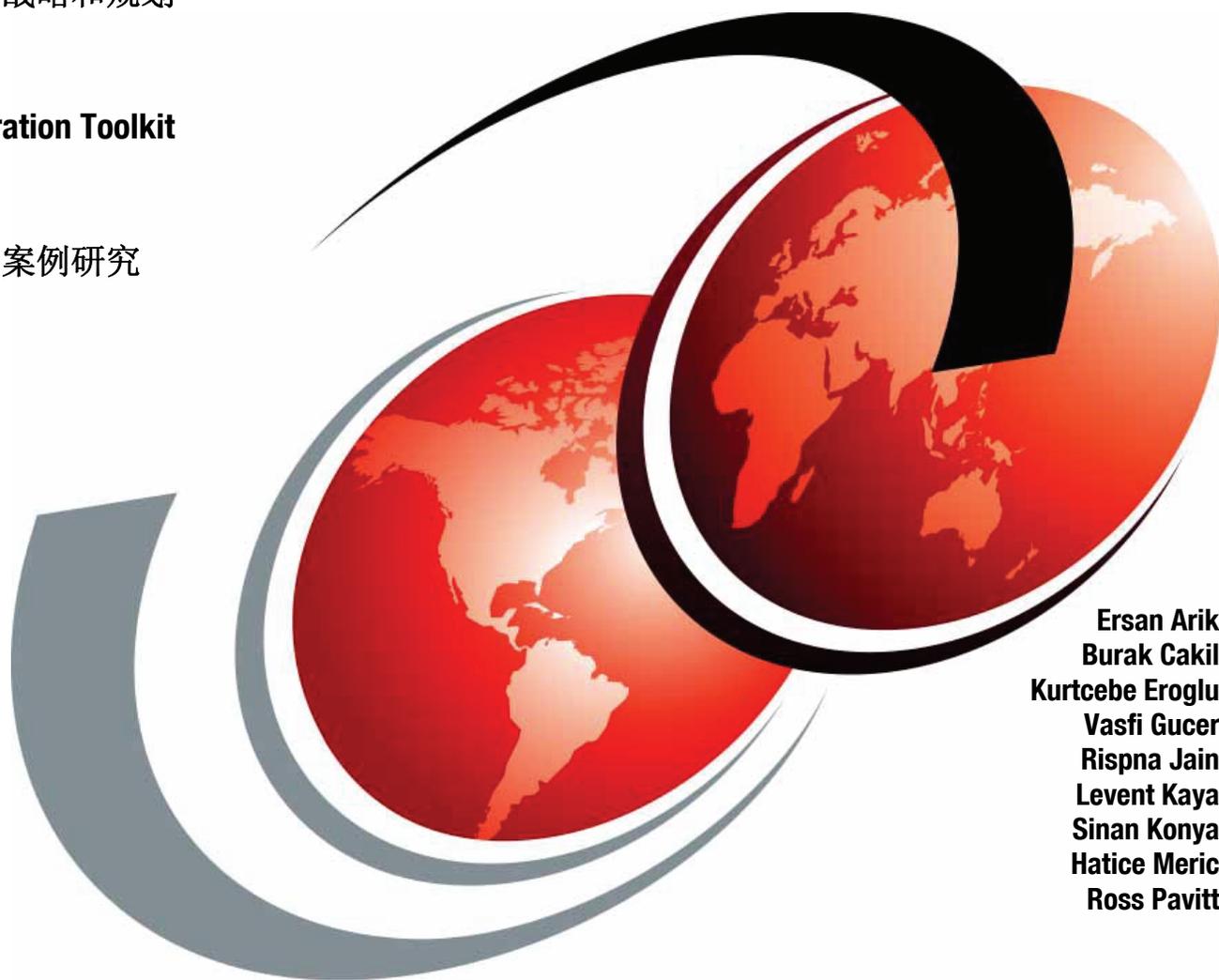


WebSphere Application Server V8.5 V2V 迁移指南

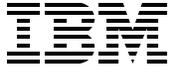
了解迁移战略和规划

体验 Migration Toolkit

复审迁移案例研究



Ersan Arik
Burak Cakil
Kurtcebe Eroglu
Vasfi Gucer
Rispna Jain
Levent Kaya
Sinan Konya
Hatice Meric
Ross Pavitt



国际技术支持组织

WebSphere Application Server V8.5 V2V 迁移指南

2013 年 2 月

注意： 在使用本资料及其支持的产品之前，请阅读“声明”在第 ix 页上中的信息。

第一版 (2013 年 2 月)

此版本适用于以下 IBM 产品：

IBM WebSphere Application Server V8.5

IBM WebSphere Application Server Migration Toolkit V3.5

IBM Rational Application Developer V8.5

此文档创建或更新于 2013 年 5 月 20 日。

© Copyright International Business Machines Corporation 2013. All rights reserved.

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

目录

声明	ix
商标	x
本文编写组	xii
现在，您也能够成为一名出版作家！	xiii
欢迎评论	xiii
与 IBM Redbooks 保持连接	xiii
第 1 章. 迁移战略及规划	1
1.1 迁移过程概述	2
1.1.1 迁移类型	2
1.1.2 重要方面及注意事项	3
1.2 迁移评估	4
1.2.1 迁移调查问卷	4
1.2.2 迁移调查问卷评审会	4
1.2.3 迁移评估报告	5
1.3 迁移规划	5
1.3.1 角色和职责	5
1.3.2 迁移项目计划	5
1.4 迁移实施	6
1.4.1 实施注意事项	6
1.4.2 实施阶段	6
1.4.3 应用程序迁移	8
1.4.4 IDE 迁移	9
1.4.5 运行时迁移	13
1.5 后部署	14
1.5.1 性能调整及优化	14
1.5.2 获取帮助	14
1.6 总结	16
第 2 章. 常见迁移问题	17
2.1 J2EE 至 Java EE 迁移注意事项	18
2.1.1 Java 消息服务 (JMS)	21
2.1.2 JavaServer Page	21
2.1.3 Servlet	21
2.1.4 JavaServer Faces	22
2.1.5 Web 服务	22
2.1.6 Java 持久性 API (JPA)	24
2.2 运行时迁移问题	24
2.2.1 同时迁移其他产品	25
2.2.2 资源定义	25
2.2.3 开发环境问题	27
2.3 互操作性和集成	28
第 3 章. Application Migration Tool 的安装和配置	29
3.1 IBM WebSphere Application Server Migration Toolkit 概述	30
3.1.1 Migration Toolkit 基础知识	32
3.2 在 Eclipse 上安装 Application Migration Tool	37
3.3 在 Rational Application Developer V8.5 上安装 Application Migration Tool	40
3.4 命令行安装	41

3.5 配置 Application Migration Tool.....	42
3.5.1 运行分析.....	42
3.5.2 配置和运行分析.....	44
3.5.3 生成报告.....	48
3.6 故障诊断.....	50
第 4 章 . Application Migration Tool - WebSphere Version to Version 的安装和配置.	51
4.1 Application Migration Tool - WebSphere Version to Version 概述.....	52
4.1.1 Application Migration Tool V3.5.....	52
4.1.2 WebSphere Application Server V8.5 Liberty 概要文件.....	53
4.2 在 Eclipse 上安装 Application Migration Tool.....	53
4.2.1 使用 Eclipse Marketplace 安装 Application Migration Tool.....	53
4.2.2 使用 CLI 安装 Application Migration Tool.....	55
4.2.3 使用 Rational Application Developer V8.5 安装 Application Migration Tool.....	56
第 5 章 . 从较早版本的 WebSphere Application Server 迁移.	59
5.1 概念.....	60
5.1.1 整体迁移过程.....	60
5.1.2 轻松迁移.....	63
5.1.3 已知问题.....	63
5.2 Plants by WebSphere 样例应用程序迁移.....	73
5.2.1 应用程序概述.....	74
5.2.2 修改和重建 Plants by WebSphere 样例.....	77
5.2.3 删除及重新安装 PlantsByWebSphere.ear 文件.....	78
5.2.4 从管理控制台安装 Plants By WebSphere.....	79
5.2.5 运行样例应用程序.....	81
5.3 Plants By WebSphere 的迁移.....	82
5.3.1 导出 PlantsByWebSphere.ear 以及将其导入 Eclipse 以进行分析.....	82
5.3.2 配置和运行 Application Migration Tool - WebSphere Version to Version.....	85
5.3.3 修正由 Application Migration Tool 报告的错误.....	91
5.3.4 Application Migration Tool 未检测到的其他迁移问题.....	93
5.3.5 导出经修正的 Plants By WebSphere 应用程序并在 WebSphere Application Server V8.5 上运行该应用程序.....	94
5.4 概率分布样例.....	95
5.4.1 将应用程序导入 Rational Application Developer V8.5 中.....	95
5.4.2 在 WebSphere Application Server V6.1 上安装应用程序.....	95
5.4.3 在 WebSphere Application Server V6.1 上测试应用程序.....	96
5.4.4 运行 Application Migration Tool - WebSphere Version to Version.....	96
5.4.5 解决发现的迁移问题.....	97
5.4.6 在 Rational Application Developer 中准备应用程序在 WebSphere Application Server V8.5 上安装.....	99
5.4.7 在 WebSphere Application Server V8.5 上安装应用程序.....	99
5.4.8 在 WebSphere Application Server V8.5 上测试应用程序.....	99
5.4.9 迁移后升级应用程序的其他选项.....	100
5.5 Web 服务 Axis 2 股票报价样例.....	100
5.5.1 概述.....	101
5.5.2 在 WebSphere Application Server V6.1 上安装.....	101
5.5.3 在 WebSphere Application Server V6.1 上测试.....	102
5.5.4 将应用程序源代码导入到 Eclipse.....	103
5.5.5 运行 Application Migration Tool - WebSphere Version to Version.....	103
5.5.6 准备在 WebSphere Application Server V8.5 上进行安装.....	105
5.5.7 在 WebSphere Application Server V8.5 上安装.....	106
5.5.8 配置 WebSphere Application Server V8.5 以支持应用程序.....	106

5.5.9 在 WebSphere Application Server V8.5 上测试应用程序	108
5.5.10 Web 服务 Axis 2 股票报价应用程序迁移的摘要	108
附录 A. 迁移调查问卷	109
业务要求	110
一般信息	110
应用程序体系结构	110
依赖关系	111
持久性	111
国家语言	111
代码	112
Java	112
EJB 使用情况	112
Servlet 和 JSP	112
Web 服务	113
数据库访问	113
JMS	114
JNDI 命名	114
应用程序跟踪和日志	114
Struts	114
事务	115
线程	115
套接字	115
XML	115
开发迁移调查问卷	116
工作站配置	116
集成开发环境	116
开发测试配置	116
软件开发技能	116
开发方法论	117
构建和包装	117
Ant	117
运行时迁移调查问卷	118
概述	118
当前硬件	118
软件	119
HTTP 服务器	119
网络边缘服务器 (Edge Component)	119
可用性	119
推出问题	120
管理	120
安全性	120
测试迁移调查问卷	121
硬件	121
实践和工具	121
附录 B. 附加材料	123
查找 Web 材料	123
使用 Web 材料	123
下载 Web 材料的系统要求	123
下载并解压缩 Web 材料	124

相关出版物	125
IBM Redbooks	125
其他出版物	125
在线资源	125
IBM 提供的帮助	126

声明

本信息是为在美国提供的产品和服务编写的。

IBM 可能在其他国家或地区不提供本文档中讨论的产品、服务或功能特性。有关您当前所在区域的产品和服务的信息，请向您当地的 IBM 代表咨询。任何对 IBM 产品、程序或服务的引用并非意在明示或暗示只能使用 IBM 的产品、程序或服务。只要不侵犯 IBM 的知识产权，任何同等功能的产品、程序或服务，都可以代替 IBM 产品、程序或服务。但是，评估和验证任何非 IBM 产品、程序或服务，则由用户自行负责。

IBM 公司可能已拥有或正在申请与本文档内容有关的各项专利。提供本文档并未授予用户使用这些专利的任何许可。您可以用书面方式将许可查询寄往：

IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

本条款不适用英国或任何这样的条款与当地法律不一致的国家或地区： International Business Machines Corporation “按现状”提供本出版物，不附有任何种类的（无论是明示的还是暗含的）保证，包括但不限于暗含的有关非侵权、适销和适用于某种特定用途的保证。某些国家或地区在某些交易中不允许免除明示或暗含的保证。因此本条款可能不适用于您。

本信息中可能包含技术方面不够准确的地方或印刷错误。此处的信息将定期更改；这些更改将编入本出版物的新版本中。IBM 可以随时对本信息中描述的产品和 / 或程序进行改进和 / 或更改，而不另行通知。

本信息中对非 IBM Web 站点的任何引用都只是为了方便起见才提供的，不以任何方式充当对那些 Web 站点的保证。那些 Web 站点中的资料不是 IBM 产品资料的一部分，使用那些 Web 站点带来的风险将由您自行承担。

IBM 可以按它认为适当的任何方式使用或分发您所提供的任何信息而无须对您承担任何责任。

涉及非 IBM 产品的信息可从这些产品的供应商、其出版说明或其他可公开获得的资料中获取。IBM 没有对这些产品进行测试，也无法确认其性能的精确性、兼容性或任何其他关于非 IBM 产品的声明。有关非 IBM 产品性能的问题应当向这些产品的供应商提出。

本信息包含在日常业务操作中使用的数据和报告的示例。为了尽可能完整地说明这些示例，示例中可能会包括个人、公司、品牌和产品的名称。所有这些名字都是虚构的，若现实生活中实际业务企业使用的名字和地址与此相似，纯属巧合。

版权许可证：

本信息包含以源语言编写的样本应用程序，它们演示各种操作平台上的编程技巧。为了开发、使用、推广或分发符合编写样本程序的操作系统的应用程序编程接口的应用程序，您可以任何形式复制、修改和分发这些样本程序，而无需向 IBM 付费。这些示例未在所有情况下彻底测试。因此，IBM 不能保证或暗示这些程序的可靠性、可维护性或功能。

商标

IBM、IBM 徽标和 `ibm.com` 是 International Business Machines Corporation 在美国和 / 或其他国家或地区的商标或注册商标。 These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (or), indicating US registered or common law trademarks owned by IBM at the time this information was published. 这些和其他带 IBM 商标的术语在本文中第一次出现时使用商标符号 (or) 进行标记, 这些符号表示在本文发布之时它们是 IBM 的美国注册商标或普通法商标。这些商标也可能是在其他国家或地区的注册商标或普通法商标。 IBM 商标的最新列表可在下述网页的“版权和商标信息”中查看: <http://www.ibm.com/legal/copytrade.shtml>。

以下术语是 International Business Machines Corporation 在美国和 / 或其他国家或地区的商标:

AIX 5L	IBM	System p
AIX	MQSeries	System x
CICS	Passport Advantage	System z
DB2	Rational	WebSphere
developerWorks	Redbooks	z/OS
GDDM	Redpaper	
Global Technology Services	Redbooks (徽标) 	

ITIL 是英国内阁办公室部长的注册商标和欧盟注册商标, 且已在美国专利和商标局注册。

Linux 是 Linus Torvalds 在美国和 / 或其他国家或地区的商标。

Microsoft、Windows 及 Windows 徽标是 Microsoft Corporation 在美国和 / 或其他国家或地区的商标。

Java 和所有基于 Java 的商标和徽标是 Oracle 和 / 或其子公司的商标或注册商标。

UNIX 是 The Open Group 在美国和其他国家或地区的注册商标。

其他公司、产品或服务名称可能是其他公司的商标或服务标记。

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Trademarks

IBM, the IBM logo, and [ibm.com](http://www.ibm.com) are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

AIX 5L™	IBM®	System p®
AIX®	MQSeries®	System x®
CICS®	Passport Advantage®	System z®
DB2®	Rational®	WebSphere®
developerWorks®	Redbooks®	z/OS®
GDDM®	Redpaper™	
Global Technology Services®	Redbooks (logo)  ®	

ITIL is a registered trademark, and a registered community trademark of The Minister for the Cabinet Office, and is registered in the U.S. Patent and Trademark Office.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java, and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.

编写组

本文是 IBM Redbooks Publication 的一部分，其标题为 **WebSphere Application Server V8.5: 竞争性迁移指南**，该书由在位于土耳其伊斯坦布尔的 IBM® 办公室编写。

Ersan Arik 是任职于土耳其 Akbank 银行的 IT 专家，担任解决方案架构师一职，负责 WebSphere® Application Server 产品。Ersan Arik 在 IT 和银行业拥有 9 年的专业经验，最近五年专门负责通过 WebSphere Application Server 和 IBM WebSphere MQ 进行集成解决方案的实施。其专业领域包括基础体系结构、功能规划、任务关键环境的集群和高可用性体系结构以及跨平台性能调整。他还是擅长 IBM Content Management and OnDemand Server 等文档管理系统产品。他拥有伊斯坦布尔技术大学计算机工程学士学位。

Burak Cakil 是一名 IT 专家，拥有 6 年的 IT 从业经验。他是土耳其 Web Middleware Enablement Service Line Component in Integrated Technology Delivery (ITD) SSO 团队的主管，在 WebSphere Application Server、IBM Content Manager 及 J2EE 技术的实施、设计和管理方面拥有 4 年的经验。同时，他已通过了 WebSphere Application Server 管理及 IBM Content Manager 解决方案设计方面的认证。他负责针对土耳其 ITD SSO 的内部需要设计相应的 J2EE 应用程序。他曾在土耳其多所大学就 WebSphere Application Server、IBM DB2® 及 J2EE 组件做过多次演讲。他拥有 Yildiz 技术大学计算机工程理学学士学位，目前正在攻读同一专业的理学硕士学位。

Kurtcebe Eroglu 是一名 IT 专家，供职于 Sade Yazilim Ve Danismanlik。他拥有 10 多年的 IT 领域从业经验，主要从事应用程序集成中间件及企业软件开发。他拥有中东技术大学环境工程理学学士、计算机工程理学硕士及工程管理理学硕士学位，目前正在攻读哲学学位。

Vasfi Gucer 担任 IBM 国际技术支持部门的项目主管。他于 1999 年 1 月加入 ITSO。他在系统管理、网络硬件及大型机和分布式平台上所用软件等领域有超过 12 年的从业经验。他曾就 IBM 产品编著过多本出版物并世界各地的在 IBM 课堂上任教。Vasfi 还是已取得 IBM 认证的高级 IT 专家、PMP 及 ITIL 专家。

Rispna Jain 是 IBM 全球技术服务部 WebSphere 产品系列的技术软件部署经理，负责北美地区的客户工作。她在 WebSphere Application Server 产品开发方面拥有 7 年的从业经验，曾在 IBM Software Group 担任多个职位，包括开发、3 级支持和测试。Rispna 还是各种 WebSphere 会议上 WebSphere Application Server 相关主题的技术发言人。她是已取得 IBM 认证的 SOA 的会员，拥有计算机科学硕士学位。

Levent Kaya 是一名 IT 专家，已在土耳其的 IBM Software Services for WebSphere (ISSW) 部门工作达 2 年时间。他拥有 5 年的 IT 行业从业经验。他在 IBM WebSphere Process Server 方面拥有 2 年的开发经验。他拥有 Yildiz 技术大学计算机工程理学学士学位。

