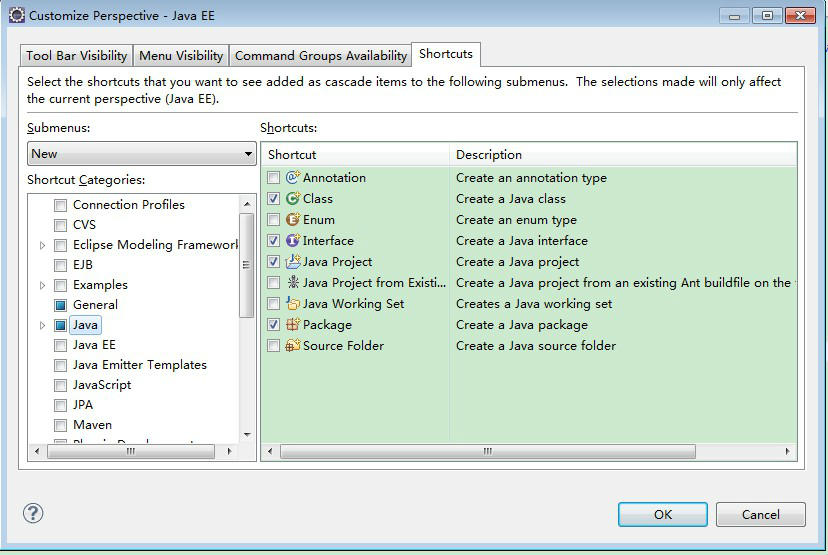
# Eclipse

## Eclipse右键New菜单项的自定义设置

大家都知道一般在Eclipse中新建一个项目或者文件都是通过菜单项的File-New来创建，然而有些右键出来的选项可能从来都不会使用到，而有些可能会常用到但是右键中又没有，这个时候就可以自定义New中的右键选项，具体设置方式如下：

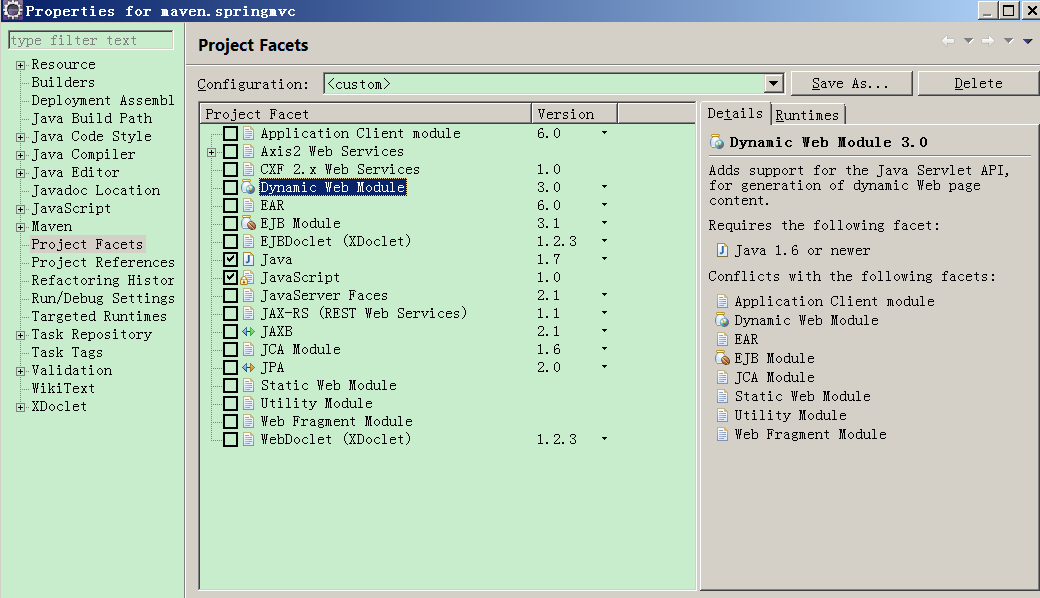
**Window --> Customize Perspective... --> Shortcuts --> 选择"New"**  进行相关设置即可(如下图)



## web项目run as 菜单中没有 run on server

eclipse中已安装好tomcat，某maven web项目原本也可以在tomcat中运行，突然发现右击该项目的 run as 菜单中，没有了 run on serven 选项，项目也无法在tomcat中跑了。

原因是该项目的Dynamic Web Module 被我不经意间删除了，如下图。MyEclipse中的Web项目导入到Eclipse后，也会出现这样的情况。主要把Dynamic Web Module勾上即可。



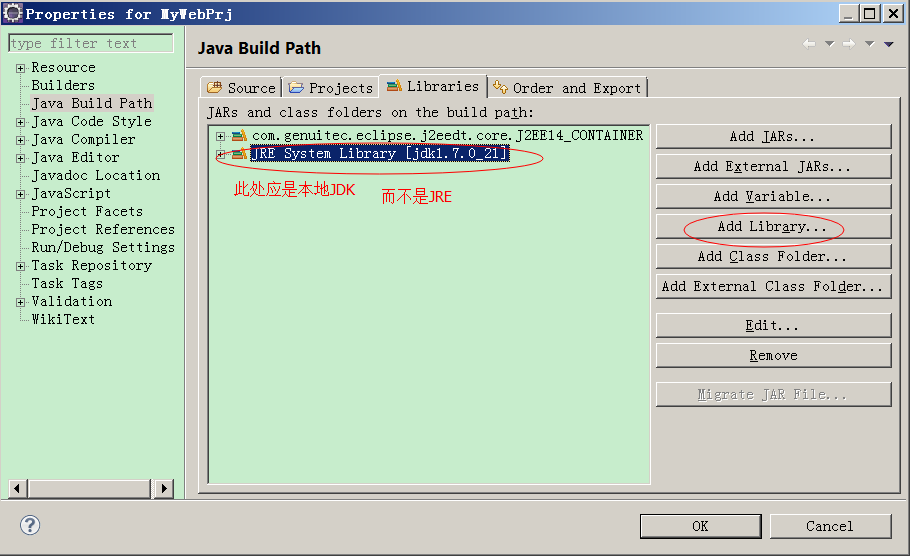
## 为什么eclipse新建项目的默认编码为何是GBK,而不是UTF-8

我们用的操作系统是中文操作系统，windows操作系统的编码是GBK，所以eclipse默认的新项目的编码是GBK的，你在Window->Preferences->General->Workspace->Text file encoding 将其改为UFT-8即可，同样，如果你想修改html,css等文件的编码格式，你在Window->Preferences->General->Content Types中可以修改

## 如何删除Eclipse弃用的工作空间？

在 eclipse\configuration\.settings\org.eclipse.ui.ide.prefs 文件里找到相应的空间，删除即可。

## eclipse因引用本地的JDK而不是JRE。

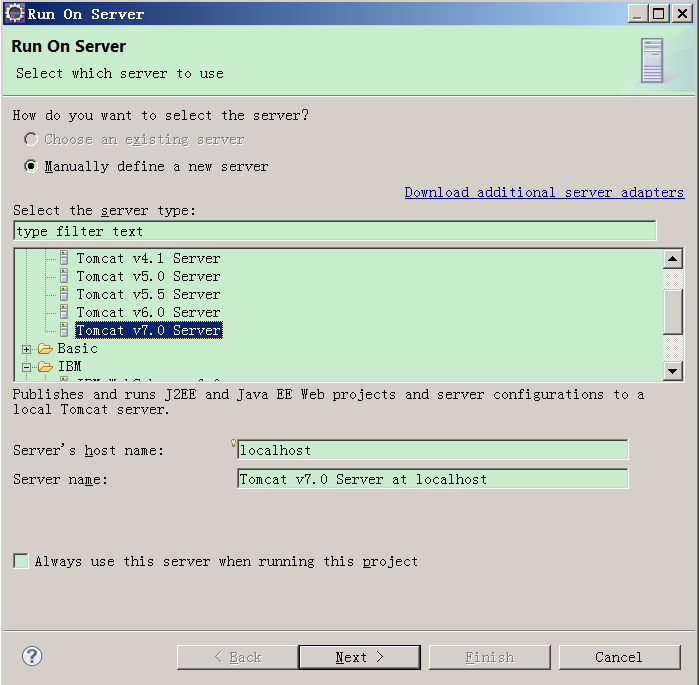


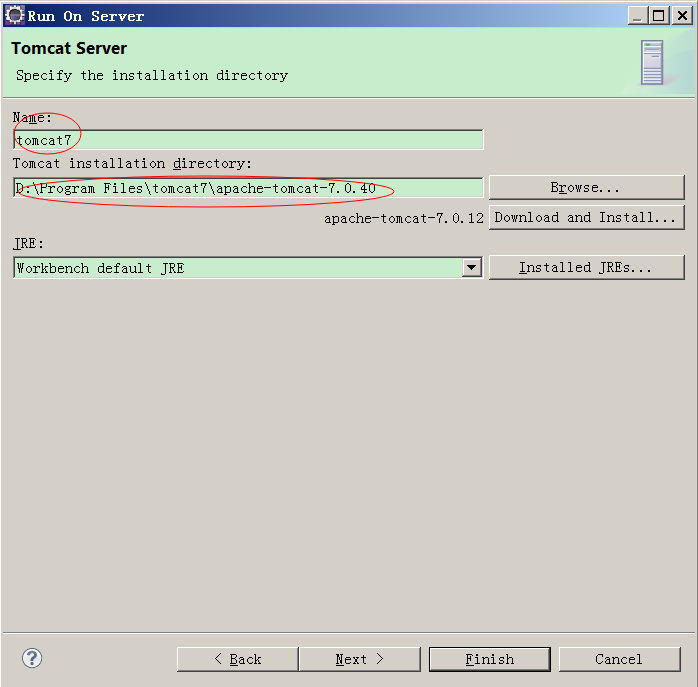
## Eclipse中安装Tomcat

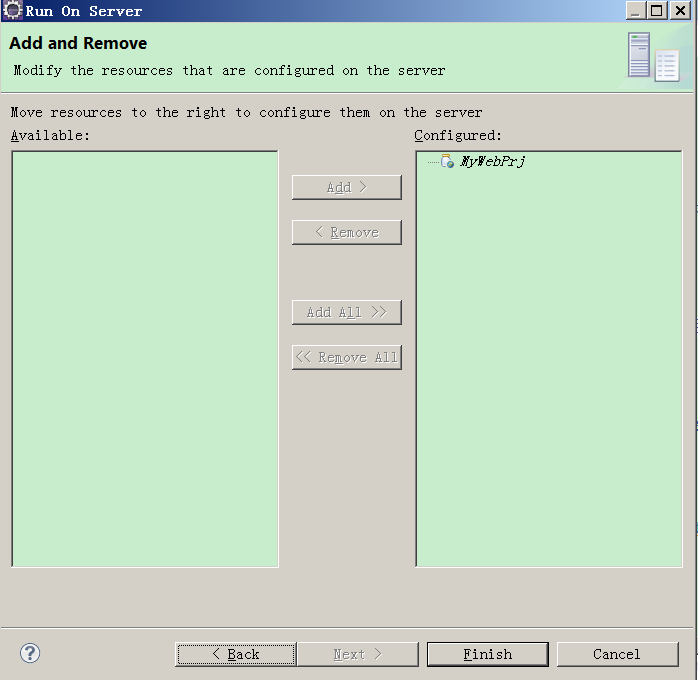
1、在本地安装并配置好tomcat

2、新建一个Web项目，右击该项目，选择Run as -> run on server.

3、在弹出的run on server框上选择要安装的tomcat版本

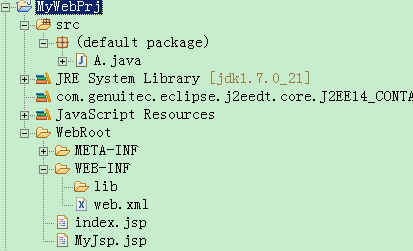






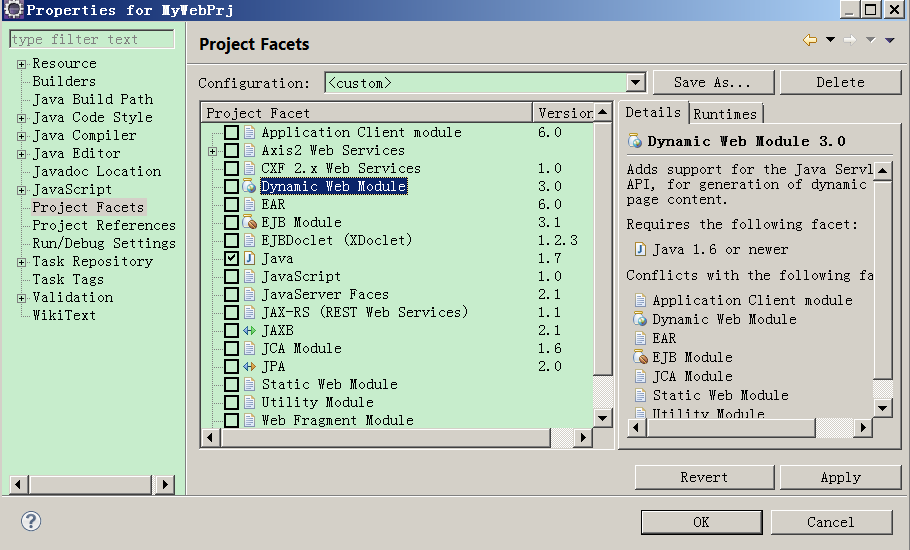
## 将MyEclipse的Web项目导入到Eclipse

将一个MyEclipse的Web项目导入到Eclipse后，项目并没有报错，奇怪的是，右击该项目，选择run as菜单，此时并没有出现run on server 选项。项目结构如下：

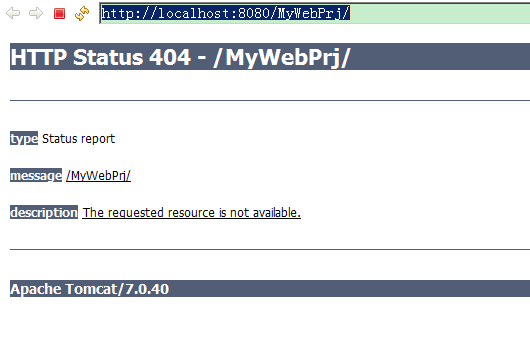


解决的办法如下：

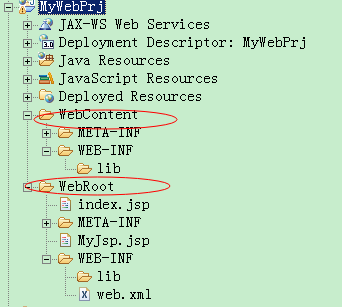
1、为项目添加Dynamic Web Module



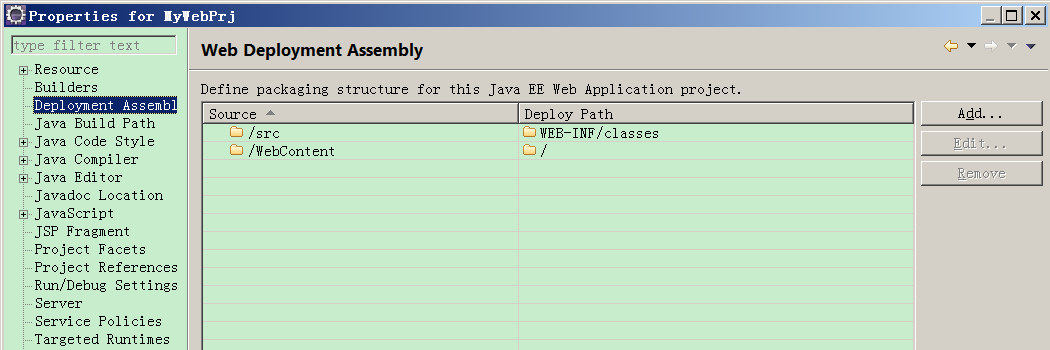
2、添加Dynamic Web Module后，run as菜单出现run on server了。执行run on server，在tomcat上运行该项目，结果切报错。



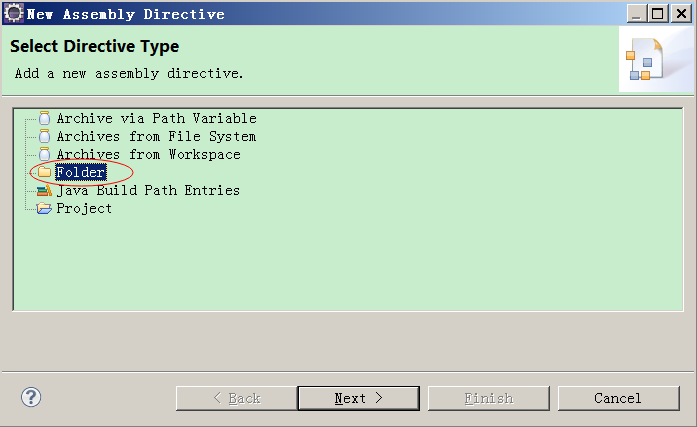
3、原来添加Dynamic Web Module后，项目结构中多了一个WebContent目录。Myeclipse中默认的Web目录是WebRoot，而Eclipse中默认的Web目录是Webcontent，因此报错。

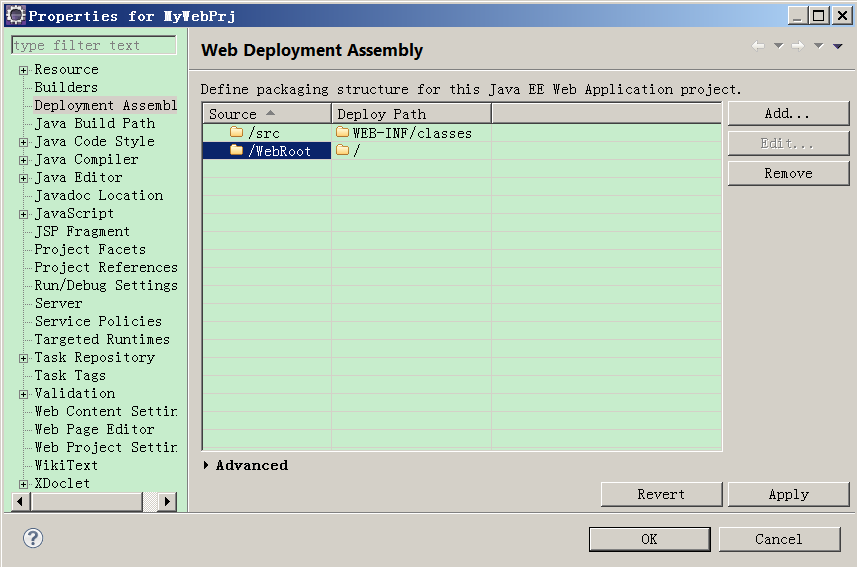


4、在项目属性中修改项目的部署目录。添加Dynamic Web Module后项目部署目录如下：

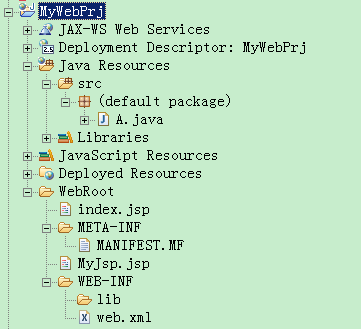


删除WebContent目录，然后点击Add按钮添加Folder，让Folder指向WebRoot





5、在项目的目录结构中，删除WebContent目录，删除后如下：



6、重新执行run on server即可。

## Eclipse中Project Explorer和Package Explorer这两个有什么区别

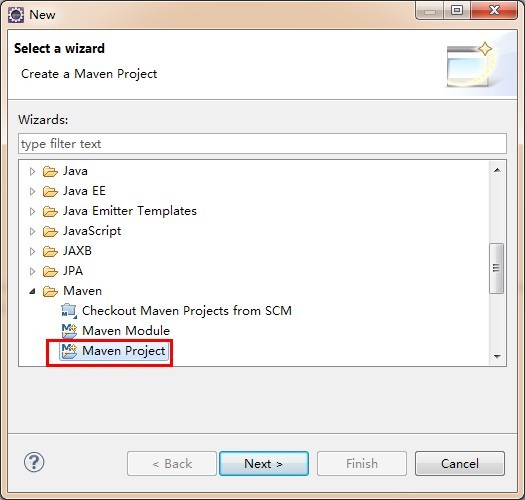
Package Explorer不会显示修改某些文件后会自动更改的文件，如class文件，在web工程里web-inf下的classes文件夹在Package Explorer下会隐藏  
  
Project Explorer会显示所有文件

# maven

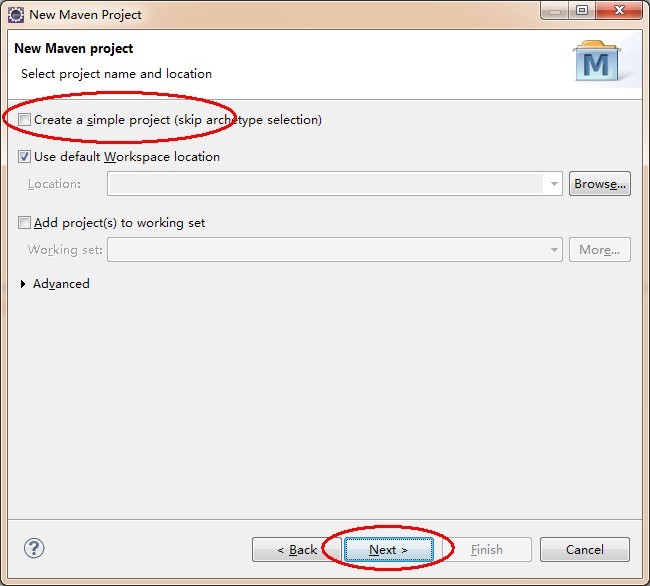
## Eclipse+Maven创建webapp项目

Eclipse+Maven创建webapp项目<二>

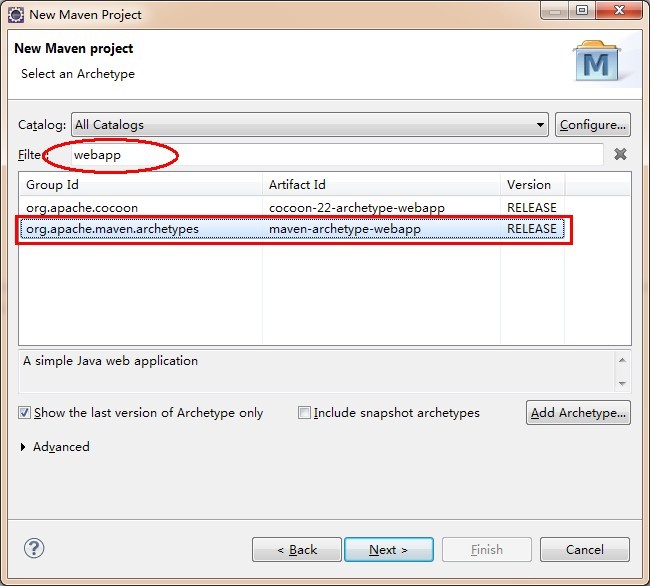
1、开启eclipse，右键new——》other，如下图找到maven project



