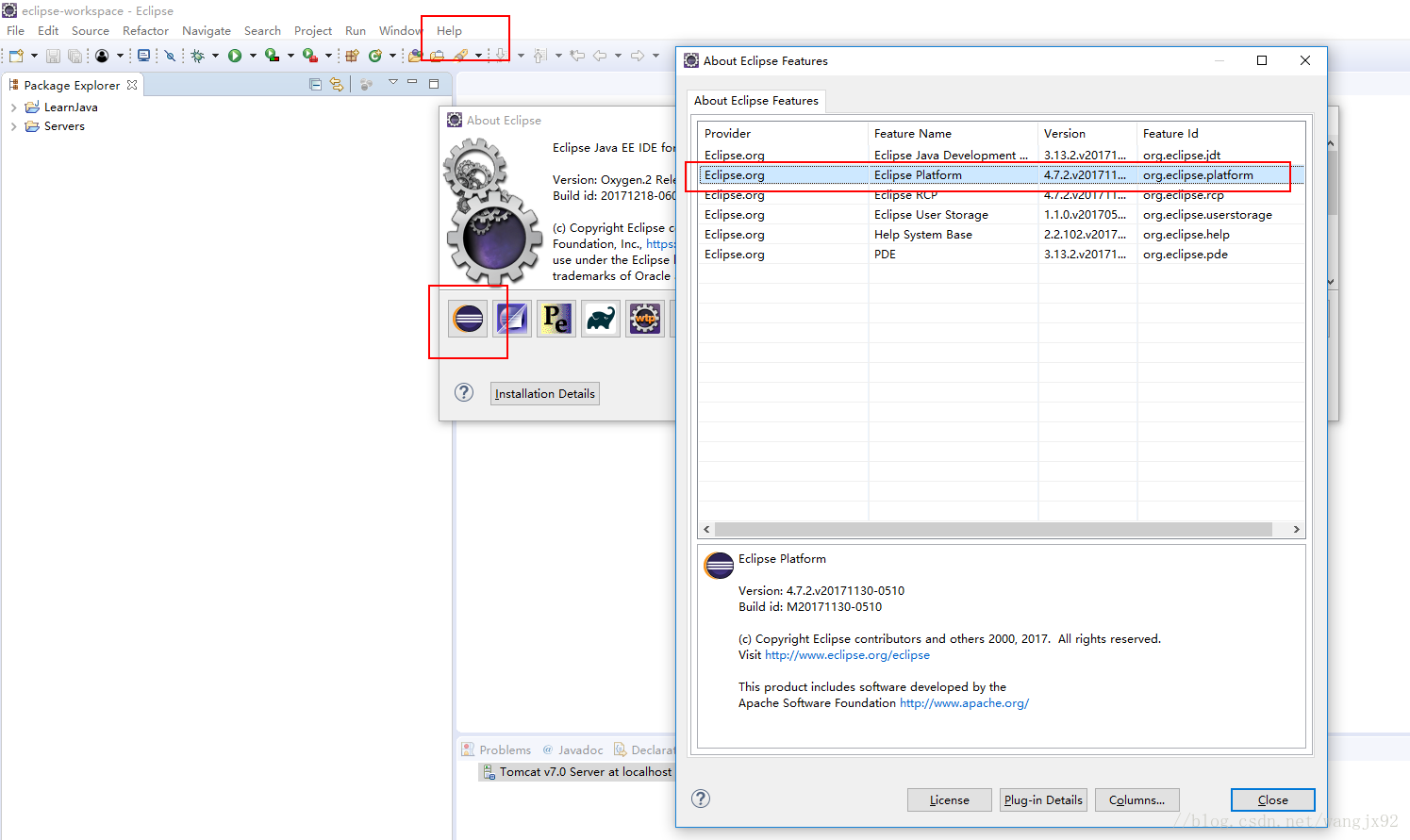
用于在编写XML的时候提示输出

1. 手动安装插件：