Sinan Konya 是 eteration a.s 的首席架构师和创立人。他也是一名对象导师及教育家。他是 ObjectLearn 的创立成员之一，于 2000 年编写了 ObjectWeb Lomboz，该软件是用于 JavaEE 开发的开放式源代码软件。在 eteration 工作之前，Sinan 曾就职于 BEA Systems Inc. 和 The Object People，担任导师一职。他在企业级和任务关键系统的设计和实施方面拥有超过 17 年的从业经验。他曾在各个行业教授对象技术及使用 Java 和 Java EE 技术进行企业软件开发，相关经验超过 15 年之久。他拥有伊斯坦布尔技术大学控制与计算机工程理学硕士学位。他曾担任 Eclipse WTP 评委。

Hatice Meric 是一名技术顾问，就职于土耳其的 IBM 伊斯坦布尔创新中心。在 IBM 供职期间，曾担任 IBM 土耳其分公司的软件服务部门的 IT 专家。她拥有 Yildiz 技术大学计算机工程理学学士学位。

Ross Pavitt 是英国 IBM Hursley 的软件工程师。他于 2012 年加入 WebSphere Application Server Liberty 概要文件团队，担任测试主管一职，主要负责客户方案测试。之前，他曾是 IBM CICS® Transaction Gateway 团队的成员之一。Ross 于 2008 年加入 IBM，主要负责 IBM System z®、IBM System x® 及 IBM System p® 等多个平台的功能验证及其他自动化测试。

Dave Vines 是位于英国的 IBM Hursley 实验室的 IBM 创新骨干及高级软件工程师，供职于 WebSphere Application Server 开发部门，主要负责 Liberty 概要文件及 IBM Workload Deployer。他于 1984 年加入该实验室，曾参与过多个 IBM 许可程序的工作，其中包括 GDDM®、IBM MQSeries®、LANDP；1998 年以后，负责 Component Broker 及 WebSphere Application Server。他于 1984 年获得英国埃克塞特大学的理学学士学位。

Hakan Yildirim 是土耳其 Garanti Teknoloji 平台及应用程序管理方面的高级专家。Hakan 在金融领域拥有超过 14 年的从业经验。他毕业于 Yildiz 技术大学的电子与通信工程系。他于 1998 年加入 Emlak Bank，职业生涯由此开启。他曾在土耳其伊斯坦布尔先后担任工程师、专家和高级专家等职。他主要从事 WebSphere Application Server、WebSphere Portal Server、WebSphere Process Server 及 WebSphere ESB 等 WebSphere 系列产品相关工作。他的核心技术领域为 WebSphere Application Server 管理及 WebSphere Application Server 环境的规划和设计。他于 2009 年加入 Garanti Teknoloji，担任平台及应用程序管理高级专家一职。他曾与开发人员一起负责 Java 应用程序的调试，积累了大量经验。他曾参与过多个大型银行项目，主要负责网上银行及电子商务解决方案。目前，Hakan 负责 WebSphere 和 WebLogic 等 Java 应用程序服务器的管理事宜。

Tayfun Yurdagul 是 VBT Vizyon Information Technologies 的解决方案架构师。他在 IBM WebSphere、IBM Rational® 产品、DB2 及 Java 技术方面拥有超过 9 年的经验。他曾参与过多个大型任务关键软件项目。同时，Tayfun 还拥有 DB2、WebSphere 及 Rational 产品证书。

非常感谢下述人士对此次项目的卓越贡献：

Susan Hanson、David Bennin、Emma Jacobs、Wade Wallace、Erica Wazewski
国际技术支持部

Dana Duffield、Surya Duggirala、Cindy High、Roman Kharkovski、Tom Kristek、
Dr. Alex Mulholland、Mike Morin、Rengan Sundararaman、Donald Vines、Nathan Ziemann
IBM 美国

Graham Hopkins 和 Ian Robinson
IBM 英国

Jale Akyel、Gulsun Emuler、Yesim Mutlu、Serkan Sahin、Engin Turpcular、Kivanc Uslu
IBM 土耳其

Andrea Bauer 和 Tim deBoer
IBM 加拿大

Ping Shao
IBM 中国

本书编写组对项目执行发起人 Richard Baird（IBM WebSphere 基础开发副总裁）在整个项目过程中提供的大力支持表示衷心感谢。

同时，也对本文之前各个版本的编者表示由衷感谢。

- ▶ 第一版 *WebSphere Application Server V7: 竞争性迁移指南*，SG24-7870-00，2012 年 7 月出版的编者包括：

Joao Emilio Santos Bento da Silva、Kurtcebe Eroglu、Kiran Mantripragada、Hamdy Eed
和 Fabio Xavier Albertoni

现在，您也能够成为一名出版作家！

这里我们为您提供展示技能，迈上职业生涯新台阶并成为一名出版作家的机会！赶快加入 ITSO 高级培训项目并协助编写您所擅长领域的出版物，这也是利用最前沿的技术积累经验的绝佳机会。您的付出将能极大地扩展技术联系人及关系网，从而有助于提升产品认可度和客户满意度。培训项目为期 2 到 6 周，您可以亲临现场参与培训，也可以在家接受远程训练课程。

您可以通过以下网址了解有关高级培训计划的更多信息，浏览高级培训索引并在线申请：

ibm.com/redbooks/residencies.html

欢迎评论

您的评论对我们而言至关重要！

我们希望本文能够为您提供尽可能多的帮助。您可以采用下列方式的任何一种向我们发送您对本文或其他 IBM Redbooks® 出版物的评论：

- ▶ 使用在线**联系我们 Redbooks** 评审表，其网址为：

ibm.com/redbooks

- ▶ 使用电子邮件将您的评论发送到：

redbooks@us.ibm.com

- ▶ 将您的评论邮寄到：

IBM 集团公司国际技术支持部
HYTD Mail Station P099
2455 South Road
Poughkeepsie, NY 12601-5400

与 IBM Redbooks 保持连接

- ▶ Facebook 联系方式：

<http://www.facebook.com/IBMRedbooks>

- ▶ Twitter 联系方式：

<http://twitter.com/ibmredbooks>

- ▶ LinkedIn 联系方式：

<http://www.linkedin.com/groups?home=&gid=2130806>

- ▶ 您可以通过 IBM Redbook 每周新闻简报浏览最新 Redbooks 出版物、高级培训项目及研讨会的相关信息，其网址为：

<https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm>

- ▶ 您可以通过以下网址随时了解最近的 IBM Redbooks 出版物及 RSS 源：

<http://www.redbooks.ibm.com/rss.html>



迁移战略及规划

本章节提供的信息，您可以用来创建迁移流程的相关计划。本章所含信息并非最终方法，仅供您在组织迁移过程、处理特定任务相关人员、了解何处需要寻求帮助并决定如何实施项目时做参考之用。

本章介绍了启动迁移和制定项目计划的相关指南以及一些可用作影响迁移能否成功的输入信息的工件。

本章所述主题如下：

- ▶ 迁移过程概述
- ▶ 迁移评估
- ▶ 迁移规划
- ▶ 迁移实施
- ▶ 后部署
- ▶ 总结

1.1 迁移过程概述

影响迁移执行决策的因素有很多，包括确保公司适应不断变化且越来越具挑战性的 IT 环境的业务决策和技术评估等等。

Java EE 应用程序的迁移不仅仅是从一个应用程序服务器更改为另一个服务器或者仅仅更改源代码这么简单。迁移是一个涉及到多个阶段以及来自多个领域、具备各种技能的人员的过程。之所以要迁移到新的应用程序服务器，您的出发点可能是为了充分利用新功能、改善应用程序管理方式，重新编写代码、提升性能或者减少停机时间。

尽管每次迁移都很复杂，但最好的迁移方式是遵循能够对应应用程序开发环境、源代码、设计注意事项等等加以评估的正确流程，并就如何将应用程序迁移到 WebSphere 环境给出建议。因此，应在有序的环境中实施这些变化，从而通过适当的战略规划和实施来避免出现问题。

图 1-1 对迁移过程进行了简要概述。

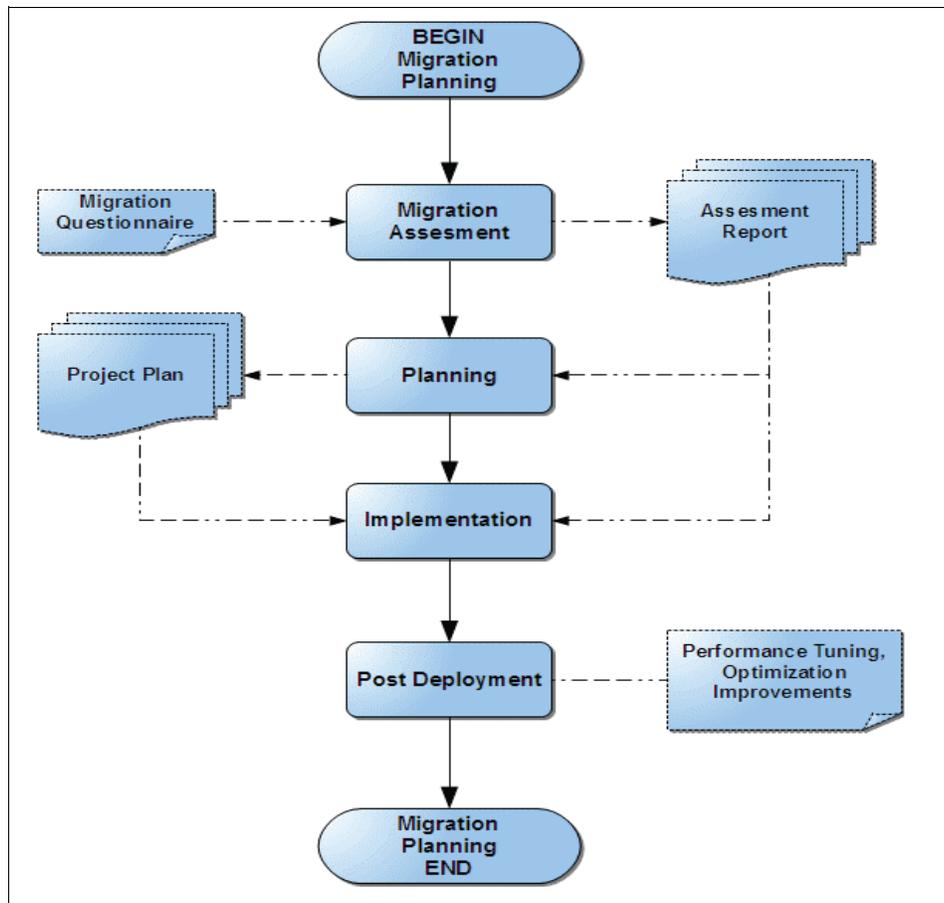


图 1-1 迁移过程概述

1.1.1 迁移类型

在确定迁移方式时，有很多方法可供遵循。本章介绍了三种迁移方法，其中涵盖了大多数类型的应用程序迁移备用方案。我们需要根据实际的情况来决定采用哪种方式。运行时迁移与具体类型无关。在各种迁移方法中，可能均需进行迁移评估。（有关迁移评估，请参阅第 4 页上的“1.2 迁移评估”）。

您也可能会决定制定相应的项目管理方案，具体视您的情况而定。

样板迁移

样板迁移是指能够代表其他应用程序的单个应用程序的迁移。代表性应用程序是指与企业内迁移所涉及的大多数应用程序在体系结构和实施方面均具有类似性的应用程序。

第一个应用程序的迁移完成后，可针对更多的应用程序重复该流程。此次迁移为基于样板应用程序迁移时所遭遇的诸多挑战的未来迁移奠定了基础，因此此次迁移可以说是最为常见的应用程序迁移类型。您也可以确定迁移的速度并突出所面临的问题，用以制定简要的计划。

部分迁移

部分迁移是指采用分阶段实施的方式一次性迁移多个或少量应用程序，所有的迁移都涵盖在迁移计划表之中，直至项目完成。当您的团队拥有共享环境中的应用程序或者您的某个团队拥有多个应用程序时，由于时间冲突问题而导致无法同时迁移全部应用程序的情况下，这种迁移类型是比较理想的选择。

基于相同的体系结构编写并采用模块化结构的应用程序也可采用此种类型的迁移（例如，在迁移业务组件之前首先迁移用户界面模块，反之亦然）。

完全迁移

完全迁移的含义很明显，但也有其自身的特性。在决定采用完全迁移时，您拥有专门的环境供应用程序使用，或者由于支持终止、某些应用程序无法互操作或其他可能出现的问题，您必须迁移整个环境。

1.1.2 重要方面及注意事项

本节给出了一些与 **Java EE** 应用程序迁移有关的注意事项。假定在迁移完成后，均能在新版本的代码、编译器、新功能、运行时管理和性能提升方面有所改善。

了解迁移过程中可能面对的各个挑战，也至关重要。明智的决策能够避免很多问题。每次迁移都各不相同。在复杂性、规模和其他方面都存在诸多差异。为了便于您顺利实施迁移，下面列出了我们根据自身的迁移经验以及编著本书时思考得出的一些注意事项：

- ▶ 若想要制定出适合于各种环境的迁移路线图，是不可能的事情。
- ▶ 应使迁移尽可能简单化，避免一次性实施多个变化。
- ▶ 切记除生产环境以外还有很多其他环境，例如开发环境、QA 环境、测试环境等等。
- ▶ 迁移应用程序时应确保尽可能少的变化，而且迁移实施要尽可能地迅速。
- ▶ 在开始迁移之前，切勿忽略迁移评估，无论要进行的迁移是简单还是复杂。
- ▶ 评估时，应审核代码并评估代码与新平台 (**Java SE**) 的兼容性。
- ▶ 评估时，应审核代码并评估应用程序符合或偏离 **Java EE** 规范的程度。
- ▶ 切勿忘记集成开发环境 (**IDE**) 的迁移。
- ▶ 尽管应用程序代码和应用程序服务器（旧版本和新版本）均符合 **Java EE** 规范，但各个供应商可按照自己的方式实施其代码，也可能对规范进行扩展，必须予以迁移。
- ▶ 不要一次性实施过多变化。应采用迭代的方法进行迁移。首先迁移一小部分代码并加以测试，然后返回并重复此过程。在较小的受控环境中，更容易发现问题。
- ▶ 应尽量确保变化的变数最小化。举例来说，除了与迁移有关的变化外，不要做出代码变化。仅评估构建脚本及必需的运行时配置所需的变化。
- ▶ 明确如何处理 **Java SE** 和 **Java EE** 规范。是应采用与之前版本兼容的规范进行编译还是采用升级后的规范？
- ▶ 切勿从之前的应用程序服务器中复制您的性能调整设置。使用 **WebSphere** 做出具体的性能调整决策。

- ▶ 组织相关各方对迁移进行讨论，并使其相互交流，以借鉴之前迁移实施中经验。
- ▶ 切勿忘记有很多工具可供在迁移过程中做辅助之用。
- ▶ **WebSphere Application Server V8** 能够与 **J2EE 1.2** 之后的全部版本兼容。不过，大多数较早版本的应用程序代码都包含第三方库，且使用供应商特定的扩展及非公共 **API**。因此，必须重新设计、替换并重新实施这些代码。
- ▶ 我们可能会遇到导致迁移较为复杂的外部因素，因此相关各方均应予以注意。这些外部因素包括与现有系统的整合及依赖于其他应用程序功能或服务的迁移应用程序之间的互操作性。
- ▶ 务必制定回滚计划。回滚计划对业务而言至关重要，用以确保迁移过程中不会出现任何计划外停机时间。在试图对生产系统进行迁移之前，必须实施回滚计划，这一点同样重要。此外，在开始迁移前应备份您的代码。

1.2 迁移评估

在某些情况下，您可能在信息不足的情况下仓促进行迁移实施和规划，随后就会出现可预测的意外问题。通过执行迁移评估即能对这些问题进行预测并降低其影响。

迁移评估是迁移的首要步骤。迁移团队要想了解迁移的范围、预期、当前环境、相关团队及所有其他相关信息，除了进行评估研究之外，别无他法。评估阶段收集的信息有助于了解应用程序环境。这些信息可供在规划阶段使用。

本节介绍了评估的方法及评估阶段能够产生的工件。

1.2.1 迁移调查问卷

因为评估是迁移的首要步骤，且在某些情况下，您可能根本无实际客户的之前经验可供借鉴，所以，您必须了解项目启动所需的信息，并确定相关各方的角色和职责。利益相关方必须尽心尽责完成项目，确保迁移的成功实施。

若要获取相关的知识，应善加利用迁移调查问卷，该调查问卷是一系列问题集，用以收集与迁移、业务要求和决策、硬件和软件、拓扑、基础结构、应用程序代码等有关的必要信息。在此步骤中，切记确保所有的问题都得到妥善的回答，不要使用诸如“是”或“否”等非常简单的答案。

在收集调查问卷所用信息时，可使用 **IBM WebSphere Application Server Migration Toolkit**。对于样板迁移的调查问卷，请参阅第 109 页上的附录 A “迁移调查问卷”。

Migration Toolkit 具有报告功能，能够发现已知 **Java** 问题并将其导出为 **PDF** 报告文件。**XML**、**JSP** 和类路径等结果可导出为 **XML** 文件。

1.2.2 迁移调查问卷评审会

所需信息收集完毕后（通常由各个团队单独收集），必须与相关方对此类信息进行讨论，这一点至关重要。应定期举办此类评审会，并组织利益相关方共同讨论收到的回应，解决问题并组织相关活动，例如代码审核及概念验证。这些评审会可通过电话、**Web** 会议和面对面磋商等方式进行。

1.2.3 迁移评估报告

在完成调查问卷后，将收到的信息整理成报告形式，以便各个相关人士了解达成决策的关键指标。该报告的内容应包括以下章节：引言、培训、规模确定、拓扑、应用程序迁移、运行时（服务器）迁移、测试、部署、调整及性能、风险以及与项目计划相关的其他事宜。

通过该报告能够将项目风险降至最低并提高项目的成功率。该报告能让您了解您可能未考虑到的多个问题：

- ▶ 项目的目标是什么？
- ▶ 项目能够给业务带来哪些益处？
- ▶ 开发人员和管理员是否熟悉 IBM WebSphere Application Server V8.5？
- ▶ 测试人员是否了解测试的关键部分？
- ▶ 对于迁移活动的执行而言，需要哪些必要资源和技能？
- ▶ 迁移需要多长时间？
- ▶ 计划使用何种拓扑？
- ▶ 相关的其他环境有哪些？
- ▶ 应用程序代码迁移的相关建议有哪些？
- ▶ 在运行时迁移过程中必须进行哪些配置？
- ▶ 是否会创建部署和管理脚本？

以上罗列了评估报告中的一些样例，将有助于迁移，报告越完善，项目计划和实施就越顺利。

1.3 迁移规划

进行迁移评估后，接下来的步骤即为迁移规划。此时，评估报告已经完成，您对项目的利益相关方及待完成工作量也有所了解。在制定正式的项目计划时，应以迁移评估结果为依据。

1.3.1 角色和职责

在迁移过程中，确定各个角色及其职责至为关键。首先应分派的角色之一是项目经理。项目经理负责项目的规划和实施。其他角色包括应用程序所有者、开发人员、测试人员、管理员及决策人员。为特定活动分派相应的资源，由各个资源负责所分派的角色任务。

通常情况下，迁移的执行人员已提前确定，但在评估报告完成后，可根据报告结果对计划进行修改。所分派的团队会对项目计划产生一定的影响。举例来说，由于培训问题或之前的经验及项目的复杂性等因素，均需要或多或少的时间和资源，具体视迁移的执行人员而定。

