

BD操作之-getProperty.getValues().addPropertyValue()研究

(a) 案例演示

```
233     }
234     if (parameters != null) {
235         beanDefinition.getPropertyValues().addPropertyValue( propertyName: "parameters", parameters);
236     }
237     return beanDefinition;
238 }
```

DubboBeanDefinitionParser

没有属性，无中生有是不行的哇

Caused by: org.springframework.beans.NotWritablePropertyException: Invalid property 'munualStr' of bean class [com.source.prop.PropStringAndRef]: Bean property 'munualStr' is not writable or has an invalid setter method. Does the parameter type of the setter match the return type of the getter?



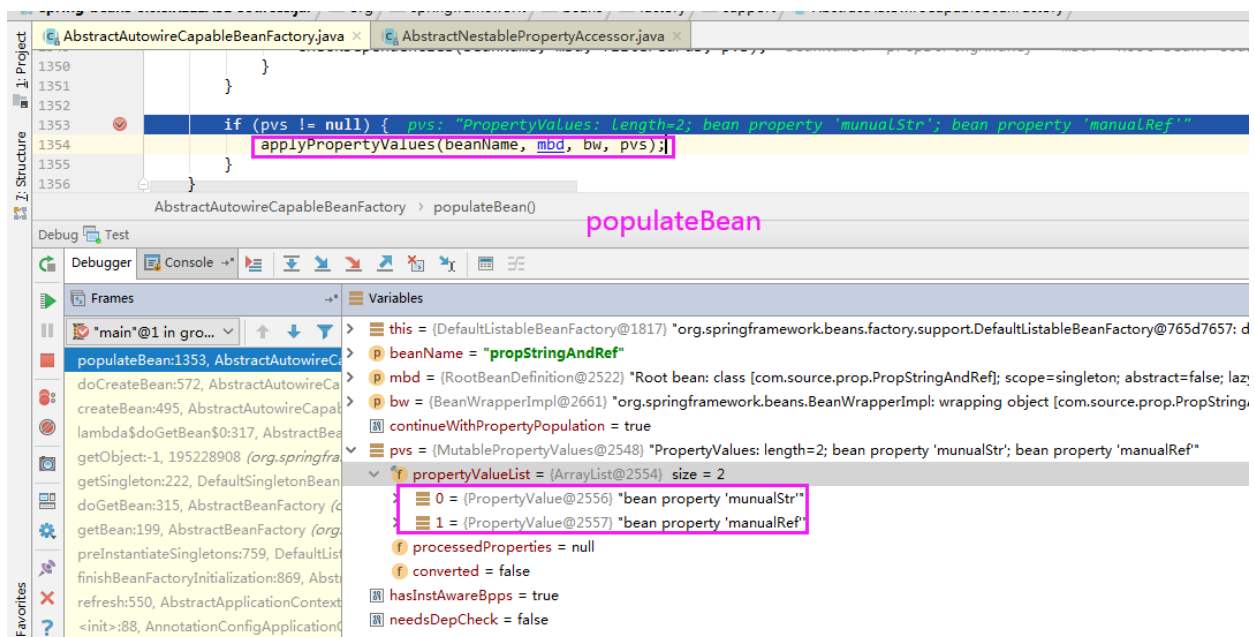
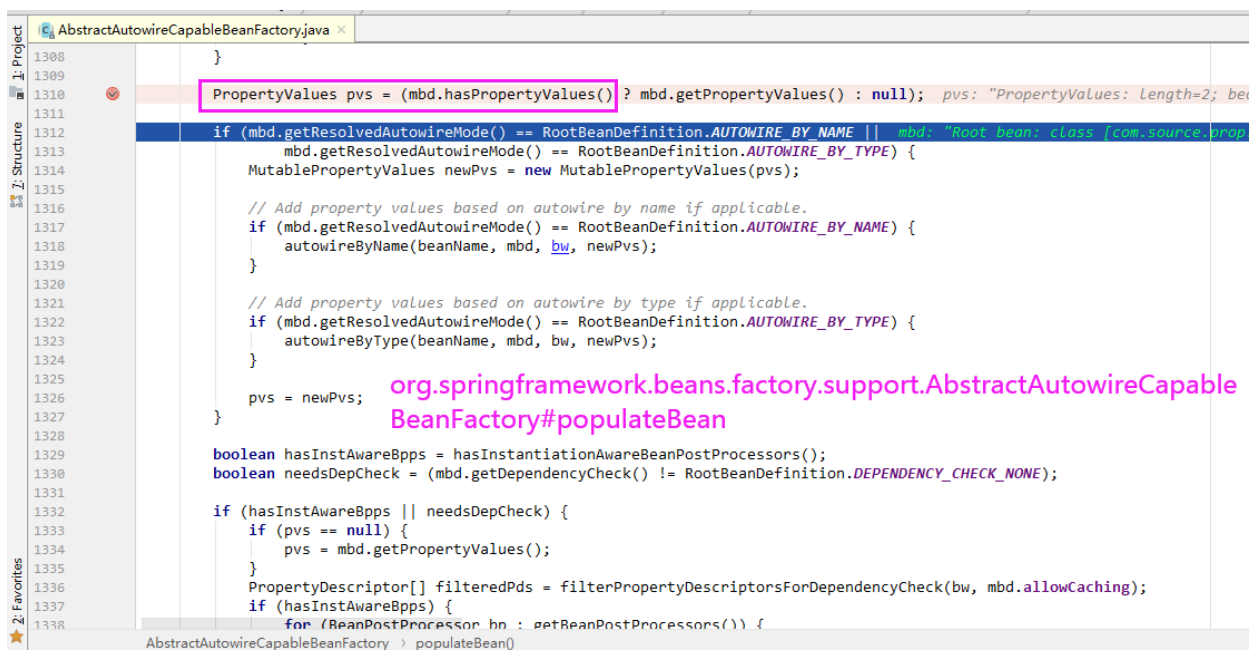
```
96     parserContext.getRegistry().registerBeanDefinition(id, beanDefinition);
97     beanDefinition.getPropertyValues().addPropertyValue( propertyName: "id", id);
98
99     if (ProtocolConfig.class.equals(beanClass)) {
100         for (String name : parserContext.getRegistry().getBeanDefinitionNames()) {
101             BeanDefinition definition = parserContext.getRegistry().getBeanDefinition(name);
102             PropertyValue property = definition.getPropertyValues().getPropertyValue( propertyName: "protocol");
103             if (property != null) {
104                 Object value = property.getValue();
105                 if (value instanceof ProtocolConfig && id.equals(((ProtocolConfig) value).getName())) {
106                     definition.getPropertyValues().addPropertyValue( propertyName: "protocol", new RuntimeBeanReference(id));
107                 }
108             }
109         }
110     } else if (ServiceBean.class.equals(beanClass)) {
111         // 注意spring注入属性的手法
```

