
Subject: Re: Inheriting Properties (or something similar) in IDLDOC
Posted by [tom.grydeland](#) on Mon, 11 Feb 2013 09:48:59 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Monday, February 11, 2013 9:43:40 AM UTC, Tom Grydeland wrote:
> Here is a method for DOCTREECLASS that lists a class's inherited methods,

actually, `_here_` it is (below).

My thought was that the `getVariable` method could grow a couple of new variables, such as `'n_inherited_methods'` and `'inherited_methods'`:

```
'inherited_methods': begin
  methods = self->getInheritedMethods()
  return, methods
end
```

and the template 'profile.tt' etc could be modified to use these variables instead:

```
[% IF has_class %][% FOREACH class IN classes %][% SCOPE class %]
<h2>Inherited methods</h2>
  [% IF n_inherited_methods gt 0L %]
  [% FOREACH r IN inherited_methods %][% SCOPE r %]
    etc etc etc

  [% END %][% END %][% END %]
[% END %][% END %][% END %]
```

but I have not been able to figure out how to make this work. I may be missing something obvious.

```
function doctreeclass::getInheritedMethods
  compile_opt strictarr, hidden

; First, find all methods of all ancestors in inheritance order
inhm = obj_new('MGcoHashtable', key_type=7, value_type=11)
n_anc = self.ancestors->count()
for a = 0L, n_anc - 1L do begin
  anc = self.ancestors->get(position=n_anc-a-1L)
  methods = anc->getVariable('visible_methods')
  for m = 0L, n_elements(methods) - 1L do begin
    m_name = methods[m]->getVariable('name')
    parts = strsplit(m_name, '::', /extract)
    if n_elements(parts) ne 2 then continue
    inhm->put, parts[1], methods[m]
  endfor
endfor
```

```
; Next, if there is a direct method of the same name, delete the inherited
; method
n_meth = self.methods->count()
if n_meth ne 0 then begin
  keys = self.methods->keys()
  for m = 0L, n_meth - 1L do begin
    parts = strsplit(keys[m], '::', /extract)
    inhm->remove, parts[1]
  endfor
endif

methods = inhm->values()
; cleanup
return, methods
end
```