1.3.2 迁移项目计划

迁移项目计划应根据评估报告制定。通过评估报告，您能够了解各项活动之间的联系、可并行执行的活动以及所迁移各个部分的复杂性。已做出的决策及必需做出的决策均应列于活动列表之中。

任一项目的项目经理均应明确了解如何执行活动，但由于本书的读者所承担的角色不一，因此下文给出了良好的迁移项目计划应包含的部分：

- ▶ 日期（推出及里程碑）
- ▶ 任务
- ▶ 所分派资源的数量
- ▶ 测试计划
- ▶ 依赖关系
- ▶ 风险
- ▶ 生产投产

- ▶ 会议及检查点
- ▶ 培训活动
- ▶ 关键路径

某些任务可并行进行（培训、设置、测试等等）。

该列表并非最终列表，您可以根据您的具体需求进行删减。总结并整理迁移实施过程中得出的经验对于以后的迁移实施非常有益。

生产投产时所分派的资源可能与迁移时所分派的资源不同。

1.4 迁移实施

迁移实施与迁移规划密不可分。在迭代项目中，迭代的最终阶段为迭代输入阶段。该方法有助于调整项目计划中遇到的问题，使其更为准确，也有助于确定将来推出时的预期。

在规划阶段，需做出很多决策（例如：首先迁移哪个应用程序；使用何种类型的迁移（样板迁移、部分迁移或完全迁移））。此外，还必须做出目前能够处理的实施决策，大多数情况为技术决策。

1.4.1 实施注意事项

应考虑并行实施运行时（服务器）和应用程序的迁移。这样在完成代码更改及调整后，您的环境也将准备就绪，可供部署使用。应利用评估报告完成以下任务：

- ▶ 识别环境方面可能出现的变化。
- ▶ 与相关各方进行互动沟通。

对运行时和集成调查问卷进行研究，以了解如何利用这些答案来确保迁移的顺利实施。举例来说：

- ▶ 检查拓扑及网络相关的答案。
- ▶ 识别可能妨碍您的应用程序适当运行的防火墙规则。
- ▶ 检查是否需要服务器之间的互操作性或者是否正在处理与现有系统的集成事宜。
- ▶ 确保功能及非功能要求均包括在测试计划之内。
- ▶ 在将应用程序移至生产环境时，尝试特定的情景并将审核源相互隔离，以避免发生问题。

1.4.2 实施阶段

在本章的前面部分，我们曾提到，对于迁移项目而言，迭代过程是个不错的选择。通过这一过程能够明确识别各个阶段之间的关键路径并为将来的迭代提供反馈，从而有助于降低风险。有关更多常见问题，请参阅第 17 页上的第 2 章“常见迁移问题”。

考虑图 1-2 中所示的 5 个阶段。

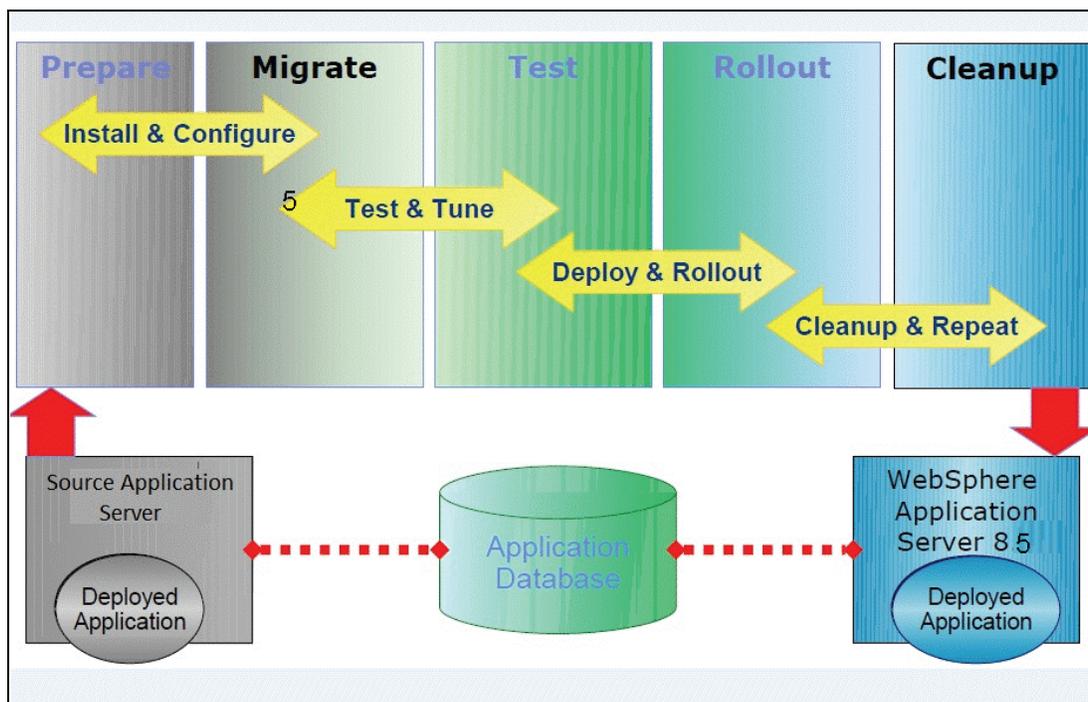


图 1-2 迁移关键活动

► 准备阶段

在该阶段中，需检查产品和应用程序的先决条件。应确保硬件、软件及人员均已到位。设置开发环境、制定会议计划、提供培训并在必要时寻求帮助。

► 迁移阶段

开始应用程序代码、运行时环境及集成开发环境 (IDE) 的迁移。切记评估报告是迁移的依据；您能够从中找到与所做的决策相关的信息，有时候甚至还能够找到有关说明迁移中具有挑战性领域的代码审核反馈的重要信息。

您必须确定迁移类型，因此应立即对所发生的任何事情及从反馈中得出的信息予以记录。对于项目经理或其代表而言，在处理某个人员与其他各方（诸如软件和硬件供应商或第三方服务提供商）互相讨论或交互时所提出的问题方面，其角色至关重要。

迁移和测试阶段紧密相连。在完成迁移所有内容之前，首先测试一小部分功能，这样可能会非常有益。这种方法有两个优势：

- 第一个已迁移功能的相关挑战可用作其他迁移的输入，从而加快迁移过程的速度。
- 测试与代码迁移同时进行，并为迁移完成后的 QA 和负载测试做好准备。

► 测试阶段

测试阶段比较灵活且与迁移阶段和推出阶段均有交叉。在迁移过程中包括三种主要的测试类型：

- 交接测试，该测试与应用程序迁移同步进行
- 功能 / 回归测试，用以确保全部功能均能符合预期
- 负载测试，用以评估应用程序 / 服务器的性能是否能达到或超越之前的环境

► 投产阶段

在前述的全部测试均已完成并且您确定应用程序能够提供与之前的环境同等或更好的响应之后，即可开始推出阶段。我们都知道可能有些问题仅在生产过程中出现，这就是为什么需要制定计划并进行之前其他阶段的原因，旨在防止出现推出中断的情况。

您需要将应用程序部署到生产环境中并再次运行测试，在一切事宜都准备就绪后，再确定是否继续。通常，需要针对这些推出活动制定包括里程碑、所有者及检查点在内的详细计划，以便组织全部团队成员来解决出现的任何问题或启动中止计划。

带有互操作性和界面的推出活动比较难以实施，因为存在潜在的与现有应用程序的兼容性问题。要时刻牢记这些挑战并制定相应计划以防此类问题的发生。在这一方面比较好的方法就是在维护期限内完成迁移，以便能够中止向受影响系统的外部输入并测试集成和互操作性，从而将进行已与实际有效的请求混为一体的测试的风险将至最低程度。可能还需要整理数据库或消息队列。

► 整理阶段

在将已迁移的应用程序成功部署到生产环境后即可开始本阶段。有很多性能和系统稳定性监控工具可供本阶段使用。同时，分派相关专业人员对可能需要的设置进行适当的调整。在生产环境中作出变化之前，应首先在非生产环境中作出相应的变化，并重新执行测试计划。

1.4.3 应用程序迁移

有三种截然不同的应用程序迁移方式：

- 将服务器运行时从应用程序服务器迁移至 IBM WebSphere Application Server V8.x。
- 将开发环境从供应商特定 IDE 迁移至 WebSphere Application Server V8.x Developer Tools for Eclipse 或 Rational Application Developer for WebSphere Software。
- 迁移在应用程序服务器上运行的应用程序。

Java EE 应用程序迁移的复杂性各不相同，具体取决于在迁移评估过程中识别的一系列因素：

- 源和目标 Java SE 规范（JRE 1.4 至 JRE 1.6）
- 源和目标 Java EE 规范（J2EE 1.4 至 Java EE 6）
- 特定 Java 组件的使用（EJB、JSP 和 标记库）
- 特定于应用程序服务器的专用代码的使用
- 特定于供应商的部署描述符
- 扩展功能
- 特定于供应商的服务器配置（JMS 提供商或 JDBS 连接池）
- 类装入问题
- 与 CRM 或 SAP 等复杂外部系统的集成

上述变化可手动或使用工具实施以加快迁移速度。在这一方面，可使用 IBM Rational Application Developer V8.5 提供的扩展功能。此外，新的 Migration Toolkit 已发布，用以将应用程序从 Oracle WebLogic、Oracle Application Server、JBoss、Tomcat 或之前版本的 WebSphere Application Server 迁移至 IBM WebSphere Application Server V8.x。对于使用注释的应用程序，Rational Application Developer 能够为其提供 WRD 标记，用以构建和维护使用 EJB 3.x、Web 服务和其他组件的应用程序。同时，还提供有其他相关工具，能够在代码审核方面提供指导。

应用程序迁移有一些通用的步骤可供遵循。有关常见的迁移问题，请参阅第 17 页上的第 2 章“常见迁移问题”。

1.4.4 IDE 迁移

如果您使用的是特定于供应商的 IDE（此类 IDE 能够为其应用程序服务器提供相关工具和向导并包括自有的项目结构），则必须将其迁移至 **WebSphere Application Server Tools for Eclipse** 或 **Rational Application Developer**。大多数基于 Eclipse 的 IDE 都以 Eclipse WTP 为基础，使得迁移更为容易。

WebSphere Application Server Tools for Eclipse 包含有用于管理服务器、发布至本地或远程服务器以及控制增量发布的工具。这些工具包括富页面编辑器及 WYSIWYG 编辑器，使用这些工具即能轻松地编辑 HTML 文件、将 Dojo 窗口小部件添加至 HTML 页面以及创建和编辑用于移动设备的 Web 页面。

Rational Application Developer V8.5 为企业应用程序开发人员提供了从设计到部署的完整套件。**Rational Application Developer V8.5** 能够帮助 Java 开发人员快速设计、开发、编译、测试、构建和部署高质量的 Java/Java EE、Portal、Web/Web 2.0、OSGi、Web 服务及 SOA 应用程序。

本书所选的工具中，对迁移样板最为有益的工具为 **Rational Application Developer V8.5**。因此，需了解 **Rational Application Developer V8.5** 可如何帮助您改善从之前的 Java EE 规范迁移以及如何充分利用 **WebSphere Application Server V8.x** 的新功能。有关更多信息，请参阅：

- ▶ *Experience JEE Using Rational Application Developer V7.5*, SG24-7827
- ▶ *Getting Started with WebSphere Application Server Feature Pack for Service Component Architecture*, REDP-4633

如何将代码导入 Eclipse 中

您能够将单个 EAR 文件导入到项目或将项目（含多个 EAR 文件或 WAR 文件）导入 Eclipse 平台中。两者所依据的逻辑完全相同。

将 EAR 文件导入项目中

若要将 EAR 文件导入项目中，按照以下步骤操作即可：

1. 从 Java EE Eclipse Enterprise Explorer 选项卡中，右键单击并选择导入 → EAR 文件）（图 1-3）。

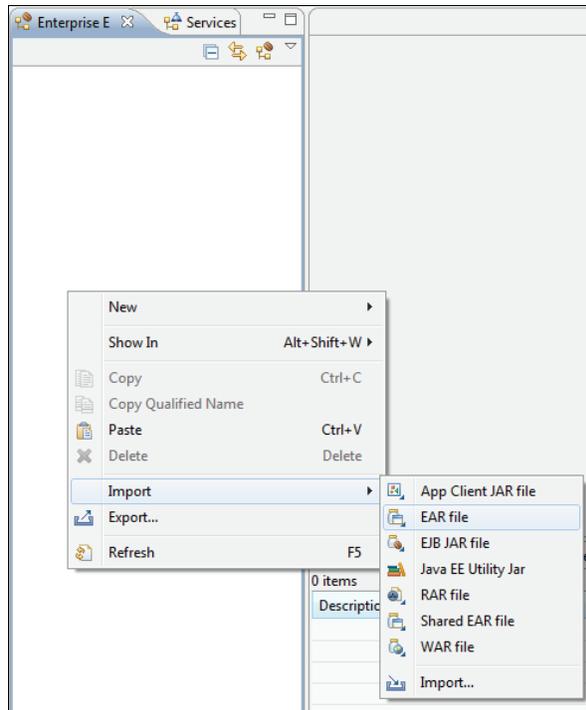


图 1-3 “选择导入源”窗口

2. 选择 EAR 文件，例如 sampleEJB.ear（图 1-4）。

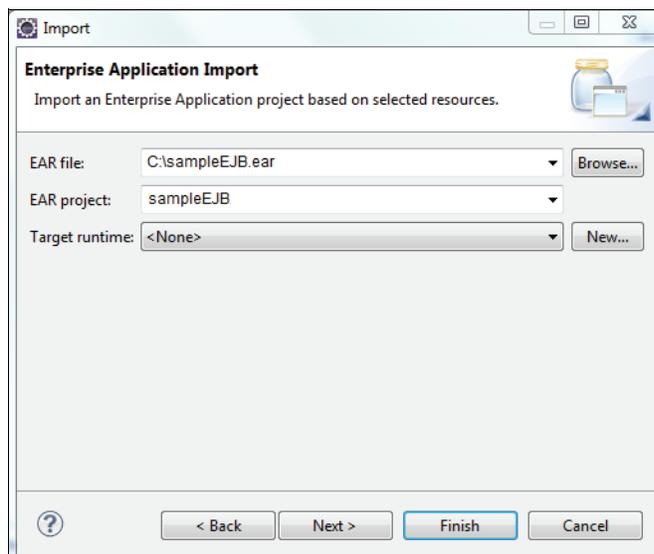


图 1-4 选择 EAR 文件

3. 在导入项目时，若未将 WebSphere Application Server 添加到工作空间中，则必须在“企业应用程序导入”窗口中单击**新建**，选择目标运行时（图 1-5）。

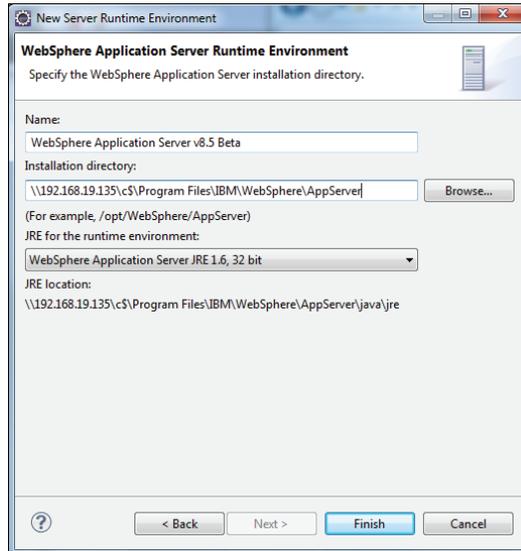


图 1-5 选择目标运行时

4. 单击**完成**。在下一个窗口中，选择待导入的 JAR 文件（图 1-6）。

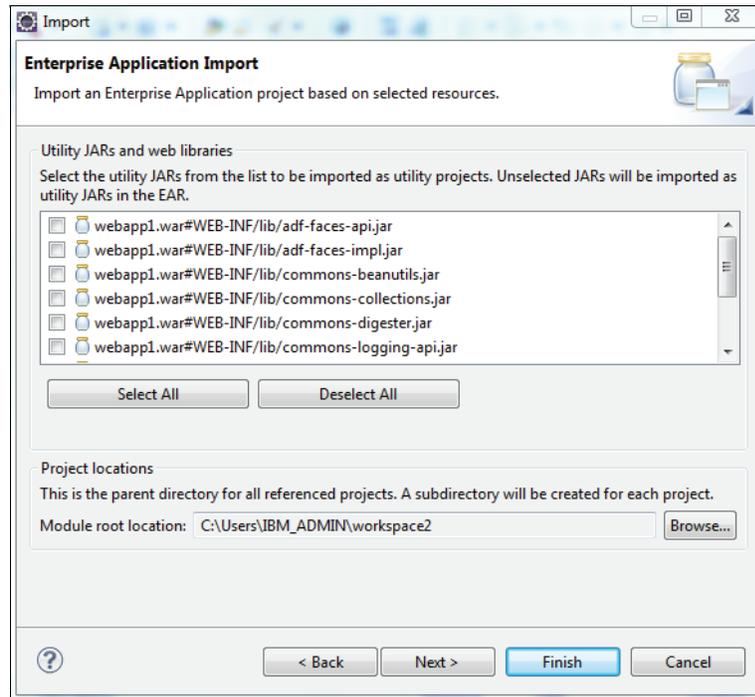


图 1-6 选择 JAR 文件

导入 Java 源代码：导入 EAR 文件即可构建应用程序、Web 模块和 EJB 模块的项目结构。通常，EAR 文件不包括 Java 源代码，因此必须通过额外的步骤将其导入其中。

将现有项目导入 *Eclipse* 中

您也可以通过以下步骤将现有项目导入 *Eclipse* 中：

1. 在 *Eclipse* 主菜单中，单击文件 → 导入。
2. 在“选择导入源”窗口中，选中“工作空间中的现有项目”并单击下一步。或者，您也可以单击常规 → 工作空间中的现有项目并单击下一步（图 1-7）。