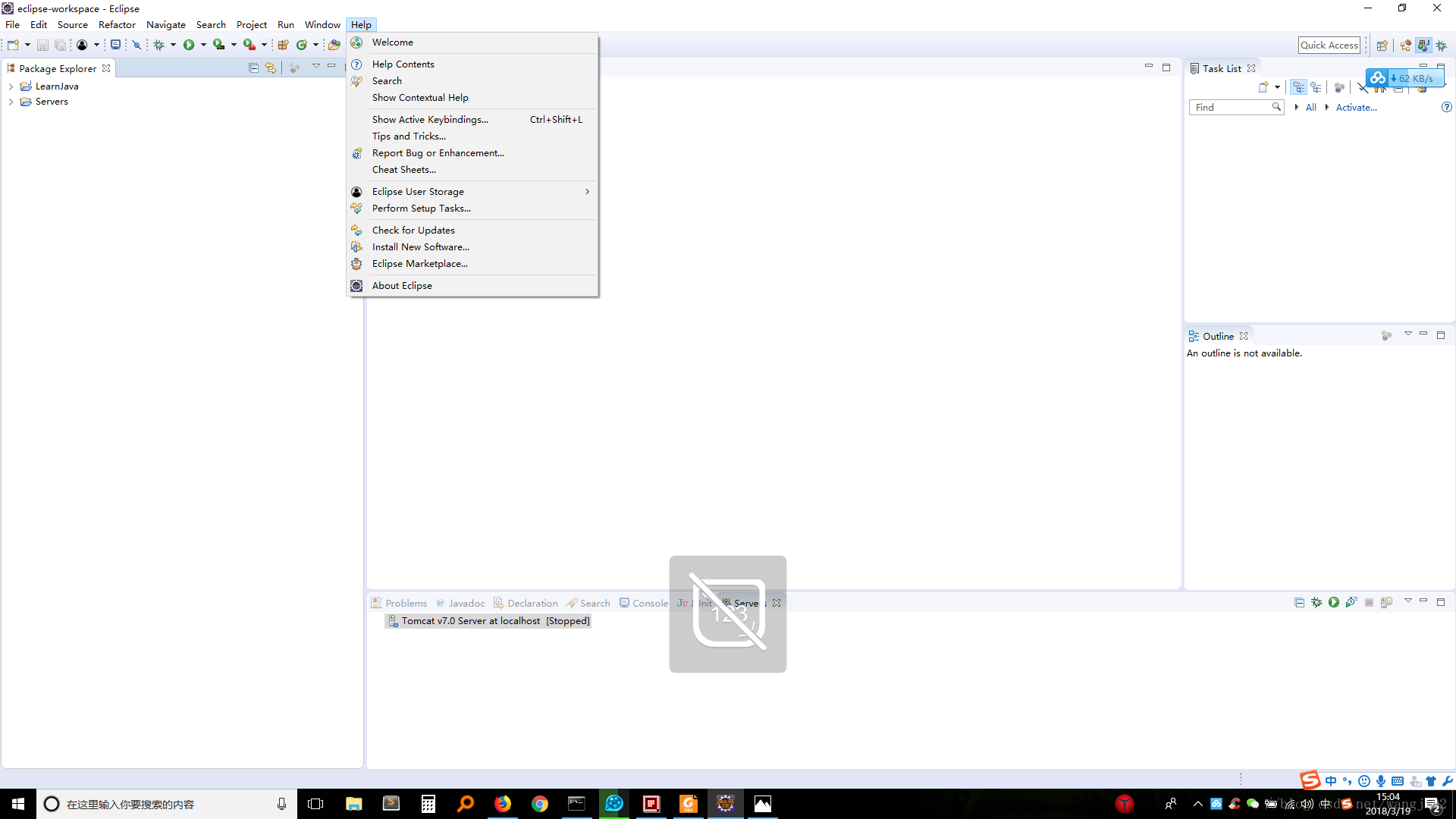
[springsource-tool-suite](https://spring.io/tools/sts/all)

