

RFT与 ring结合进行自动化脚本开发之一 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/467/2021\\_2022\\_RFT\\_E4\\_B8\\_8E\\_rin\\_c67\\_467275.htm](https://www.100test.com/kao_ti2020/467/2021_2022_RFT_E4_B8_8E_rin_c67_467275.htm) 目前，大部分自动化脚本开发者利用RFT的测试对象地图管理测试脚本的测试对象，RFT的测试对象地图有两种类型：公有测试对象地图和私有测试对象地图。(1)目前，大部分脚本采用的都是私有测试对象地图，如果被测试的程序顶层对象发生变化，这样每个脚本关联的测试对象地图都要进行修改，之所以这样，有一个原因：在RFT中测试对象地图不能够继承(仅仅可以合并)，如果两个测试对象地图之间可以进行继承，这样把公用的测试对象放在父测试对象地图中，其他的测试对象地图继承这个公用的测试对象地图(同时继承公有的测试对象)，如果被测试程序顶层框架发生变化，仅仅只修改父测试对象地图就可以了，但是目前RFT中不能实现(据我了解)。(2)RFT中还有一种类型的测试对象地图是公有的测试对象地图，公有的测试对象地图可以实现公用的测试对象重复使用，如果测试对象是一模一样的，RFT在公有的测试对象地图中只保留此测试对象的一个实例，但是把多个测试对象都插入到一个测试对象地图中，同样又面临着难以管理的困难。(3)另外，RFT中的测试对象地图与脚本紧紧的耦合，如果程序即使发生很小的变动，也要更新测试对象地图，修改脚本等等。(4)Spring具有强大的功能：Ioc,可以实现在xml中把一个测试对象注入到另一个测试对象中，再加上可以利用RFT的find方法可以把spring的配置文件配置成一张RFT测试对象地图。另外，一个spring配置文件可以被另一个spring文件import进来。这样可

以把公有的测试对象放在一个spring配置文件中，然后有其他的spring配置文件进行导入，可以实现测试对象地图(spring的配置文件)的继承。另外，在spring的配置文件中，可以注入一个对象的属性，这样可以人工的更改这个测试对象的父框架，另外可以注入测试对象的识别属性，这样可以进行人为的进行测试对象识别属性的更改(这样可以更好的应对被测测试程序的变化)。例如：xml代码

```
<!--从classpath(类路径)
中导入base-test-object-map.xml文件 可以使
用base-test-object-map.xml文件中定义的bean--> <!--导入时
候要包含完整的包名--> < import
resource="classpath:pkg/base-test-object-map.xml"/ > <!--设置待
查找对象的识别属性 格式：propertyName-propertyValue-->
< property name="objectProperties" > < list > <!--属性名称和
属性精确匹配用==--> <!--属性名称和属性用正则表达式匹配
用:--> < value > .class=Html.FORMvalue > 然后是怎么使
用spring配置文件中的测试对象 < value > .name:.*Formvalue >
list > property > <!--设置从什么对象开始查找 注入父测试对
象--> <!--browserTestObject 引用的是base-test-object-map.xml
文件中已经定义的bean--> < property name="parentTestObject"
> < ref bean="browserTestObject"/ > property > 在脚本中引用
测试对象通过如下语句： java 代码TextGuiTestObject
text_userName = new
TextGuiTestObject((TestObject)context.getBean("userNameText")
). text_userName.setText("system").
```

(5)采用spring管理测试对象，也不是说一劳永逸的，如果被测试程序发生变化，避免不了的也要修改spring配置文件。但是，至少比RFT管理测试对

象地图的修改量要小的多。(6)采用spring管理测试对象也有以下缺点：(1)需要测试脚本开发者有比较高的编程技能(熟悉springloc基本配置和对RFT API比较熟悉)(2)被测试程序结构规范，最好每个HTML元素都有其名字，还有每次修改程序，如果HTML元素名称能不修改最后就不修改(RFT可以通过组件名称查找对象)还有就是按钮上面的文本，采用RFT管理测试对象地图也有此要求。(3)要求对被测程序的组件结构有充分了解(可以通过RFT测试地图了解)(4)开发采用spring管理测试对象地图的脚本比开发有RFT管理的测试对象地图时间要长的多，但是带来的好处也是明显的。(5)测试对象执行动作的速度也没有RFT管理的测试对象地图快。

```
base-test-object-map.xml 代码 xml 代码 <!--sp-- > xml
version="1.0" encoding="gb2312"? > <!--CTYPE beans PUBLIC
"-//SPRING//DTD BEAN//EN" </sp-- >
"http://www.springframework.org/dtd/spring-beans.dtd" > <beans
> <!--注册com.rational.test.ft.script.Property属性编辑器-- >
<bean id="customEditorConfigurer"
class="org.springframework.beans.factory.config.CustomEditorCon
figurer" > <property name="customEditors" > <map > <entry
key="com.rational.test.ft.script.Property" > <bean
class="pkg.TestObjectPropertyEditor"/ > entry > map > property
> bean > <!--实例化pkg.BaseTestObjectScript类-- > <bean
id="baseTestObjectScript" class="pkg.BaseTestObjectScript"/ >
<!--调用baseTestObjectScript中的非静态方
法getBrowserTestObject产生一个BrowserTestObject测试对象--
> <bean id="browserTestObject"
```

```
factory-bean="baseTestObjectScript"
factory-method="getBrowserTestObject" singleton="false"/ > <!--
调用baseTestObjectScript中的非静态方
法getDocumentTestObject产生一个DocumentTestObject测试对
象-- > < bean id="documentTestObject"
factory-bean="baseTestObjectScript"
factory-method="getDocumentTestObject" singleton="false"/ >
beans > base-test-object-map.xml 可以作为公有的测试对象地图
，其中customEditorConfigurer向容器注册了一个属性编辑
，pkg.BaseTestObjectScript是一个RFT脚本，此脚本
有getBrowserTestObject()和documentTestObject()，可以在这个
类中放入基本的测试对象，通过Spring将这些对象封装为bean,
然后有其他的spring配置文件来import，这样就实现了对象的
继承。pkg.BaseTestObjectScript的代码如下：100Test 下载频道
开通，各类考试题目直接下载。详细请访问 www.100test.com
```