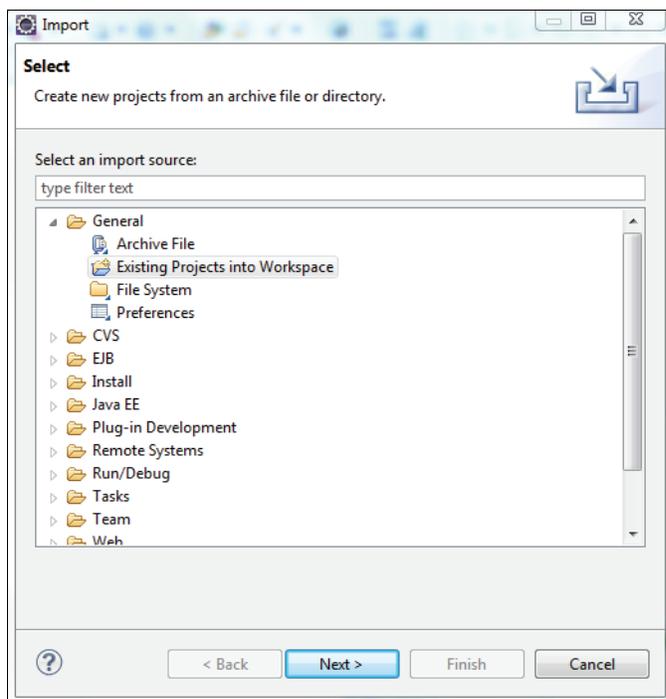


图 1-7 工作空间中的现有项目

3. 浏览并选择**选择根根目录**或**选择归档文件**，然后选择想要添加的项目（图 1-8）。

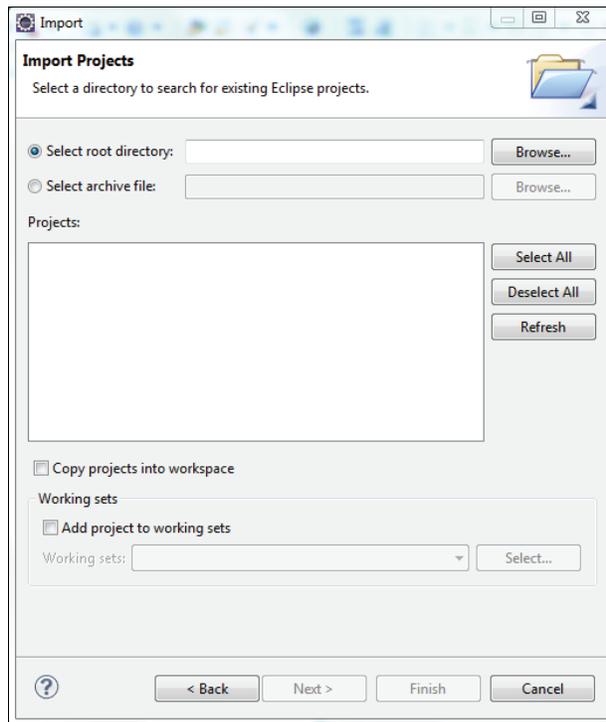


图 1-8 浏览并选择根目录或归档文件

4. 单击**完成**。

1.4.5 运行时迁移

有时我们仅仅关注于代码的变化，当认识到在其他方面也存在迁移挑战时，可能为时已晚。无论环境复杂还是简单，总有需要变化、实施或迁移的事项。具体来说，在迁移 **Java EE** 应用程序时，尽管符合规范，但各个供应商实施和提供服务及改善功能的方式各不相同。

考虑到应用程序运行于操作系统中，而操作系统也可能是迁移的一部分，因此在操作系统级和应用程序服务器级进行的更改和配置均需加以留意。运行时迁移包括以下几个要素：

- ▶ 管理脚本更改或更换
- ▶ 支持应用程序服务器功能的资源配置
- ▶ 操作系统先决条件
- ▶ 防火墙规则
- ▶ 安全配置及认证数据库
- ▶ 部署绑定
- ▶ 启动 / 停止脚本
- ▶ **JVM** 专用或附加配置
- ▶ 类装入器配置
- ▶ 库
- ▶ 池及线程配置

迁移还包括产品所特定的任务。我们将在具体产品的介绍章节对这些任务加以说明。

1.5 后部署

新环境的性能至少会达到与之前环境相同的水平，但我们的预期绝非仅此而已。为了实现更高的性能，就必须采取相应的措施。这些措施基于以下两个前提：应用程序和运行时。

在代码更改方面，应尽量避免不必要的修改。在迁移应用程序时，应将变化的数量控制在最小程度。在应用程序迁移完成后，重新访问该应用程序，以对其加以改善，确保应用程序和新环境的整齐度、正确性及快速性。

1.5.1 性能调整及优化

性能调整和优化需要相应的专业知识和信息，以为您的决策提供支持。此类信息有两个来源：应用程序代码审核和负载测试结果。只要掌握了这些结果，专业人员就有足够的信息来进行必要的调整。同样，迭代过程是理想的选择。首先选择一小部分代码或特定配置，并关注于此类要素直至达到预期的结果。如此一来，即可找到使整个应用程序或环境的性能更佳的关键路径。

若要实现更佳的性能，必须遵照项目计划的要求谨慎执行相应过程。首先至少应确保达到与之前同等的性能水平，然后在生产环境中将迁移部署完毕后，就能了解性能提升的程度。在这一方面，我们也提供了相应的辅助工具。IBM 通过单独的渠道提供此类帮助。有关入门信息，请查看 **WebSphere Application Server V8** 信息中心 **Web** 页面：

http://www14.software.ibm.com/webapp/wsbroker/redirect?version=phil&product=was-nd-mp&topic=welc_howdoi_tprf

1.5.2 获取帮助

以下团队会随时为你提供迁移过程方面的帮助：

- ▶ IBM 业务合作伙伴
- ▶ IBM 业务合作伙伴服务实践
- ▶ IBM Software Services for WebSphere Migration Practice (ISSW)
- ▶ WebSphere 竞争性迁移团队
- ▶ 第三方来源

IBM 迁移专家在这些环境方面经验丰富。迁移及目标实施环境方面的专业知识至关重要，甚至在某些情况下，对减少实施过程中的中断几率非常关键。

获取 IBM 提供的支持

使用下文所述的渠道，即可获取 IBM 提供的支持和协助。

IBM WebSphere 开发人员服务

IBM WebSphere 开发人员服务能够在 IBM 业务合作伙伴开发与 IBM WebSphere 产品相关的应用程序或解决方案时为其提供开发人员对开发人员的技术协助。若您需要协助，只需提交出被称作 *问题管理记录 (PMR)* 的票据即可。可通过以下 **Web** 页面提交 PMR：

<http://www.ibm.com/isv/tech/remoteEmail/entryForm.jsp>

IBM Passport Advantage

IBM Passport Advantage® 能够就生产问题及产品漏洞提供相应支持。若要通过 Passport Advantage 打开 PMR（或服务请求），请转至：

<http://www-01.ibm.com/software/support/probsub.html>

IBM Software Services for WebSphere (ISSW)

IBM Software Services for WebSphere 能够提供手把手支持和技能传授活动，以帮助您成功部署新的 WebSphere 软件解决方案。有关 ISSW 的更多信息，请转至：

<http://www3.software.ibm.com/ibmdl/pub/software/dw/wes/pdf/services/DevelopDeployFinal.pdf>

WebSphere 教育

WebSphere 教育能够帮助您掌握并提升 WebSphere 技能。该工具为您提供了 250 多种有关 WebSphere 软件产品组合及 SOA 的课程。培训方式很灵活，包括课堂培训、在线培训和私人培训。这些课程均由拥有第一手产品知识的知名讲师设计。

WebSphere 教育中基于角色的培训方法能够帮助您确定掌握 WebSphere 特定产品技能的最佳途径。有关 WebSphere 教育的产品列表，请转至：

<http://www-01.ibm.com/software/websphere/education/>

WebSphere 竞争性迁移团队

WebSphere 竞争性迁移团队由来自世界各地的迁移专家组成，能够为您提供后面章节中所述的服务。

迁移发现

帮助客户了解 WebSphere 所提供的价值主张及竞争性应用程序服务器所带来的优势。

迁移评估

就基于竞争性应用程序服务器的应用程序的迁移步骤和方法提供全方位的应用程序分析及具体的建议。此外，还能为客户及 ISV 提供实施协助，以提升时间效率并降低风险。

指导及技能传授

帮助客户充分利用 WebSphere 所提供的功能，为客户创造更多价值。

入门联系人

若要入门，可与您当地的 IBM 市场营销代表联系，或发送电子邮件至 wascmt@us.ibm.com。

您还能够免费获得以下部分或全部服务：

- ▶ 由 IBM 迁移专家为您精确估算迁移所需时间（需要 2 到 3 小时）。
- ▶ 由 IBM 专家亲临现场对您的环境和应用程序进行评估，并免费进行全方位的迁移评估。
- ▶ 举办迁移研讨会，为您公司的技术人员或管理人员提供为期 1 到 3 天的培训课程，期间将为其介绍 WebSphere Application Server V8.5 并由专家亲身演示，帮助其了解业务价值和迁移过程。
- ▶ 概念验证 (PoC)，在其中，IBM 迁移专家将亲临现场并使用 IBM 久经考验的迁移方法和自动化工具将您的 1 到 2 个应用程序迁移到测试环境中的 WebSphere Application Server。

1.6 总结

上述迁移过程和活动并非最终迁移方法，也不是实施迁移项目的唯一途径。其作用在于帮助您如何为迁移做准备并估计可能出现的挑战。在实施您自己的项目实施时遵照这一过程，必定是您受益匪浅。正如您所看到的，影响迁移规划的因素有很多，但最主要的因素有两个：

- ▶ 迁移评估及评估报告（用作项目计划的输入信息）
- ▶ 使用迭代过程，其中第一波或第一次迭代能够为将来的迭代提供知识依据

另外需要特别注意的是，尽管迁移与提升互为关联，但“先迁移，再提升”的方法能够将这两种活动的缺陷相互隔离，是一种很好的方法。



常见迁移问题

本章介绍了读者在从其他 Java Enterprise Edition (EE) 平台迁移至 WebSphere Application Server V8.5 时遇到的常见迁移问题。

Java EE 定义了可由不同供应商单独部署的应用程序的框架。该配置确保企业用户能够以一种不依赖于单个供应商解决方案的方式编写应用程序。尽管 Java 和 Java EE 背后的主要理念为可移植性及一次编写，到处运行 (*Write Once, Run Anywhere, WORA*)，有时亦作一次编写，随处运行 (*Write Once, Run Everywhere, WORE*) 功能，但由于各个供应商实施 Java EE 规范的方式各不相同，因此通常会导致在迁移 Java EE 应用程序时出现问题。

供应商还会实施未包含在 Java EE 规范之内的功能，这就是 Oracle 提供供应商与 Java EE 兼容产品对照表的原因。Oracle 通过测试来验证 Java EE 应用程序服务器是否符合规范。这种测试能够确保在 Java EE 应用程序服务器上部署的应用程序也能够其他供应商的应用程序服务器上运行。

Java EE 应用程序也可能与 Java EE 规范相冲突。Java EE 规范极为复杂，开发人员也许终其一生也无法掌握其全部细节。可使用诸如 Rational Application Developer V8.5 等工具来验证应用程序的编写是否符合规范。

本章所述主题如下：

- ▶ J2EE 至 Java EE 迁移注意事项
- ▶ 运行时迁移问题
- ▶ 互操作性和集成

2.1 J2EE 至 Java EE 迁移注意事项

JAVA EE 会基于其首次发行版不断发展变化，还会以新的 Java EE 规范的形式定期定义其新功能。

J2EE 1.4 解决了之前版本中的很多问题。J2EE 1.4 平台的主要变化在于对 Web 服务的全面集成。J2EE 组件能够使用 Java API for XML Remote Procedure Calls (JAX-RPC) 和 SOAP with Attachments API for Java (SAAJ 1.1) 与 Web 服务进行交互并采用标准化方式加以实施。除 Web 服务外，EJB 2.1 的主要变化包括对 EJB-QL 的重要增强、新的计时器服务及消息驱动的 Bean (MDB) 的扩展，以支持各种类型的消息传递系统（不仅仅是 JMS）。

从 J2EE 到 Java EE 是个重大的转变。EJB 3.0 作为一个新的业务组件开发模型而被引入其中，JPA 作为持久性模块替代了旧版本的 EJB 2.x 实体 Bean 模型，JAX-WS 2.0 作为 SOAP Web 服务 API 取代了 JAX-RPC，同时将 JSF 作为标准表示层框架引入其中。

Java EE 5 重点在于使用注释、POJO 编程和零配置系统并简化 XML，从而降低复杂性。Java 语言注释能够简化 Java EE 应用程序的开发。使用注释能够使很多应用程序避免使用部署描述符。正是上述变化让 Java EE 5 取得成功。

Java EE 6 是转变的另一大步，其中涉及了一系列创新，诸如新的 API、上下文和依赖关系注入 (Contexts and Dependency Injection, CDI) 1.0、WebBeans 1.0、JAX-RS1.1 及 Java Servlet 3.0。Java EE 6 主旨在于裁剪和概要文件。Java EE 6 开始删除那些过时、支持不足或部署不够广泛的 API，为平台瘦身。小范围到中等范围的 Java Web 应用程序均无需完整的 Java EE 功能。例如，使用事务、持久性、消息传递及 Web 服务等功能但无需 JSF 或 JSP 等表示层的 SOA 应用程序。概要文件就是为了解决这一问题而设计的。Java API 的子集收集成概要文件，用以更好地组织日益复杂的 Web 服务标准。作为 WebSphere Application Server 8.5 一部分的 Liberty 概要文件正是基于这一理念而构建。

表 2-1 给出了有关模块版本的更多详情。

表 2-1 Java EE 版本号

规范 /API	J2EE	Java EE 5	Java EE 6
Web	2.2	2.5	3.0
EJB	1.1 您可以将 EJB 1.1 迁移至 EJB 3 或 EJB 3.1，其中假定没有 CMP 或 BMP bean（带有 CMP 或 BMP bean 的 EJB 1.x 和 2.x 项目可保留其在 Java EE EAR 中的状态）。	3.0	3.1
JDBC	3.0	4.0	4.1
JCA	1.0	1.5	1.6
JPA	不适用	2.0	2.0
JMS	1.1	1.1	1.1
Java Servlet	2.4	2.5	3.0
JSP	2.0	2.1	2.1
JSF	1.0	1.2	2.0

规范 /API	J2EE	Java EE 5	Java EE 6
CDI	不适用	不适用	1.0
JAX-RPC	1.1	1.1	1.1
JAX-WS	不适用	2.1	2.2
JAX-RS	不适用	不适用	1.1
Bean 验证	不适用	不适用	1.0
SIP	不适用	1.1	1.1
EAR	1.2	5.0	6.0
应用程序客户机	1.2	5.0	6.0

WebSphere Application Server V8.5 支持根据 Java EE 6 编写的应用程序，也支持为之前的 Java EE 版本而编写可移植应用程序，具体说来包括 Java EE 5、Java 2 Platform、Enterprise Edition (J2EE) 1.4 及 J2EE 1.3。

若您计划维护您的应用程序，可考虑将其迁移至 Java EE 6。通过这种方式，即能实现更新，也更便于实施下一 Java EE 规范中的变化。

有关 Java EE 6 新机会的更多信息，请参阅 *Experience JEE Using Rational Application Developer V7.5*, SG24-7827。

规范级迁移

重要提示：无论是采用手动方式还是使用 Migration Toolkit 来升级 Java SE 版本，都不会改变您的应用程序的 Java EE 规范要求。若您是将您的应用程序从其他 Java EE 应用程序服务器迁移到 WebSphere Application Server，建议采用的迁移方法为：首先执行平台迁移，确保应用程序在 WebSphere Application Server 上运行不会出现任何问题后，然后再执行 Java EE 规范的迁移。规范迁移应在平台迁移之后进行，因为 WebSphere Application Server 可与较早版本的 Java EE 兼容。

若您决定升级应用程序的 Java EE 级别，可使用 Rational Application Developer 中的 Java EE 规范升级向导。本节将介绍如何进行这一升级。您可以将 Migration Toolkit 用于 Java SE 级别迁移，但不能用于 Java EE 级别迁移。

Java EE 升级向导对于所有 Java EE 模块类型都支持从 J2EE 1.2、1.3 和 1.4 规范迁移至 Java EE 5.0 规范级别。该向导还对于所有 Java EE 模块类型都支持从 Java EE 5.0 迁移至 Java EE 6.0 规范级别。

使用 Rational Application Developer V8.5 中的 Java EE 规范升级向导，完成以下步骤：

1. 在 Enterprise Explorer 视图中，右键单击想要升级的项目，并选择 **Java EE** → **规范升级向导**。
2. 按照“Java EE 规范升级向导欢迎页面”中的提示操作，并单击**下一步**。
3. 在 J2EE 版本字段中，选择想要升级到的 Java EE 版本级别并单击**下一步**。
4. 选择想要升级的模块并单击**完成**（图 2-1）。

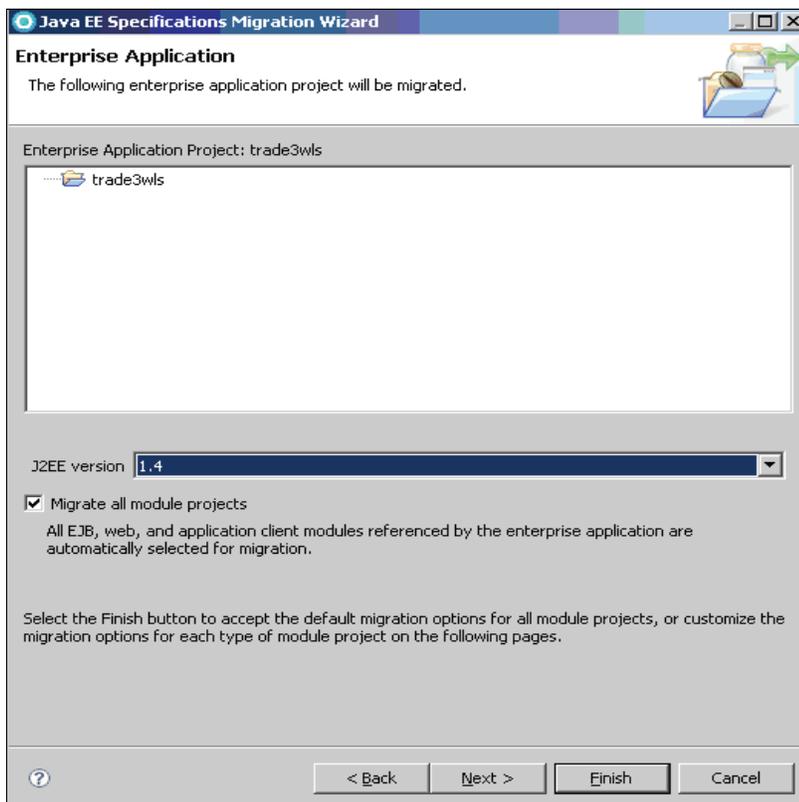


图 2-1 Java EE 规范迁移向导 - J2EE 版本选择

图 2-2 至第 21 页上的图 2-5 给出了由 Rational Application Developer 迁移过程自动完成修改。模块在迁移前后的规范级级别（图 2-2 和第 20 页上的图 2-3）以及在迁移前后应用程序包中的描述符文件（第 21 页上的图 2-4 和第 21 页上的图 2-5）在 Rational Application Developer V8.5 中如下所示。

Project Facet	Version
<input type="checkbox"/> Add WebSphere XDoclet support	6.1
<input checked="" type="checkbox"/> EJB Module	2.1 ▼
<input type="checkbox"/> EJBDodet (XDoclet)	1.2.3 ▼
<input checked="" type="checkbox"/> Java	1.4 ▼

图 2-2 增加项目面的版本级别 - 迁移前

Project Facet	Version
<input type="checkbox"/> Add WebSphere XDoclet support	6.1
<input checked="" type="checkbox"/> EJB Module	3.1 ▼
<input type="checkbox"/> EJBDodet (XDoclet)	1.2.3 ▼
<input type="checkbox"/> iWidgets	1.1
<input checked="" type="checkbox"/> Java	1.6 ▼

图 2-3 增加项目面的版本级别 - 迁移后



图 2-4 将 .xmi 文件转换为 .xml 文件 - 迁移前



图 2-5 将 .xmi 文件转换为 .xml 文件 - 迁移后

2.1.1 Java 消息服务（JMS）

Java 消息服务的消息传递是从其他供应商迁移至 **WebSphere Application Server V8.5** 过程中的常见问题之一。每个供应商实施规范的方式各不相同，因此如何在应用程序中使用消息传递有多种方法。