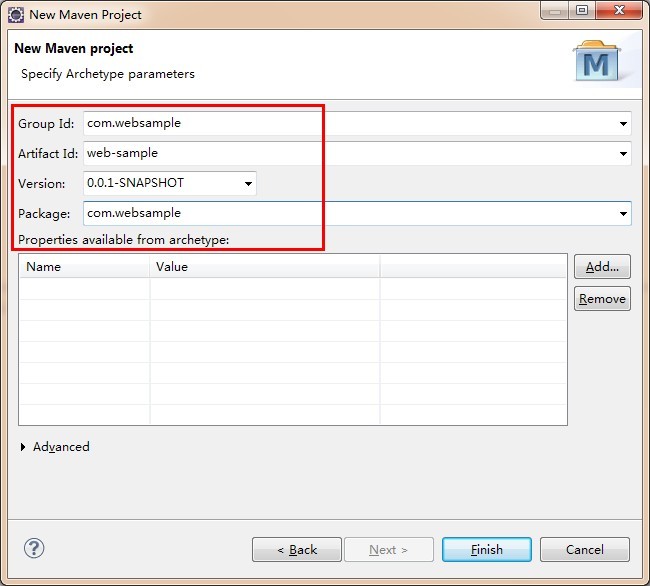
2、选择maven project，显示创建maven项目的窗口，此次不勾选Create a simple project，直接next，如图所示：



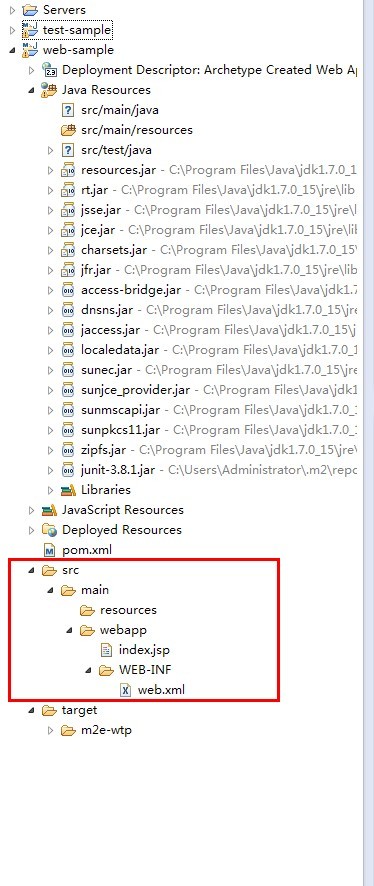
3、在Filter中搜索webapp，选择如图所示，并next：



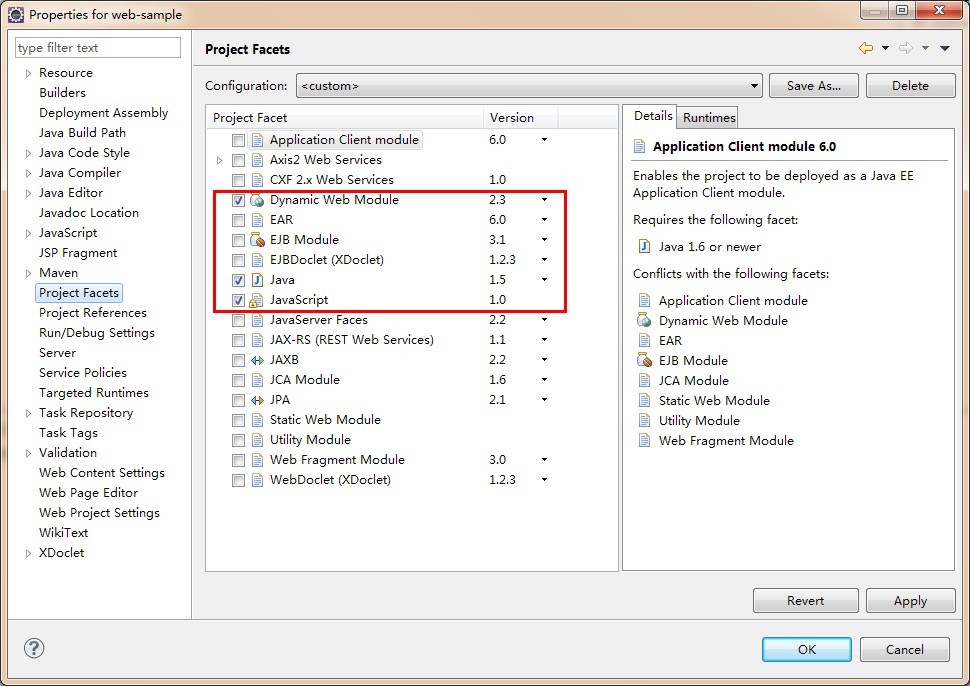
4、输入maven项目的基本信息，如下图所示：



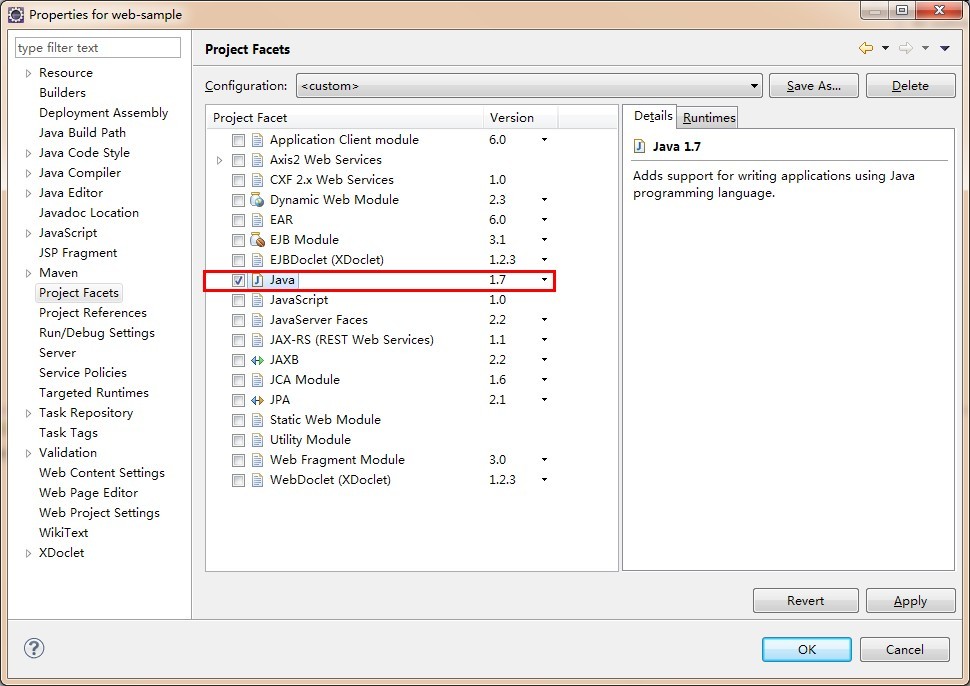
5、完成maven项目的创建，生成相应的webapp项目结构，如下所示，此处有部分结构是项目不需要的，我们需要去掉：



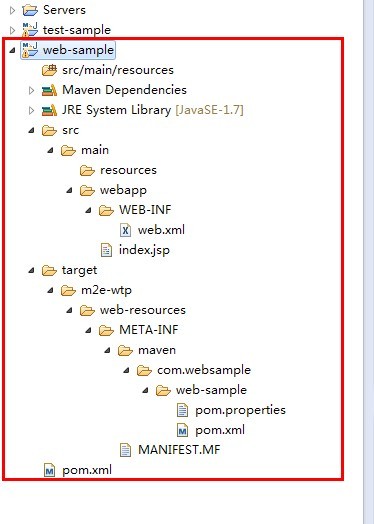
6、选择项目，右键选择Properties，进入属性页面，选择到Maven菜单下，如下图所示：



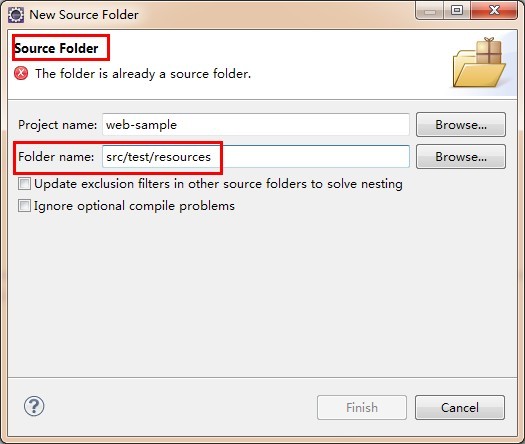
7、选择java版本为1.7，并去掉其他两项，如下图：



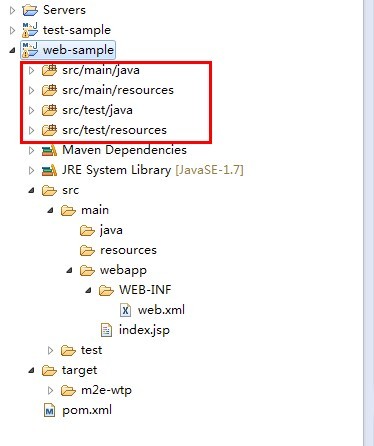
8、点击ok之后，再次回到项目结构，此时项目结构比较清晰，符合我们想要创建的maven项目：



9、创建没有显示的源代码文件夹，如下图所示：



10、最后得到完成的maven webapp项目结构如下：



## Eclipse+Maven构建springmvc项目

一.背景介绍

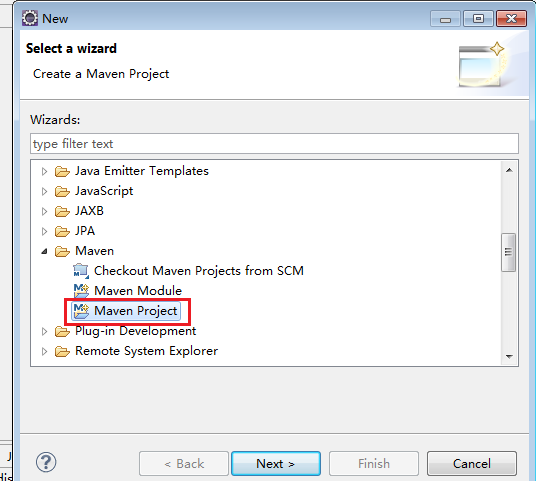
　　对于初学者，用maven构建项目并不是一件容易的事，springmvc并不是MVC中的主流，但是本人觉得springmvc比struts要好用，真正做到了零配置。一次使用，就放不下了。

二.准备工作

　1.Eclipse 3.7

　　2.maven

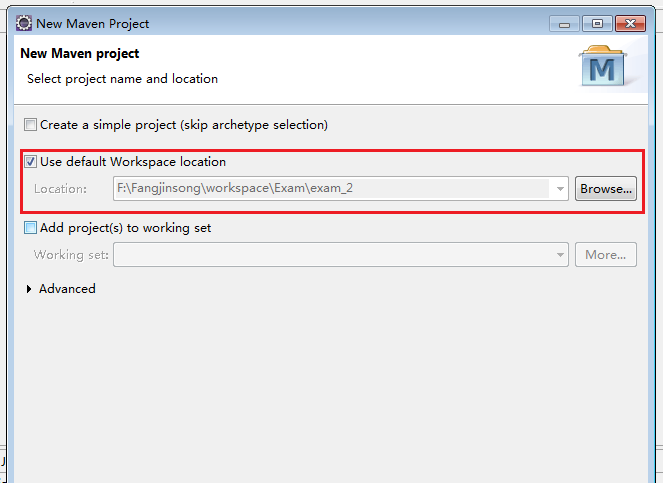
　　3.Eclipse 需要安装maven插件。url:maven - <http://download.eclipse.org/technology/m2e/releases> 。安装maven-3.0.4。并且选择本地的maven，如下图：



三.构建工程

　　1.用maven插件构建项目框架

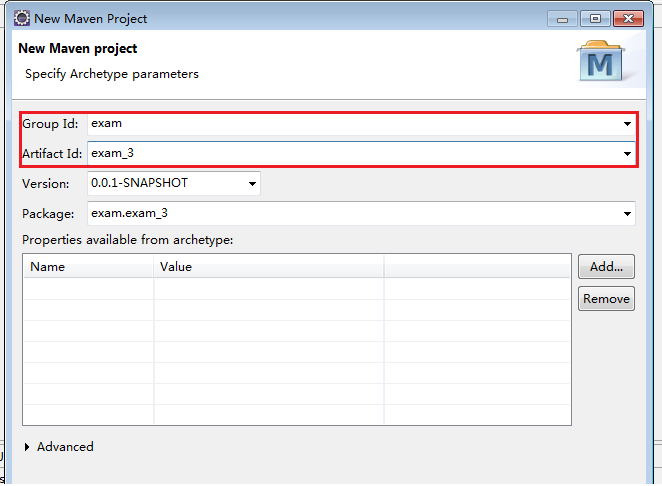
　　maven具有强大构建功能，使用maven可以构建多种不同类型的工程。这里我们构建maven-archetype-webapp类型的项目。在Eclipse->New中选择other,找到maven Project型。如下图：



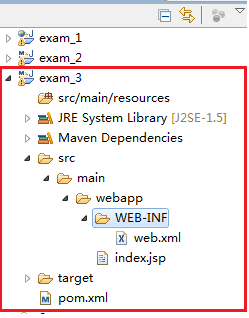
在选完路径之后，我们选择构建类型，如下图：



接下来，填写工程的Group Id，Artifact Id，如下图：



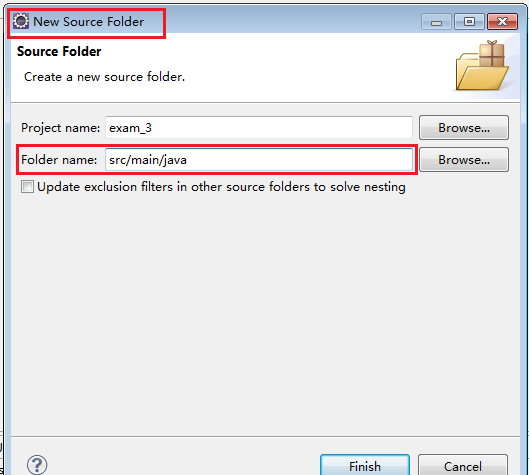
这里的Group Id就是大项目的id，Arifact Id就是该项目的Id。就像一个大项目中有许多小项目组成一样。此时，我们的项目已经成型了，样子如下图：



接下来，我们要完善项目的目录，配置。

　　2.完善项目

　　首先，完善目录，增加重要的source Folder，这个不是简单的Floder,这些文件夹是会参与编译的。增加src/main/java，src/test/resources，src/test/java目录。让目录变成标准的maven结构。如下图：



接下来，改变一些配置：

让工程的JDK用本地的jdk;

让工程的字符集为UTF-8;

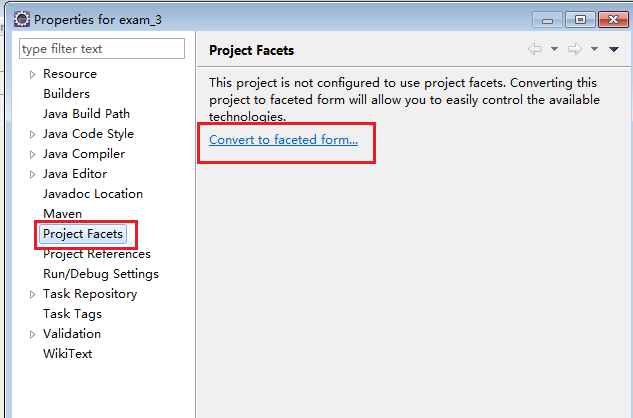
改变工程的目录顺序；

　　这些都完成之后，工程目录应该是如下的样子：

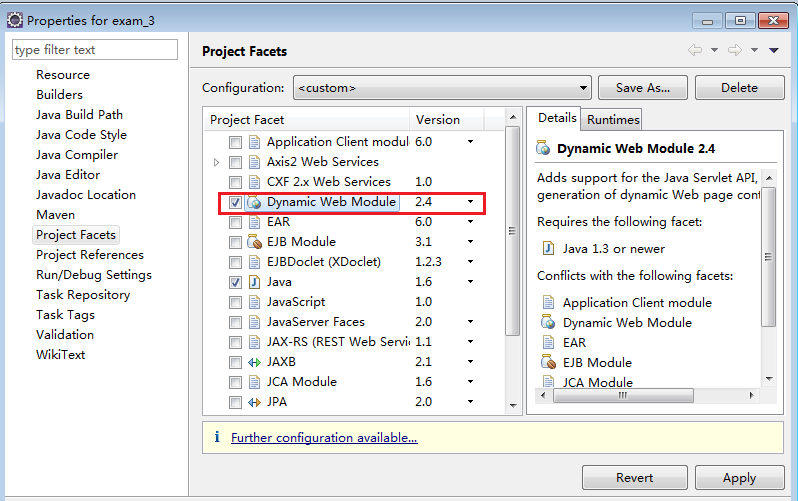


　　3.将工程变成web工程

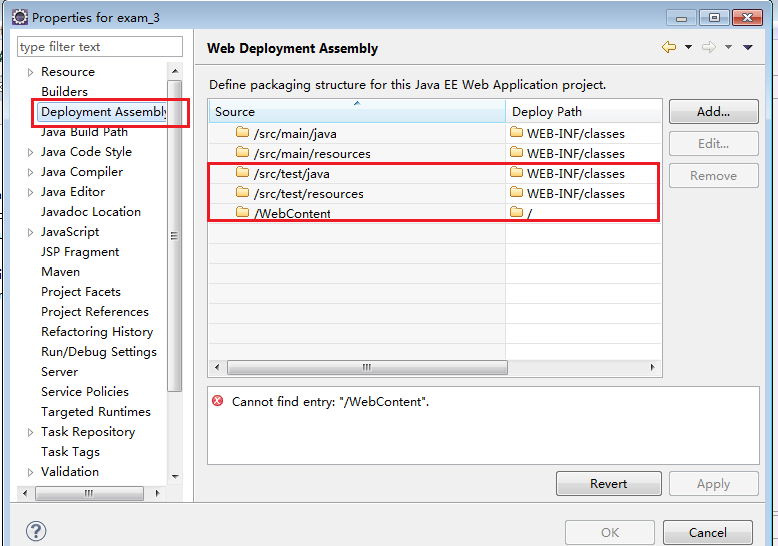
　　此时，我们的工程还不是标准的web工程，可以在eclipse中增加web工程的特性，选择工程的Properties,选Project Facets,如下图：



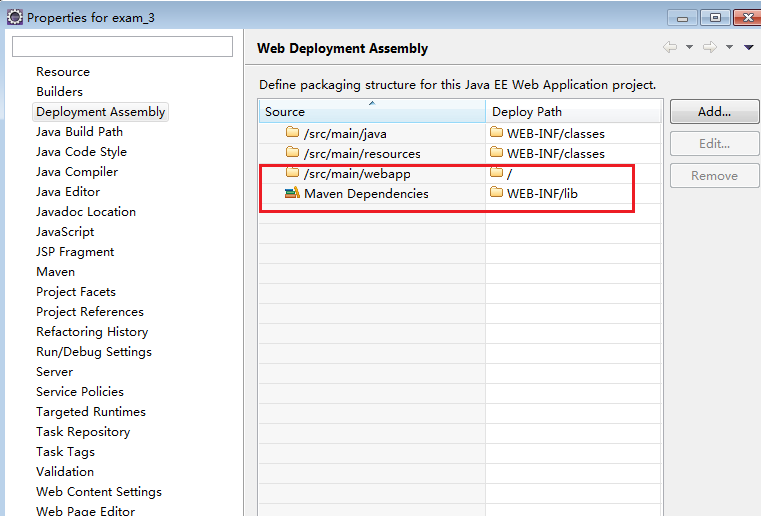
这里，我们选择Dynamic Web Module,版本选择2.4，这个版本比较通用。如下图：



此时，我们看到目录中多了一个WebContent目录，由于使用maven构建，web目录是src/main/webapp,所以我们删除WebContent目录。接下来，要配置web项目的发布目录，就是Deployment Assembly，如图：



test目录不要发布，WebContent目录没有了，所以这三项都删掉。并且增加src/main/webapp目录，和Maven Dependenices,完成之后如下图：



于是，我们的工程就完全是一个web工程了。

　　4.赋予工程的springmvc特性

　　配置web.xml，使其具有springmvc特性，主要配置两处，一个是ContextLoaderListener，一个是DispatcherServlet。代码如下：

http://images.cnblogs.com/OutliningIndicators/ExpandedBlockStart.gif

[复制代码](javascript:void(0);)

1 <?xml version="1.0" encoding="UTF-8"?>

2 <web-app version="2.4" xmlns="http://java.sun.com/xml/ns/j2ee"

3 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

4 xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee

5 http://java.sun.com/xml/ns/j2ee/web-app\_2\_4.xsd">

6

7 <listener>

8 <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>

9 </listener>

10

11 <servlet>

12 <servlet-name>exam</servlet-name>

13 <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>

14 </servlet>

15

16 <servlet-mapping>

17 <servlet-name>exam</servlet-name>

18 <url-pattern>/</url-pattern>

19 </servlet-mapping>

20

21 <welcome-file-list>

22 <welcome-file>index.jsp</welcome-file>

23 </welcome-file-list>

24 </web-app>

[复制代码](javascript:void(0);)

　　配置ContextLoaderListener表示，该工程要以spring的方式启动。启动时会默认在/WEB-INF目录下查找applicationContext.xml作为spring容器的配置文件，这里可以初始化一些bean，如DataSource。我们这里什么也不做。代码如下：

http://images.cnblogs.com/OutliningIndicators/ExpandedBlockStart.gif

1 <?xml version="1.0" encoding="UTF-8"?>

2 <!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN//EN" "http://www.springframework.org/dtd/spring-beans.dtd">

3 <beans>

4 </beans>

　　配置DispatcherServlet表示，该工程将采用springmvc的方式。启动时也会默认在/WEB-INF目录下查找XXX-servlet.xml作为配置文件，XXX就是DispatcherServlet的名字，该文件中将配置两项重要的mvc特性：