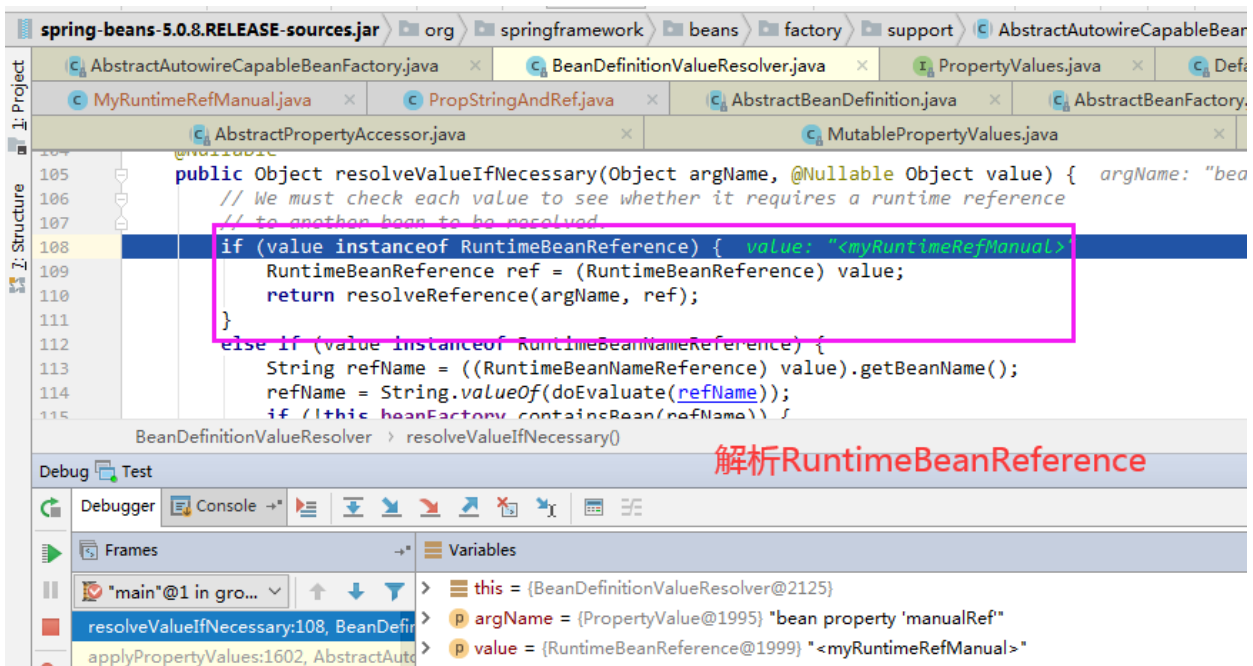
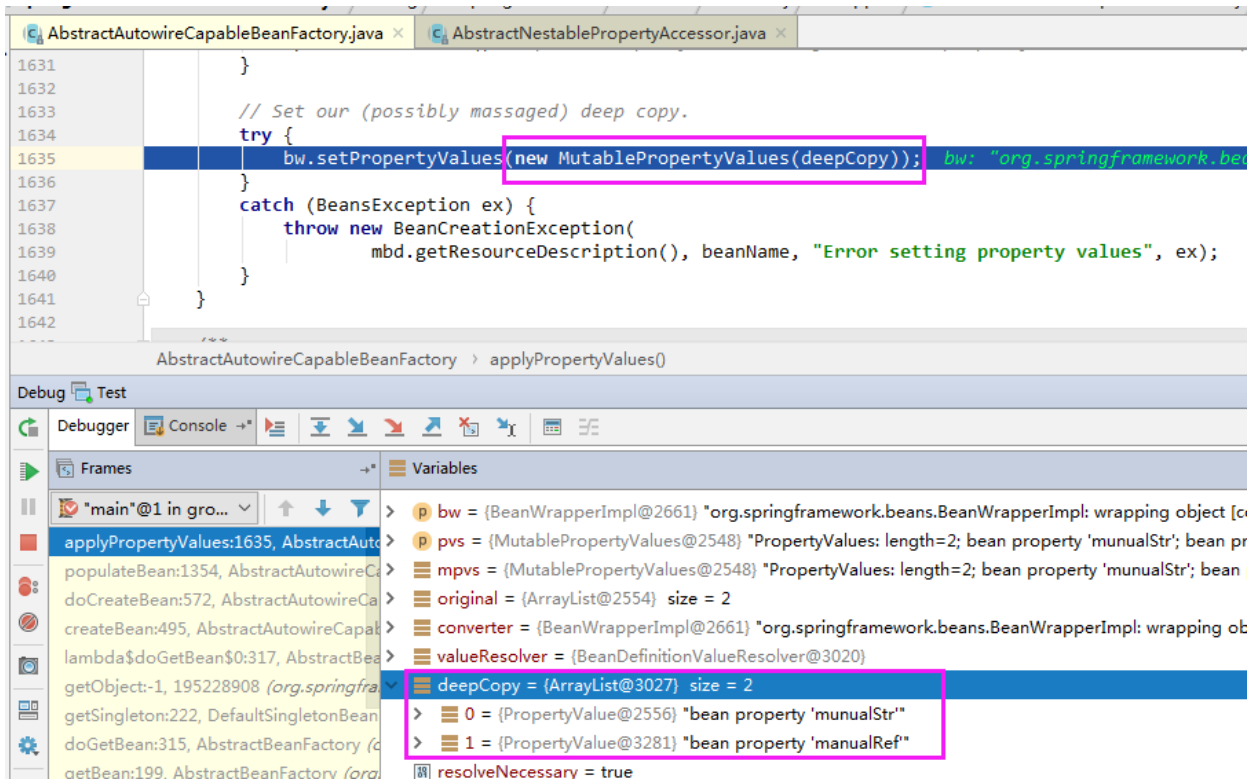
Caused by: org.springframework.beans.NotWritablePropertyException: Invalid property 'manualRef' of bean class [com.source.prop.PropStringAndRef]: Bean property 'manualRef' is not writable or has an invalid setter method. Does the parameter type of the setter match the return type of the getter?