JMS 1.1 中的变化应保留与 **JMS 1.0.2b** 的兼容性并简化和统一编程 API。请检查您的旧版本应用程序，了解是否能够使用这些新的 **JMS 1.1** 功能。

大多数与 **JMS** 相关的问题旨在确保旧环境中可用的全部资源配置能够在 **WebSphere Application Server V8.5** 中得以设置。由于系统管理员可能接受过新环境维护和支持方面的培训，因此系统管理员应承担这一分派任务。有关运行时迁移及资源配置的更多信息，请参阅第 24 页上的 2.2 “运行时迁移问题”。

2.1.2 JavaServer Page

不同供应商的 **Java Server Page (JSP)** 可能不尽相同。**WebSphere Application Server V8.5** 在验证 **DTD** 和 **XML 模式 (schemas)** 方面存在更多限制，因此不允许出现任何偏差（如区分大小写的名称）。

与 **JSP** 页面迁移有关的另一个常见问题是，一些容器在包中具有隐含的类导入，诸如 `java.util`，因此需要使用导入语句更新 **JSP**。否则，迁移会简单得多。

Java EE 6 平台需要 **JSP 2.2**。

2.1.3 Servlet

Servlet 与 **Java EE** 实施的出现时间相同，因此在我们的迁移活动中并未出现任何问题。由于我们的迁移目的地是在代码可维护性方面有较大改进的新 **Java EE** 实施，因此您可能不希望丢失注释。

新注释（如 `@servlet`）的使用有助于避免团队在更新部署描述符时所面临的问题。若您不希望维护 **XDoclet**，很适合使用这种方法将其移除。

以下所述为 Java EE 6 平台中包含的新的 Java Servlet 技术功能：

- ▶ 注释支持
- ▶ 异步支持
- ▶ 配置容易
- ▶ 对现有 API 的增强
- ▶ 可插入性

Java EE 6 平台需要 Servlet 3.0。

2.1.4 JavaServer Faces

在我们的实验室进行 Java Server Faces 应用程序迁移时，除 Java EE 6 应用程序出现了 JSP 页面相关问题外，未出现其他任何问题。这种情况并不意味着 JSF 应用程序迁移不会出现任何问题。每个供应商针对 Java EE 6 实施 JSF 的方式均不相同。

在之前版本的 WebSphere Application Server 中以及对于其他供应商，JSF 应用程序必须将 JAR 导入 Web 模块中或将其置于类路径中，例如在 JBoss lib 目录中。有关如何处理这些问题的更多信息，请参阅本书中相应的特定于供应商的章节。

下面将为您介绍 Java EE 6 平台中所含 Java Server Faces 的新功能：

- ▶ 使用注释而非配置文件来指定受管理的 bean 和其他组件
- ▶ Facelet - 使用 XHTML 文件替代 JavaServer Pages (JSP) 技术的一种显示技术
- ▶ Ajax 支持
- ▶ 复合组件
- ▶ 隐含导航

Java EE 6 平台需要 JavaServer Faces 2.0 及 Expression Language 2.2。

2.1.5 Web 服务

Web 服务的迁移涉及到多个方面的专业知识和资源技能。在我们的迁移实践中，我们发现可以采用多种方法来进行 Web 服务迁移。Web 服务迁移变更主要包括三个主要方面：

- ▶ IDE 的开发环境及迁移
- ▶ Web 服务客户机
- ▶ Web 服务部署

在迁移 Web 服务应用程序时，由于可能需要更新对这些服务的引用（在 WSDL 文件中），因此所用的 URL 是最重要的注意事项之一。

您必须考虑各种方法，具体视您的环境而定：

- ▶ 第一个方法是更新并重新部署应用程序客户机（web 服务的使用者，在图 2-6 中用数字 1 表示）。在这种方法中，您必须重新编写应用程序客户机及指向新 WSDL 提供者的引用。这一过程可能涉及到很多软件开发和部署工作，具体视使用该服务的用户数量而定。若 Web 服务在互联网上发布，这种方法就不可行。

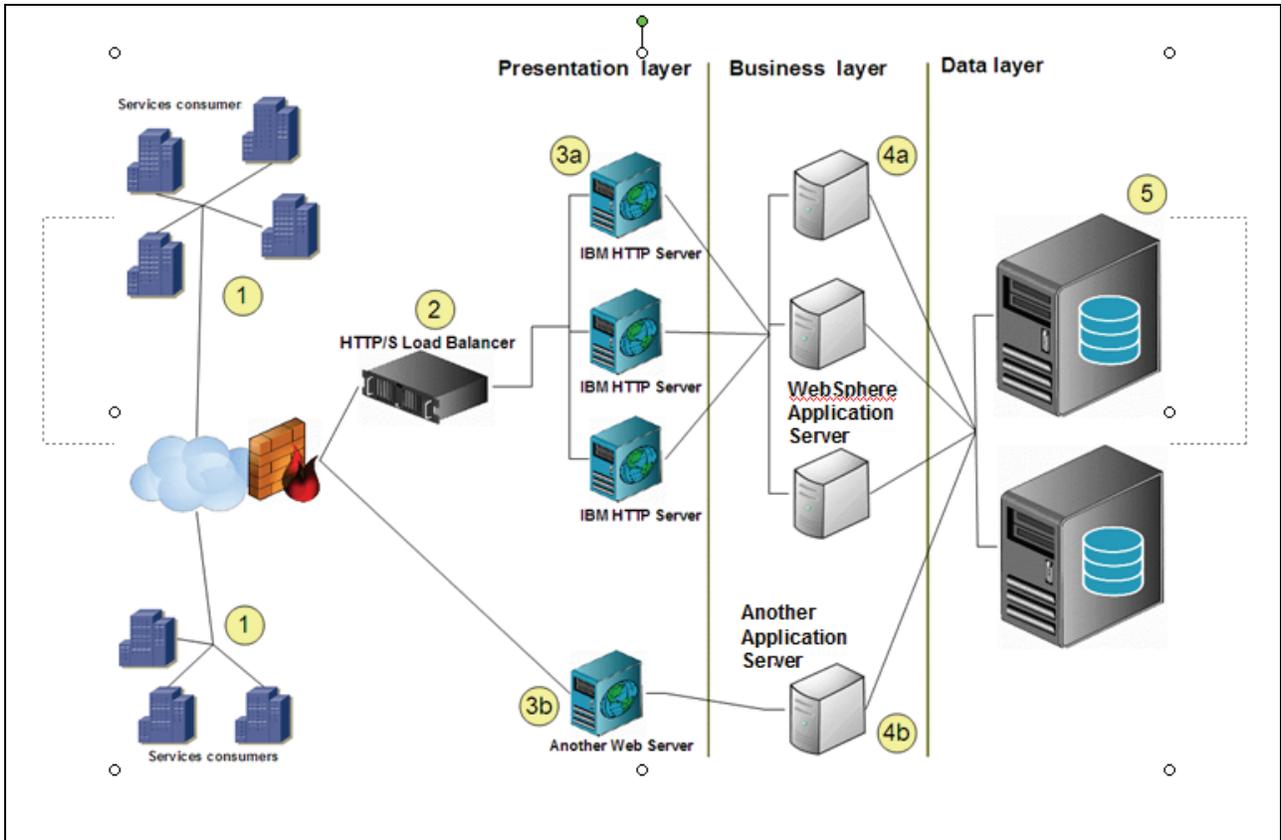


图 2-6 Web 服务应用程序的典型基础结构

- ▶ 若您的 Web 服务在一个三层基础结构上开发并安装，您也可以仅仅在前端进行更改，诸如负载均衡器、NAT 或反向代理设备（见图 2-6 中的数字 2）。在此情况下，您必须重新配置这些设备，使其指向 Web 服务的新 URL。
- ▶ 首先来看看体系结构中的第一层（见图 2-6 中的 3a 和 3b），您可能将 Web 服务器用作静态页面的前端设备，将另一个 Web 应用程序和代理用作应用程序服务器本身。举例来说，若您使用 IBM HTTP Server，那么您可以使用 HTTP 模块来用新的 URL 取代旧的 URL。有关更多信息，请访问 Apache HTTP Web 站点上的 Apache HTTP 模块（mod_redirect、mod_proxy、mod_rewrite 及 mod_alias）Web 页面，网址为：
<http://httpd.apache.org/docs/2.2/mod/>

- ▶ 由于您可能需要从一个应用程序服务器迁移至另一个应用程序服务器（第 23 页上的图 2-6 中的第 4 层），因此在 **Rational Application Developer** 上构建应用程序时，应考虑开发问题。若可能，应尽量避免更改 URL（服务器名称和服务器端口）及路径名称（上下文根和 **servlet** 映射）。若未更改这些项目，迁移过程就会较为容易，还能确保与其他客户端应用程序的连接。由于您可能需要迁移至其他服务器，甚至迁移至另一个基础结构，因此不可能保持相同的 URL。在这些情况下，您应考虑采用其他选项来重新映射 URL。

2.1.6 Java 持久性 API（JPA）

由于需要访问关系数据库，因此持久性对于企业应用程序而言至为关键。**Java 持久性 API (JPA)** 针对 **EJB 3.0** 和 **EJB 3.1** 规范提供了管理持久性及对象相关映射和功能的机制。**JPA** 使用注释或 **XML** 将对象映射到一个或多个数据库表，从而实现了对象相关映射这一重要任务的标准化。若要进一步简化持久性编程模型：

- ▶ **EntityManager API** 可用于针对数据库中的对象执行保留、更新、检索或删除操作。
- ▶ **EntityManager API** 及对象相关映射元数据能够处理大多数数据库操作，您无需编写 **JDBC** 或 **SQL** 代码即可维持持久性。
- ▶ **JPA** 提供了一种查询语言，这种语言是独立 **EJB** 查询语言的扩展（又称为 **JPQL**），您可以使用该语言检索对象，而无需编写特定于所用数据库的 **SQL** 查询。

WebSphere Application Server V8.5 中有两种内置的 **JPA** 持久性提供程序。

- ▶ **JPA for WebSphere Application Server**
- ▶ **Apache OpenJPA**

Apache OpenJPA 是与 **JPA** 规范兼容的实现。**WebSphere Application Server** 以 **OpenJPA** 作为其基础实施方法，使用扩展来为 **WebSphere Application Server** 客户提供更多功能和实用程序。因为 **JPA for WebSphere Application Server** 是从 **OpenJPA** 构建的，所以所有 **OpenJPA** 功能、扩展及配置均不会受到 **WebSphere Application Server** 扩展的影响。您无需更改 **OpenJPA** 应用程序即可在 **WebSphere Application Server** 中运行这些应用程序。若在持久性单元定义文件 (**persistence.xml**) 中未指定明确的提供程序元素，那么应用程序服务器将使用缺省持久性。除上述两个提供程序外，您也可以使用第三方的持久性提供程序。

在您决定哪种持久性实现更适合您的需要之后，您应在持久性单元定义文件中指定提供程序，以避免出现不兼容的情况。根据所用的提供程序，您可能需要为您的应用程序设置类装入器顺序，确保首先装入具有应用程序类装入器的类，在定义了第三方 **JPA** 提供程序尤其如此。

若要了解如何配置 **JPA** 持久性提供程序，请参阅以下主题：

http://www14.software.ibm.com/webapp/wsbroker/redirect?version=phil&product=was-nd-mp&topic=tejb_configjpa

2.2 运行时迁移问题

运行时迁移是迁移专业人员的专职工作。该迁移涉及多个方面、角色和职责。以下列表详细说明了与运行时迁移有关的各个团队：

- ▶ 支持团队：
 - 操作系统
 - **Web** 中间件
 - 数据库
- ▶ 测试团队
- ▶ 部署专业人员
- ▶ 基础结构架构设计师
- ▶ **DNS** 及网络团队

- ▶ 安全团队
- ▶ 项目经理
- ▶ 旧版系统
- ▶ 应用程序开发接口人
- ▶ 其他

以下所列为必须与应用程序迁移同步进行的全部活动：

- ▶ 安装
- ▶ 全部硬件和软件、操作系统、数据库产品等的配置

基础结构架构设计师必须为新环境设计相应的解决方案。这些架构设计师需就要实施的拓扑、与其他系统的接口、兼容性检查进行相应决策，例如操作系统级别与 **WebSphere Application Server V8.5** 或数据库的 **JDBC** 驱动程序、防火墙规则、**SOA** 基础结构及其他相关基础结构组件。

2.2.1 同时迁移其他产品

新环境的构建并不属于本书的范围，但我们还是列出了一些与之相关的注意事项。

通常，操作系统级别的迁移非常直接，诸如从 **IBM AIX® 5L™ V5.3** 或 **AIX V6.1** 迁移至 **AIX V7**，以满足 **WebSphere Application Server V8.5** 的先决条件。由于之前的环境可能使用的功能以及在新版本中可能不同的一些命令和参数，导致其他情况（例如迁移至 **UNIX** 实施）更具挑战性。问题较多的迁移是从 **Windows** 到 **UNIX** 的迁移，反之亦然；这两者完全是相互独立的环境，因此需要更改应用程序，重新进行培训以及更改支持运维自动化所用的脚本。

在迁移其他组件（例如数据库或消息传递产品）时，也会遇到同样的情况。在应用程序代码和资源配置中，必须使用新的驱动程序、发现新的功能以及更改不建议使用的功能；此外其他功能可能会填补之前环境中未涵盖的空白。

在从某个应用程序服务器迁移到另一个应用程序服务器时，均需对上述因素加以考虑。

2.2.2 资源定义

Java EE 容器提供了规范的实现，开发人员可使用这些 **Java EE** 容器实现与数据库、系统集成总线及 **Web** 服务的交互。此外，还提供了应用程序用户和组进行安全设置所用的附加配置、认证机制及提供程序、**SSL** 证书及管理角色和功能。以下章节给出了在应用程序服务器迁移项目中有关资源定义的推荐实践：

JMS 资源

在 **JMS** 资源迁移过程中的常见问题是各个应用程序服务器配置 **JMS** 资源的方式随供应商的不同而不同。

在将 **JMS** 定义迁移至 **WebSphere** 之前，您首先应决定使用何种消息传递引擎来实现异步消息交换。**WebSphere Application Server** 支持以下消息传递提供程序：

- ▶ **WebSphere Application Server** 的缺省消息传递提供程序（使用服务集成总线作为提供程序）。
- ▶ **WebSphere MQ** 消息传递提供程序（使用您的 **WebSphere MQ** 系统作为提供程序）。
- ▶ 实施 **JCA V1.5 - 1.6** 资源适配器或 **JMS V1.1** 规范 **ASF** 组件的第三方消息传递提供程序。

您的应用程序可使用来自任一上述 **JMS** 提供程序的消息传递资源。在 **WebSphere Application Server V8.5** 中，在您定义或创建 **JMS** 资源（例如队列、主题和连接工厂）之前，您可以先创建服务集成总线。**WebSphere Application Server V8.5** 中的服务集成总线是缺省消息传递提供程序使用的后端。

侦听器端口： 缺省消息传递提供程序 (SIB) 不使用侦听器端口。少数通过侦听器端口连接至 SIB 的应用程序可能会使用较早版本的 WebSphere MQ 类型配置，这些配置主要用于支持通过 MQClientLink 连接的 WebSphere Application Server 类型 5 应用程序 (V5 缺省消息传递提供程序)。使用 SIB 的 MDB 使用激活规范来实现连接，而连接至 WebSphere MQ 消息传递提供程序的 MDB 则可通过侦听器端口或激活规范来实现连接。从 WebSphere Application Server V8.5 起，将会除去 V5 缺省消息传递提供程序。

有关消息传递的更多信息（包括教程及示例），请访问信息中心，网址为：

http://www14.software.ibm.com/webapp/wsbroker/redirect?version=phil&product=was-nd-mp&topic=welc6tech_msg

JDBC 资源

重新配置 JDBC 资源可能是进行应用程序服务器之间迁移时最为常见的步骤。一般来说，从应用程序服务器进行的迁移可能还会需要迁移数据库产品或版本，这就需要更新 JDBC 驱动程序，因此请务必使用适当的 JDBC 驱动程序，以避免新数据库的新实施出现问题。在沿用以前数据库时，亦应确保使用正确的驱动程序。

数据库供应商或应用程序服务器供应商均可提供 JDBC 驱动程序。应使用由数据库供应商提供的驱动程序。除非未使用非标准扩展或特定于驱动程序的属性，否则驱动程序更改后对应用程序的影响通常会非常明显。若应用程序根据标准 JDBC (4.0) 编写，规范就会正确运行，且迁移所需投入也最少。

在 WebSphere Application Server V8.5 中，在设置数据源时，可供执行的配置有以下三种：

- ▶ JDBC 驱动程序，通常适用于大多数数据库
- ▶ J2C 认证数据，需配置用户和密码方可连接数据库
- ▶ 数据源本身

有关配置 JDBC 资源的更多信息，请参阅：

http://www14.software.ibm.com/webapp/wsbroker/redirect?version=phil&product=was-nd-mp&topic=welc6tech_dat

安全资源

针对已迁移应用程序及管理功能和角色的安全资源配置涉及到多个方面，诸如认证机制、授权机制及 Web 服务安全。

由于该方面的迁移较为敏感，可能涉及到必须采用的安全标准（诸如 ITC 104（IBM 内部安全标准 – 信息技术安全标准）），因此应查看提供的相应参考材料。本书中特定于产品的迁移章节给出了一些基本配置。

有关 WebSphere Application Server V8 中安全配置和资源的更多信息，请参阅：

<http://www14.software.ibm.com/webapp/wsbroker/redirect?version=phil&product=was-nd-mp&topic=welc6topsecuring>

另请参阅 *WebSphere Application Server V7.0 Security Guide*, SG24-7660。

JVM 自变量、类路径及系统属性

由于系统管理员的配置管理方式各不相同，因此运行时迁移也会视情况的不同而不同。一些管理员使用启动脚本命令，一些管理员在过程用户 ID 概要文件中设置环境变量，而另一些管理员则使用特定于应用程序服务器的配置或行为。

下文就如何避免迁移时出现问题给出了一些提示：

- ▶ 验证启动脚本并检查 JVM 自变量。
- ▶ 在 UNIX 机器上，检查过程 ID 概要文件中的任何相关环境配置。
- ▶ 在 Windows 机器上，所用的用户 ID 应在启用用户帐户控制 (UAC) 的情况下在 Windows Vista、Windows 7 或 Windows Server 2008 操作系统中的 Program Files 或 Program Files (x86) 目录中安装 WebSphere Application Server V8 时所使用的非管理员用户 ID。否则，WebSphere Application Server 将不能正常运行。
- ▶ 打开不同厂商服务器特定的配置文件（诸如 JBoss 中的 server.xml 及 Weblogic 中的 config.xml）以查看更多配置。
- ▶ 了解旧版应用程序服务器如何实施类路径。
- ▶ 确保已了解哪些 JAR 扩展了您的旧版应用程序服务器库。

有关过程定义配置的更多信息，请参阅 *WebSphere Application Server V8: Administration and Configuration Guide*, SG24-7971。

2.2.3 开发环境问题

编译、打包和部署应用程序所用的开发环境和技术有多种组合，具体视情况而定。我们总结了一些常用方式，诸如使用特定于供应商的 IDE (Rational Application Developer V8.5)、使用带有不同种类插件的 Eclipse 及 Maven、Apache Ant、XDoclet 及其他 IDE 的组合。

由于 WebSphere Application Server V8.5 完全与 Java EE 规范相兼容，因此并不依赖于任何集成开发环境 (IDE)。在我们的迁移实践中，我们采用两种方法：