HandlerMapping,负责为DispatcherServlet这个前端控制器的请求查找Controller；

ViewResolver,负责为DispatcherServlet查找ModelAndView的视图解析器。

代码如下：

http://images.cnblogs.com/OutliningIndicators/ExpandedBlockStart.gif

[复制代码](javascript:void(0);)

1 <?xml version="1.0" encoding="UTF-8"?>

2 <!-- Bean头部 -->

3 <beans xmlns="http://www.springframework.org/schema/beans"

4 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:p="http://www.springframework.org/schema/p"

5 xmlns:mvc="http://www.springframework.org/schema/mvc" xmlns:context="http://www.springframework.org/schema/context"

6 xmlns:util="http://www.springframework.org/schema/util"

7 xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-3.0.xsd

8 http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-3.0.xsd

9 http://www.springframework.org/schema/mvc http://www.springframework.org/schema/mvc/spring-mvc-3.0.xsd

10 http://www.springframework.org/schema/util http://www.springframework.org/schema/util/spring-util-3.0.xsd">

11

12 <!-- 激活@Controller模式 -->

13 <mvc:annotation-driven />

14 <!-- 对包中的所有类进行扫描，以完成Bean创建和自动依赖注入的功能 需要更改 -->

15 <context:component-scan base-package="cc.monggo.web.controller" />

16

17 <bean class="org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandlerAdapter" />

18

19 <bean id="viewResolver" class="org.springframework.web.servlet.view.InternalResourceViewResolver">

20 <property name="prefix">

21 <value>/WEB-INF/jsp/</value>

22 </property>

23 <property name="suffix">

24 <value>.jsp</value>

25 </property>

26 </bean>

27 </beans>

[复制代码](javascript:void(0);)

　　5.让maven自动配置jar包

　　在用maven生成框架时，就生成了pop.xml，这就是maven的配置文件。我们要引入spring-web,servlet等特性的包。代码如下：

http://images.cnblogs.com/OutliningIndicators/ExpandedBlockStart.gif

[复制代码](javascript:void(0);)

1 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

2 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4\_0\_0.xsd">

3 <modelVersion>4.0.0</modelVersion>

4 <groupId>exam</groupId>

5 <artifactId>exam\_3</artifactId>

6 <packaging>war</packaging>

7 <version>0.0.1-SNAPSHOT</version>

8 <name>exam\_3 Maven Webapp</name>

9 <url>http://maven.apache.org</url>

10 <dependencies>

11 <dependency>

12 <groupId>junit</groupId>

13 <artifactId>junit</artifactId>

14 <version>3.8.1</version>

15 <scope>test</scope>

16 </dependency>

17

18 <dependency>

19 <groupId>org.springframework</groupId>

20 <artifactId>spring-web</artifactId>

21 <version>3.0.5.RELEASE</version>

22 </dependency>

23

24 <dependency>

25 <groupId>org.springframework</groupId>

26 <artifactId>spring-webmvc</artifactId>

27 <version>3.0.5.RELEASE</version>

28 </dependency>

29

30 <dependency>

31 <groupId>org.apache.geronimo.specs</groupId>

32 <artifactId>geronimo-servlet\_2.5\_spec</artifactId>

33 <version>1.2</version>

34 </dependency>

35

36 </dependencies>

37 <build>

38 <finalName>exam\_3</finalName>

39 </build>

40 </project>

[复制代码](javascript:void(0);)

　　maven就是这么简单，一旦保存，maven就会自动下载pop.xml的jar包。此时可以看到目录中Maven Dependencies下生成了jar包。

　　更多的jar包可以在maven中心库下载：[http://mvnrepository.com](http://mvnrepository.com/)。

　　6.做个测试

　　说了一大堆，只有运行起来才有意思，下面写个简单的测试。先写Controller。编写两个类，LoginControler.java，LoginForm.java。代码如下：

http://images.cnblogs.com/OutliningIndicators/ExpandedBlockStart.gif

[复制代码](javascript:void(0);)

1 package cc.monggo.web.controller;

2

3

4 import javax.servlet.http.HttpServletRequest;

5 import javax.servlet.http.HttpServletResponse;

6

7 import org.springframework.stereotype.Controller;

8 import org.springframework.web.bind.annotation.RequestMapping;

9 import org.springframework.web.servlet.ModelAndView;

10

11 import cc.monggo.domain.LoginForm;

12

13 @Controller

14 public class LoginController {

15 @RequestMapping(value="login")

16 public ModelAndView login(HttpServletRequest request,HttpServletResponse response,LoginForm command ){

17 String username = command.getUsername();

18 ModelAndView mv = new ModelAndView("/index/index","command","LOGIN SUCCESS, " + username);

19 return mv;

20 }

21 }

[复制代码](javascript:void(0);)

http://images.cnblogs.com/OutliningIndicators/ExpandedBlockStart.gif

[复制代码](javascript:void(0);)

1 package cc.monggo.domain;

2

3

4 public class LoginForm {

5 private String username;

6 private String password;

7 public String getUsername() {

8 return username;

9 }

10 public void setUsername(String username) {

11 this.username = username;

12 }

13 public String getPassword() {

14 return password;

15 }

16 public void setPassword(String password) {

17 this.password = password;

18 }

19 }

[复制代码](javascript:void(0);)

　　再增加一些jsp,首页的index.jsp,主要是做跳转，代码如下：

http://images.cnblogs.com/OutliningIndicators/ExpandedBlockStart.gif

1 <%

2 request.getRequestDispatcher("/WEB-INF/jsp/login/login.jsp").forward(request,response);

3 %>

　  还有两个jsp，做些简单的功能，一个表单login.jsp,一个表单提交的返回index.jsp，代码如下：

http://images.cnblogs.com/OutliningIndicators/ExpandedBlockStart.gif

[复制代码](javascript:void(0);)

1 <%@ page language="java" contentType="text/html; charset=utf-8"

2 pageEncoding="utf-8"%>

3 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

4 <html>

5 <head>

6 <meta http-equiv="Content-Type" content="text/html; charset=utf-8">

7 <title>Insert title here</title>

8 </head>

9 <body>

10 <div>

11 <form action="login" methed="get">

12 <input type="text" name="username">

13 <input type="submit" value="SUBMIT">

14 </form>

15 </div>

16 </body>

17 </html>

[复制代码](javascript:void(0);)

http://images.cnblogs.com/OutliningIndicators/ExpandedBlockStart.gif

[复制代码](javascript:void(0);)

1 <%@ page language="java" contentType="text/html; charset=utf-8"

2 pageEncoding="utf-8"%>

3 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

4 <html>

5 <head>

6 <meta http-equiv="Content-Type" content="text/html; charset=utf-8">

7 <title>Insert title here</title>

8 </head>

9 <body>

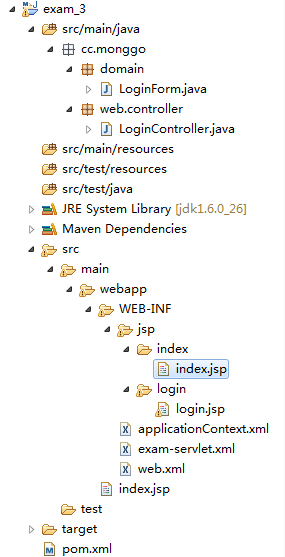
10 ${command}

11 </body>

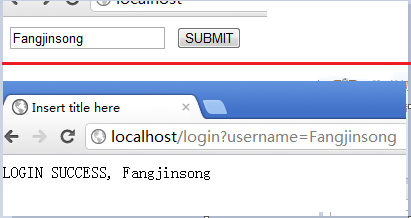
12 </html>

[复制代码](javascript:void(0);)

　　整个项目的目录结构如下：



　　在tomcat中运行，即可。不要使用Tomcat插件，可能有问题。就用普通的方式。运行效果如下图：



## Maven项目打包缺少XML文件

解决办法，在pom.xml中把xml文件包含进去。

<build>

<resources>

<resource>

<directory>src/main/java</directory>

<includes>

<include>com/sun/mapping/ChargeMapper.xml</include>

<include>com/sun/mapping/ConsumeMapper.xml</include>

<include>com/sun/mapping/RefundMapper.xml</include>

<include>com/sun/mapping/UserMapper.xml</include>

</includes>

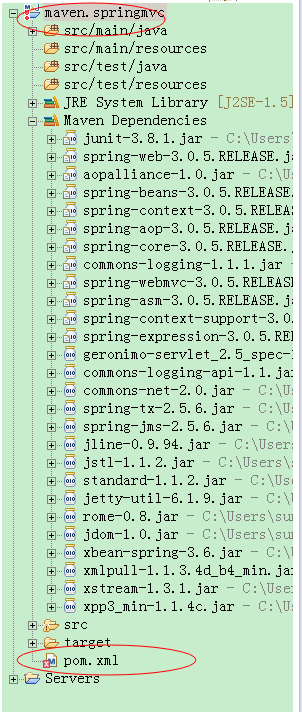
<filtering>true</filtering>

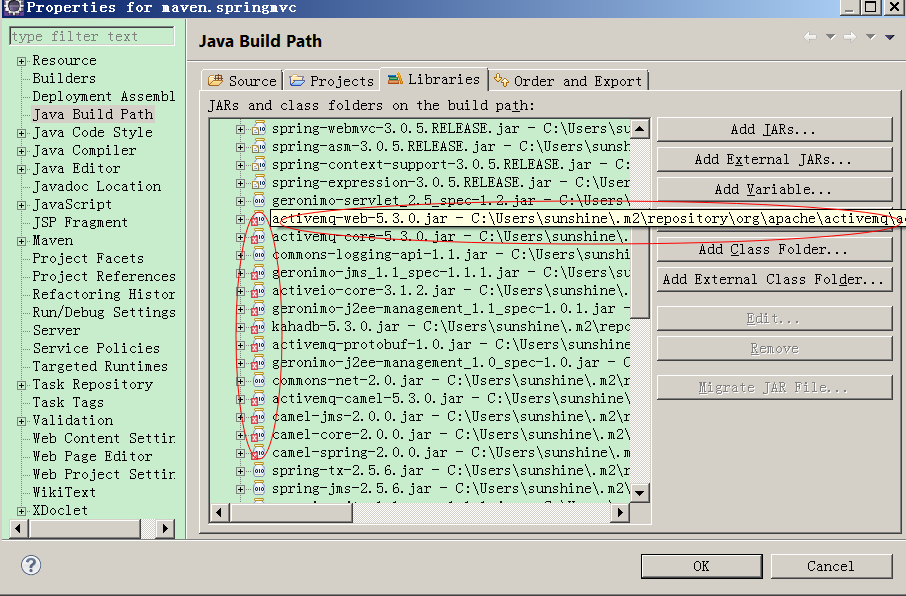
</resource>

</resources>

</build>

## pom.xml更新下载jar包失败





1.问题描述

在通过pom.xml中的依赖坐标,从私服下载依赖包时,往往会遇到如下的类似问题:

The container 'Maven Dependencies' references non existing library '${groupid}/${artifactid}-${version}.jar'

e.g.

|  |  |
| --- | --- |
| 1 | The container 'Maven Dependencies' references non existing library 'E:\Maven\repo\org\apache\hadoop\hadoop-yarn-api\2.0.0-cdh4.5.0\hadoop-yarn-api-2.0.0-cdh4.5.0.jar' |

2.解决方案

上面问题往往是在下载依赖过程中网络出现问题导致的.

此时我们本机已经开始下载依赖代码,但是下载失败,本机仓库中会在${MAVEN\_repo}/${groupid}/${artifactid}/${version} 路径下面生成 \*.lastUpdated 的文件.

\*.lastUpdated实在可恶,下次再下载时由于它的存在Maven不会再去私服中下载

2.1 解决方法:(针对PC与Maven私服之间网络传输问题)

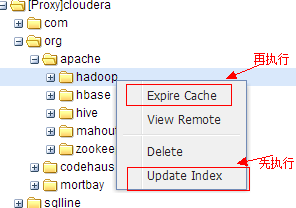
1. 打开本地仓库所在目录, 通过win文件夹的搜索功能,查找 \*.lastUpdated ,然后将找到的文件全部删除
2. 重新 Maven Update Project,
   1. 若在eclipse中操作时勾选上 Force Update of Snapshots / Releases
   2. 若在命令行操作时增加 -U 参数

一般情况下上面方法即可解决问题. 若问题依然存在,请确认下主机与Maven私服之间网络是否畅通.可以通过在浏览器中打开私服地址确认.

若本机与私服之前网络畅通,那这个问题可能是由于私服与中央仓库(或者其他第三方仓库)之间下载依赖时网络故障导致的.

2.2 解决方法:(针对私服与中央仓库或第三方仓库之间网络传输问题)

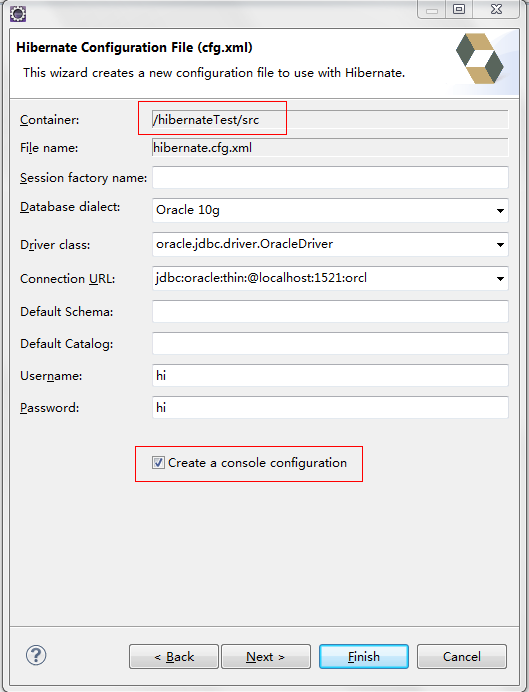
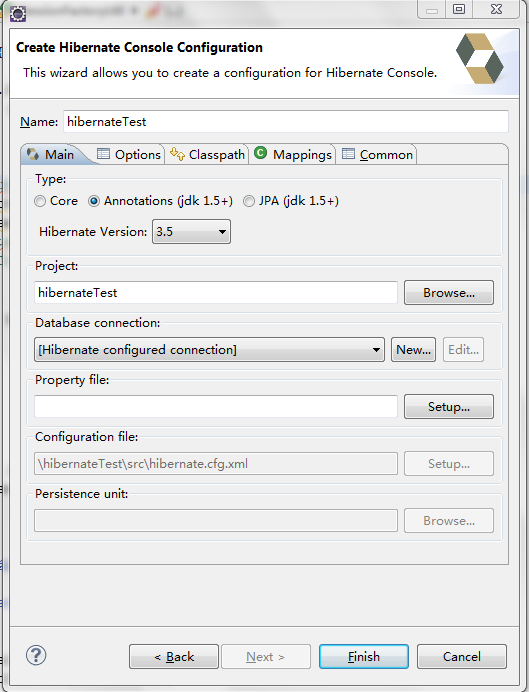
1. 使用管理员登陆私服页面(Sonatype Nexus)
2. 在仓库的浏览页面找到下载失败的包, 右键菜单 "Update Index", 右键菜单 "Expire Cache"

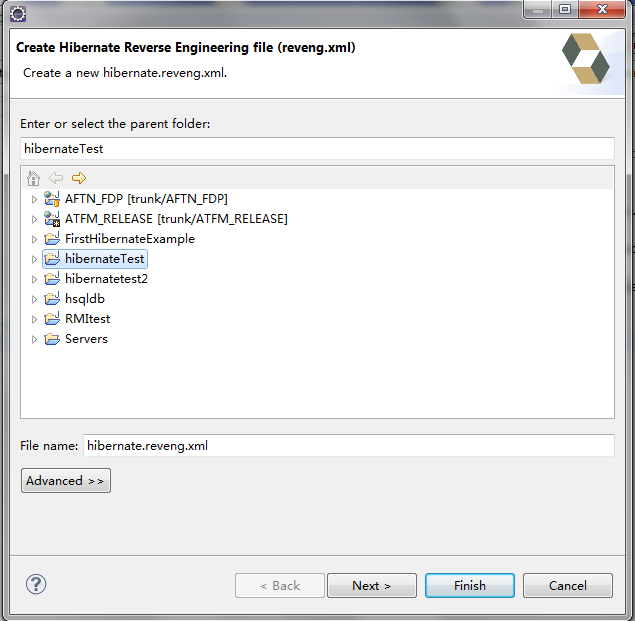
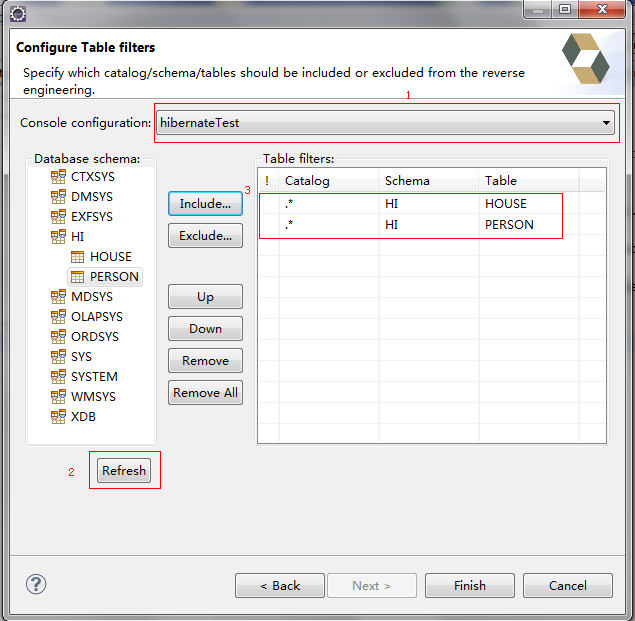
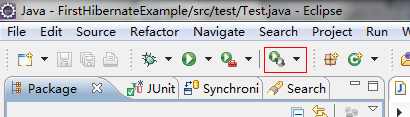
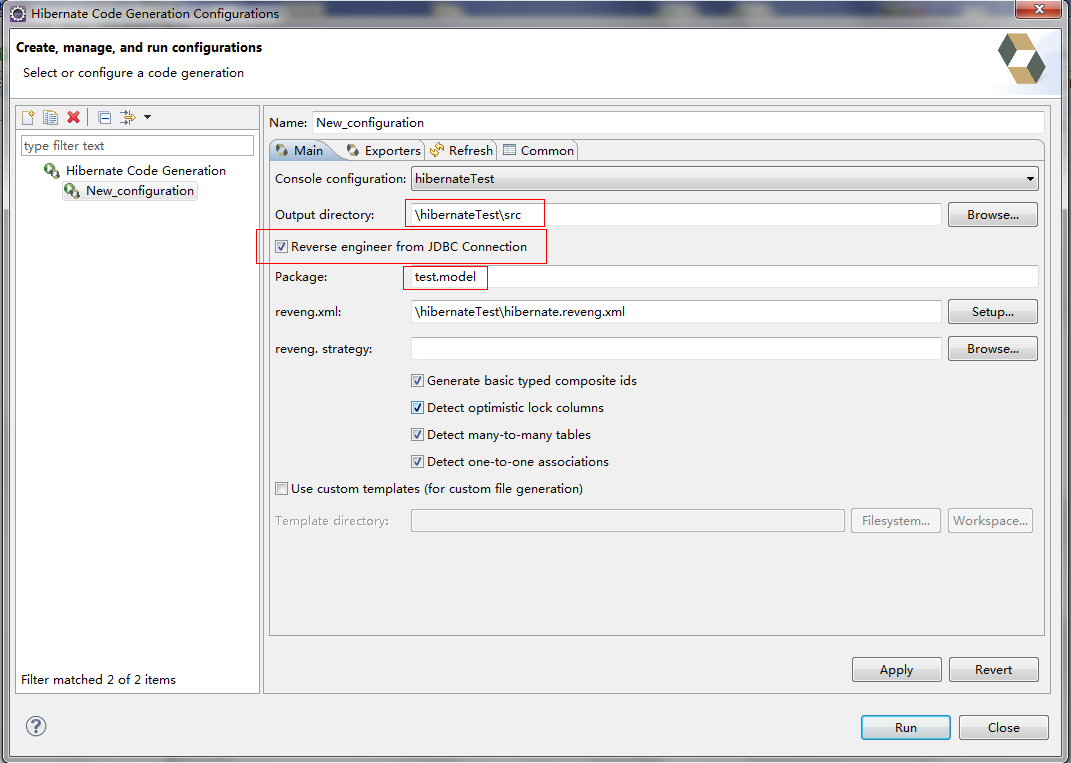
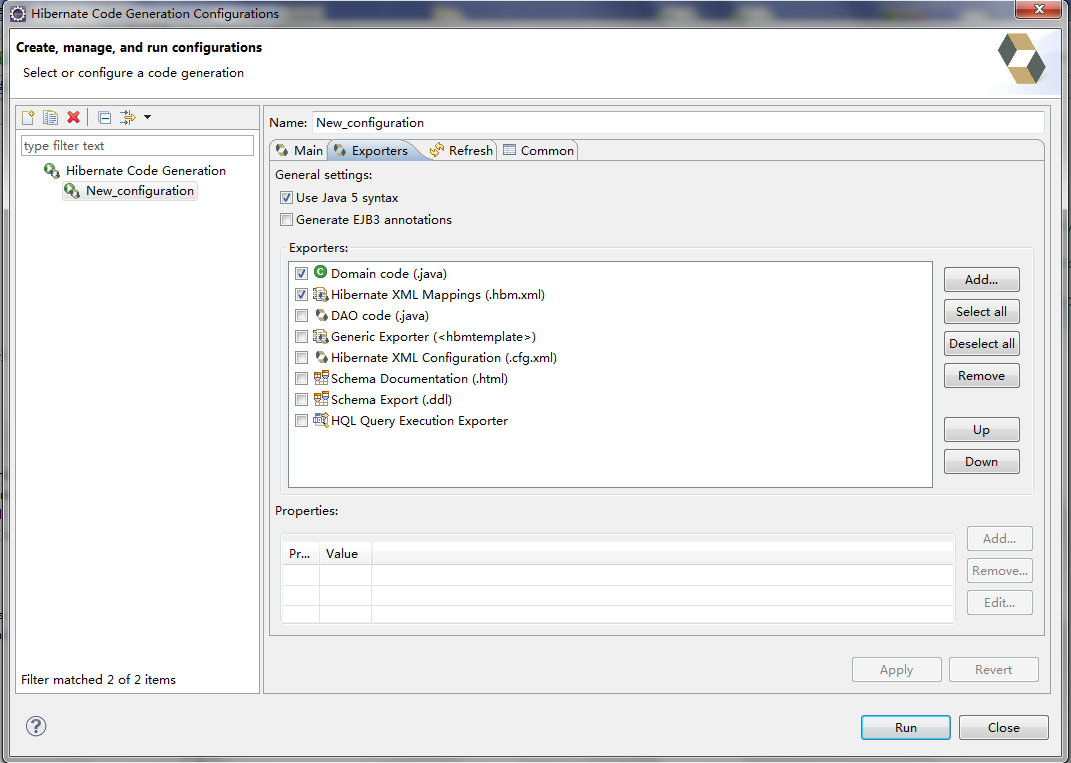
[](http://static.oschina.net/uploads/space/2013/1206/162311_NoW2_1189023.png)

1. 按照PC与私服之间网络问题的解决步骤处理一遍

# Hibernate

## Eclipse 使用Hibernate Tools插件生成POJO

1. 很多时候我们已经设计好了数据库，需要使用hibernate来做数据持久化，因此需要根据数据库中的表结构生成相应的POJO.
3. 本例使用hibernatetools来自动创建pojo。
5. 测试环境：eclipse3.7（indigo）+hibernate4.1+hibernatetools3.3+oracle10g
6. 生成POJO只需要导入数据库驱动包，不需要导入其他Hibernate包。
7. **1.下载hibernatetools插件**
9. hibernatetools可从官网下载，但是需要注意下载和自己使用的eclipse版本对应的版本包，然后安装到eclipse中取，link安装方式参见：<http://maimode.iteye.com/blog/1164524>
11. 下载地址:
12. 1、首先，要去Hibernate的网站下载这个工具包。一个大约14M的ZIP压缩文件。  
    <http://jaist.dl.sourceforge.net/sourceforge/jboss/HibernateTools-3.2.4.Beta1-R200810311334.zip>  
    2、解压缩下载好的文件，释放出features和plugins这两个文件夹和其中的内容。完成后将这两个文件夹复制到Eclipse的目录下（这两个文件夹在Eclipse中是默认存在的，直接复制就行了）。  
    3、到这步为止HibernateTools的安装就算结束了
13. **2.创建hibernate配置文件及console配置文件**
15. 新建java项目，建立包test,test.hibernate,test.model。
16. 选中项目，右键-》new-》other-》hibernate-》hibernate configuration file：
17. 
19. 然后 next，配置console文件：
20. 

