

```

/*
Pro Spring
By Rob Harrop
Jan Machacek
ISBN: 1-59059-461-4
Publisher: Apress
*/

////////////////////////////////////
////
//File: lazy.xml
<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN//EN" "http://www.springframework.org/dt
d/spring-beans.dtd">
<beans>
  <bean id="destructiveBean" class="DestructiveBeanWithInterface">
    <property name="filePath">
      <value>c:/test.txt</value>
    </property>
  </bean>
  <bean id="shutdownHook" class="ShutdownHookBean" lazy-init="false"/>
</beans>

////////////////////////////////////
////
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStream;

import org.springframework.beans.factory.DisposableBean;
import org.springframework.beans.factory.InitializingBean;
import org.springframework.beans.factory.config.ConfigurableListableBeanFactory;
import org.springframework.beans.factory.xml.XmlBeanFactory;
import org.springframework.core.io.FileSystemResource;

public class DestructiveBeanWithInterface implements InitializingBean,
    DisposableBean {

    private InputStream is = null;

    public String filePath = null;

    public void afterPropertiesSet() throws Exception {

        System.out.println("Initializing Bean");

        if (filePath == null) {
            throw new IllegalArgumentException(
                "You must specify the filePath property of " + DestructiveBean.
class);
        }

        is = new FileInputStream(filePath);
    }

    public void destroy() {

```

```

System.out.println("Destroying Bean");

    if (is != null) {
        try {
            is.close();
            is = null;
        } catch (IOException ex) {
            System.err.println("WARN: An IOException occurred"
                + " trying to close the InputStream");
        }
    }
}

public void setFilePath(String filePath) {
    this.filePath = filePath;
}
}

////////////////////////////////////
////

import org.springframework.beans.BeansException;
import org.springframework.beans.factory.BeanFactory;
import org.springframework.beans.factory.BeanFactoryAware;
import org.springframework.beans.factory.config.ConfigurableListableBeanFactory;

public class ShutdownHookBean implements BeanFactoryAware,
    Runnable {

    private ConfigurableListableBeanFactory factory;

    public void setBeanFactory(BeanFactory factory) throws BeansException {

        if (factory instanceof ConfigurableListableBeanFactory) {
            this.factory = (ConfigurableListableBeanFactory) factory;
            Runtime.getRuntime().addShutdownHook(new Thread(this));
        }
    }

    public void run() {
        if (factory != null) {
            System.out.println("Destroying Singletons");
            factory.destroySingletons();
            System.out.println("Singletons Destroyed");
        }
    }
}

////////////////////////////////////
////

import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStream;

import org.springframework.beans.factory.InitializingBean;

```

```

import org.springframework.beans.factory.config.ConfigurableListableBeanFactory;
import org.springframework.beans.factory.xml.XmlBeanFactory;
import org.springframework.core.io.FileSystemResource;

public class DestructiveBean implements InitializingBean {

    private InputStream is = null;

    public String filePath = null;

    public void afterPropertiesSet() throws Exception {

        System.out.println("Initializing Bean");

        if (filePath == null) {
            throw new IllegalArgumentException(
                "You must specify the filePath property of " + DestructiveBean.
class);
        }

        is = new FileInputStream(filePath);
    }

    public void destroy() {

        System.out.println("Destroying Bean");

        if (is != null) {
            try {
                is.close();
                is = null;
            } catch (IOException ex) {
                System.err.println("WARN: An IOException occurred"
                    + " trying to close the InputStream");
            }
        }
    }

    public void setFilePath(String filePath) {
        this.filePath = filePath;
    }
}

////////////////////////////////////
////
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.FileSystemXmlApplicationContext;

public class LazyInitDemo {

    public static void main(String[] args) {
        ApplicationContext ctx = new FileSystemXmlApplicationContext(
            "build/lazy.xml");

        DestructiveBeanWithInterface bean = (DestructiveBeanWithInterface) ctx
            .getBean("destructiveBean");
    }
}

```