
Subject: Re: Smart way to extract the same attribute from a list of objects?

Posted by [PMan](#) on Thu, 04 Oct 2012 16:53:54 GMT

[View Forum Message](#) <> [Reply to Message](#)

Well, I will share with you what I did. Not sure about if it qualifies as better/faster/more-elegant, but here you go.

I created a new object that is just a list:

```
pro newObj__define
  compile_opt idl2
```

```
  _ = {newObj, $
    inherits list $
  }
```

Then, in my init, I created a struct like

```
function newObj::init, nSamples = nSamples
  compile_opt idl2
```

```
  newObjStruct = {newObjStruct, $
    radiance:0, $
    channel:0}
```

```
  self->add, replicate(newObjStruct, nSamples), /extract
```

```
  return, 1
end
```

Then I overloaded the setProperty and getProperty functions.

```
pro newObj::setProperty, _extra=extra
  compile_opt idl2
```

```
  if self->isEmpty() then return
```

```
  if (~n_elements(extra)) then return
```

```
  num = n_elements((tags=tag_names(extra)))
```

```
  lArray = self->toArray()
  self->remove, /all
```

```
  for ii=0, num-1 do begin
    ok = execute("lArray."+tags[ii]+ " = extra.(ii)")
  endfor
```

```
    self.add, lArray, /extract
end

pro newObj::getProperty, _ref_extra=extra
  compile_opt idl2

  if self->isEmpty() then return

  if (~n_elements(extra)) then return

  lArray = self.toArray()

  for ii=0, n_elements(extra)-1 do begin
    ok = execute("(scope_varfetch(extra[ii], /ref_extra)) = lArray."+extra[ii])
  endfor

end
```