Caused by: org.springframework.beans.ConversionNotSupportedException: Failed to convert property value of type 'com.source.prop.MyRuntimeRefManual' to required type 'java.lang.String' for property 'manualRef'; nested exception is java.lang.IllegalStateException: Cannot convert value of type 'com.source.prop.MyRuntimeRefManual' to required type 'java.lang.String' for property 'manualRef': no matching editors or conversion strategy found



(b) 源码分析，这些属性什么时候赋上值的？





```

spring-beans-5.0.8.RELEASE-sources.jar > org > springframework > beans > factory > support > BeanDefinitionValueResolv
AbstractAutowireCapableBeanFactory.java x BeanDefinitionValueResolver.java x PropertyValues.java x
MyRuntimeRefManual.java x PropStringAndRef.java x AbstractBeanDefinition.java x AbstractBeanFact
AbstractPropertyAccessor.java x MutablePropertyValues.java x Abstract
51 @Nullable
52 @ private Object resolveReference(Object argName, RuntimeBeanReference ref) {
53     try {
54         Object bean;
55         String refName = ref.getBeanName();
56         refName = String.valueOf(doEvaluate(refName));
57         if (ref.isToParent()) {
58             if (this.beanFactory.getParentBeanFactory() == null) {
59                 throw new BeanCreationException(
60                     this.beanDefinition.getResourceDescription(), this.beanName,
61                     "Can't resolve reference to bean '" + refName +
62                     "' in parent factory: no parent factory available");
63             }
64             bean = this.beanFactory.getParentBeanFactory().getBean(refName);
65         }
66         else {
67             bean = this.beanFactory.getBean(refName);
68             this.beanFactory.registerDependentBean(refName, this.beanName);
69         }
70         if (bean instanceof NullBean) {
71             bean = null;
72         }
73         return bean;
74     }
}

```

BeanDefinitionValueResolver > resolveReference()