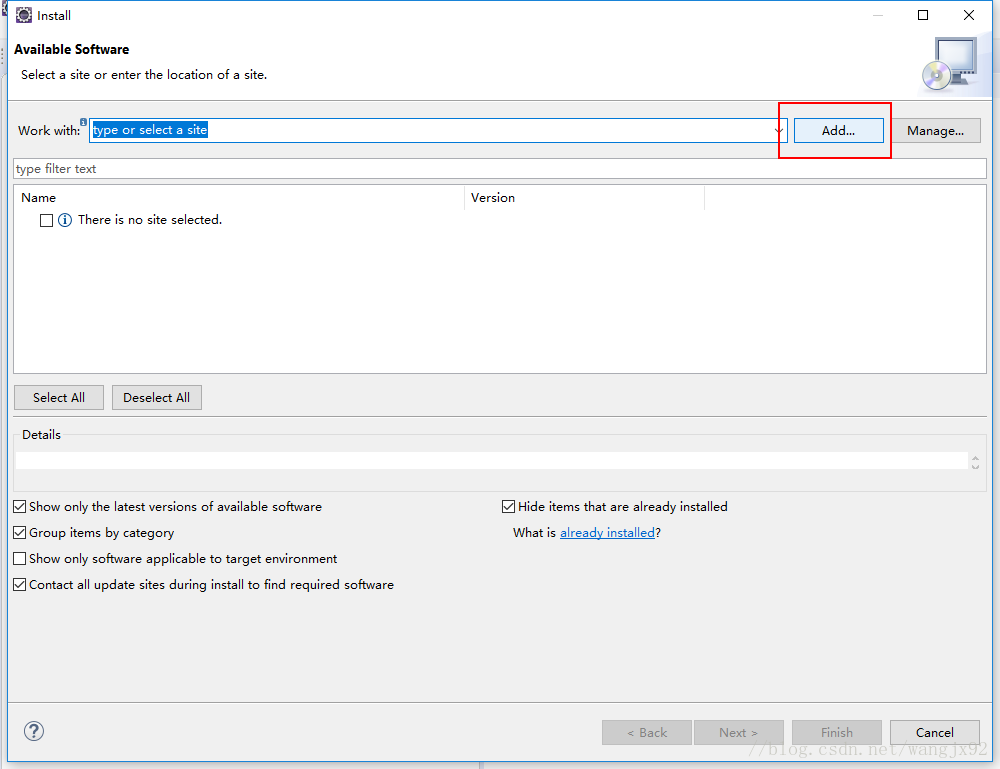
根据Eclipse的版本选择合适的安装包

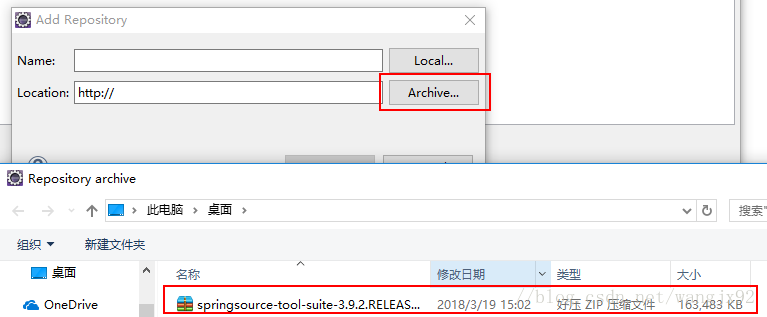


下载完成后

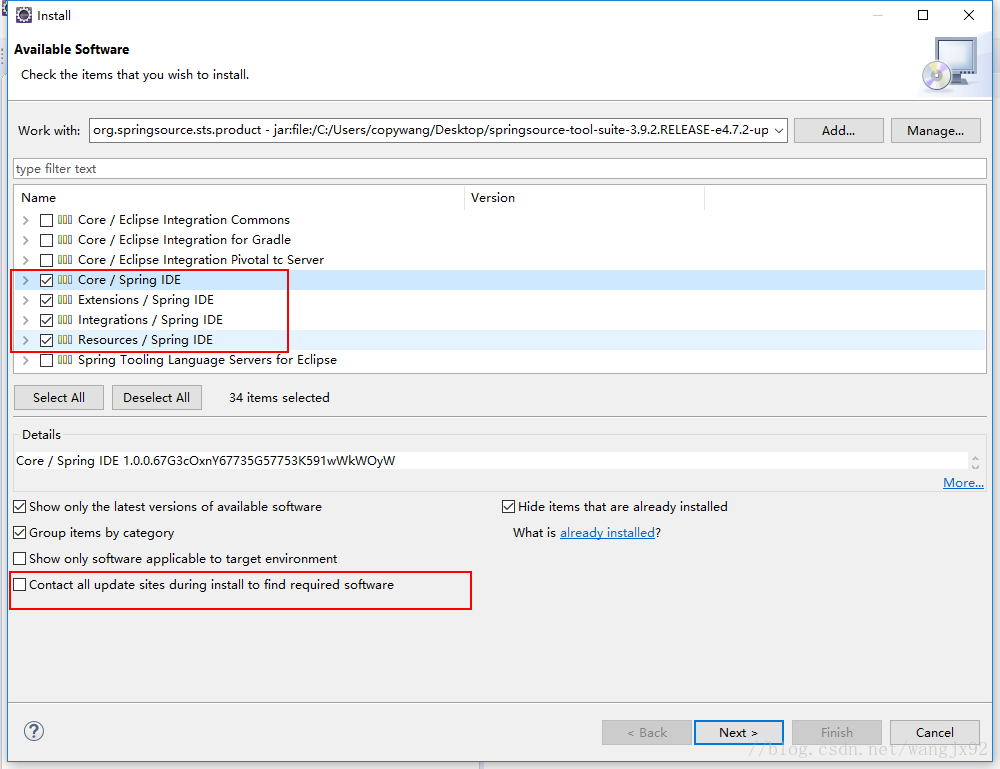
安装







勾选4个IDE，取消自动更新

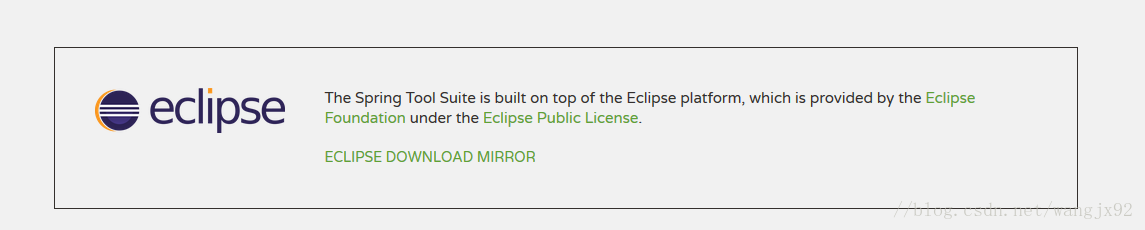


