

使用WeblogicIntegration的应用程序架构 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/233/2021_2022__E4_BD_BF_E7_94_A8WebI_c67_233932.htm 在一个已经使用Weblogic

Integration和Workshop开发出多个应用程序的环境中，您可能希望考虑一种支持以下功能的方法：组件重用；将多个应用程序共同部署到同一个WebLogic域上 本文关注单

个Workshop/Weblogic Integration应用程序即一个部署单元（一个ear）的应用程序架构。要记住，多个应用程序可以部署到单个Weblogic Integration域上。所提出的应用程序架构支持：代码的重用、维护和扩展 团队工作 根据子元素的可变性，在一个应用程序内对项目进行重新组织，使其从一个应用程序变为多个应用程序 稍后您将看到，在一个应用程序内组织Workshop项目将对性能调优造成直接影响。 我将分3部分介绍：应用程序各层以及将代码织入各层中的方法 在一个简单示例中应用这些方法 对性能调优和部署的影响 第1部分：应用程序各层以及将代码织入各层中的方法 在定义应用程序构件块时，一种好的方法是使用逻辑层，这些逻辑层随后可能被扩展为SOA层。对于复杂的应用程序，我们通常定义3个层，如下图所示。层与层之间可能包含一个由消息总线（如AquaLogic Service Bus）组成的中介体。在本文中，我们将不讨论中介体，因为我们主要关注的是单个应用程序、单个ear中的架构，要记住，随后该应用程序在开发过程中将被拆分为多个应用程序，每个应用程序都被公开为web服务，成为一个SOA构件块。之后可以使用一个中间层进行服务与服务之间的整合，而系统与系统之间的整合基本上使

用Weblogic Integration来完成。这些层是SOA相关的，它们在单个ear文件中也有意义。类似地，当在企业范围内应用SOA时，在单个应用程序中考虑无疑是提供构件块的最简单的方式。

复合层(Composite layer)：向企业合作伙伴和外部客户提供对服务的访问，提供必要的转换、过滤等。该层可以聚合来自编排层的业务服务，以提供对添加约束、过滤和安全性等服务的外部访问。该层将用于编排层中的标准XML模型转换为适用于外部业务合作伙伴的简化模型。该层向外部客户端公开Web服务。该层可能包含作为信息组合的页面流。页面流可作为面向业务合作伙伴的Web Interface提供，或者使用Weblogic Portal和WSRP将其公开为Web服务。

编排层(Orchestration layer)：该层包含了Weblogic Integration业务流程。业务流程使用标准的XML数据格式。该层用于编排许多后端系统。在该层中使用标准的模型提供了该层与上下层之间的无关性。在对一个后端系统进行更改以针对不同的业务合作伙伴提供适当的服务时，这就提供了很大的灵活性和无关性。

连通性层(Connectivity layer)：该层提供到后端系统的访问。它提供从数据的后端表示法到用于编排层中的标准数据格式之间的相互转换。在编排层中使用标准数据模型是一种好方法。它提供了应用程序代码与该应用程序所连接的其他系统之间的无关性。标准的模型可能由XML模式中所表示的UML域模型组成。流程之间这种XML数据的传输应该通过粗粒度消息（定义在UML域模型中、由一组对象组成的消息）来完成。XML中所表示的与DTO (data transfer object)模式相关的相同理念应该是一个好的起点。使用多个层并不意味着每次都必须通过各个层。例如，从表示层进入连通性层，而

不经过编排层的任何组件。应该根据对公开于各个层的服务的重用需求对此进行考虑。在每一层中，代码和Workshop项目的逻辑编排应该仔细考虑，以便提供更多的灵活性。那么如何在一个Workshop应用程序中定义项目呢？一个Workshop应用程序由多个不同类型的项目组成。项目的类型可以是主要包含Weblogic Integration流程的“流程”项目，主要包含java页面流的“Web”项目，用于需要在不同的项目之间共享的控件的“控件”项目等。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