

Eclipse远程调试Weblogic运行的源代码 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/267/2021_2022_Eclipse_E8_BF_9C_E7_c104_267212.htm

远程调试的好处: 我们现在开发的模式大多是本机用Eclipse集成source管理环境，集成ant的编译环境，用weblogic插件集成运行环境，这样的好处就是开发环境配置集中，普通的程序员只要按步骤做就可以了，开发步骤简单，对整个项目是一件好事。那么站在普通的程序员角度？什么也没有学到，就是在模仿而已，如果本机的硬件配置如果很低的话，意味着开发效率的降低，同时一点开发乐趣都没有，就是在实现业务而已，是很难了解工具本身的特点。远程调试的好处就是把集中的配置分散开来，让资源分配在不同的机器上，降低本机资源的使用。换一句话，就是将运行环境和开发环境的彻底的分离出来，对于那些实时对应和保守的程序员来说，简直就是一个福音，用户环境一旦出现什么bug的话，你个人可以在最短的时间内再现和调试，而不用运行本机的测试环境。

远程调试的准备知识: 使用Java平台调试架构，Java Platform Debugger Architecture，简称JPDA，具体的可以参照API文档里面的Tool Support里面的Java Platform Debugger Architecture (JPDA)(这里的JDK文档是1.42的，如果是别的版本，你搜索一下jpda就可以找到相关的资料)。换一句话，我们借助于这个平台来debug应用程序，其实我们也会看到eclipse里面也有debug的，那么它们是不是一回事呢？搜索了一下，基本上eclipse的调试就是使用和扩充这个工具平台的，只不过远程调试需要我们自己手动配置而已，而你在本地调试的时候，那些配置都是插件帮你完成的，你本

人就是一个使用者而已，所以没有关心的那么多。具体的实现步骤: 配置服务器的启动参数，如果熟悉weblogic的话，会发现weblogic更多是用java实现的。对于我们来说，如果是Windows用户的话，你只需要修改domain里面的startWebLogic.cmd批处理文件，你在这个文件里设置一个系统变量，白话就是你添加一行 set JAVA_DEBUG=-Xdebug -Xnoagent -Xrunjdwp:transport=dt_Socket,address=4000,server=y,suspend=n。这样是不够的，你还需要将这句话添加到 %JAVA_HOME%\bin\java %JAVA_VM% %MEM_ARGS% %JAVA_OPTIONS% -Dweblogic.Name=%SERVER_NAME% -Dweblogic.ProductionModeEnabled=%PRODUCTION_MODE% -Djava.security.policy="%WL_HOME%\server\lib\weblogic.policy" weblogic.Server 修改后的结果： %JAVA_HOME%\bin\java %JAVA_VM% %JAVA_DEBUG% %MEM_ARGS% %JAVA_OPTIONS%weblogic.Name=%SERVER_NAME% -Dweblogic.ProductionModeEnabled=%PRODUCTION_MODE% -Djava.security.policy="%WL_HOME%\server\lib\weblogic.policy" weblogic.Server 这样做的目的就是让你启动weblogic的同时，把JPDA也启动起来，这么写的目的就是让JPDA的参数也加载到你的启动参数列表里，启动weblogic的同时也将JPDA启动起来，具体的修改还是根据自己的实际情况来定，但是目的是相同的。 配置客户端: 这边使用的客户端工具自然就是eclipse了，原先你在本地的工程配置保持不变，换一句话说

你也必须把自己的工程import到你的eclipse，使用和操作上和本地的没有任何的变化。但是你需要建立一个remote java Application,然后配置一下就可以了。简单地说，在你的debug里有java application,java Applet,junit当然那里也有remote java Application，你新建一个就可以了，简单配置一下远程主机的地址和监听端口，就是你在服务器配置过程指定的(4000)，选择一下你的工程，然后给这个remote java Application起个名字之类。需要注意的地方:在你debug的时候，一定要保证你本地的source和服务服务器上运行的环境上的跑的class一致，换一句话说一定要及时更新source，否则debug的结果就是莫名其妙的。还有你把自己的source编译后的jar包或者文件夹放到服务器上，一定要每次都发布一遍，如果你偶尔忘记一次，那么虽然你服务器上跑的class和本地的source一致，由于你没有发布，那么debug的结果，也不一致了，所以保持一致性是两重的含义。小结 其实配置远程debug并不困难，基本的原理很简单，服务器端在启动的服务器的同时，也把JPDA明确的启动起来，分配一个端口号，那么你在客户端只要连接上JPDA服务(sun称为工具平台)便能测试了，感觉更像是一个socket编程实现的服务。至于其他的服务器的远程调试?原理一定是一样的，只不过你需要根据它们提供的文档，相应的修改远程调试的参数而已。 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