安装后重启eclipse，然后xml文件上右键Open With可以看到有Spring Config editor的提示

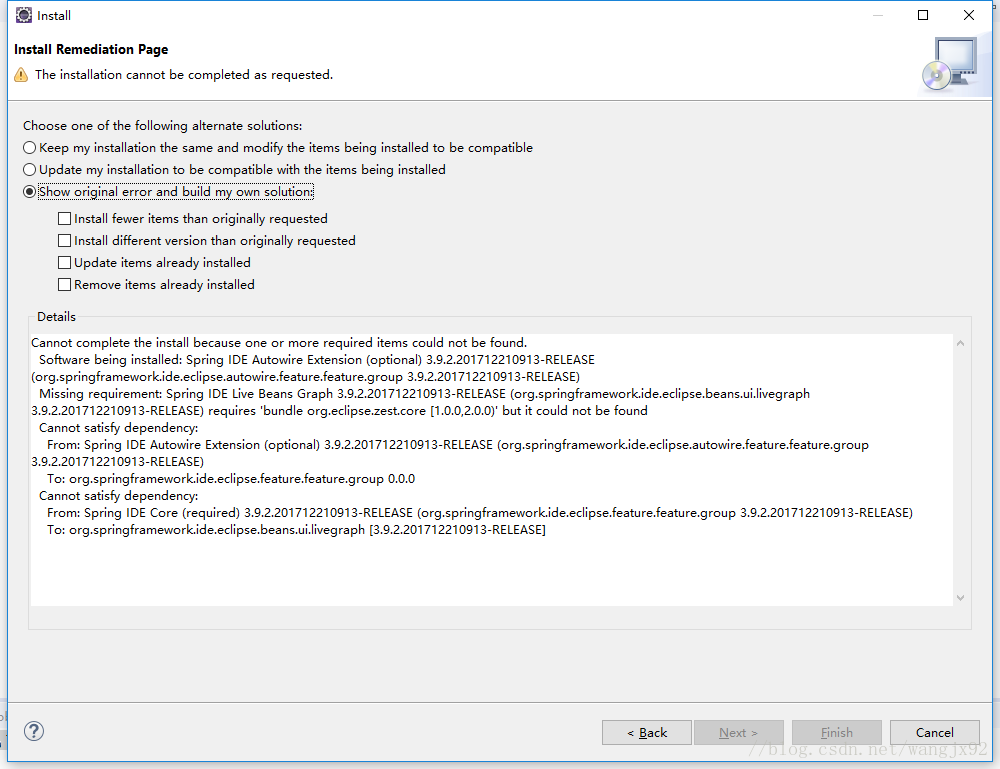
2. 使用Spring安装好的插件包：（新手推荐方式！！）

[Spring Tool Suite™](https://spring.io/tools)