23. **3.创建reverse文件**
25. 选中项目，右键-》new-》other-》hibernate-》hibernate reverse engineering file
26. 
28. 下一步，配置要生成POJO对应的表：
29. 
31. **4.反向工程创建pojo**
33. 在hibernate code generation configuration菜单中配置
34. 
36. 
38. 
40. 执行run后便生成了相应的pojo文件：
42. 

45. 一般自动生成的mapping文件（XXX.hbm.xml）都需要修改后才能使用。

## 日期类型

某字段数据库中数据类型为date时，Hibernate的实体类中，该字段可以是Date类型，也可以是String类型。String类型时，默认是yyyy-MM-dd HH:mm:ss格式。

# spring

## 通过ClassPathXmlApplicationContext读取配置文件

ApplicationContext context =

**new** ClassPathXmlApplicationContext(**new** String[]

{ "file:D:/eclipse-jee-juno/mavenspace/maven.springmvc/src/main/webapp/WEB-INF/applicationContext.xml" });

SpringPublisher publisher = (SpringPublisher) context

.getBean("springPublisher");

## Windows下ActiveMQ下载、安装部署

1、下载：http://activemq.apache.org/download.html 最新Windows版本

2、安装

(1) 首先配置JAVA环境变量

JAVA\_HOME=D:\Program Files\Java\jdk1.5.0

CLASSPAHT=.;D:\Program Files\Java\jdk1.5.0\lib\tools.jar;D:\Program Files\Java\jdk1.5.0\lib\td.jar;D:\Program Files\Java\jdk1.5.0\lib

PATH=D:\Program Files\Java\jdk1.5.0\bin;

(2)直接解压至任意目录(例如：D:\apache-activemq-5.3.0)

3、启动ActiveMQ服务器：直接运行\bin\win32\activemq.bat

当运行成功后

4、打开ActiveMQ消息管理后台系统 http://localhost:8161/admin/

## 消息队列：Spring JMS

### [SPRING与JMS结合的实例](http://www.cnblogs.com/phoebus0501/archive/2011/02/28/1966896.html)

|  |
| --- |
| package jms.activemq.myexample.spring;  import java.util.Date;  import javax.jms.JMSException;  import javax.jms.Message;  import javax.jms.Session;  import javax.jms.TextMessage;  import org.springframework.jms.core.MessageCreator;    public class MyMessageCreator implements MessageCreator {        /\*\*       \* 消息序号       \*/      private int msgNo;        public MyMessageCreator(int no) {          this.msgNo = no;      }        @Override      public Message createMessage(Session session) throws JMSException {          TextMessage textMsg = session.createTextMessage();          textMsg.setText(new Date() + "第" + this.msgNo + "条消息发出");            return textMsg;      }    } |
| package jms.activemq.myexample.spring;    import javax.jms.\*;    /\*\*   \* Text消息监听   \*   \* @author xiaochuanyu   \*   \*/  public class TextListener implements MessageListener {        /\*\*       \* Casts the message to a TextMessage and displays its text.       \*       \* @param message       \*            the incoming message       \*/      public void onMessage(Message message) {          TextMessage msg = null;            try {              if (message instanceof TextMessage) {                  msg = (TextMessage) message;                  System.out.println("Reading message: " + msg.getText());              } else {                  System.out.println("Message of wrong type: "                          + message.getClass().getName());              }          } catch (JMSException e) {              System.out.println("JMSException in onMessage(): " + e.toString());          } catch (Throwable t) {              System.out.println("Exception in onMessage():" + t.getMessage());          }      }  } |

|  |
| --- |
| package jms.activemq.myexample.spring;    import org.springframework.context.ApplicationContext;  import org.springframework.context.support.ClassPathXmlApplicationContext;    public class SpringJmsTestMain {        public static void main(String[] args) throws InterruptedException {            ApplicationContext context = new ClassPathXmlApplicationContext(                  new String[] { "SpringJms/SpringJms.xml" });            SpringPublisher publisher = (SpringPublisher) context                  .getBean("springPublisher");          publisher.start();      }  } |
| package jms.activemq.myexample.spring;    import javax.jms.Destination;    import org.springframework.jms.core.JmsTemplate;    public class SpringPublisher {      /\*\*       \* Jms模板       \*/      private JmsTemplate template;        /\*\*       \* Topic       \*/      private Destination topic;        public JmsTemplate getTemplate() {          return template;      }        public void setTemplate(JmsTemplate template) {          this.template = template;      }        public Destination getTopic() {          return topic;      }        public void setTopic(Destination topic) {          this.topic = topic;      }        /\*\*       \* Start       \*       \* @throws InterruptedException       \*/      public void start() throws InterruptedException {            int messageCount = 10;            while ((--messageCount) > 0) {              sendMessage(messageCount);              Thread.sleep(1000);          }      }        /\*\*       \* 消息发送       \*/      protected void sendMessage(int msgNO) {            this.template.send(this.topic, new MyMessageCreator(msgNO));      }  } |

[复制代码](javascript:void(0);)

<?xml version="1.0" encoding="UTF-8"?>  
<beans xmlns="http://www.springframework.org/schema/beans"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-2.0.xsd">  
  
  
  
 <!-- jms 连接工厂 -->  
 <bean id="jmsFactory" class="org.apache.activemq.ActiveMQConnectionFactory">  
 <property name="brokerURL">  
 <value>tcp://localhost:61616</value>  
 </property>  
 </bean>  
  
  
 <!-- jms 连接池 -->  
   
 <!--   
 <bean id="pooledJmsFactory" class="org.apache.activemq.pool.PooledConnectionFactory">  
 <property name="connectionFactory">  
 <ref local="jmsFactory" />  
 </property>  
 </bean>  
 -->  
  
 <!-- jms Topic -->  
 <bean id="myTopic" class="org.apache.activemq.command.ActiveMQTopic"  
 autowire="constructor">  
 <constructor-arg value="STOCKS.JAVA" />  
 </bean>  
  
  
 <!-- 消息监听器 -->  
 <bean id="myTextListener" class="jms.activemq.myexample.spring.TextListener">  
 </bean>  
  
  
 <!-- jms Consumer -->  
 <bean id="javaConsumer"  
 class="org.springframework.jms.listener.DefaultMessageListenerContainer">  
 <property name="connectionFactory" ref="jmsFactory" />  
 <property name="destination" ref="myTopic" />  
 <property name="messageListener" ref="myTextListener" />  
 </bean>  
  
 <!-- jms 模板 -->  
 <bean id="jmsTemplate" class="org.springframework.jms.core.JmsTemplate">  
 <property name="connectionFactory">  
 <ref local="jmsFactory" />  
 </property>  
 </bean>  
  
  
 <!-- 消息发布器 -->  
 <bean id="springPublisher" class="jms.activemq.myexample.spring.SpringPublisher">  
 <property name="template">  
 <ref local="jmsTemplate" />  
 </property>  
 <property name="topic">  
 <ref local="myTopic" />  
 </property>  
 </bean>  
</beans>

### [Spring+JMS例子](http://www.blogjava.net/wangbing/archive/2010/03/31/317093.html)

第一: 在applicationContext.xml

<?xml version="1.0" encoding="UTF-8"?>  
<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN//EN" "http://www.springframework.org/dtd/spring-beans.dtd">

<beans>    
 <bean id="connectionFactory" class="org.apache.activemq.ActiveMQConnectionFactory">  
   <property name="brokerURL">  
     <value>tcp://192.168.3.48:61616</value>  
   </property>  
    
 </bean>  
   
 <bean id="jmsTemplate" class="org.springframework.jms.core.JmsTemplate">  
    <property name="connectionFactory">  
      <ref bean="connectionFactory"/>  
    </property>  
 </bean>  
   
 <bean id="destination" class="org.apache.activemq.command.ActiveMQQueue">  
   <constructor-arg index="0">  
     <value>HelloWorldQueue</value>  
   </constructor-arg>  
 </bean>  
</beans>

2.写发送方

package ch13.JMS;

import javax.jms.Destination;  
import javax.jms.JMSException;  
import javax.jms.Message;  
import javax.jms.Session;

import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
import org.springframework.jms.core.JmsTemplate;  
import org.springframework.jms.core.MessageCreator;

public class HelloWorldSender  
{  
 public static void main(String args[]) throws Exception  
 {  
  ApplicationContext context = new ClassPathXmlApplicationContext(new String[] { "ch13/JMS/applicationContext.xml" });  
  JmsTemplate jmsTemplate = (JmsTemplate) context.getBean("jmsTemplate");  
  Destination destination = (Destination) context.getBean("destination");  
  jmsTemplate.send  
  (  
   destination, new MessageCreator()  
   {  
    public Message createMessage(Session session) throws JMSException  
    {  
     return session.createTextMessage("大家好这个是测试！");  
    }  
   }  
  );  
 }

}  
3.写接收方

package ch13.JMS;

import javax.jms.Destination;  
import javax.jms.TextMessage;

import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
import org.springframework.jms.core.JmsTemplate;

public class HelloWorldReciver  
{

 public static void main(String args[]) throws Exception  
 {  
  ApplicationContext context = new ClassPathXmlApplicationContext(new String[] { "ch13/JMS/applicationContext.xml" });  
  JmsTemplate jmsTemplate = (JmsTemplate) context.getBean("jmsTemplate");  
  Destination destination = (Destination) context.getBean("destination");  
  System.out.println("will wait:" + jmsTemplate.getReceiveTimeout()+ " seconds for message");  
  TextMessage msg = (TextMessage) jmsTemplate.receive(destination);  
  System.out.println("reviced msg is:" + msg.getText());  
 }

}

4.启动activemq中bin 下的activemq.bat

5.先运行 HelloWorldSender

6.再运行 HelloWorld

7.结果:

will wait:-1 seconds for message  
reviced msg is:大家好 这个是测试！

## Spring IOC

1、获取Bean的方法

ApplicationContext ctx = WebApplicationContextUtils.*getWebApplicationContext*(

request.getSession().getServletContext() );

TestService testService = (TestService)ctx.getBean("testService");

testService.sayHello();

# PowerDesigner

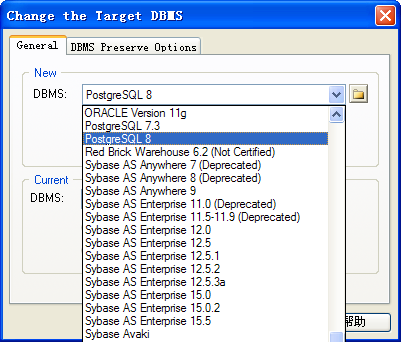
## pdb和pdm文件有什么区别

早期的PowerBuilder编译生成的动态链接库的后缀名是PDB，  
但后来PowerDesigner出来后，它的物理模型图（数据表结构）文件后缀名是PDM，但同时它会生成这个文件的备份文件，就是后缀名为PDB的文件。附文件名意思：  
PDB PowerBuilder Dynamic Library   
PDB PowerDesigner Physical Model Backup   
PDM Sybase Power Designer File

## [PowerDesigner导出sql](http://housen1987.iteye.com/blog/1476160)

1 按照数据库类型，切换数据库。

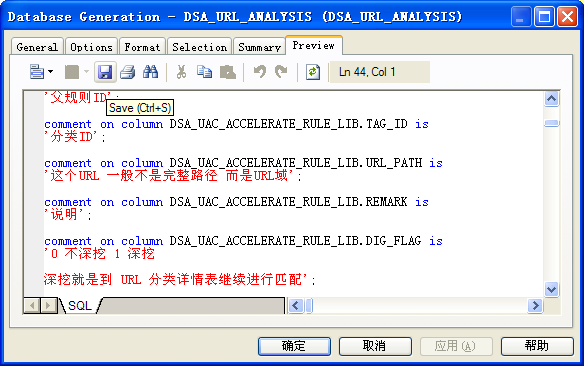
Database-> Change Current DBMS...



2 生成sql脚本

Database -> Database Generation 的Preview选项卡

点击保存按钮，保存产生的sql脚本到本地。



# Word 2007

## 标题与编号

1、设置一级标题

输入标题内容，选中内容，点击“标题1”，设置为一级标题。

在选中标题内容，右键-编号，选中要使用的编号即可。

后续同级别的标题，可使用格式刷进行设置。如现在要设置编号为3的标题，则选中编号为2的标题然后点格式刷，注意不能选中编号为1的标题点格式刷，以免编号错乱。

2、设置二级标题

输入标题内容，选中内容，点击“标题2”，设置为一级标题。

在选中标题内容，右键-编号，选中要使用的编号即可。

后续同级别的标题，可使用格式刷进行设置。如现在要设置编号为3的标题，则选中编号为2的标题然后点格式刷，注意不能选中编号为1的标题点格式刷，以免编号错乱。

## 标题多级列表

设置标题的时候希望出现多标题并且自动编号的标题，如下

1. XXXXXXXXXXX

1.1 XXXXXXXXXXX

2 XXXXXXXXXXX

2.1 XXXXXXXXXXX

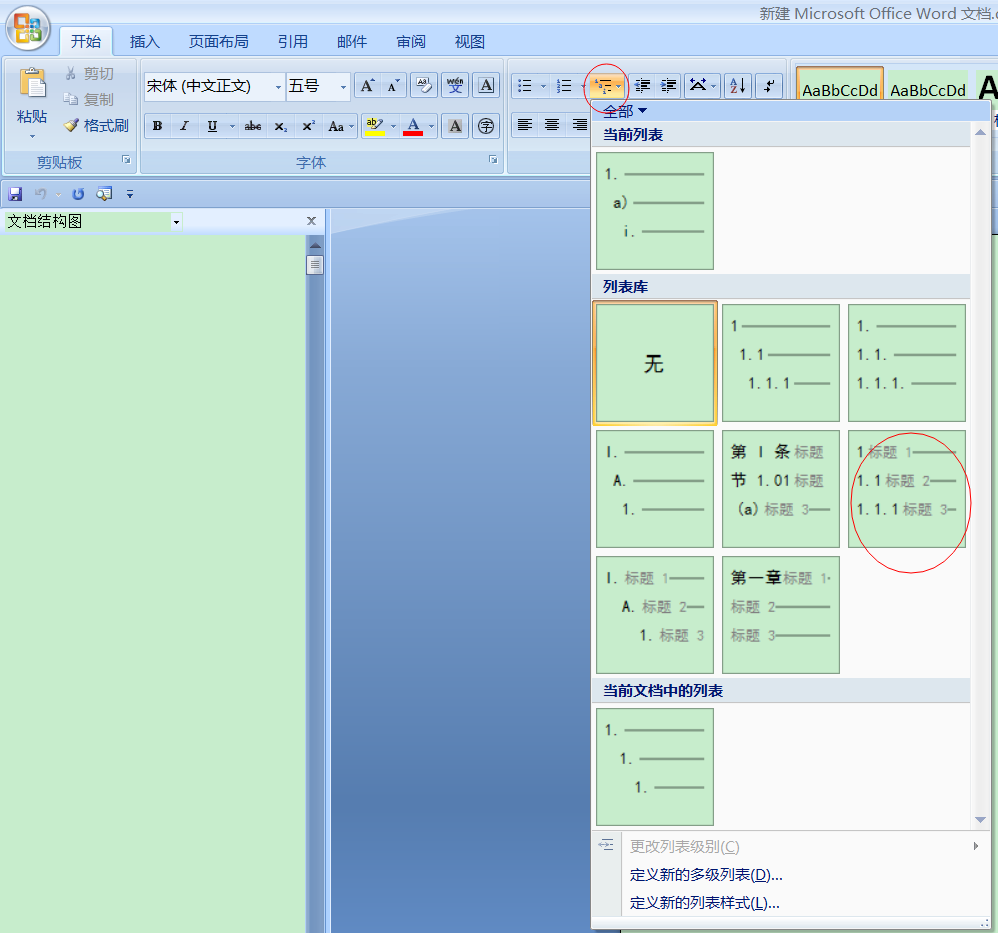
2.2 XXXXXXXXXXX

2.2.1 XXXXXXXXXXX

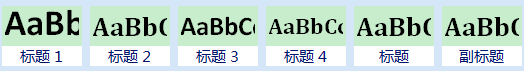
2.2.2 XXXXXXXXXXX

 2.3 XXXXXXXXXXX

输入一个标题内容，如下图将它设置为多级列表，则该标题自动变为一级标题。

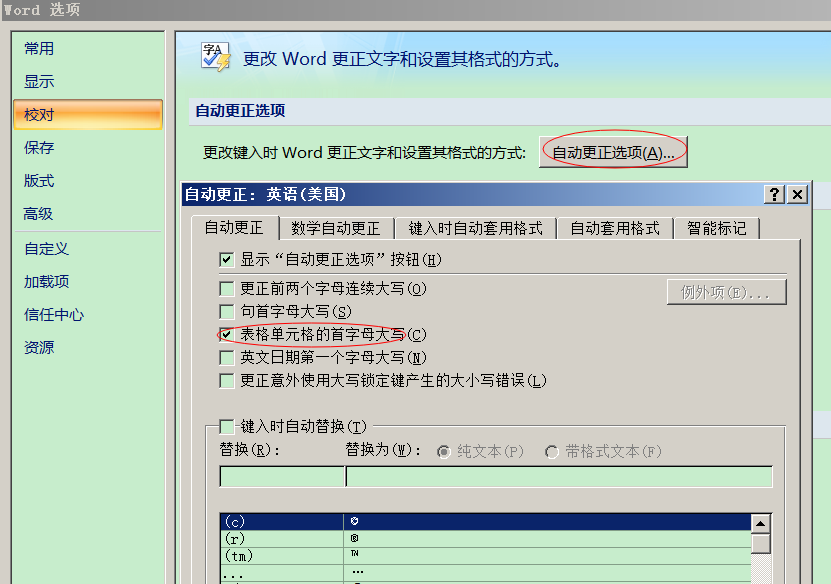


其他各级别的标题，只要选中标题内容，点击 标题1、标题2、标题3即可设置成功。

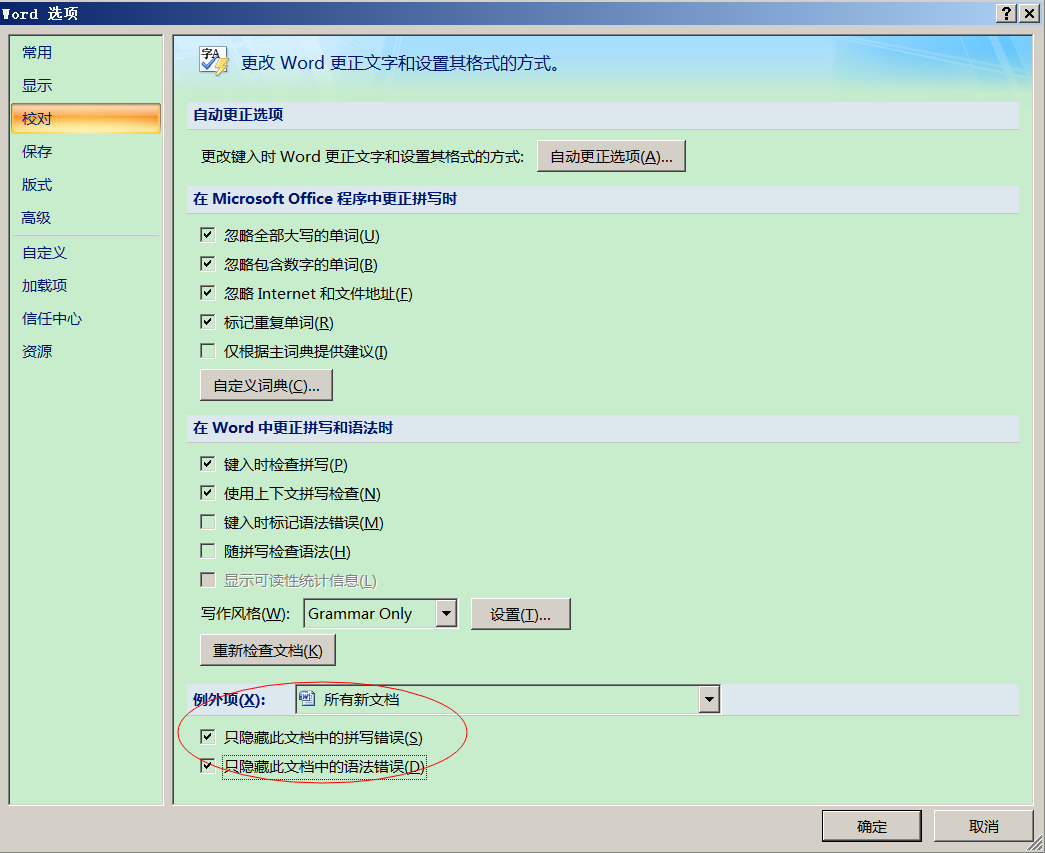


## word取消首字母大写





## 关闭语法检查功能



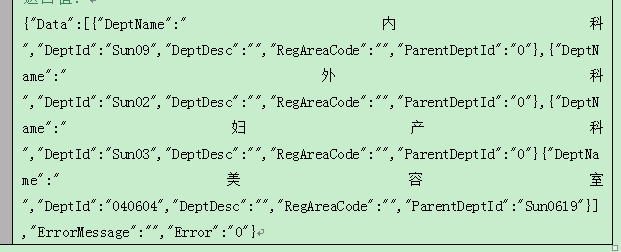
## 格式刷

格式刷的快捷键是 Ctrl + Shift + C/V

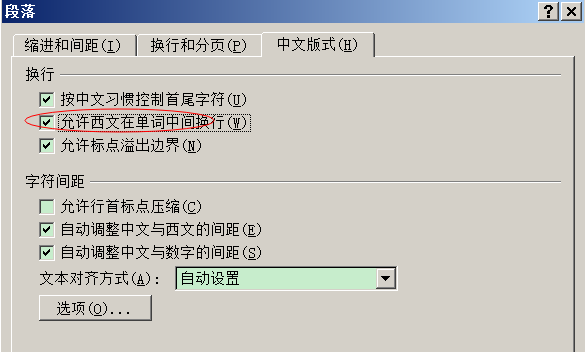
在使用“格式刷”时，不能先输入“页码”，那样的话，格式刷就不能翻页使用了。使用快捷键时，不受页码限制。

