

PackageScope transformation

Groovy's convention for properties is that any *field* without any visibility modifier is exposed as a property, with a getter and a setter transparently generated for you. For instance, this Person class exposes a getter getName() and a setter setName() for a private name field:

```
class Person {  
    String name  
}
```

Which is equivalent to this Java class:

```
public class Person {  
    private String name;  
    public String getName() { return name; }  
    public void setName(name) { this.name = name; }  
}
```

That said, this approach has one drawback in that you don't have the possibility to define a field with package-scope visibility. To be able to expose a field with package-scope visibility, you can now annotate your field with the @PackageScope annotation.