Subject: Re: Inheriting Properties (or something similar) in IDLDOC
Posted by tom.grydeland on Wed, 30 Jan 2013 12:04:08 GMT
View Forum Message <> Reply to Message

On Wednesday, August 29, 2012 8:14:48 PM UTC+2, Mike Galloy wrote:

> But, you also want a listing of inherited routines?

This was exactly what we thought we would like for IDLDoc.

Given that the nicest possible way of asking for a new feature is to implement it, I present a pair of functions called SUPERCLASSES and METHOD_NAMES, and somehow I attach to them a hope for the requested feature to materialize in a future version of IDLdoc. METHOD_NAMES makes no attempt at distinguishing between function and procedure methods, but the extension to handle that should be straightforward.

> Michael Galloy

--Tom Grydeland

```
; docformat = 'rst'

;+
; :Copyright:
;   (c) 2013, Northern Research Institute Tromsø AS (Norut),
;                  All rights reserved.
;            This code can be redistributed in source and binary form
;            under the same terms as the rest of the IDLdoc system.
;
; :Requires:
;     IDL 8.0
;
; :Author: Tom Grydeland <tom.grydeland@norut.no>
;-

;+
;  List all superclasses of an object or class name, in method search order
;  (depth-first, left-to-right)
;
;  Will attempt to compile class definitions in order to find their
;  superclasses.
;
; :Params:
;   obj : required, in
;     an object, or the name of a class (a string)
;
; :Returns:
;   A string array (may be empty)
```

```
;
; :Example:
;   List all superclasses of 'gsar_reader'::
;     IDL> print, transpose(superclasses('gsar_reader'))
;-
function superclasses, obj
  compile_opt idl2, logical_predicate

  case size(obj, /type) of
    7 : classname=obj          ;string
    11 : classname=obj_class(obj)
    else: message, 'Not an object' + string(obj)
  endcase

  catch, error_status
  if error_status ne 0 then begin
    message, 'unable to resolve ' + classname, /info
    return, []
  endif

  resolve_routine, classname + '__define', /no_recompile
  super = obj_class(classname, count=nsuper, /super)

  if nsuper eq 0 then return, []

  supersuper = []
  for ii=0, nsuper-1 do begin
    supersuper = [supersuper, super[ii], superclasses(super[ii])]
  endfor

  return, supersuper
end



; docformat = 'rst'

;+
; :Copyright:
;   (c) 2013, Northern Research Institute Tromsø AS (Norut),
;                 All rights reserved.
;           This code can be redistributed in source and binary form
;           under the same terms as the rest of the IDLdoc system.
;
; :Requires:
;     IDL 8.0
;
; :Author: Tom Grydeland <tom.grydeland@norut.no>
```

```
;-

;+
;  List all methods of an object or class name, including inherited methods
;
; :Params:
;   obj: required, in
;     an object, or the name of a class (a string)
;   whence: optional, out
;     A hash that maps method name to superclass name
;
; :Returns:
;   a string array of method names
;-
function method_names, obj, whence
  compile_opt idl2, logical_predicate

  case size(obj, /type) of
    7 : classname=obj          ;string
    11 : classname=obj_class(obj)
    else: message, 'Not an object' + string(obj)
  endcase

  routines = [routine_names(/proc), routine_names(/func)]
  whence = hash()

  ;; find inherited methods and own methods
  foreach class, [reverse(superclasses(classname)), classname] do begin
    resolve_routine, class + '__define', /compile_full, /no_recompile
    ;; ii = where(strcmp(routines, super + '::', strlen(super)+2))
    foreach method, routines[where(strcmp(routines, class + '::', strlen(class)+2, /fold_case))] do
begin
      parts = strsplit(method, '::', /extract)
      if n_elements(parts) eq 2 then begin
        whence[parts[1]] = parts[0]     ; = class
      endif
    endforeach
  endforeach

  return, (whence.keys()).toArray()
end
```