- ▶ 保留现有的开发环境并手动调整配置文件及 Apache Ant 构建文件，并在适用时手动替换 XDoclet 标记。根据应用程序代码和部署描述符所用的 Java EE 组件、基于之前应用程序服务器的扩展以及特定配置，这些变化的复杂性可能会由简单到相当耗时。
- ▶ 将应用程序导入 Rational Application Developer V8.5 中并使用 WebSphere 版本到版本迁移的 Migration Toolkit。Rational Application Developer V8.5 有助于应用程序服务器、Java EE 规范的迁移（诸如 J2EE 1.4 到 Java EE 6），也有助于解决老版本 WebSphere 应用服务器到 WebSphere Application Server V8.5 的迁移过程中出现的问题。

2.3 互操作性和集成

我们可能会遇到导致迁移较为复杂的外部因素，因此相关各方均应予以注意。这些外部因素包括与现有系统的集成及取决于其他应用程序功能或服务的已迁移应用程序之间的互操作性。

互操作性是指应用程序、系统或产品与另外的应用程序、系统或产品以可以共存且无需进行任何特殊更改的方式进行交互的能力。举例来说，您所迁移的应用程序可通过 XML 或 Web 服务与外部 CRM 或 ERP 系统进行交互。在这种情况下，您必须了解应用程序与这些外部应用程序交互的方式并确定是否需要更改外部应用程序的接口。

再举例来说，若您在同一网络中同时拥有 IBM WebSphere Application Server V8.5 和 Oracle WebLogic Vx 且两者之间没有防火墙，同时您决定将 Oracle WebLogic 应用程序迁移至 WebSphere Application Server V8.5。该应用程序与由 Oracle WebLogic Server 托管的另一个应用程序中所部署的 EJB 有依赖关系。在您将第一个应用程序迁移至 WebSphere Application Server V8.5 时，就可能出现这样的情况，即要两者配合工作，要求不要进行更改或只进行极少量更改。不过，应切记迁移过程中服务器之间的依赖关系可能会带来问题，因为 J2EE 规范规定这些规范的实施由个别供应商负责。

使用这个方法时，也存在出现问题的情况（例如，选择为 EJB 等其他应用程序提供组件或服务的样板应用程序迁移）。

对于迁移的复杂性讨论，若存在与现有系统集成，则需要投入一定量的时间来确定如何进行迁移。即便某些标准、工具及久经考验的体系结构能够将复杂性降至最低，但集成对于中间件而言仍旧较为敏感。在管理或参与迁移项目时，切记集成并非更改配置或几行代码那样简单。集成需要对标准和兼容性进行修改。在大多数情况下，团队应将其视作单独的项目考虑。



Application Migration Tool 的安装和配置

本章介绍了 IBM WebSphere Application Server Migration Toolkit (Migration Toolkit) 中的工具的安装和配置步骤，这些工具可指导您将您的应用程序迁移至 IBM WebSphere Application Server V8.5。我们还介绍了使用 Eclipse 和 Rational Application Developer V8.5 需完成的相关步骤。为确保您在这两种环境中做出正确选择，我们还解释了这两者之间的区别。

重要提示：本章不包括 WebSphere Application Server V8.5 的安装方法。有关如何安装 WebSphere Application Server V8.5 的信息，请转至 WebSphere Application Server V8.5 信息中心，并搜索“安装和配置应用程序服务器环境”：

<http://publib.boulder.ibm.com/infocenter/iisinfsv/v8r5/index.jsp>

本章所述主题如下：

- ▶ IBM WebSphere Application Server Migration Toolkit 概述
- ▶ 在 Eclipse 上安装 Application Migration Tool
- ▶ 在 Rational Application Developer V8.5 上安装 Application Migration Tool
- ▶ 命令行安装
- ▶ 配置 Application Migration Tool
- ▶ 故障诊断

3.1 IBM WebSphere Application Server Migration Toolkit 概述

由于不同的平台是根据不同的理念和概念开发的，因此从一个平台到另一个平台的应用程序迁移通常需要花费很多时间和精力。甚至 Java 本身在其运行的不同应用程序服务器中的修改和使用都有所不同。

在您转换 Java EE 应用程序使其在另一应用程序服务器（非原设计使用的应用程序服务器（图 3-1））中运行时，必须遵循一系列步骤。其中包括迁移的初步评估和规划、Java 类、XML 文件、部署描述符、JavaServer Pages (JSP) 文件的修改等等。由此可见，迁移过程是一项风险大、成本高的操作。

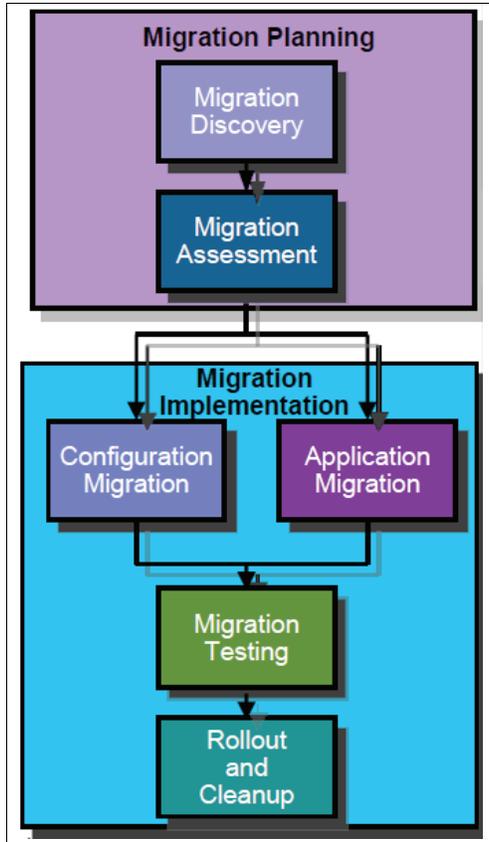


图 3-1 应用程序迁移的规划和实施步骤

如图 3-2 中所示，应用程序的更改和测试占整个迁移过程的 42%。

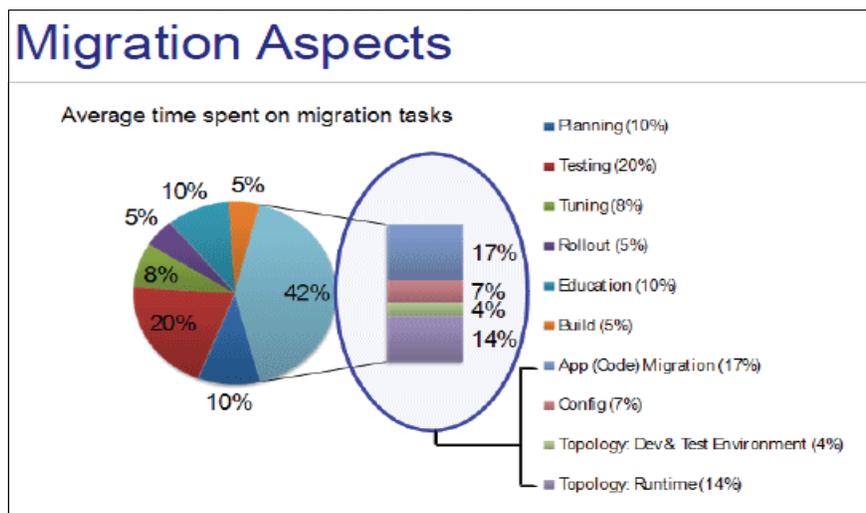


图 3-2 迁移相关问题

Migration Toolkit 是一整套用于 Eclipse 和 Rational Application Developer 产品的插件，可降低应用程序迁移过程的成本，缩短迁移时间。其中包括数个可用于从不同的应用程序服务器迁移应用程序的迁移工具。Migration Toolkit 包括 Application Migration Tool - WebSphere Version to Version (请参阅第 52 页上的 4.1 “Application Migration Tool - WebSphere Version to Version 概述”)。

这些工具以 IBM Rational Software Analyzer 为基础。IBM Rational Software Analyzer 具有代码扫描功能，可帮助识别应用程序迁移时必须检查并可能需要更改的代码构造。

Java 应用程序的转换过程可能涉及修改 Java 源代码、JSP 文件、部署描述符和 XML 文件以及类路径信息，这是一个成本高、耗时长过程。这些工具还包含 Java SE 版的迁移规则，其中识别了更改 Java 运行时版本时可能发生的迁移问题。此外，还有关于 Java 编码最佳实践的代码改善附加规则。

总之，Migration Toolkit 简化了迁移过程，缩短了开发者实现价值的时间，是将应用程序迁移至 WebSphere Application Server 的有用工具。

提示：IBM Education Assistant 包括一个无需附加费用的 WebSphere Application Server Migration Toolkit 多媒体教育模块。有关更多信息，请参阅：

<http://publib.boulder.ibm.com/infocenter/ieduasst/v1r1m0/topic/com.ibm.iea.wasmt/wasmt/3.5/0verview.html>

3.1.1 Migration Toolkit 基础知识

Migration Toolkit 的应用程序迁移工具有预定义的规则和规则集，各自特定于工具的目标应用程序服务器（图 3-3）。创建应用程序分析配置后，运行分析开始根据这些规则扫描应用程序，找出与这些规则匹配的应用程序部分。工具将列出应用程序迁移所需的更改。很多规则具有自动修正功能，这就意味着迁移工具可自动修正问题。对于工具不能修正的规则，工具将会提供详细的帮助，说明需实施的必需更改。

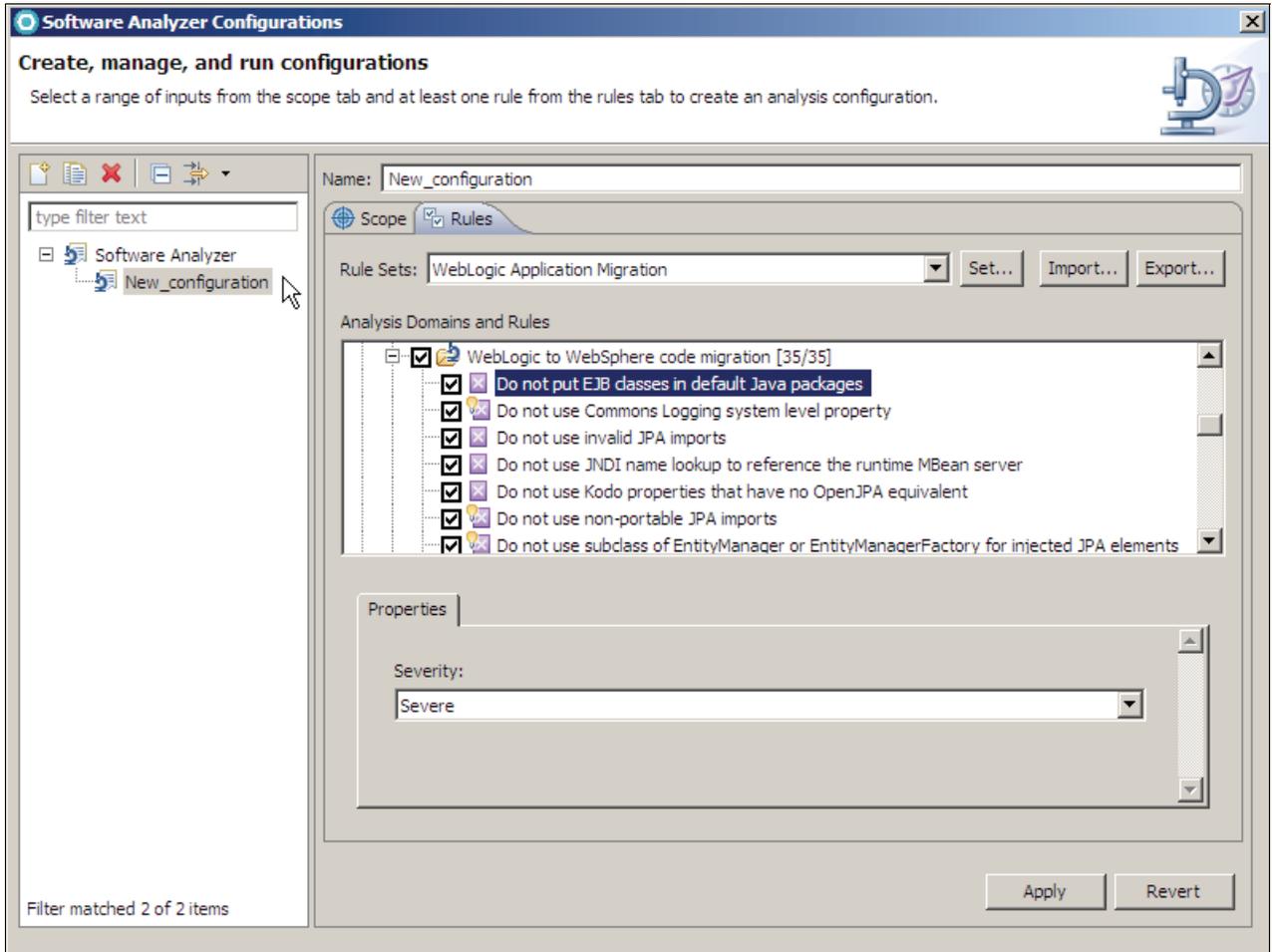


图 3-3 从 WebLogic 迁移至 WebSphere 的示例配置（WebSphere 版本到版本的迁移界面类似）

进行更改后，在 Enterprise Explorer 中的应用程序名称后应没有红色的 X（图 3-4）标识，表示不存在妨碍应用程序在目标应用程序服务器中运行的错误。

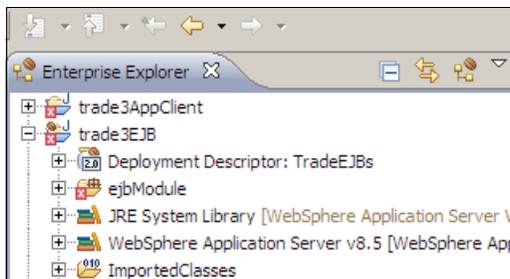


图 3-4 模块有红色的 X 标识时，说明存在一个或多个错误。

在图 3-5 中列出了将竞争性应用程序服务器应用程序（或之前版本的 WebSphere Application Server）迁移至 WebSphere Application Server V8.5 应用程序的工作流程。

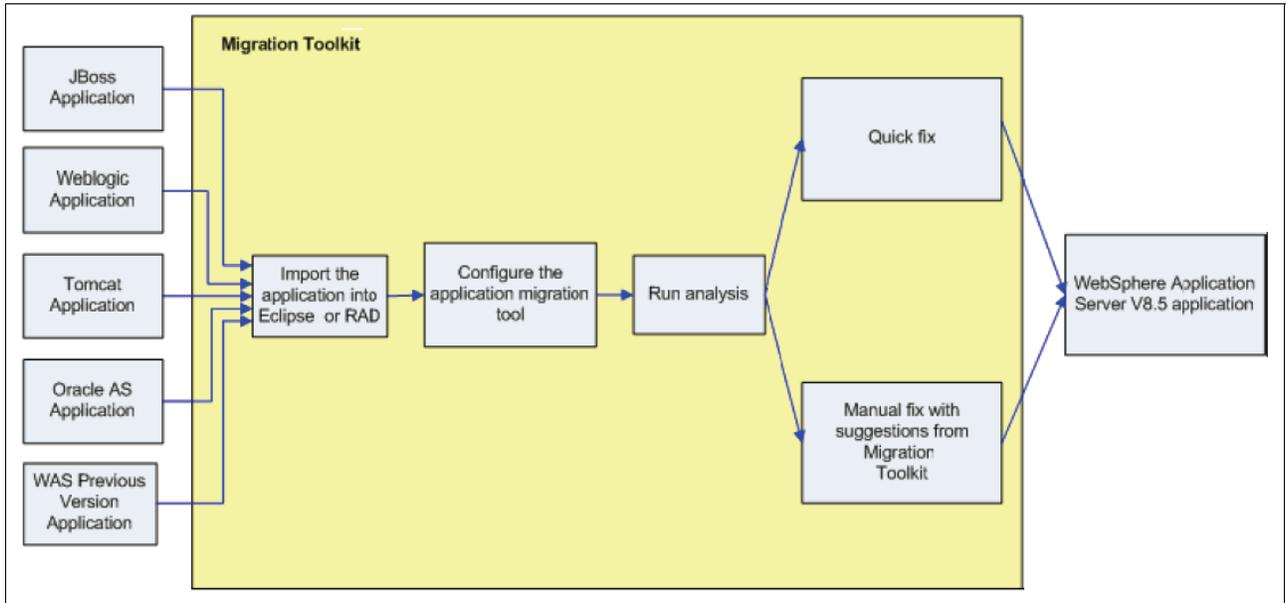


图 3-5 Migration Toolkit 的高级工作流程

Migration Toolkit 基于以下概念：

- ▶ **规则**，指可能妨碍应用程序在目标应用程序服务器中运行的已知问题。Migration Toolkit 使用这些规则扫描应用程序。
- ▶ **规则集**，指组合在一起来完成具体任务的一组规则。例如，各类竞争性应用程序服务器迁移和诸如 Java 最佳实践的违例识别等一般任务都使用规则集。这些规则集可在不同的开发环境间导入、导出。
- ▶ **代码复查**，指按所扫描的文档类型组织的规则，例如 Java、JSP、XML 和 Manifest (.MF) 文件。
- ▶ **快速修正**，指用于修正应用程序中的迁移问题的措施。您可以选择快速修正所有问题、单个问题或单个类别中的所有问题。

您可以使用 Migration Toolkit 创建多个配置，以避免为不同应用程序服务器或迁移战略制定迁移分析计划时多次删除然后重新创建配置。

Migration Toolkit 也可为您生成报告，以便更好地估算应用程序的迁移范围。报告将汇总过去的分析记录，并提供应用程序代码质量、复杂性和结构的高级输出。单击“Software Analyzer 结果”窗口右侧窗格中的图标（图 3-6）运行分析后，即可生成分析报告。您还可生成历史扫描报告。