## 允许西文在单词中间换行

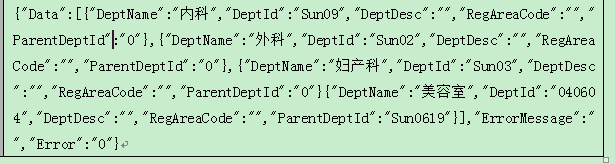
有时候由于西文单词不能换行导致一些段落格式十分混乱，例如：



可以选定内容然后在段落设置中勾上 允许西文在单词中间换行。

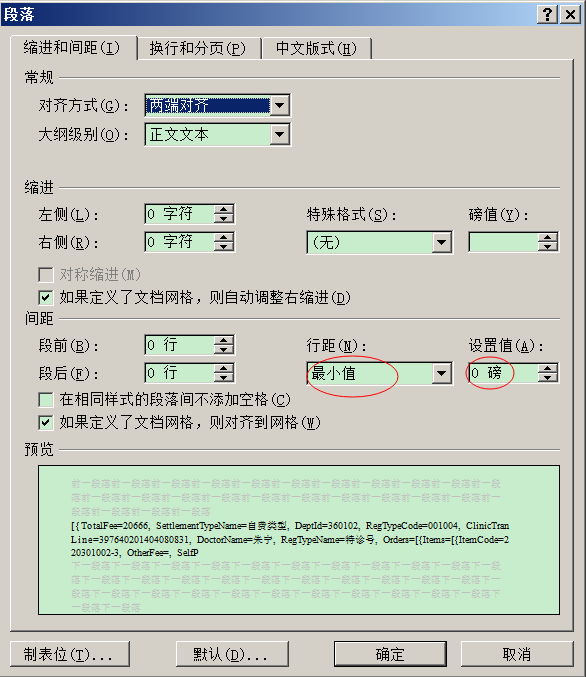


格式就会变得整齐了。



## 设置行距为0

如果要把段落的行距设置为0，必须要先把 行距 这个选项设置为“最小值”，然后才能设置值为0磅。



# 版本管理

## git

## github

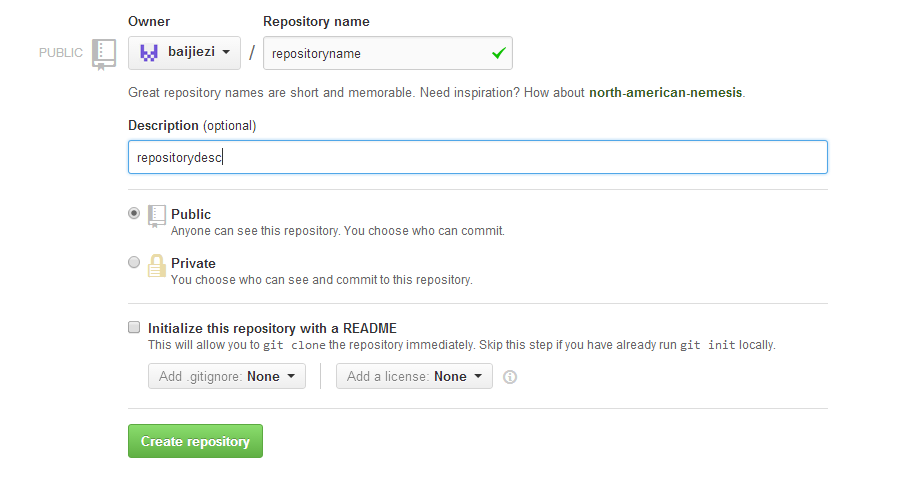
文档、电子书、个人笔记等最好放到版本管理空间，而不是放到云存储空间。因为文档和电子书常会做笔记修改，有版本管理功能的空间比较容易管理，而云空间就会造成笔记内容被覆盖丢失。

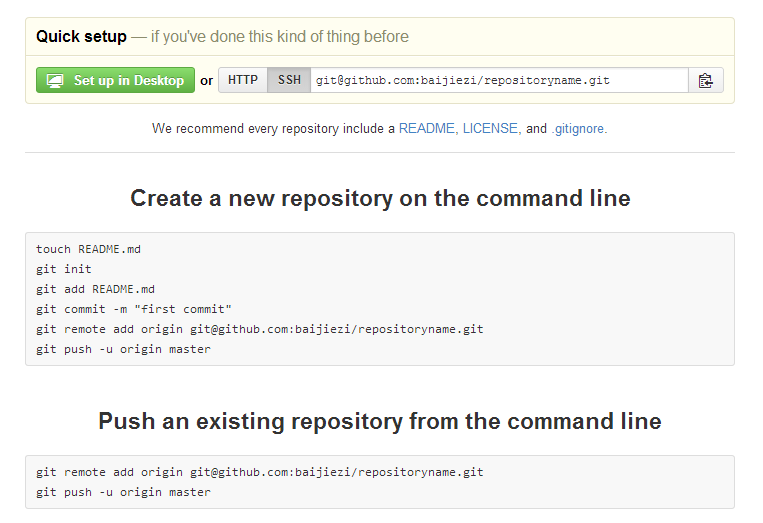
### 注册并登陆github

https://github.com/

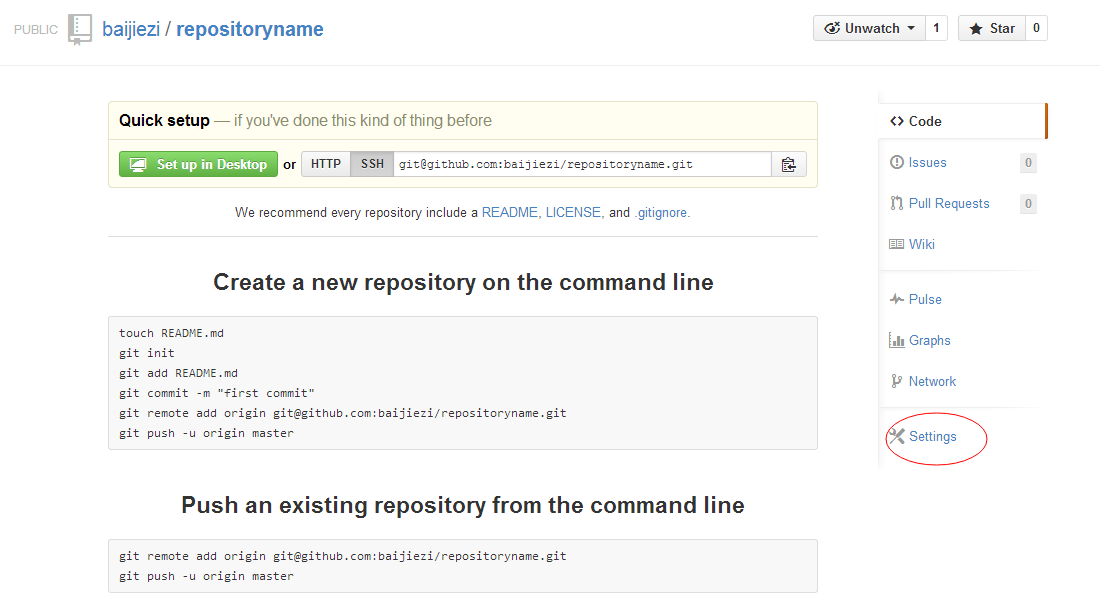
### 新建仓库

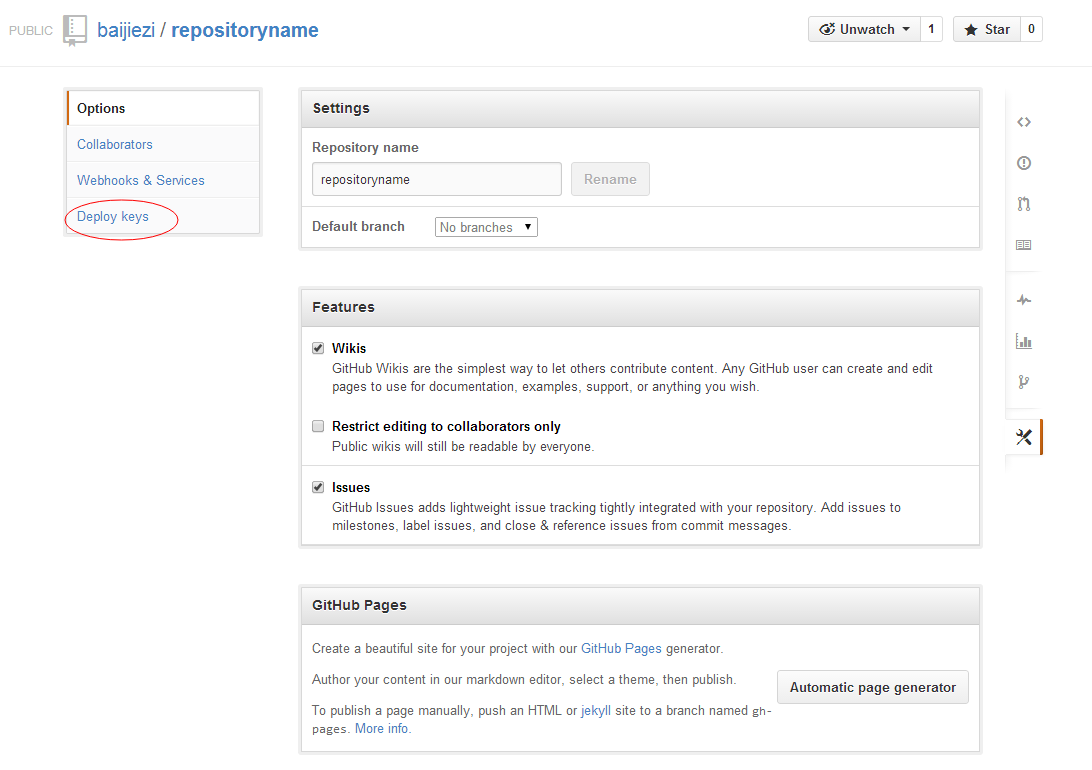


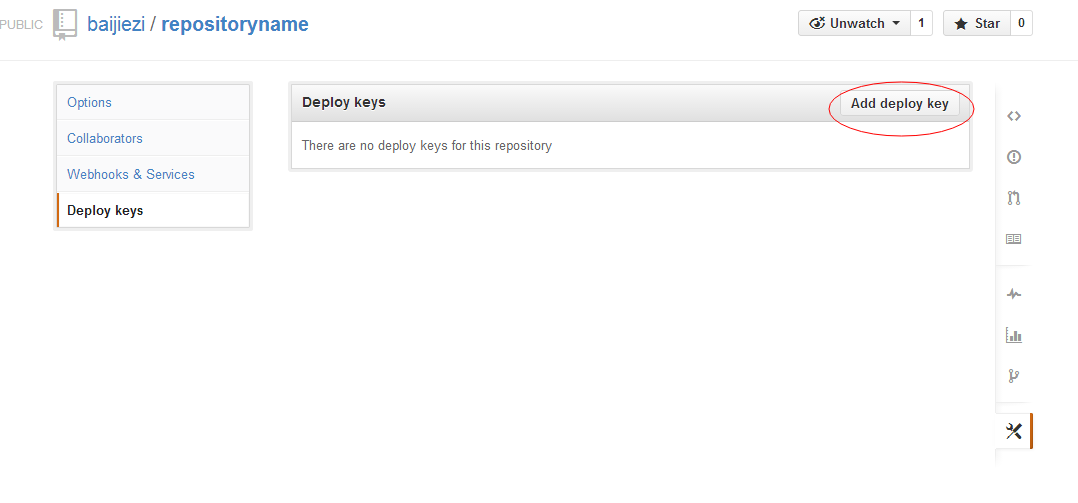




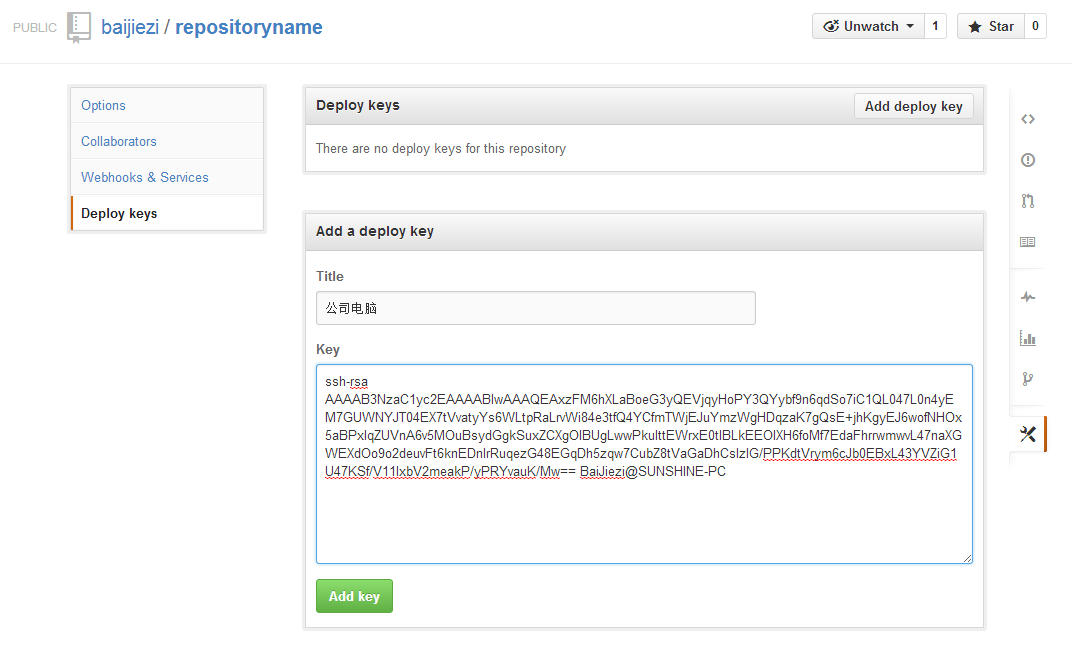
### 配置仓库的用户和key

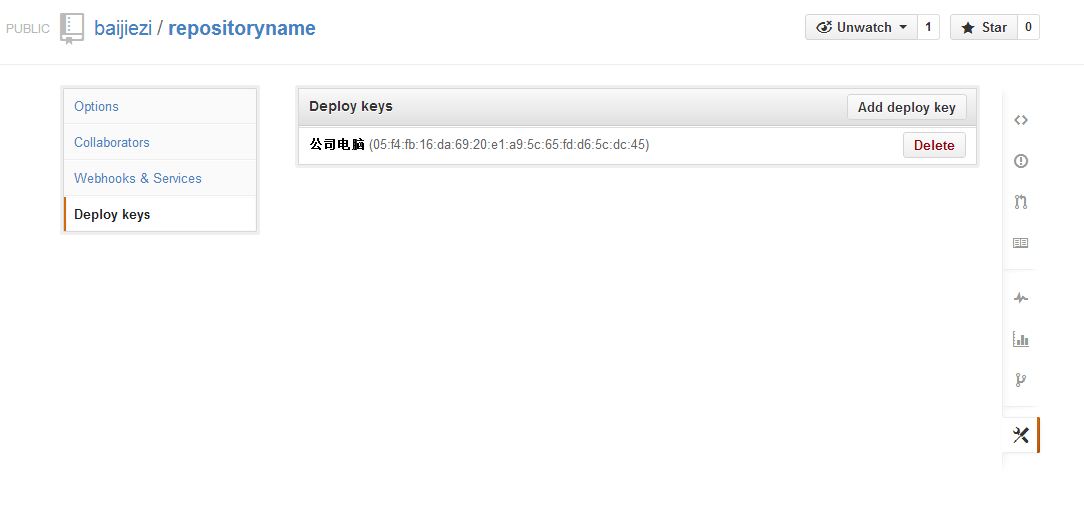




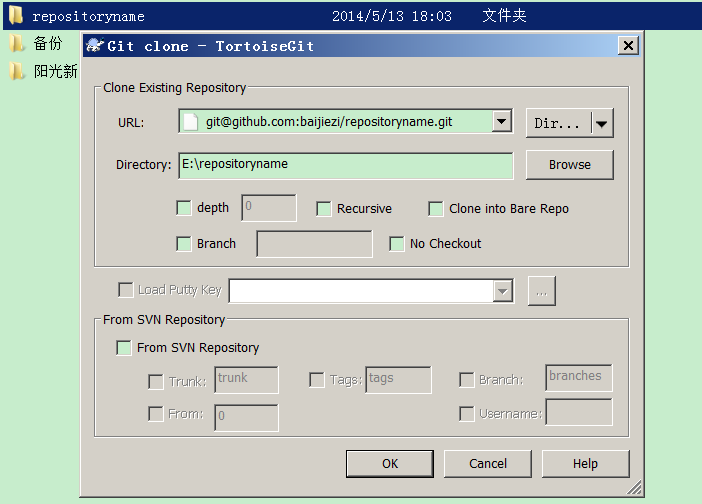


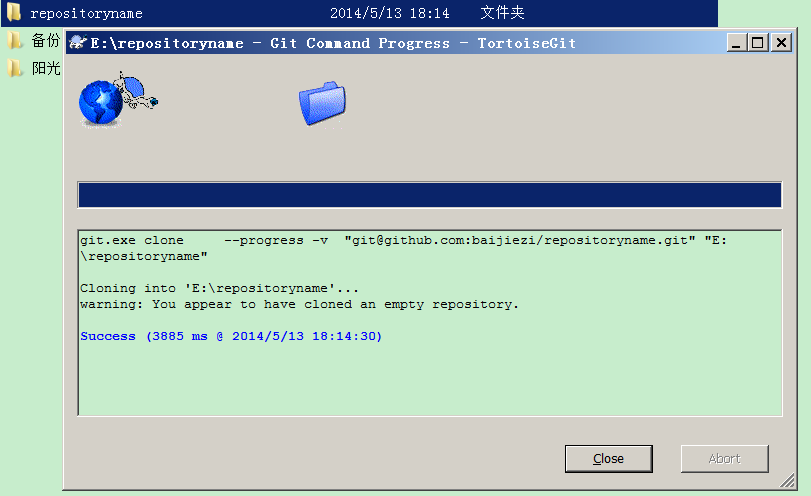
**在要配置的用户电脑上找到id\_rsa.pub文件，例如C:\Users\sunshine\.ssh\id\_rsa.pub**





