

理解类加载器：J2EE环境下的log4j PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/145/2021_2022__E7_90_86_E8_A7_A3_E7_B1_BB_E5_c104_145163.htm

大纲 Log4j是一个很流行的日志记录工具。根据java项目不同的日志需求和调试需求。它可以提供非常灵活的控制。它可以对输出信息进行分级控制。也可以在运行时改变输出信息的表现格式。而这一切都不需要改变源程序。Log4j用三个对象来控制日志的记录。Logger, appenders,和layouts。logger会以特定的layouts（样式）将日志记录到appenders中。在这个操作过程中常常需要指定具体的外部配置文件。（指定外部配置文件也是最好的一种方式）。这些配置信息会在你的应用开始时加载进来，也可能在运行的时候被改变。在使用log4j的时候将会用到以下这些步骤：1. 写配置文件，在文件中需要：#8226. 指定appender的属性 #8226. 为appender指定一个layout 2.在你的代码中，需要根据class或name得到一个logger.典型的一种做法是将logger与当前的类的进行关联。3. 用你从第二步得到的logger的任何一种方法（log.debug(),log.info（）,log.warn（）,log.error（））进行写日志。设置安装在这本书的例子是运行在weblogic 7.0 sp2 demo上。我们之所以用weblogic代替开源的产品（如jboss）是因为weblogic为在项目组中配置log4j提出了更多的挑战。它也是最流行的应用服务器。自从我发表的之前的文章。我收的绝大多数问题都是关于它的。但是这里决不是暗示推荐使用weblogic。这只是提供一个测试环境。说明在j2ee环境下如何配置log4j Weblogic的设置安装 Weblogic7.0sp2可以从BEAs trial page下载。选择你需要的版本

类型进行下载。注意你必须要在bea上面进行一些个人信息的注册。而且下载的文件很巨大。大约有150M。当然你也有别的选择。你可以选择免费的测试CD 如果你得到了安装文件。安装就很简单了。你只需要按照安装向导一步一步安装就可以了。我们的测试环境选择典型安装。当安装到最后，安装向导会问你是否需要配置Domain.选择yes然后就会运行Domain的配置向导。第一屏、选择Domain的类型和名字。选择WLSDomain作为Domain的类型，给它起个名字叫mydomain.第二屏、设置server的类型为单一server(standalone server).再下一屏选择缺省路径做为Domain的路径，再点击next为server进行配置。然后一路next就可以了。最后需要你为domain输入管理员帐号和密码。我一般就是以admin作为管理员帐号，password最为密码（很原始的做法）。接下来选择是否将service注册成windows的services(在windows环境下安装)，一般选择no。最后选择yes在windows开始菜单上方置快捷方式。最后点击create建立Domain.运行weblogic就很简单了。从开始菜单Start->All Programs->BEA WebLogic Platform 7.0->User Projects->mydomain->Start Server这样可以启动server.不过更常用的方式是从dos命令行，进入BEA的HOME目录。进入user_projects\mydomain后，运行startWebLogic.cmd.就可以了。启动的时候会要求你输入管理员帐号和密码。最后你可以看到server开始启动。当你看到Server start in running mode表示启动成功。如果要关闭你只要关闭Dos窗口就可以了。如果你安装运行成功。你可以在浏览器中输入如下地址http://localhost:7001/console.你会进入weblogic的管理界面。你可以对applications和server进行各种

配置。当然你要输入管理员帐号和密码。就和启动时输入的一样。进入后你可以看到各种管理任务的界面。你也可以执行相应的管理任务。log4j的安装和前面提到的一样。Log4j可以从log4j web site. 下载。请按照前面的文章介绍进行下载和安装log4j的二进制代码。在这章最后我们会介绍log4j的配置。

类加载器的概念有什么不一样。虽然有关类加载器的讨论在本文以外已经有很多了。但是我还是尽力解释一下类加载器是如何影响应用服务器中log4j的配置的。类加载器。顾名思义，表示在java虚拟机中加载classes. 在我们的class执行和被访问之前。它必须通过类加载器加载使之有效。给定一个class名字，类加载器会定位class并且将它加载到java虚拟机。但是类加载器本事就是class。这就带来一个问题。是谁来加载这些类加载器呢。当你运行一个java程序。（例如在命令行内输入java命令），它就会执行并且启动一个本地的native java launcher(我叫他java 启动者，估计不太准确注明一下)。这里本地的意思是指针对你的平台和运行环境而言。这个本地的java启动者包含一个类加载器,这个类加载器的名字叫做bootstrap 类加载器.(引导类加载器).这个引导类加载器.是本地的和你的环境有关。而且它不是用java写成的。这个引导类加载器的主要功能是加载java的核心类。 Figure 1. 类加载器委托层次 Java虚拟机缺省会执行其他两个类加载器.引导类加载器会加载extension(扩展)类加载器和application(应用)类加载器到内存中。这些都是用java写成的。和前面提到的一样。引导类加载器会加载java的核心类（例如：java.util包下的类）。扩展类加载器会加载扩展的java核心类（例如：javax.包下的类，或者是在运行环境中类路径ext目录下的java类。）应用类

加载器会加载你应用下的各种class文件。所有三种类加载器都遵循委托模式。当低一级的类加载器需要定位一个类。它会将这个任务委托给父一级类加载器.当你的应用需要一个特殊的类。应用类加载器会将这个任务委托给扩展类加载器.而扩展类加载器又会将这个任务委托个引导类加载器。如果你请求的类是一个核心类。那么引导类加载器会将这个类加载，使你可以使用这个类。但是如果找不到这个类。这个请求就会返回到扩展类加载器.最后在返回到应用类加载器.这样的做法就使每个类加载器会先让它的父一级的类加载器搜索需要的类。只有当父一级的类加载器没有找到。才会让子一级的类加载器为它自己搜索所需的类。 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