Subject: Re: odd behaviour from array_equal() with NaN, Inf values
Posted by wallabadah on Mon, 23 Jan 2012 23:27:25 GMT
View Forum Message <> Reply to Message

Here's a quick wrapper that checks for NaNs in the input, and returns true for the above test case. I haven't tested it extensively, but it seems to solve my current problem.

```
function wjp_array_equal, op1, op2, no_typeconv = no_typeconv
 compile_opt idl2

 op1NaN = where(finite(op1, /NaN), op1NaNCount, complement = op1Finite)
 op2NaN = where(finite(op2, /NaN), op2NaNCount, complement = op2Finite)

 ; if no NaN values are present, use the normal array_equal() function
 if op1NaNCount eq 0 && op2NaNCount eq 0 then $
  return, array_equal(op1, op2, no_typeconv = no_typeconv)

 ; if counts of NaN values are equal, and results of the two where()
 ; functions are the same, compare the non-NaN values using array_equal()
 if op1NaNCount eq op2NaNCount && array_equal(op1Finite, op2Finite) then $
  return, array_equal(op1[op1Finite], op2[op2Finite], no_typeconv = no_typeconv)

 ; otherwise return 0
 return, 0
end
```