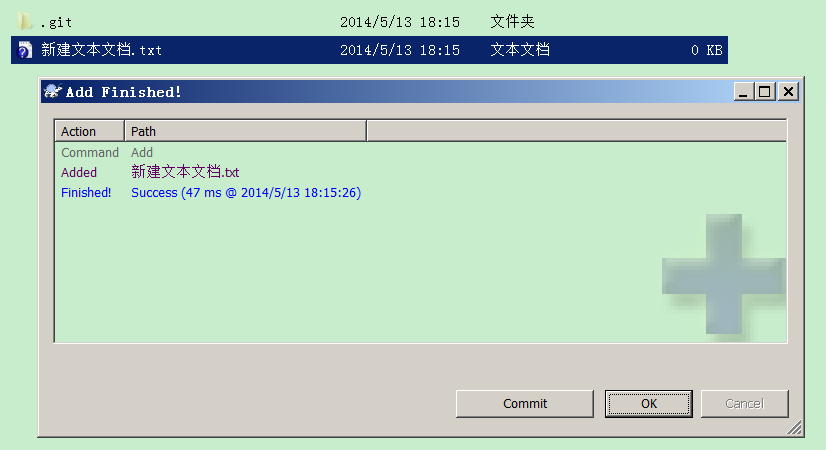
### 克隆仓库到本地

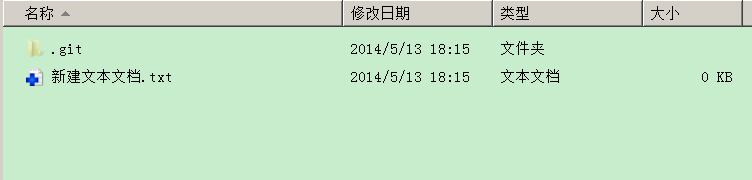


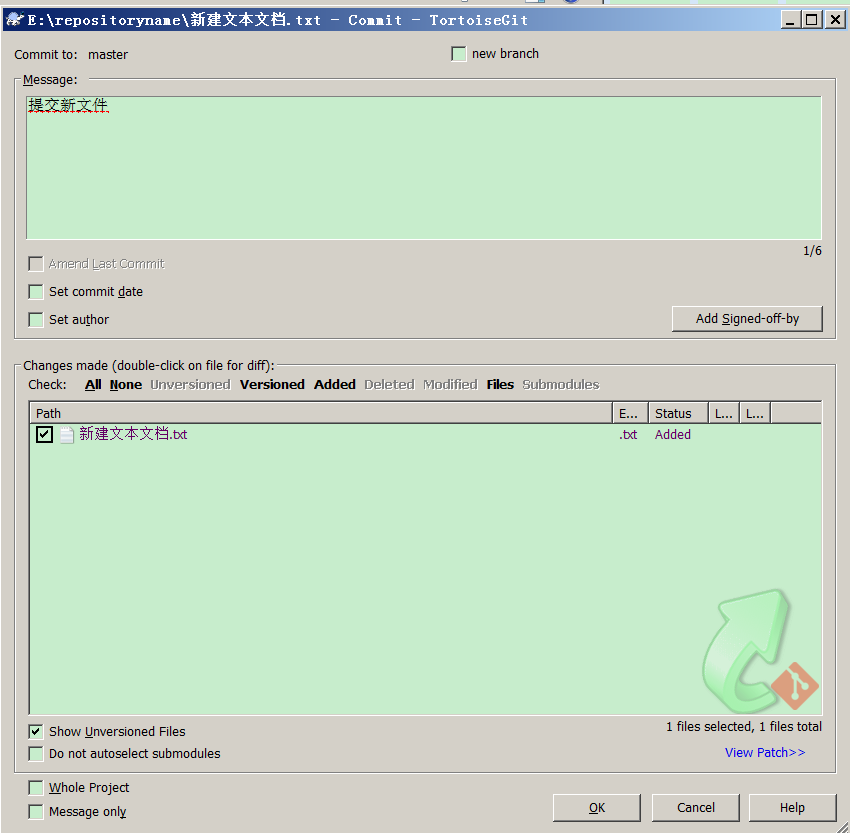


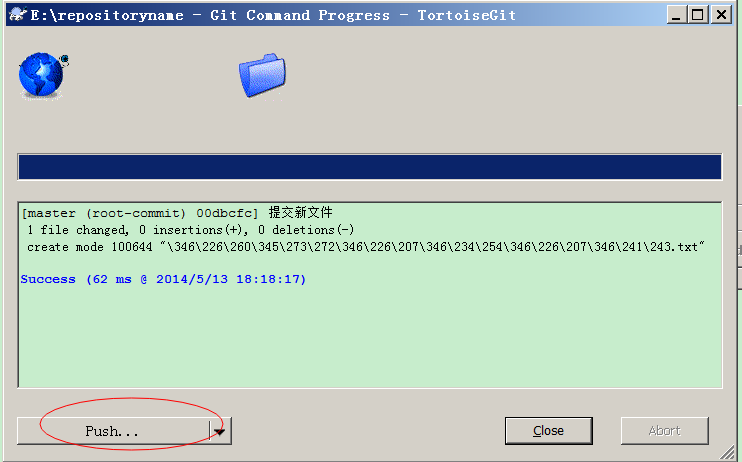
### add、commit、push操作

新建文本文档.txt

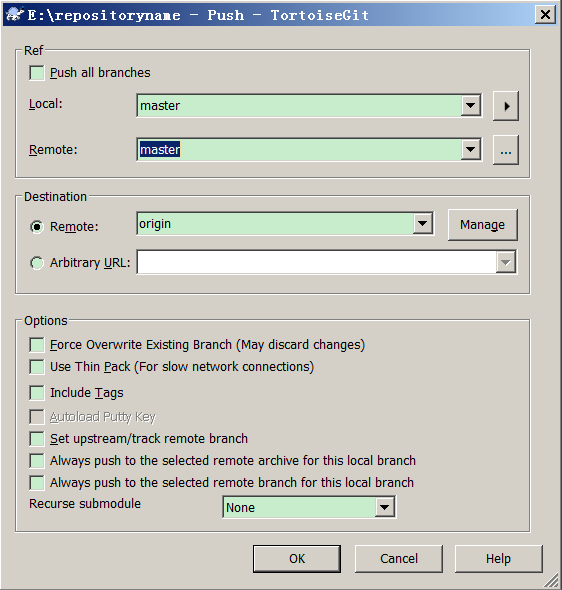


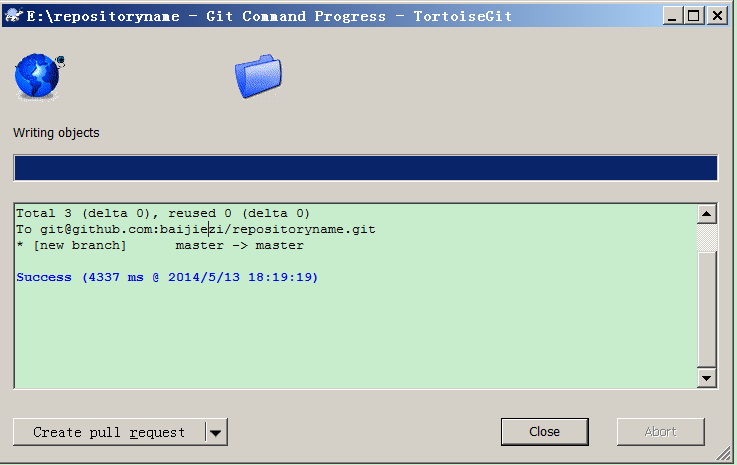






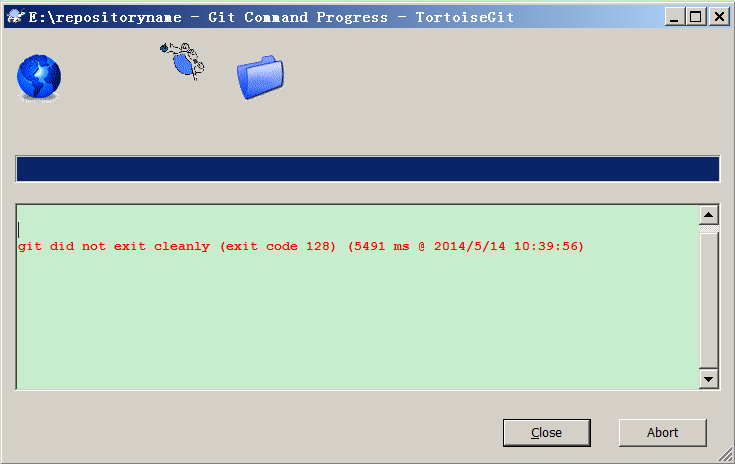
github的push操作，是针对整个分支的。如果这个分支此前commit过多次，不能选择只push其中某次commit的内容，而是会全部commit的内容一起push.





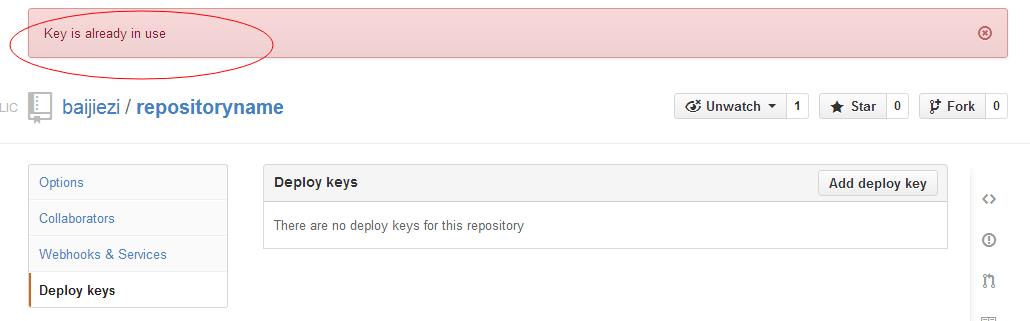
push成功。

当key还没有配置的时候push操作会出现这个异常：



### 在多个github账号上添加SSH公钥

如果一台计算机的SSH公钥已经在一个github账号（仓库）上配置了，现在又想在另一个账号（仓库）上也配置这个公钥，则github会提示这个信息：Key is already in use.



那么如何在在多个github账号上添加SSH公钥呢？下面是一篇博文（http://justjavac.com/git/2012/04/13/multiple-ssh-keys.html）。

--------------------------------------------------------博文开始--------------------------------------------------------

github使用SSH与客户端连接。如果是单用户（first），生成密钥对后，将公钥保存至github， 每次连接时SSH客户端发送本地私钥（默认~/.ssh/id\_rsa）到服务端验证。 单用户情况下，连接的服务器上保存的公钥和发送的私钥自然是配对的。

但是如果是多用户（first，second），我们在连接到second的帐号时， second保存的是自己的公钥，但是SSH客户端依然发送默认私钥，即first的私钥， 那么这个验证自然无法通过。 不过，要实现多帐号下的SSH key切换在客户端做一些配置即可。

首先cd到~/.ssh 使用 ssh-keygen -t rsa -C 'second@mail.com' 生成新的SSH key：id\_rsa\_second，生成完后将新的SSH public key添加到github。

ssh-keygen -t -rsa -C 'second@mail.com'

默认SSH只会读取id\_rsa，所以为了让SSH识别新的私钥，需要将其添加到SSH agent

ssh-add ～/.ssh/id\_rsa\_second

该命令如果报错：Could not open a connection to your authentication agent. 无法连接到ssh agent，可执行ssh-agent bash命令后再执行ssh-add命令。

完成以上步骤后在~/.ssh目录创建config文件，该文件用于配置私钥对应的服务器。内容如下：

# Default github user(first@mail.com)

Host github.com

HostName github.com

User git

IdentityFile C:/Users/username/.ssh/id\_rsa

# second user(second@mail.com)

Host github-second

HostName github.com

User git

IdentityFile C:/Users/username/.ssh/id\_rsa\_second

配置完成后，在连接非默认帐号的github仓库时，远程库的地址要对应地做一些修改， 比如现在添加second帐号下的一个仓库test，则需要这样添加：

git remote add test git@github-second:second/test.git

#并非原来的git@github.com:second/test.git

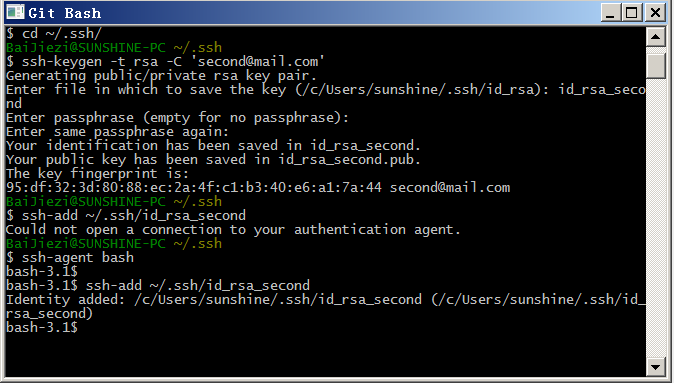
这样每次连接都会使用id\_rsa\_second与服务器进行连接。至此，大功告成！

**注意：** github根据配置文件的user.email来获取github帐号显示author信息， 所以对于多帐号用户一定要记得将user.email改为相应的email(second@mail.com)。

--------------------------------------------------------博文结束--------------------------------------------------------

以下是根据博文介绍在本机实际操作的过程记录：

用git命令生成新的公钥并添加公钥：



config文件内容：

# Default github user(first@mail.com)

Host github.com

HostName github.com

User git

IdentityFile C:/Users/sunshine/.ssh/id\_rsa

# second user(second@mail.com)

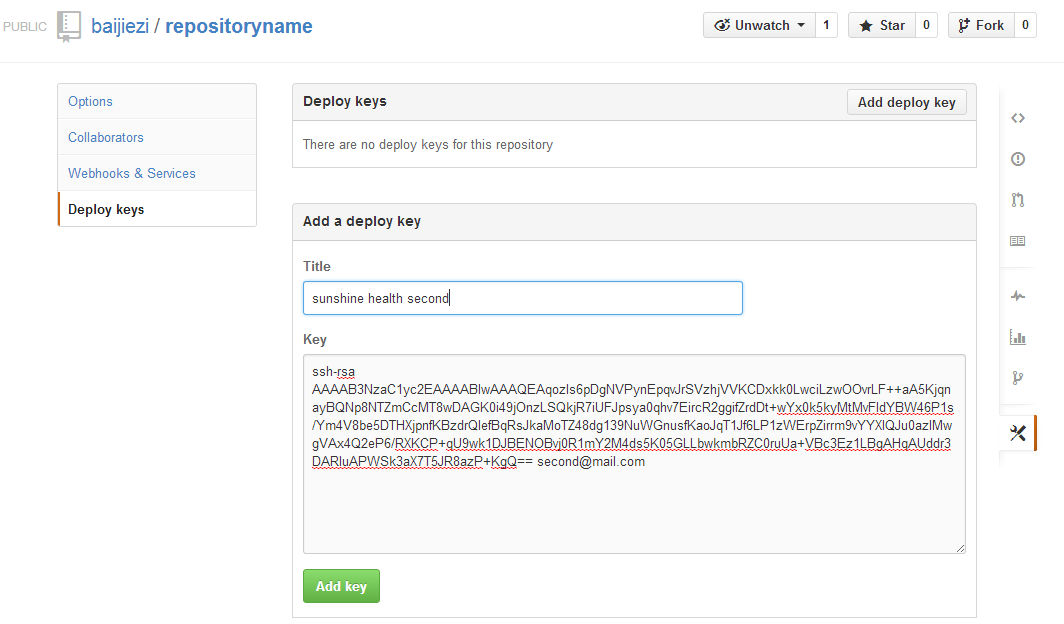
Host github-second

HostName github.com

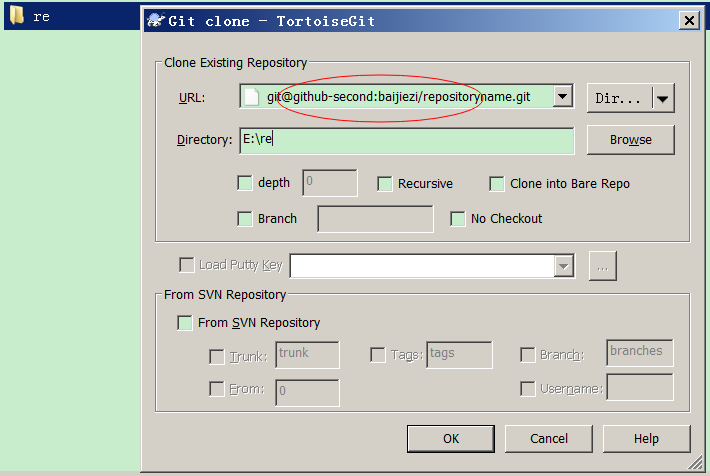
User git

IdentityFile C:/Users/sunshine/.ssh/id\_rsa\_second

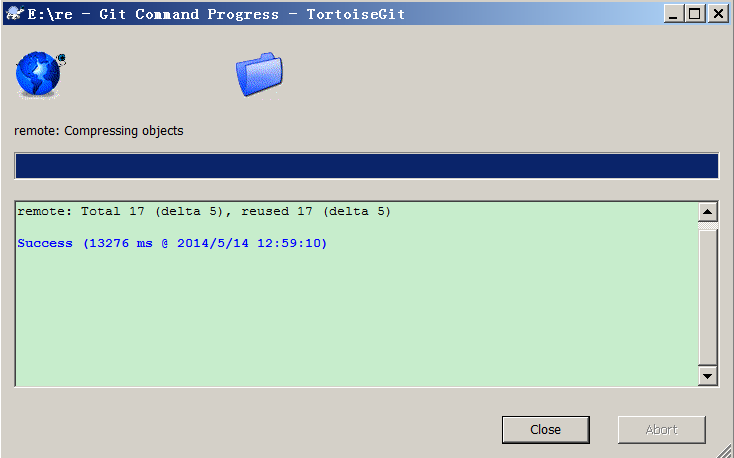
将新生成的公钥配置到github



使用替换后的URL克隆github仓库：

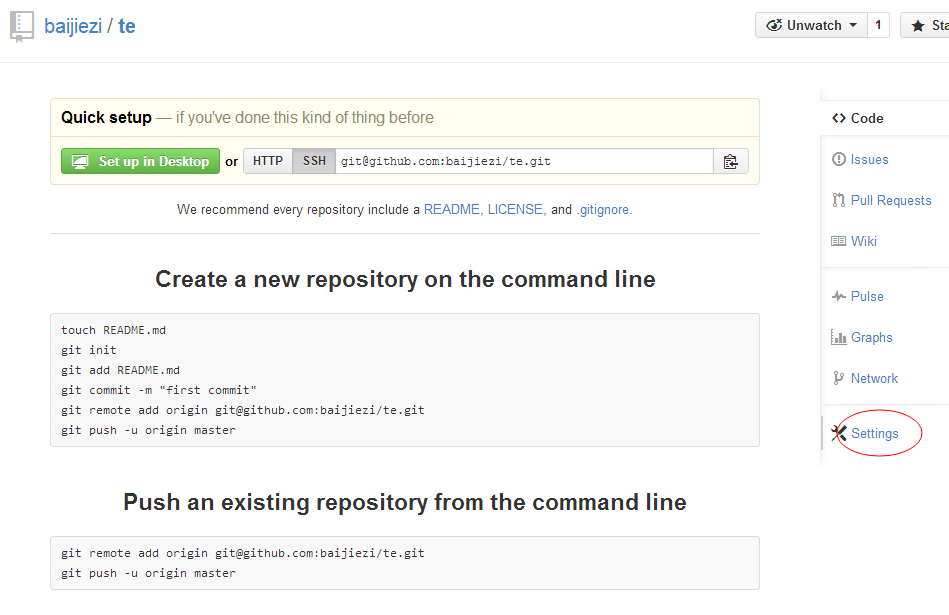


克隆成功：

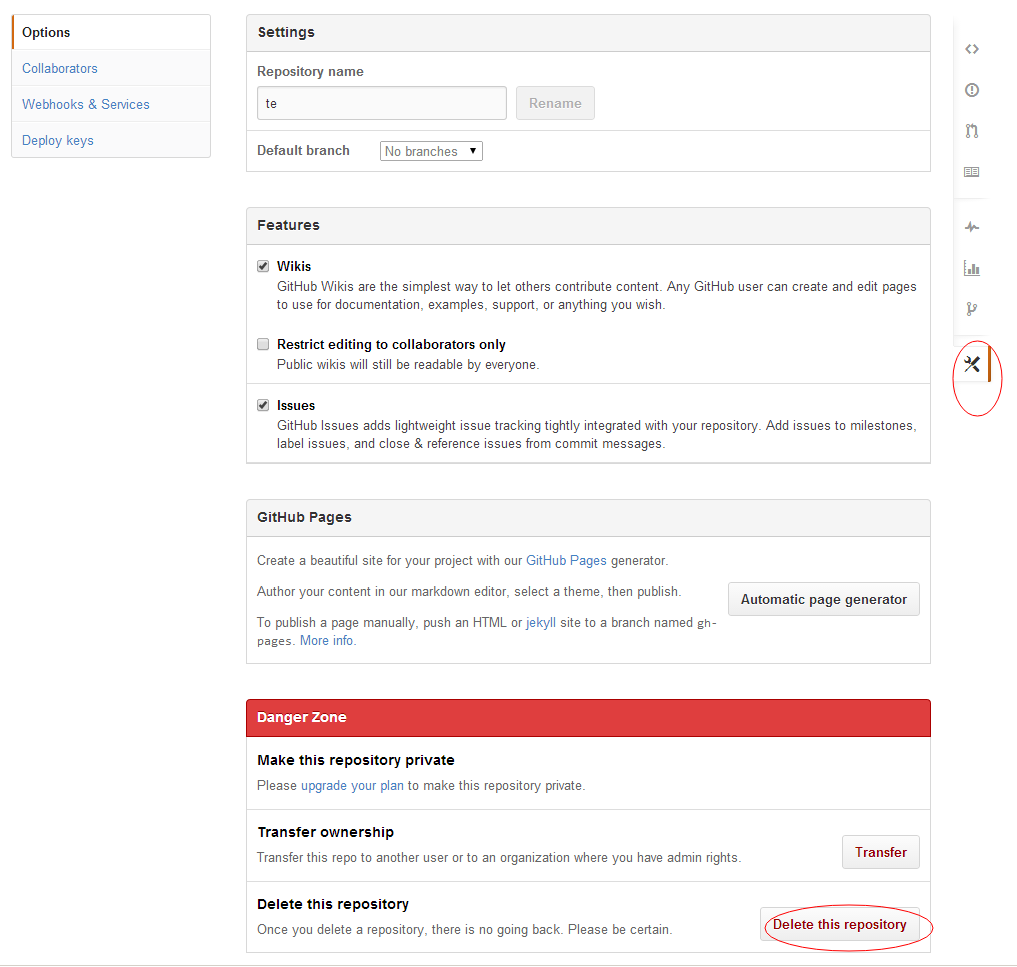


### 如何删除github仓库

1、例如要删除项目te，登陆后进入到该项目的页面，点击右边的Settings:



2、进入项目的Setting页面后，页面拉到最下面，就可以看到Delete this repository按钮。



### git如何只更新版本库某个目录的文件？

## svn

# 快捷键

使用键盘而不使用鼠标，这样有个很大的好处，可以让人的注意力更加集中在工作上，而不会把注意力分散在手和鼠标上。

## word 文档快捷键

1、在word中添加一个更改字体颜色的快捷键

点“Word选项”，点“自定义”，在最下面有个键盘快捷方式“自定义”，在“指定命令类别”下选择“开始选项卡”，在右边“命令”下选择Fontcolourpicker，在“请按新快捷键”下面按下组合键，点“指定”，再点关闭就行了，之后设定默认的字体颜色就行。

## Eclipse快捷键

Eclipse快捷键，熟悉快捷键可以帮助开发事半功倍，节省更多的时间来用于做有意义的事情。

Ctrl+Shift+L 查看所有快捷键说明

Ctrl+1 快速修复(最经典的快捷键,就不用多说了)

Ctrl+D: 删除当前行

Ctrl+Alt+↓ 复制当前行到下一行(复制增加)

Ctrl+Alt+↑ 复制当前行到上一行(复制增加)

Alt+↓ 当前行和下面一行交互位置(特别实用,可以省去先剪切,再粘贴了)

Alt+↑ 当前行和上面一行交互位置(同上)

Alt+← 前一个编辑的页面

Alt+→ 下一个编辑的页面(当然是针对上面那条来说了)

Alt+Enter 显示当前选择资源(工程,or 文件 or文件)的属性

Shift+Enter 在当前行的下一行插入空行(这时鼠标可以在当前行的任一位置,不一定是最后)

Shift+Ctrl+Enter 在当前行插入空行(原理同上条)

Ctrl+Q 定位到最后编辑的地方

Ctrl+L 定位在某行 (对于程序超过100的人就有福音了)

Ctrl+M 最大化当前的Edit或View (再按则反之)

Ctrl+/ 注释当前行,再按则取消注释

Ctrl+O 快速显示 OutLine

Ctrl+T 快速显示当前类的继承结构

Ctrl+W 关闭当前Editer

Ctrl+K 参照选中的Word快速定位到下一个

Ctrl+E 快速显示当前Editer的下拉列表(如果当前页面没有显示的用黑体表示)

Ctrl+/(小键盘) 折叠当前类中的所有代码

Ctrl+×(小键盘) 展开当前类中的所有代码

Ctrl+Space 代码助手完成一些代码的插入(但一般和输入法有冲突,可以修改输入法的热键,也可以暂用Alt+/来代替)

Ctrl+Shift+E 显示管理当前打开的所有的View的管理器(可以选择关闭,激活等操作)

Ctrl+J 正向增量查找(按下Ctrl+J后,你所输入的每个字母编辑器都提供快速匹配定位到某个单词,如果没

有,则在stutes line中显示没有找到了,查一个单词时,特别实用,这个功能Idea两年前就有了)

Ctrl+Shift+J 反向增量查找(和上条相同,只不过是从后往前查)

Ctrl+Shift+F4 关闭所有打开的Editer

Ctrl+Shift+X 把当前选中的文本全部变为大写

Ctrl+Shift+Y 把当前选中的文本全部变为小写

Ctrl+Shift+F 格式化当前代码

Ctrl+Shift+P 定位到对于的匹配符(譬如{}) (从前面定位后面时,光标要在匹配符里面,后面到前面,则反之)

下面的快捷键是重构里面常用的,本人就自己喜欢且常用的整理一下(注:一般重构的快捷键都是Alt+Shift开头的了)

Alt+Shift+R 重命名 (是我自己最爱用的一个了,尤其是变量和类的Rename,比手工方法能节省很多劳动力)

Alt+Shift+M 抽取方法 (这是重构里面最常用的方法之一了,尤其是对一大堆泥团代码有用)

Alt+Shift+C 修改函数结构(比较实用,有N个函数调用了这个方法,修改一次搞定)

Alt+Shift+L 抽取本地变量( 可以直接把一些魔法数字和字符串抽取成一个变量,尤其是多处调用的时候)

Alt+Shift+F 把Class中的local变量变为field变量 (比较实用的功能)

Alt+Shift+I 合并变量(可能这样说有点不妥Inline)

Alt+Shift+V 移动函数和变量(不怎么常用)

Alt+Shift+Z 重构的后悔药(Undo)

1编辑

编辑

作用域

功能

快捷键

全局

查找并替换

Ctrl+F

文本编辑器

查找上一个

Ctrl+Shift+K

文本编辑器

查找下一个

Ctrl+K

文本编辑器

删除当前行

Ctrl+D

文本编辑器

当前行的下一行插入空行

Shift+Enter

文本编辑器

当前行插入空行

Ctrl+Shift+Enter

文本编辑器

定位到最后编辑的位置

Ctrl+Q

全局

恢复上一个选择

Alt+Shift+↓

全局

快速修正

Ctrl+1

全局

内容辅助（代码提示）

Alt+/

全局

全部选中

Ctrl+A

全局

删除

Delete

全局

上下文信息

Alt+/

Alt+Shift+?