根据不同的eclipse版本下载即可



STS安装报错一般是因为GEF，报错信息如下

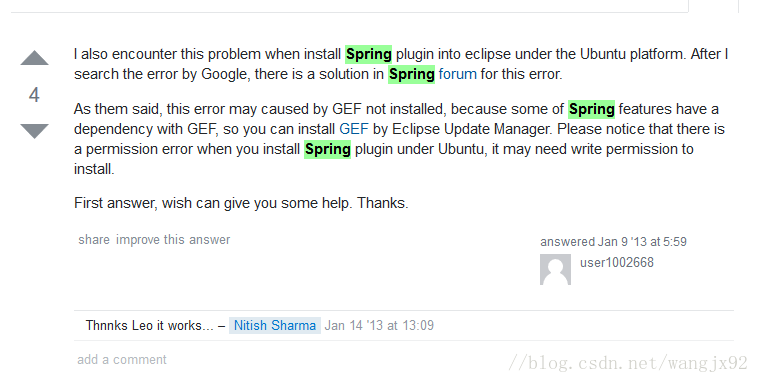


解决方案：

安装GEF

网址：[GEF](https://eclipse.org/gef/)

参考链接：[Stack Overflow](http://cache.baiducontent.com/c?m=9f65cb4a8c8507ed19fa950d100b96204a05d93e788090412189d65f93130a1c187ba0fc7063565f8e993d7a02a94b58e0f337752a4262f7c094dd5dddcdc97e789f27422b40c707418d47fc8d1d789d37902cbeef59bafae477ceee84dc800f128b08592f97f0fa1a57578d34b0436eb4fadf5f152e51f8ad6b39aa4e7758c47517a14788be692a5096f6ac0c1b9076917611e6fa75e37c48&p=91759a46d6c915eb0be29629614983&newp=8d769a4787934eac58e7c8621749bb231610db2151d6da01298ffe0cc4241a1a1a3aecbf20231001d6c1786d00af495deef63272350834f1f689df08d2ecce7e7e&user=baidu&fm=sc&query=missing+requirement+spring+IDE+Live+Beans+Graph&qid=88a914870001d087&p1=7)



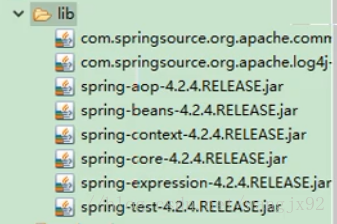
从GEF官网下载一个GEF包，安装所有

然后再安装STS插件即可

Spring的Junit支持

不需要每次自己手动创建容器对象

最小包导入要求：



其中test包是支持Junit的

代码写法：

比如Demo.java

**[java]** [view plain](http://blog.csdn.net/wangjx92/article/details/79611654) [copy](http://blog.csdn.net/wangjx92/article/details/79611654)

1. import javax.annotation.Resource;
3. import org.junit.Test;
4. import org.junit.runner.RunWith;
5. import org.springframework.context.ApplicationContext;
6. import org.springframework.context.support.ClassPathXmlApplicationContext;
7. import org.springframework.test.context.ContextConfiguration;
8. import org.springframework.test.context.junit4.SpringJUnit4ClassRunner;
10. //帮我们创建容器
11. @RunWith(SpringJUnit4ClassRunner.class)
12. //指定创建容器时使用哪个配置文件
13. @ContextConfiguration("classpath:applicationContext.xml")
14. public class Demo {
15. //将名为user的对象注入到u变量中
16. @Resource(name="user")
17. private User u;
19. @Test
20. public void fun1(){
22. System.out.println(u);
24. }
26. @Test
27. public void fun2(){
29. System.out.println(u);
31. }
33. @Test
34. public void fun3(){
36. System.out.println(u);
38. }
39. }