

Property proposal

@Property proposal

key

symbol	meaning
★	private
★	protected
★	public
★	final

Current

code	field	getter	setter
def x	★		
final x	★★		
public x	★		
protected x	★		
private x	★		
public final x	★★		
@Property x	★	★	★
other permutations?			

Proposed

code	field	getter	setter	note
def x	★	★	★	
final x	★★	★		the field doesn't need to be final IMHO (dk) My view is that the field should be final(tug)
public x	★			
protected x	★			
private x	★			
public final x	★★			a public final field like in Java
@Property x	★	★	★	if it's still supported (MrG) My proposal is to remove it (tug)
other permutations?				what happens with static? (MrG) Static behave exactly the same(tug)

Migration notes

Basically if you have @Property at the moment then you will need to remove it. If the declaration is typed then you are done. If it's not then you need to replace the @Property with def.

If you have field declarations which have default visibility (`def x` or `String y`, for example) then these will become properties. If you are using them from Groovy then you won't *have* to do anything. If they are accesses from Java then you will need to make them fields again. If you want them to still be fields you need to put `public` in front of the declaration.

Fields which are declared `public`, `protected` or `private` can be left alone.

If you would like to prepare for this then you can start to declare visibility for all your fields. The when we remove `@property` you can run a `sed` script which removes all the changes `@Property` to `def` and then compile your source. The compiler will than complain about the use of `def` and a type and you will need to remove all these defs.