Ctrl+Shift+Space

Java编辑器

显示工具提示描述

F2

Java编辑器

选择封装元素

Alt+Shift+↑

Java编辑器

增量选择上一个同级元素

Alt+Shift+←

Java编辑器

增量选择下一个同级元素

Alt+Shift+→

文本编辑器

增量查找

Ctrl+J

文本编辑器

增量逆向查找

Ctrl+Shift+J

java编辑器

自动生成get set方法

Alt+Shift+s 再按 r

java编辑器

列出所有实现此接口的类

ctrl+T

2查看

编辑

作用域

功能

快捷键

全局

放大

Ctrl+=

全局

缩小

Ctrl+-

3窗口

编辑

作用域

功能

快捷键

全局

激活编辑器

F12

全局

关闭所有编辑器

Ctrl+Shift+W

全局

上一个编辑器

Ctrl+Shift+F6

全局

上一个视图

Ctrl+Shift+F7

全局

上一个透视图

Ctrl+Shift+F8

全局

下一个b编辑器

Ctrl+F6

全局

下一个视图

Ctrl+F7

全局

下一个透视图

Ctrl+F8

文本编辑器

关闭当前窗口

Ctrl+W

全局

显示视图菜单

Ctrl+F10

全局

显示系统菜单

Alt+-

4导航

编辑

作用域

功能

快捷键

Java编辑器

打开结构

Ctrl+F3

全局

打开类型

Ctrl+Shift+T

全局

打开类型层次结构

F4

全局

打开声明

F3

全局

打开外部javadoc

Shift+F2

全局

打开资源

Ctrl+Shift+R

全局

后退历史记录

Alt+←

全局

前进历史记录

Alt+→

全局

上一个

Ctrl+,

全局

下一个

Ctrl+.

Java编辑器

显示大纲

Ctrl+O

全局

在层次结构中打开类型

Ctrl+Shift+H

全局

转至匹配的括号

Ctrl+Shift+P

全局

转至上一个编辑位置

Ctrl+Q

Java编辑器

转至上一个成员

Ctrl+Shift+↑

Java编辑器

转至下一个成员

Ctrl+Shift+↓

文本编辑器

转至行

Ctrl+L

5搜索

编辑

作用域

功能

快捷键

全局

出现在文件中

Ctrl+Shift+U

全局

查找目标文件

ctrl+shift+R

全局

打开搜索对话框

Ctrl+H

全局

工作区中的声明

Ctrl+G

全局

工作区中的引用

Ctrl+Shift+G

工作区域的类

查看某一个类的继承类或者实现类

ctrl+T

6文本编辑

编辑

作用域

功能

快捷键

文本编辑器

改写切换

Insert

文本编辑器

上滚行

Ctrl+↑

文本编辑器

下滚行

Ctrl+↓

7文件

编辑

作用域

功能

快捷键

全局

保存

Ctrl+S

全局

打印

Ctrl+P

全局

关闭

Ctrl+F4

全局

全部保存

Ctrl+Shift+S

全局

全部关闭

Ctrl+Shift+F4

全局

属性

Alt+Enter

全局

新建

Ctrl+N

8项目

编辑

作用域

功能

快捷键

全局

全部构建

Ctrl+B

9源代码

编辑

作用域

功能

快捷键

Java编辑器

格式化

Ctrl+Shift+F

Java编辑器

添加/取消注释

Ctrl+/

Java编辑器

添加导入

Ctrl+Shift+M

Java编辑器

组织导入

Ctrl+Shift+O

Java编辑器

使用try/catch块来包围

未设置，太常用了，所以在这里列出，建议自己设置。

也可以使用Ctrl+1自动修正。

Java编辑器

将所选区域字母设置为小写

Ctrl+Shift+Y

Java编辑器

将所选区域字母设置为大写

Ctrl+Shift+X

Java编辑器

方法添加注释

Alt+Shift+J

10运行

编辑

作用域 功能 快捷键

全局 单步返回 F7

全局 单步执行 F6

全局 单步跳入 F5

全局 单步跳入选择 Ctrl+F5

全局 调试上次启动 F11

全局 继续 F8

全局 使用过滤器单步执行 Shift+F5

全局 添加/去除断点 Ctrl+Shift+B

全局 显示 Ctrl+D

全局 运行上次启动 Ctrl+F11

全局 运行至行 Ctrl+R

全局 执行 Ctrl+U

11重构 编辑

作用域 功能 快捷键

全局 撤销重构 Alt+Shift+Z

全局 抽取方法 Alt+Shift+M

全局 抽取局部变量 Alt+Shift+L

全局 内联 Alt+Shift+I

全局 移动 Alt+Shift+V

全局 重命名 Alt+Shift+R

全局 重做 Alt+Shift+Y

切换控制台 Alt+Shift+Q

12修改快捷键

编辑

依次打开window->preferrence->general->keys,在 binding一栏选择修改快捷键。

## Google Chrome 浏览器快捷键

以下是整理的[Google Chrome 键盘快捷键大全](http://www.ipc.me/google-chrome-hotkeys.html),Chrome为网页浏览设置了许多快捷键,许多复杂的鼠标操作可以用键盘轻松完成.  
合理地使用快捷键可以让你的操作事半功倍. (来自[吖Bee’s非常Blog](http://youz.32o.cn/archives/42.html))

可以使用页面的查找功能来打开某个链接，按Ctrl+F输入查找内容，定位到链接的文字，然后按Esc退出查找，再按Enter即可打开链接。查找的过程中使用Enter或Shift+Enter进行查找上一个或查找下一个。

同样的方法也可用在复制页面内容上。

Alt + F 打开菜单栏

**窗口和标签页快捷方式**

|  |  |
| --- | --- |
| Ctrl+N | 打开新窗口 |
| 按住 Ctrl‎ 键，然后点击链接 | 在新标签页中打开链接 |
| 按住 Shift 键，然后点击链接 | 在新窗口中打开链接 |
| Alt+F4 | 关闭当前窗口 |
| Ctrl+T | 打开新标签页 |
| Ctrl+Shift+T | 重新打开上次关闭的标签页。 谷歌浏览器可记住您关闭的最后 10 个标签页。 |
| 将链接拖动到标签页内 | 在指定标签页中打开链接 |
| 将链接拖动到两个标签页之间 | 在新标签页横条上的指定位置打开链接 |
| Ctrl+1 到 Ctrl+8 | 切换到指定位置编号的标签页。 您按下的数字代表标签页横条上的位置。 |
| Ctrl+9 | 切换到最后一个标签页 |
| Ctrl+Tab | 切换到下一个标签页 |
| Ctrl+Shift+Tab | 切换到上一个标签页 |
| Ctrl+W 或 Ctrl+F4 | 关闭当前标签页或弹出式窗口 |

### 地址栏快捷方式

在地址栏，进行下列操作之一：

|  |  |
| --- | --- |
| 键入搜索字词 | 使用默认搜索引擎进行搜索 |
| 键入网址中”www.”和 “.com”之间的部分，然后按 Ctrl+Enter | 为您在地址栏中输入的内容添加”www.”和”.com”，然后打开网址 |
| 键入搜索引擎关键字或网址，按Tab 键，然后键入搜索字词 | 使用与关键字或网址相关联的搜索引擎进行搜索 如果谷歌浏览器可以识别您要使用的搜索引擎，将会提示您按 Tab 键。 |
| F6 或 Ctrl+L | 突出显示网址区域中的内容 |
| 输入网址，然后按 Alt+Enter | 在新标签页中打开网址 |

### 打开谷歌浏览器功能的快捷方式

|  |  |
| --- | --- |
| Ctrl+B | 打开和关闭书签栏 |
| Ctrl+H | 查看历史记录页 |
| Ctrl+J | 查看下载页 |
| Shift+Escape | 查看任务管理器 |
| Shift+Alt+T | 将键盘焦点设置在谷歌浏览器工具栏上。 使用键盘上的向右和向左箭头，导航至工具栏的不同区域。 |
| Ctrl+L | 定位到地址栏 |

### 网页快捷方式

|  |  |
| --- | --- |
| Ctrl+P | 打印当前页 |
| F5 | 重新加载当前页 |
| Ctrl+F5 或 Shift+F5 | 重新加载当前页，但忽略缓存内容 |
| 按住 Alt 键，然后点击链接 | 下载链接 |
| Ctrl+F | 打开”在网页上查找”框 |
| Enter或Ctrl+G 或 F3 | 查找与您在”在网页上查找”框中输入的内容相匹配的下一个匹配项 |
| Shift+Enter  或Ctrl+Shift+G 或 Shift+F3 | 查找与您在”在网页上查找”框中输入的内容相匹配的上一个匹配项 |
| Ctrl+U | 查看源代码 |
| 将链接拖动到书签栏 | 将链接加入书签 |
| Ctrl+D | 将当前网页加入书签 |
| Ctrl++ | 放大文字 |
| Ctrl+- | 缩小文字 |
| Ctrl+0 | 还原到正常文字大小 |

### 文字快捷方式

|  |  |
| --- | --- |
| 突出显示内容，然后按 Ctrl+C | 将内容复制到剪贴板 |
| 将光标置于文字字段中，然后按 Ctrl+V | 从剪贴板粘贴当前内容 |
| 将光标置于文字字段中，然后按 Ctrl+Shift+V | 从剪贴板粘贴当前内容的纯文字部分 |
| 突出显示文字字段的内容，然后按 Ctrl+X | 删除内容 |

## 远程桌面快捷键

在窗口与全屏幕之间切换 Ctrl+Alt+Break

关闭远程桌面连接 退出全屏，按Alt+空格，再按C

## QQ快捷键

Ctrl+Tab 不同窗口切换

Tab 焦点位置切换，可用于查看复制当天聊天内容

Shift+Tab 焦点位置反向切换

Alt+H 历史聊天记录

# Windows Server 2003

## 在Windows Server 2003上安装MySql

安装MySql时提示一些错误信息，需要先为server 2003 安装一些插件，比如Windows 图像组件 (WIC)、.Net Framework 4.0、.Net Framework 2.0 、.Net Framework 3.5、NET Framework 4.0.30319.1等。

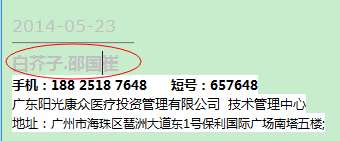
最后是安装了NET Framework 4.0.30319.1之后才成功安装了MySql，注意安装了NET Framework 4.0还不可以。

# 其他

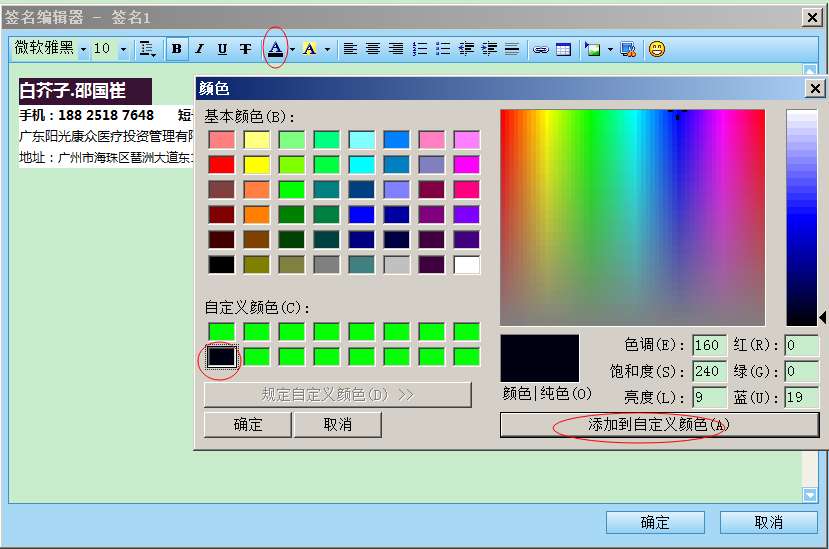
## Foxmail

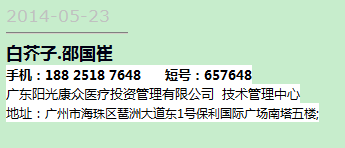
### 签名管理

每次打开写邮件窗口，签名中姓名总是浅灰色，即使在签名管理中设置为纯黑色也是如此。



解决办法：工具 - 签名管理 中设置姓名的字体颜色，一定要设置为自定义颜色，而且不能是纯黑色，可以设置成与纯黑色极相近的颜色。



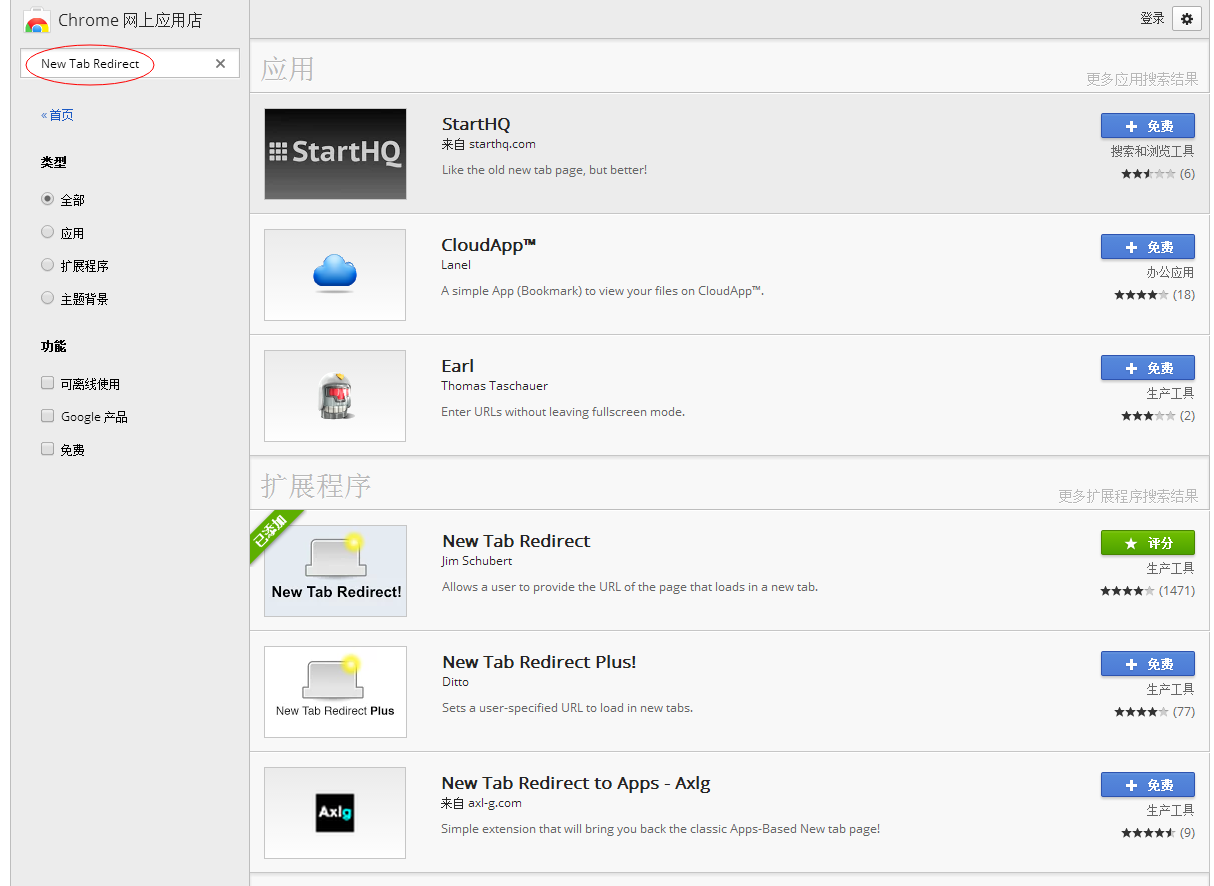


## google crome浏览器

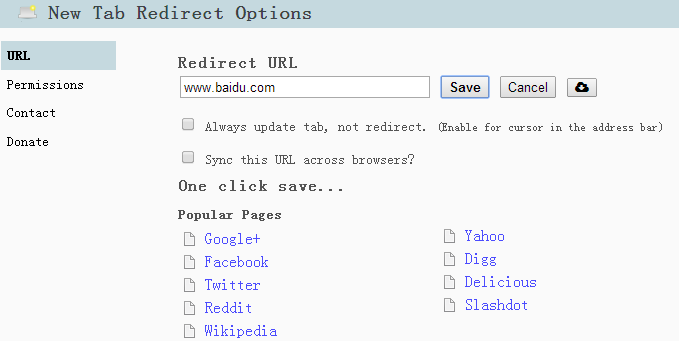
### New Tab Redirect 扩展程序

new tab redirect可用于设置每次打开新标签时默认打开页面。打开设置页面，在扩展程序中点击获取更多扩展程序进行安装。





安装完成后，在URL中输入www.baidu.com，则每次打开新标签都会默认进入百度页面。



## 百度云

2T超大免费空间，可用于软件、相片、视频等文件备份。

上传下载速度较快。

缺点：没有版本管理功能，对于有修改操作的文档、电子书不方便。

## QQ

### qq备忘录

可用于各种提醒功能，如每个周五下午四点提醒写周报。





# oracle

## oracle表名大写

oracle数据表名要用大写，否则查询的时候可能报错。

# PHP

## win7(64位)php5.5-Apache2.4-mysql5.6环境安装

*工具/原料*

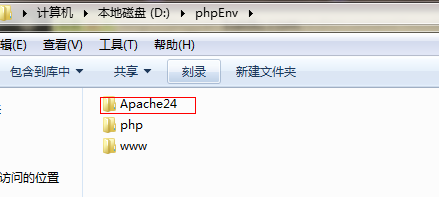
* php-5.5.10-Win32-VC11-x64.zip 下载地址: http://windows.php.net/download/
* httpd-2.4.7-win64-VC11.zip 下载地址: http://www.apachelounge.com/download/
* mysql-5.6.16-winx64.zip 下载地址: http://dev.mysql.com/downloads/mysql/

*方法/步骤*

1. 1

**一、安装配置Apache2.4.7(httpd-2.4.7-win64-VC11.zip )**

1、解压下载的安装包:httpd-2.4.7-win64-VC11.zip将其放到自己的安装目录(我的目录D:\phpEnv\Apache24)

[](http://jingyan.baidu.com/album/9faa723152c5d6473d28cb47.html?picindex=1)

1. 2

2、然后对http.conf(D:\phpEnv\Apache24\conf\http.conf)配置文件进行修改-使用记事本打开就行

**(1)**修改ServerRoot Apache的根路径:

        (37行)ServerRoot"c:/Apache24"改成=>ServerRoot "D:/phpEnv/Apache24"

**(2)**修改ServerName你的主机名称:

        (217行)ServerNamewww.example.com:80将前面的#去掉,该属性在从命令行启动Apache时需要用到。

**(3)**修改DocumentRoot Apache访问的主文件夹目录,就是php、html代码文件的位置。Apache默认的路径是在htdocs(D:\phpEnv\Apache24\htdocs)下面,里面会有个简单的入口文件index.html。这个路径可以自己进行修改,我这里将其配置在我自己新建的文件夹www(D:\phpEnv\www)下。

(247行) DocumentRoot "c:/Apache24/htdocs"

        <Directory"c:/Apache24/htdocs">

改为=>

        DocumentRoot "D:\phpEnv\www"

        <Directory "D:\phpEnv\www">

**(4)**修改入口文件配置:DirectoryIndex一般情况下我们都是以index.php、index.html、index.htm作为web项目的入口。Apache默认的入口只有index.html需要添加其他两个的支持，当然这个入口文件的设置可以根据自己的需要增减,如果要求比较严格的话可以只写一个index.php,这样在项目里面的入口就只能是index.php

(274行)<IfModuledir\_module>

        DirectoryIndexindex.html

       </IfModule>

改为=>

       <IfModuledir\_module>

        DirectoryIndex  index.php index.htm index.html

        </IfModule>

**(5)**设定serverscript的目录:

        (358行)ScriptAlias/cgi-bin/ "c:/Apache24/cgi-bin/"改为=> ScriptAlias/cgi-bin/ "D:/phpEnv/Apache24/cgi-bin"

**(6)**(380行)

        <Directory"c:/Apache24/cgi-bin">

          AllowOverride None

          Options None

          Require all granted

        </Directory>

改为=>

        <Directory"D:/phpEnv/Apache24/cgi-bin">

          AllowOverride None

          Options None

          Require all granted

        </Directory>

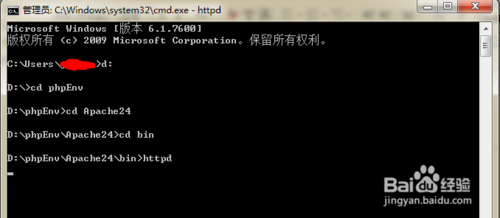
1. 3

3、接下来就可以启动Apache了

开始---运行,输入cmd,打开命令提示符。接着进入D:\phpEnv\Apache24\bin目录下回车httpd回车，如图所示.