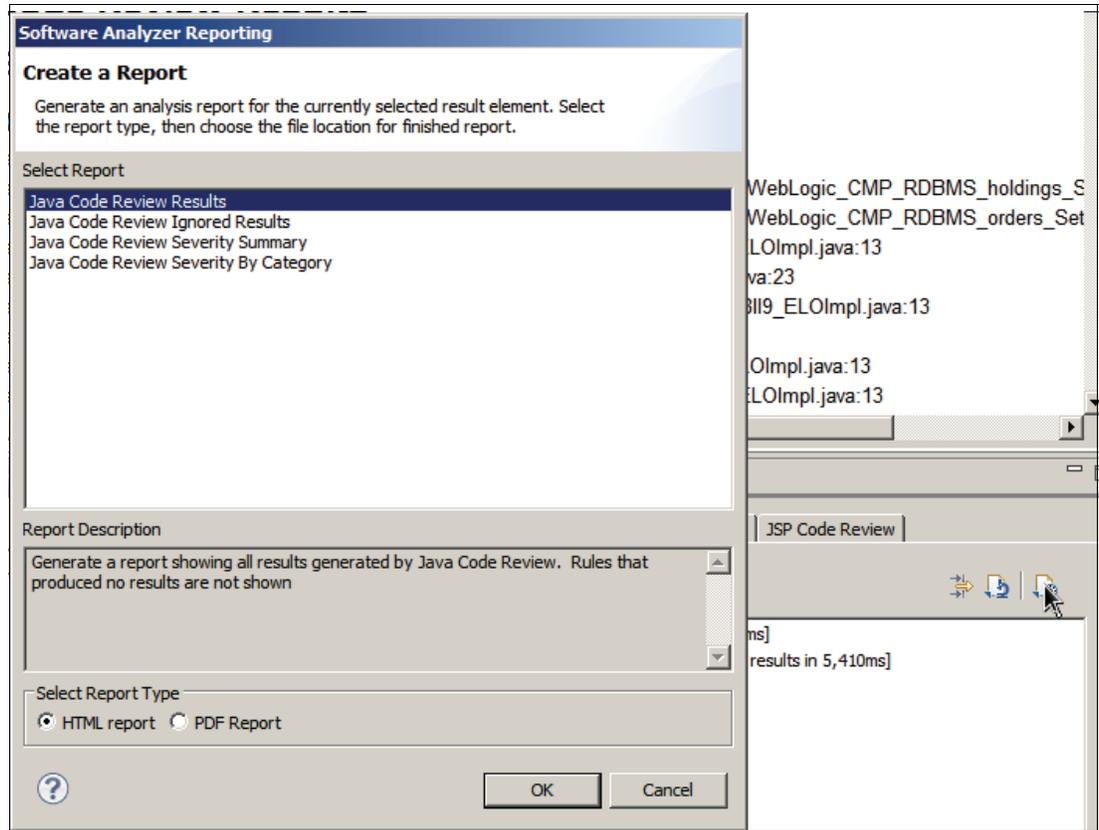


图 3-6 用于创建分析报告的“Software Analyzer 报告”窗格

对于 Java 代码复查，您可以创建 PDF 或 HTML 格式的分析报告。图 3-7 所示为 HTML 报告示例。对于 Java、XML、JSP 和类路径代码复查，您可以采用 XML 格式导出分析记录。

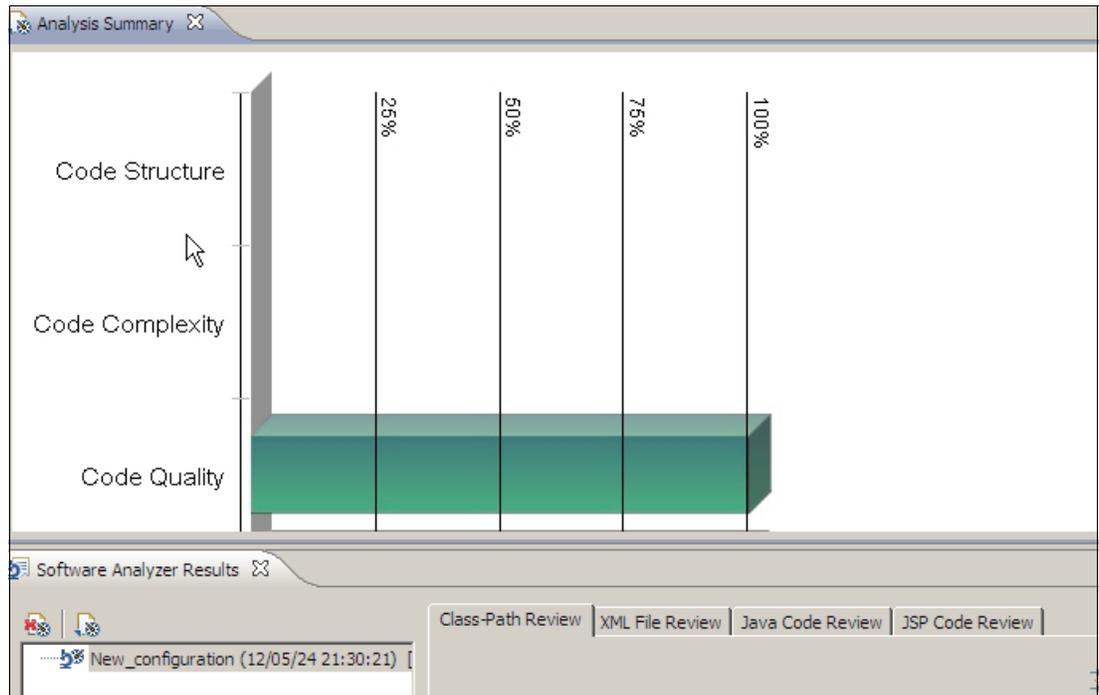


图 3-7 HTML 报告示例

在编写本书时，Migration Toolkit 的最新版本为 V3.5，目前支持以下所列版本间的迁移操作：

- ▶ 将应用程序迁移至 WebSphere Application Server V7、V8 或 V8.5
- ▶ 将应用程序迁移至较新版本的 Java SE 运行时，从 Java SE 1.4、5 或 6 迁移至 Java SE 6 或 7

Migration Toolkit 免费面向公众提供，同时提供详细的文档，以便于将 Java EE 应用程序从各种应用程序服务器迁移至 WebSphere Application Server。同时还为 WebSphere Application Server 客户提供了 Migration Toolkit 支持。

总之，Migration Toolkit 所提供的基于知识的迁移协助能够提升迁移的可靠性。如此一来，在您迁移应用程序时，您不必费力地处理文档及源代码，也不必进行试运行和错误测试来确定薄弱点。

获取帮助

在使用 Migration Toolkit 时，您也可以使用上下文相关功能。当您使用 Eclipse 或 Rational Application Developer 时，按下 F1 即可查看有关当前问题的帮助主题（图 3-8）。您也可以单击帮助 → 动态帮助来查看帮助内容。

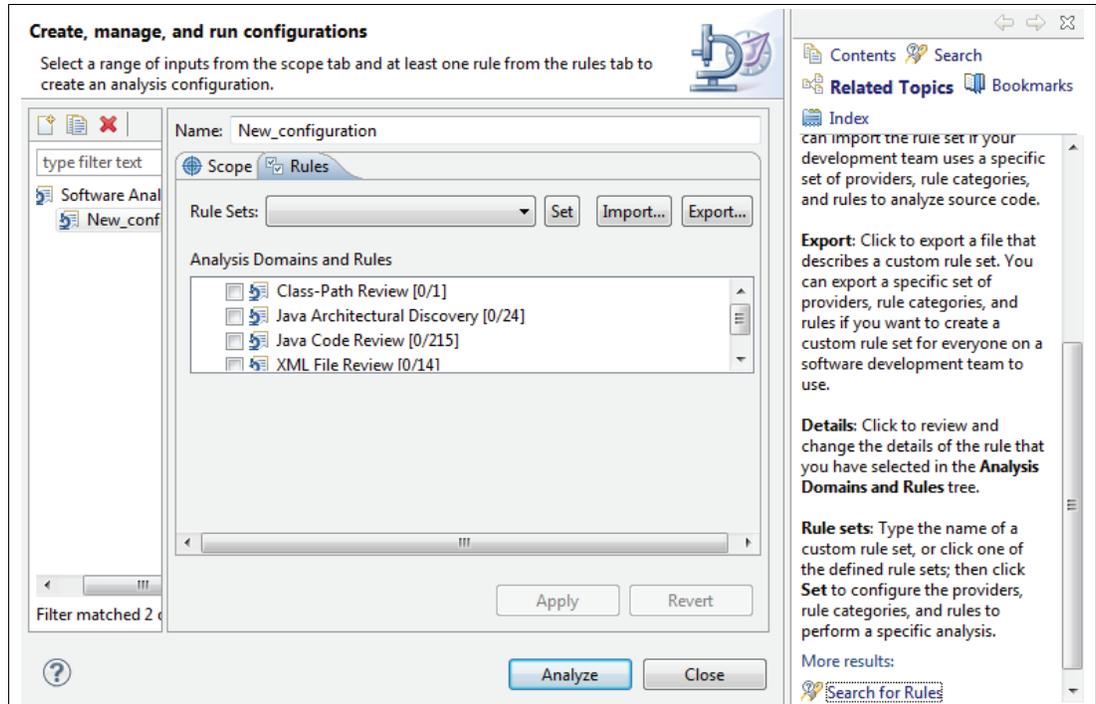


图 3-8 帮助查看

若要获取有关 Migration Toolkit 的在线信息，请访问：

<http://www.ibm.com/developerworks/websphere/downloads/migtoolkit/index.html>

同时还提供有 Migration Toolkit 论坛，以供用户输入和回答问题：

<http://www.ibm.com/developerworks/forums/forum.jsps?forumID=2106>

3.2 在 Eclipse 上安装 Application Migration Tool

各个 Application Migration Tool 均可作为 Eclipse 功能进行安装。接下来我们将介绍如何在 Eclipse 环境中安装 Application Migration Tool。

提示： 在安装 V3.5 之前，请先卸载 1.x、2.0 或 3.5 Beta 版本的 Migration Tool。2.1、3.0 和 3.1 等版本可直接升级至 V3.5。

首先，您应下载适用于待迁移应用程序的 Application Migration Tool。我们以 WebLogic to WebSphere Application Migration Tool 为例。有关分析阶段的更多信息，请参阅本书中的相应章节。有关各个应用程序服务器的迁移步骤，请参阅各自章节中的详细介绍。

您可以通过完成以下步骤下载并安装 Application Migration Tool：

1. 下载适用于待迁移应用程序服务器的 Application Migration Tool。我们以 Application Migration Tool - WebLogic to WebSphere 为例。Application Migration Tool 和相关文档可通过以下网址下载：

<http://www.ibm.com/developerworks/websphere/downloads/migtoolkit/compmig.html>

2. 在菜单栏中，单击**帮助** → **安装新软件**。
3. 在“安装”窗口中，单击**添加**。
4. 在“添加站点”窗口中，输入以下信息（图 9-6）：
 - 名称：输入工具的相应名称，例如， Application Migration Tool - WebLogic to WebSphere
 - 位置：输入所下载压缩文件的位置，例如：
C:\MigrationToolkit\Application_Migration_Tool_WebLogic_to_WebSphere_v3.5.0.zip。

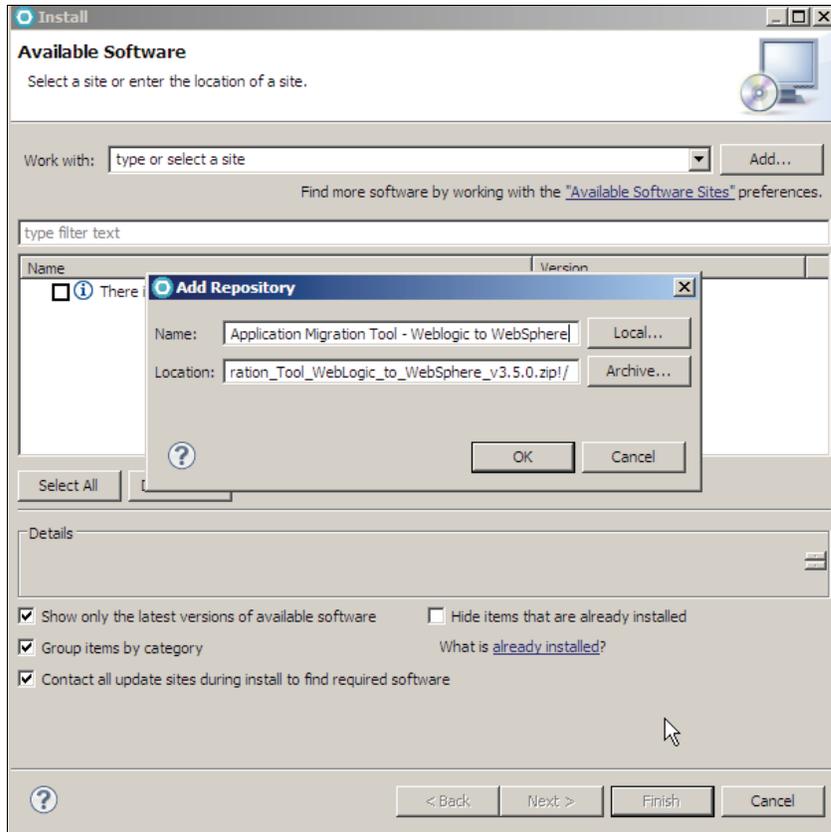


图 3-9 “添加存储库”窗口（用于将 Application Migration Tool 添加为存储库）

5. 在“安装”窗口中，选择您想要安装的插件（图 9-7）。单击下一步。

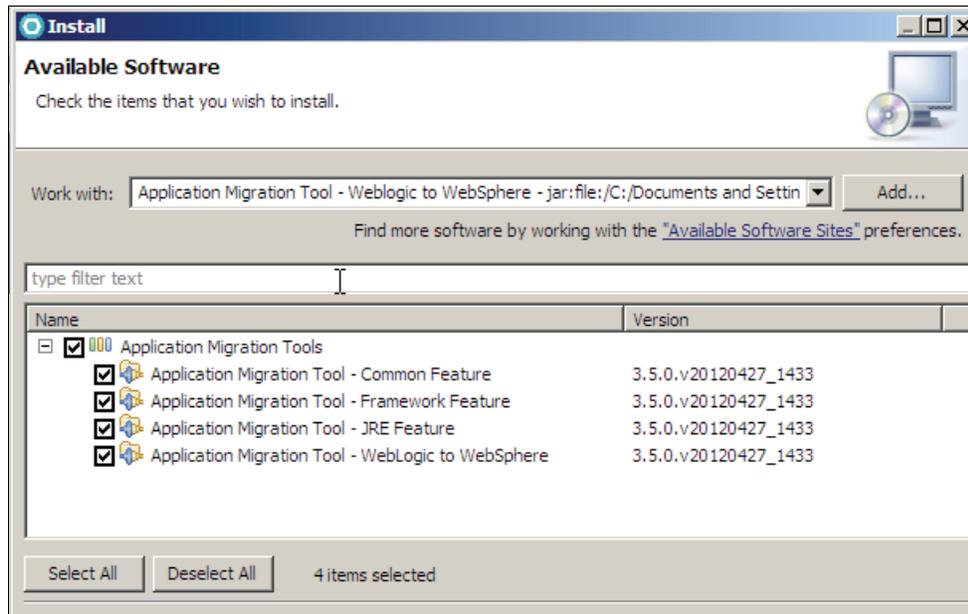


图 3-10 在“安装”窗口中选择插件

6. 接受许可协议并单击**完成**，即可开始安装。

7. 安装完成后，单击**立即重新启动**，以重新启动 Eclipse。

此时，**Application Migration Tools** 安装完毕，可供使用。您可以单击**运行 → 分析**并创建相应的运行配置，从而对您的应用程序进行分析。您可以在与要从中迁移的应用程序服务器相关的章节中查看有关分析阶段的详细信息。

3.3 在 Rational Application Developer V8.5 上安装 Application Migration Tool

本节介绍了如何在 Rational Application Developer V8.5 上安装 Application Migration Tool。请完成以下步骤：

1. 下载适用于要从中迁移的应用程序服务器的 Application Migration Tool。我们以 Application Migration Tool - WebLogic to WebSphere 为例。Application Migration Tool 和相关文档可通过以下网址下载：
<http://www.ibm.com/developerworks/websphere/downloads/migtoolkit/compmig.html>
2. Rational Application Developer 应可在 WebSphere Application Server V8.5 安装后自动在本机上发现它（图 3-11）。请检查信息是否准确无误，然后单击**完成**。可以在**服务器**选项卡中找到您创建的服务器。

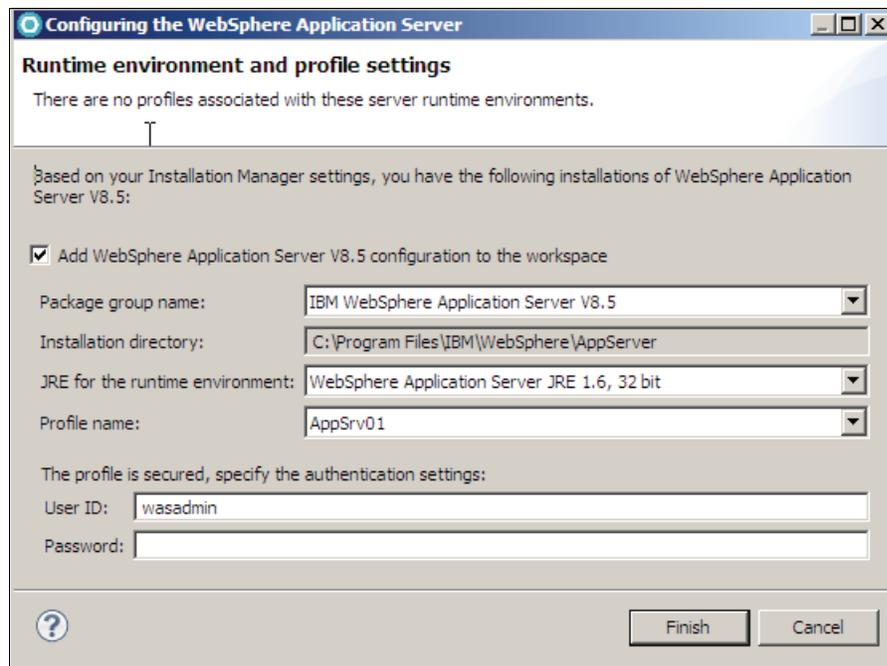


图 3-11 应用程序服务器设置

3. 对于剩余安装过程，您可以按照第 37 页上的 3.2 “在 Eclipse 上安装 Application Migration Tool” 中所述的步骤执行，从第 2 页上的第 37 步骤开始。

安装 Application Migration Tool: 除了发现步骤（第 40 页上的 3.3 “在 Rational Application Developer V8.5 上安装 Application Migration Tool” 中的步骤 2）之外，在 Eclipse 和 Rational Application Developer 上安装 Application Migration Tool 的步骤相同。

3.4 命令行安装

Application Migration Tool 也可使用命令行安装。首先应下载适当的 Application Migration Tool，然后运行示例 3-1 中的命令。

示例 3-1 通过命令行安装 Migration Toolkit

```
<installLocation>\eclipse.exe  
-application org.eclipse.equinox.p2.director -metadataRepository  
jar:file:C:/<amtDownloadDir>/Application_Migration_Tool_Weblogic_to_WebSphere_vn.n  
.n.zip!/ -artifactRepository  
jar:file:C:/<amtDownloadDir>/Application_Migration_Tool_Weblogic_to_WebSphere_vn.n  
.n.zip!/ -installIU com.ibm.ws.appconversion_feature.weblogic.feature.group  
-nosplash
```

将存储库自变量中的 n.n.n 替换为版本号，例如 3.5.0。metadataRepository 和 artifactRepository 的自变量以 jar:file: 开头并以 !/ 结尾。installIU 的自变量以附加的 .feature.group 结尾。

在所举示例中，我们以其中一个 Application Migration Tool - WebLogic to WebSphere 迁移为例。若选择了其他 Application Migration Tool，则可以从表 3-1 中选择适当的 installIU 类型。

表 3-1 Application Migration Tool 与 installIU

Application Migration Tool	installIU
JBossAS	com.ibm.ws.appconversion_feature.jboss. feature.group
Oracle AS	com.ibm.ws.appconversion_feature.oracle. feature.group
WebLogic	com.ibm.ws.appconversion_feature.weblogic. feature.group
WebSphere Version to Version	com.ibm.ws.appconversion_feature.was2was. feature.group
Apache Tomcat	com.ibm.ws.appconversion_feature.tomcat. feature.group

