[Does child bean automatically become prototype if parent bean defined as prototype](http://stackoverflow.com/questions/27513802/does-child-bean-automatically-become-prototype-if-parent-bean-defined-as-prototy)

In spring framework, if I have one bean defined as scope "protoype" and it is a parent to another bean. Does child bean automatically become prototype?

Example:

<bean id="a" class="..." scope="prototype"/>

<bean id="b" class="..." parent="a"/>

What will be the scope for b?

As specified in doc : <http://static.springsource.org/spring/docs/3.1.x/spring-framework-reference/html/beans.html#beans-child-bean-definitions>

A child bean definition inherits constructor argument values, property values, and method overrides from the parent, with the option to add new values. Any initialization method, destroy method, and/or static factory method settings that you specify will override the corresponding parent settings.

The remaining settings are always taken from the child definition: depends on, autowire mode, dependency check, singleton, scope, lazy init.

I see contradicting answers given by multiple people. That made me curious to test it out. Here is the result.

if Parent is of scope="prototype" and Child do not have a scope defined, Child will inherit prototype scope from Parent.

If Parent is of scope="singleton" and child do not have a scope defined, obviously scope of child will be Singleton

You can always define scope in child, and it will take precedence, for both singleton and prototype.

Spring documentation is misleading but if you do a quick test it will be clear.