没有报错的话就可以测试了（保持该命令窗口为打开的状态）。

    把Apache24\htdocs目录下的index.html放到D:\phpEnv\www目录下，用浏览器访问http://localhost会出现“It works”那么就说明apache已经正确安装并启动了。也可以自己写一个简单的index.html文件也可以打开。

[](http://jingyan.baidu.com/album/9faa723152c5d6473d28cb47.html?picindex=2)

1. 4

4、将Apache加入到window服务启动项里面并设置成开机启动

先关闭httpd的服务（将命令窗口关闭即可）

重新打开一个新的命令窗口进入到D:\phpEnv\Apache24\bin目录下：

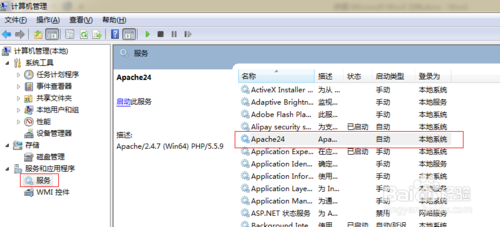
添加HTTP服务的命令是：httpd.exe -kinstall -n "servicename"  servicename是服务的名称，我添加的是：httpd.exe -k install -n "Apache24"命令成功后会有成功的提示，此时你可以在window服务启动项中看到Apache24这个服务

然后点击启动就可以了，如果不想设置成开机启动的话也可以将启动类型修改为手动。

如果要卸载这个服务的话，先要停止这个服务，然后输入httpd.exe -k uninstall -n "Apache24"卸载这个服务。

当然也可以通过D:\phpEnv\Apache24\bin下面的ApacheMonitor.exe来启动Apache这里就不多说了

如此Apache的配置就基本完成了。

[](http://jingyan.baidu.com/album/9faa723152c5d6473d28cb47.html?picindex=3)

END

*方法/步骤2*

1. 1

**二、安装配置php5.5.10（php-5.5.10-Win32-VC11-x64.zip）**

1、将下载的php-5.5.10-Win32-VC11-x64.zip  解压到安装目录下我的是（D:\phpEnv\php）

（注意PHP的安装目录不能有空格）

2、将目录下的php.ini-development文件复制一份并改名为php.ini他是php的配置文件

3、为Apache服务添加php支持

打开Apache的配置文件http.conf在最后加上

# php5 support

LoadModulephp5\_module D:/phpEnv/php/php5apache2\_4.dll

AddTypeapplication/x-httpd-php .php .html .htm

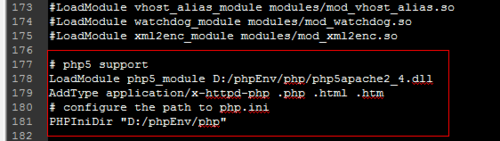
# configure thepath to php.ini

PHPIniDir "D:/phpEnv/php"

这里我添加在LoadModule下面

添加的时候要保证你的php5apache2\_4.dll文件确实存在php5.5的早期版本里面是没有这个文件的，不过高点版本里面已经有了，可以打开php安装目录找下这个文件

PHPIniDir"D:/phpEnv/php"这个就是你的php根目录

[](http://jingyan.baidu.com/album/9faa723152c5d6473d28cb47.html?picindex=4)

1. 2

4.重启Apache服务器。

5.测试。删除www中其他文件，新建一个index.php，内容为<?php phpinfo(); ?>保存，访问出现php的信息就说明php已经成功安装。

*备注：*

*Php的一些常用配置修改：（D:\phpEnv\php\php.ini）*

*时区的设置：date.timezone = Asia/Shanghai*

*错误报告等级：error\_reporting = E\_ALL这个在开发模式下可以全部打开。*

[](http://jingyan.baidu.com/album/9faa723152c5d6473d28cb47.html?picindex=5)

END

*方法/步骤3*

1. 1

**三、安装配置mysql5.6.16（mysql-5.6.16-winx64.zip）**

1、安装mysql

64位的mysql暂时没找到msi的安装包，因此直接解压到安装目录下，然后配置相关的环境变量，修改配置文件，添加window服务就行，这里就不详细写了。这里把我的配置文件贴出来给大家参考下：

    [mysqld]

    loose-default-character-set = utf8

    basedir = D:/program/mysql-5.6

    datadir = D:/program/mysql-5.6/data

    port = 3306

    sql\_mode=NO\_ENGINE\_SUBSTITUTION,STRICT\_TRANS\_TABLES

    character\_set\_server = utf8

    [client]

    loose-default-character-set = utf8

注：basedir是mysql的根目录，datadir是mysql的数据存储目录。其他的我就不做解释了

安装完mysql是没有图形用户界面的，可以安装个Navicat for MySQL之类的软件，这样看起来会比较方便点。

1. 2

2、安装完成mysql之后，为php添加mysql支持

打开php的配置文件php.ini（D:\phpEnv\php\php.ini）

（1）（721行）; extension\_dir = "ext"，去掉前面的“;”，并改为extension\_dir ="D:\phpEnv\php\ext"打开php的扩展支持，ext文件夹下有很多php的扩展支持.dll文件，感兴趣的同学可以看一下。

（2）然后就是打开php的mysql扩展了

    （875、876行）去掉前面的“;”

        extension=php\_mysql.dll

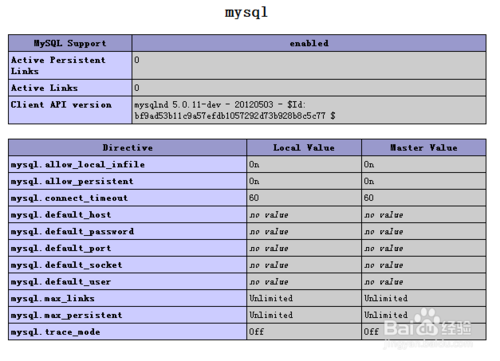
        extension=php\_mysqli.dll

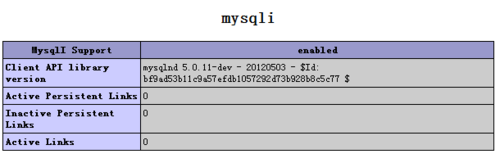
当然也可以打开881行的php\_pdo\_mysql.dll启用php的pdo支持我一般都用这个。

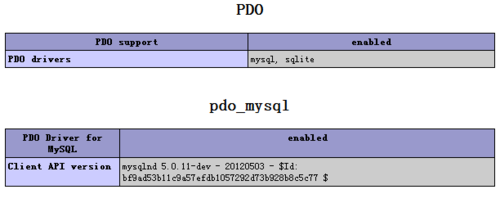
注：在第863行到第888行有很多扩展选择，你要用到什么，去掉前面的“;”就可以了。当然如果要添加其他的扩展支持如redis支持，php本身可能没有提供相应的dll文件，就需要自己去找到相应版本的dll添加到ext文件夹中，然后在配置文件中添加一个extension=…

完成之后，重启Apache

在访问phpinfo的时候就可以看到

[](http://jingyan.baidu.com/album/9faa723152c5d6473d28cb47.html?picindex=6)

[](http://jingyan.baidu.com/album/9faa723152c5d6473d28cb47.html?picindex=7)

[](http://jingyan.baidu.com/album/9faa723152c5d6473d28cb47.html?picindex=8)

END

*方法/步骤4*

1. 1

四、小结：

    Php环境的安装网上已经有很多资料了，自己在接触php的时候也是翻天覆地的查，在自己的电脑上也装过不少次了，开始的时候也用集成的软件包，WAMP、AppServ等，后来php版本升级后，发现要想升级集成包的php很麻烦，所以强烈建议自己动手单独配置下环境，更好的了解Apache、php、mysql之间的联系。

END

*注意事项*

* 安装的电脑要支持VC11
* 涉及到的配置文件行数可能会随着版本的提升或者个人的修改会有些许变动

## Windows下安装多个Apache服务,配置不同的端口和目录

有时，在一台电脑，我们需要不同端口而且不同网站目录的Apache。

或者我们可以考虑安装多个Apache ，但实际上这是不可能的，而且也没有必要，我们只需要多一个配置文件，在Windows服务中多添加一个名字不同的服务即可；

假设，我们现在已经有一个安装好了一个Apache，端口是80，目录是E:\phpweb，Apache安装目录为D:\Apache2.2，Windows中Apache的名称为Apache2.2

现在我就需要另外一个Apache，假设我要的是端口为8080，目录为E:\htdocs，Windows中Apache的名称为httpd2.2（这里不能和前一个Apache的服务同名，不然不能添加新的服务）

下面就是步骤：

1. 在D:\Apache2.2\conf\服务一份httpd.conf，改名为httpd8080.conf

2. 修改httpd8080.conf中的端口 ，即将Listen 80 改为Listen 8080 如果ServerName有用到同时也改成如这里 localhost:8080，后面的端口是8080，再改目录，DocumentRoot “E:/htdocs” 为新的地址；

3. 关键的一步：用windows命令进入D:\Apache2.2\bin\中执行httpd.exe -k install -n “Apache8080″ -f “d:\apache2.2\conf\httpd8080.conf”

4. 提示安装成功，在testing的时候可能会出错，没关系，找到Windows服务启动Apache8080服务。

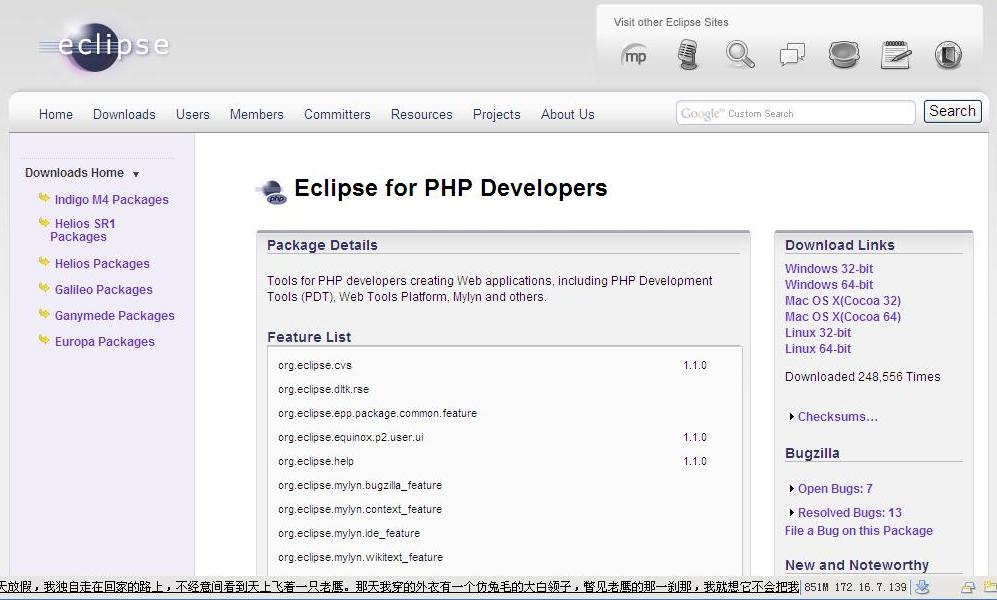
5. 在浏览器输入http://localhost:8080/这时就能看到新的目录，新的端口的Apache了。

如果要卸载这个服务的话，先要停止这个服务，然后输入httpd.exe -k uninstall -n " Apache8080"卸载这个服务。

## [如何用PHP-Eclipse调试PHP程序](http://blog.csdn.net/sleeplessljy/article/details/6153416)

笔者刚刚接触PHP-Eclipse，今天刚把debug PHP的方法弄好，发出来与大家共享一下。

    第一步：到Eclipse的官网去下载PHP-Eclipse：<http://www.eclipse.org/downloads/packages/eclipse-php-developers/heliossr1>



    第二步：下载Xdebug (我用的是php\_xdebug-2.1.0-5.2-vc6.dll，下载地址位于：<http://xdebug.org/download.php>)，并且放入到"PHP5/ext"中

    第三步：在PHP5里面找到php.ini，在这个文件的最后面加入如下配置信息：

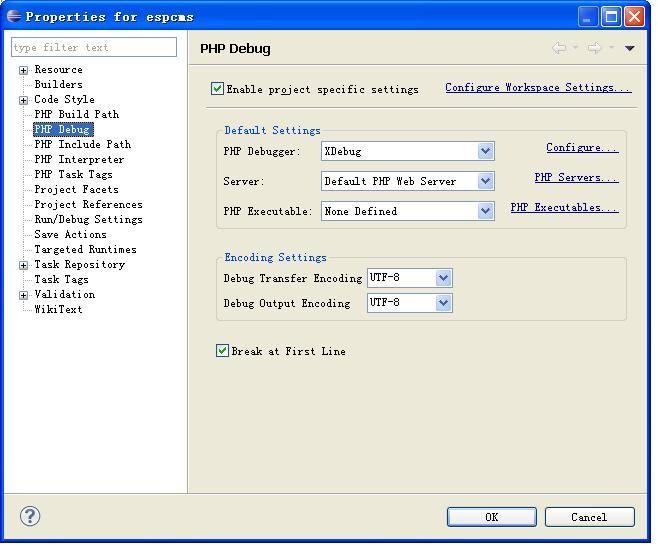
**[xhtml]** [view plaincopyprint?](http://blog.csdn.net/sleeplessljy/article/details/6153416)

1. [Xdebug]
2. zend\_extension\_ts="C:/Program Files/phpStudy/PHP5/ext/php\_xdebug.dll"
4. xdebug.auto\_trace = On
6. xdebug.show\_exception\_trace = On
8. xdebug.remote\_autostart = On
10. xdebug.remote\_enable = On
12. xdebug.remote\_host = 127.0.0.1
14. xdebug.remote\_port = 9000
16. xdebug.remote\_handler = dbgp
18. xdebug.profiler\_enable = on
20. xdebug.profiler\_output\_dir="C:/Program Files/phpStudy/PHP5/log"

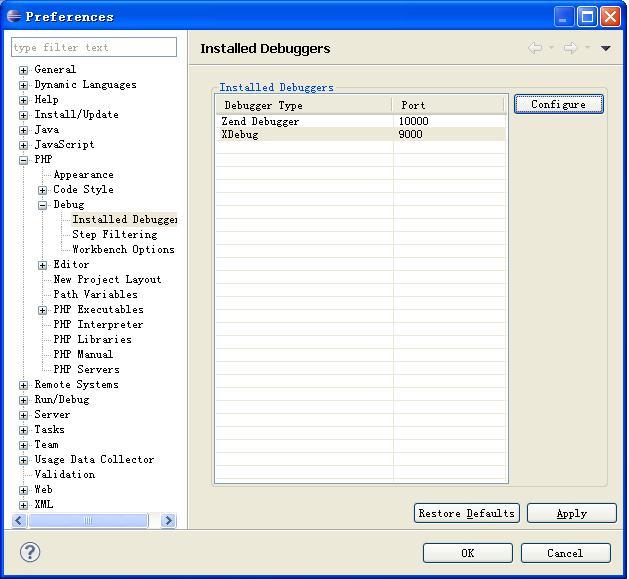
    这里要特别注意：如果在文件里有其他的zend\_extension\_ts，需要注释掉，否则apache无法启动。

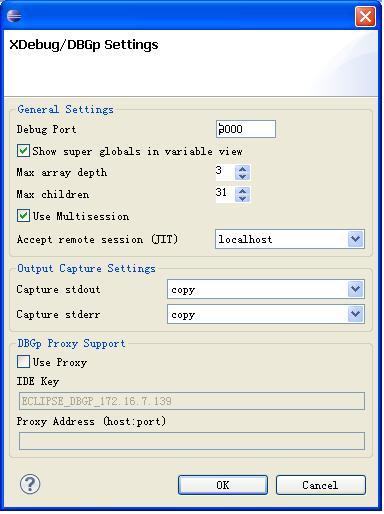
    第四步：启动apache。

    第五步：在eclipse的项目里的Properties里面找到"PHP Debug"，在PHP Debugger里面选择"XDebug"

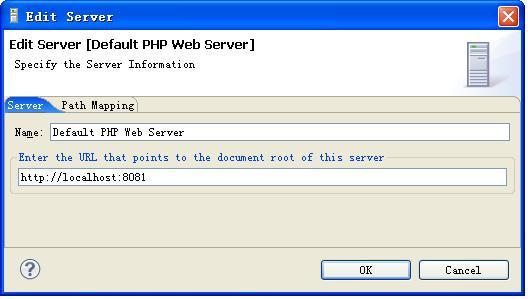


    再配置一下：





    第六步：再在Windows->Preferences里面配置下PHP Servers





     接下来，进入debug界面，就可以开始设置断点进行debug了。

## phpMyAdmin

### phpMyAdmin的安装配置

1 解压phpMyAdmin-2.7.0-beta1.tar.bz2文件到“E:\myweb\wwwroot\phpMyAdmin”目录；  
2 在“E:\myweb\wwwroot\phpMyAdmin”目录中找到config.default.php文件，将其改名为config.inc.php，然后用记事本打开；

*步骤/方法*

1. 1找到$cfg['blowfish\_secret'] = ''，将其值改为你自己想要的任意字符，如$cfg['blowfish\_secret'] = 'owndownd'；
2. 2

找到$cfg['Servers'][$i]['auth\_type'] = 'config'，改为$cfg['Servers'][$i]['auth\_type'] = 'cookie'；

1. 3

找到$cfg['DefaultLang'] = 'en-iso-8859-1'，改为$cfg['DefaultLang'] = 'zh-utf-8'；

1. 4

找到$cfg['DefaultCharset'] = 'iso-8859-1'，改为$cfg['DefaultCharset'] = 'zh-utf-8'；

1. 5

$cfg['PmaAbsoluteUri'] = 'http://您的域名/phpmyadmin/'; （phpmyadmin的URL）

1. 6

$cfg['Servers'][$i]['host'] = '数据库主机地址'； 本机localhost

1. 7

$cfg['Servers'][$i]['user'] = '站点ID'； root

1. 8

$cfg['Servers'][$i]['password'] = '密码'； zoneidc

1. 9

$cfg['Servers'][$i]['only\_db'] = '数据库名（站点ID)'； zoneidc

1. 10

修改完成后，保存config.inc.php文件；修改配置文件后，将phpmyadmin目录及其目录下的所有文件，通过ftp上传到客户网站的站点上

1. 11

通过以上设置，就可以通过http://localhost/phpmyadmin/访问到phpMyAdmin登录界面了；

1. 12

使用root账号登录，密码为空。登录后选择更改密码，输入自己需要的密码，点击“更改”按钮，即完成phpMyAdmin最后配置。现在已经可以通过phpMyAdmin来管理MySQL

### 同时管理多个mysql服务器。

**1，将phpMyAdmin根目录下的config.sample.inc.php，重命名为config.inc.php**

**2，修改config.inc.php文件**

/\*

\* First server

\*/

//如果要管理，更多个mysql服务器，就修改$connect\_hosts这个数组就行了

$connect\_hosts = array(

'1'=>array(

"host" => "localhost", //服务器1

"user" => "root",

"password" => ""

),

'2' => array(

"host" => "192.168.0.11", //服务器2

"user" => "wordpress",

"password" => "\*\*\*\*\*\*\*"

)

);

for ($i=1;$i<=count($connect\_hosts);$i++) {

/\* Authentication type \*/

$cfg['Servers'][$i]['auth\_type'] = 'cookie';

/\* Server parameters \*/

$cfg['Servers'][$i]['host'] = $connect\_hosts[$i]['host']; //修改host

$cfg['Servers'][$i]['connect\_type'] = 'tcp';

$cfg['Servers'][$i]['compress'] = false;

/\* Select mysqli if your server has it \*/

$cfg['Servers'][$i]['extension'] = 'mysql';

$cfg['Servers'][$i]['AllowNoPassword'] = true;

$cfg['Servers'][$i]['user'] = $connect\_hosts[$i]['user']; //修改用户名

$cfg['Servers'][$i]['password'] = $connect\_hosts[$i]['password']; //密码

/\* rajk - for blobstreaming \*/

$cfg['Servers'][$i]['bs\_garbage\_threshold'] = 50;

$cfg['Servers'][$i]['bs\_repository\_threshold'] = '32M';

$cfg['Servers'][$i]['bs\_temp\_blob\_timeout'] = 600;

$cfg['Servers'][$i]['bs\_temp\_log\_threshold'] = '32M';

}

注意一点，**数组下标不要从0开始，不然会提示错误的，无效的服务器索引：“0”**

登录前，

phpmyadmin 修改config.inc.php 多服务器登录

登录后，

phpmyadmin 连接多个mysql服务器，登录后

有一点要注意，用localhost登录后，选择上图下拉中的192.168.0.11后，还会让你登录，**都登录后，在多个服务器这间切换就不要在登录了**。

### [修改phpmyadmin中的默认超时时间](http://blog.csdn.net/qq7342272/article/details/6653617)

找到 phpMyAdmin / libraries / config.default.php 文件，打开，修改

$cfg['LoginCookieValidity'] = 1440;

将1440修改成更大的值即可。