern", "%.g");  
c.SetProperty("grid" ValueArray(Vector<int>() << 5 << 6));  
  
//would map to GetData / SetData directly i.e.  
c.SetProperty("data", 345);  
c.GetProperty("data", v);  
  
c.GetProperty("count", v);  
  
c.SetProperty("enable", true);  
c.GetProperty("enable", v);  
  
c.SetProperty("show", false);  
c.SetProperty("title", "MyTitle");  
c.SetProperty("tip", "GoHelp");  
  
//query all gettable properties  
c.GetProperty("rprops", v);  
//query all settable properties
```

```
c.GetProperty("wprops", v);
```

```
..
```

and many more..

this would ease things like a custom gui creation / live edit of controls..

this is just an idea, where std Ctrl could already map to a lot of things by itself.. and derived Ctrl's simply extending the 'dictionary' of properties and managing the properties..

i imagine to extend it on template base if not desired in upp, but having a uniform polymorphic interface already from Ctrl level would help a great deal in this.

i might provide an example implementation soon..

just wanted to know your opinion.

cheers