Application Migration Tool 可使用相应的 installIU 参数予以卸载，卸载方法与安装相同（示例 3-2）。

示例 3-2 通过命令行卸载 Application Migration Tool

```
<installLocation>\eclipse.exe  
-application org.eclipse.equinox.p2.director  
-uninstallIU com.ibm.ws.appconversion_feature.weblogic.feature.group  
-nosplash
```

有关 Eclipse p2 director 应用程序的更多信息，请参阅：

http://help.eclipse.org/indigo/index.jsp?topic=/org.eclipse.platform.doc.isv/guide/p2_director.html

3.5 配置 Application Migration Tool

任何 Application Migration Tool 在 Eclipse 或 Rational Application Developer 上安装完成后，可使用新的分析选项来配置和运行分析。分析范围可配置为项目、工作集或整个工作空间等等。可对该工具进行配置，以定义一系列要运行的规则（即妨碍应用程序在目标应用程序服务器上运行的已知问题）。

3.5.1 运行分析

下面一节介绍了使用 Application Migration Tool 运行分析的几种不同方式。

工具示例：在本节的示例中，我们以 Application Migration Tool - WebLogic to WebSphere 为例，但这些示例同样适用于其他 Application Migration Tool。

若要从 Eclipse 菜单各种启动分析，请单击 Eclipse 菜单中的 **运行 → 分析**（图 3-12）。

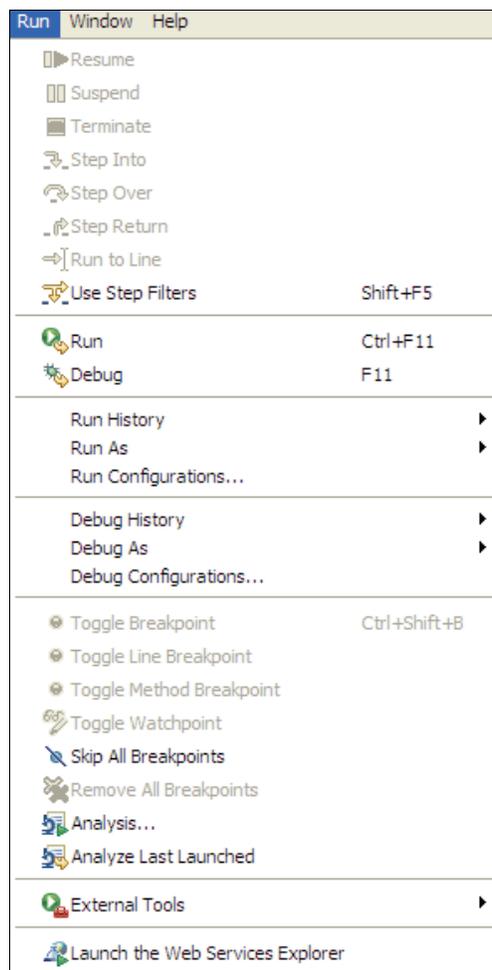


图 3-12 分析选项

若要从“启动”工具栏中启动分析，请找到图 3-13 中所示的图标并单击它。

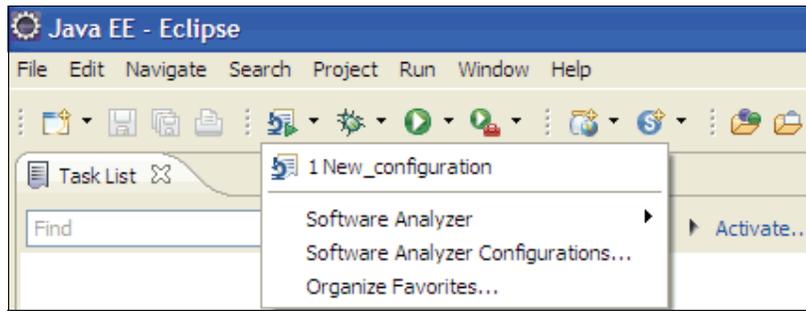


图 3-13 Software Analyzer

若要从 Eclipse 中的 Enterprise Explorer 中启动分析，请右键单击应分析的项目（图 3-14）。

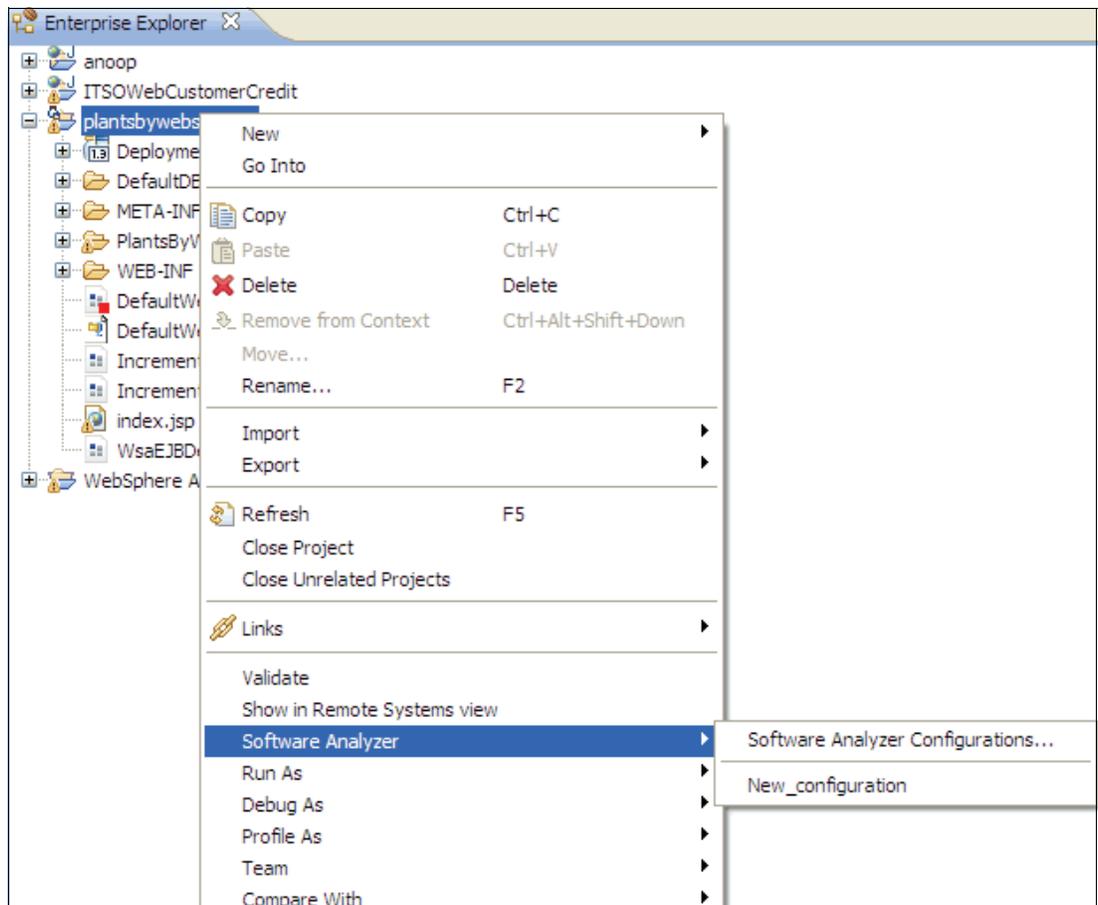


图 3-14 Software Analyzer

3.5.2 配置和运行分析

在按照第 42 页上的 3.5.1 “运行分析” 中所述的任一种方法启动分析后，完成以下步骤来配置新分析：

1. 右键单击项目并选择 **Software Analyzer**。
2. 选择 **Software Analyzer 配置 ...**。
3. 右键单击左侧的 **Software Analyzer** 并选择**新建**（图 3-15）。

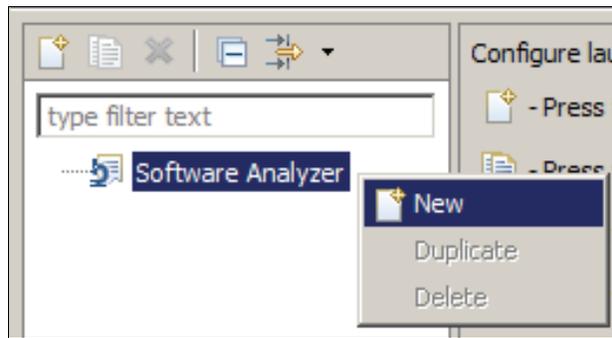


图 3-15 创建 Software Analyzer 配置

此时，即可看到“Software Analyzer 配置”窗口（图 3-16）。

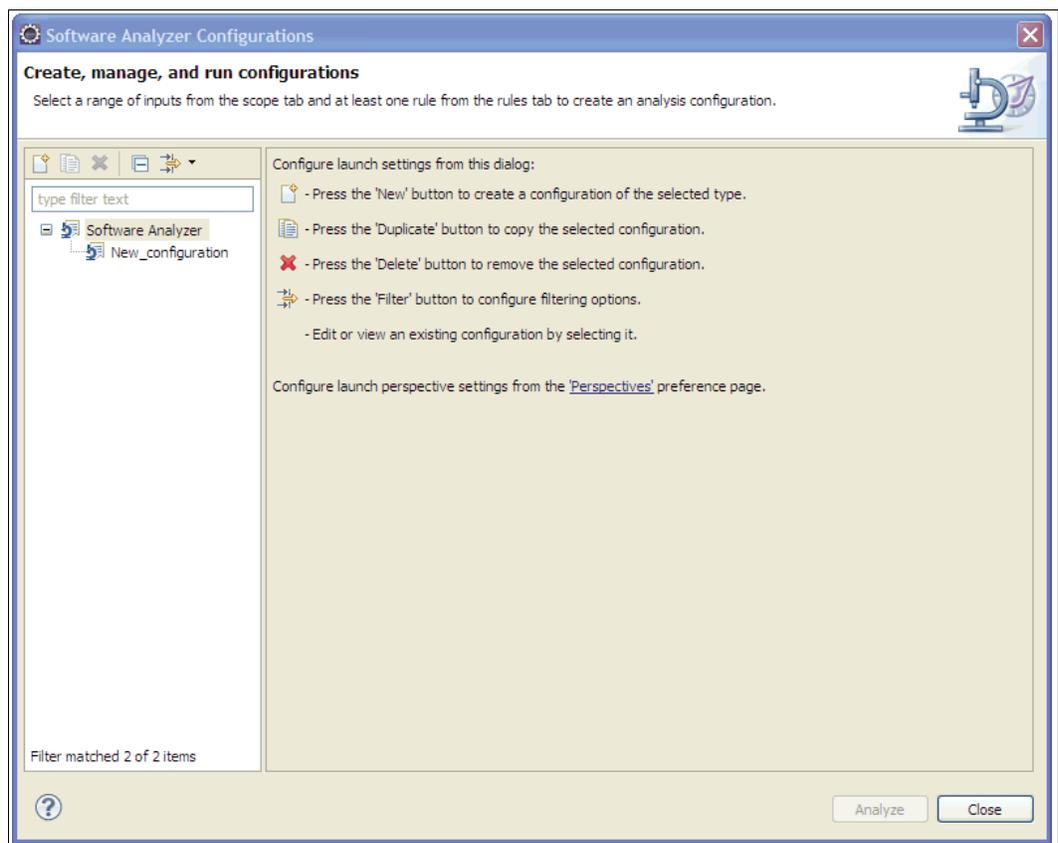


图 3-16 Software Analyzer 配置

4. 单击**新建配置**，对话框变为基本配置界面（图 3-17）。

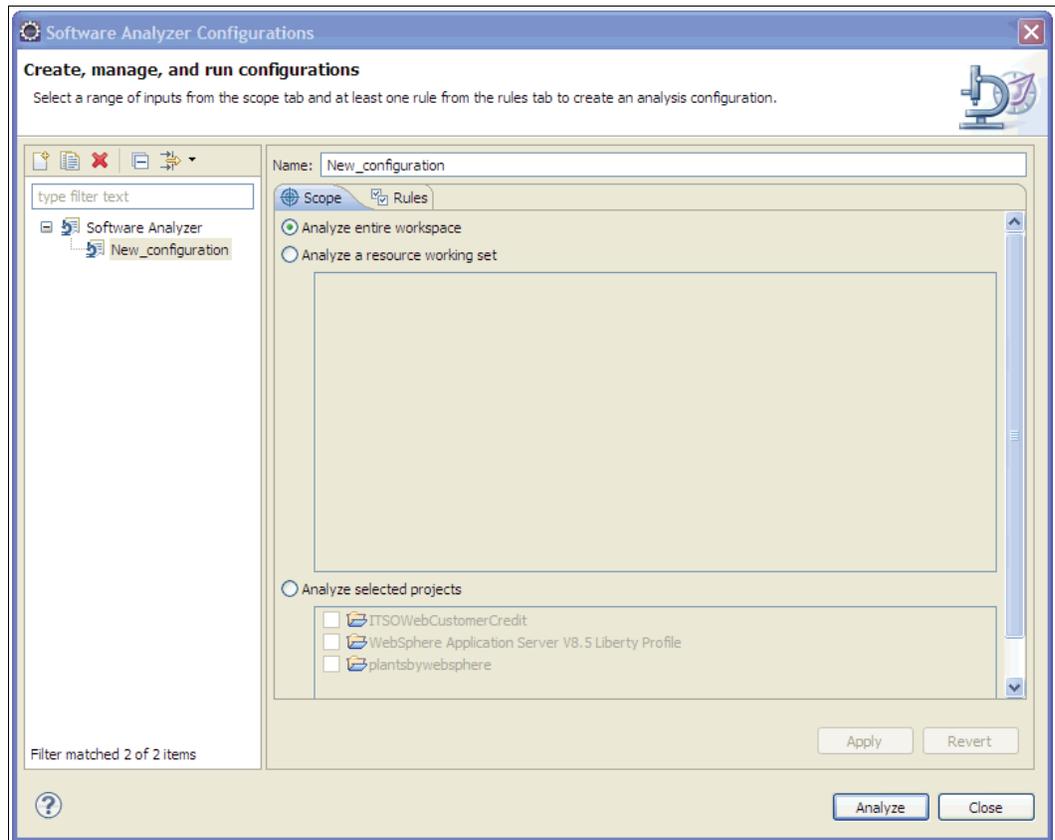


图 3-17 新建配置

在“Software Analyzer 配置”对话框中，输入配置名称，如 **SampleAppMigration**。在“范围”选项卡中，单击**分析整个工作空间**，即可扫描工作空间内的全部项目。选择该对话框中的其他选项，即可将分析范围限制为对工作集或所选项目进行分析。

提示：当使用 **Enterprise Explorer** 菜单运行分析时，分析范围仅限于所选菜单项对应项目中的节点。您可以对有限代码集执行快速分析。

5. 在“规则”选项卡中，使用“规则集”列表选择待执行的分析类型。若为 **WebLogic** 到 **WebSphere Application Server** 的迁移，则在“规则集”菜单中，选择 **WebLogic 应用程序迁移**。

该规则集包含了以下分析域内的规则：

- **Java 代码复查：**检查 **Java** 代码，确定其整体质量。该域中的规则涵盖从简单的代码样式实施到用于检测空指针问题的较复杂规则。

提示：由于某些规则会导致某些结果，因此并非所有的规则都适用于所有环境。在这些情况下，应避免运行这些规则，或者忽略其所带来的结果。

- **JSP 代码复查：**此类别包含与 **JSP** 代码相关的规则。
- **XML 文件复查：**此类别包含与 **XML** 及关联文件相关的规则。
- **类路径复查：**此类别包含与类路径问题及所含库相关的规则。

提示：若要获得某条规则的更多信息，请突出显示该规则并按 **F1**。然后，在配置对话框中即会显示该规则的帮助。初始帮助对话框中包括简短描述及指向更多信息的链接。

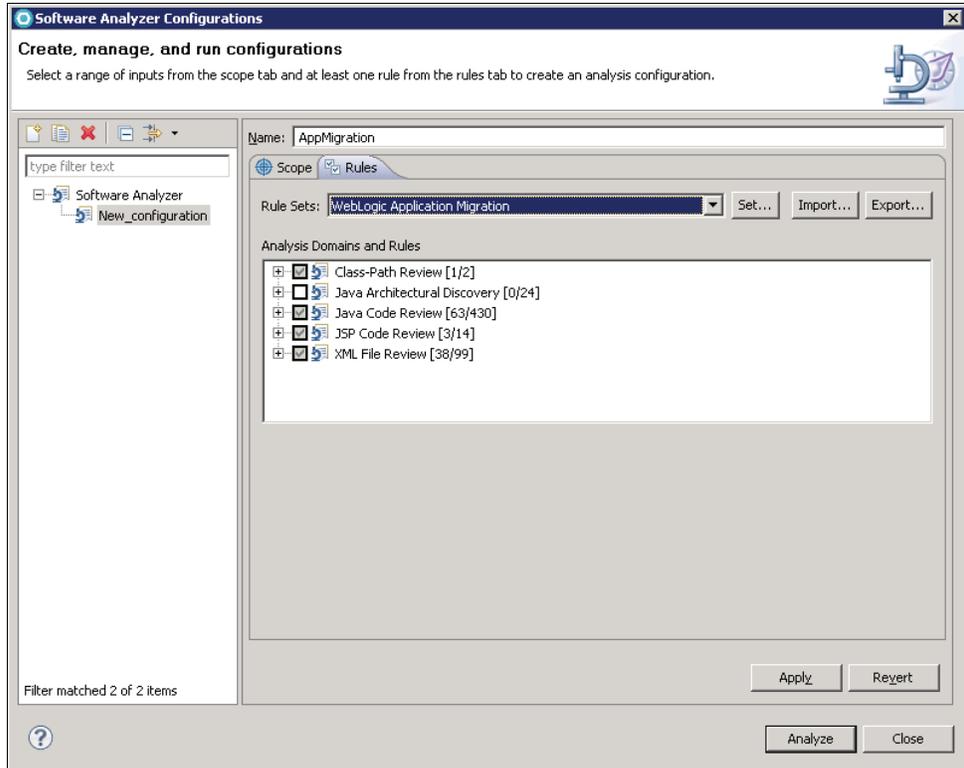


图 3-18 WebLogic 应用程序迁移规则

6. 在选择了“WebLogic 应用程序迁移”规则集后，单击**设置**即可打开相应的对话框，用以配置要迁移至的 WebSphere 以及在 WebLogic 中使用的 Java 版本（图 3-19）。

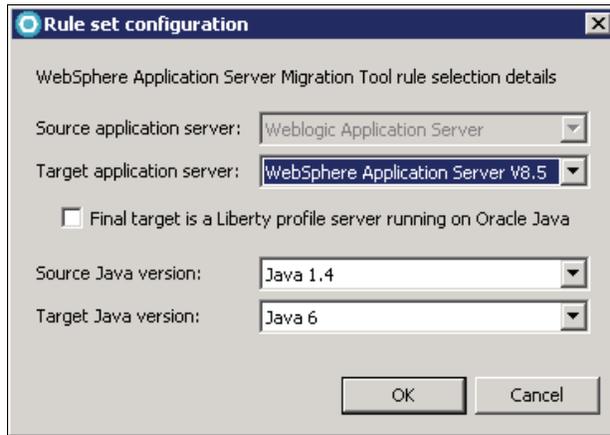


图 3-19 规则集配置

目标 WebSphere Application Server 版本可以是 V7、V8 或 V8.5。同时，选择之前所用的源 Java 版本。若目标 WebSphere 平台是 V8.5，选择 Java V6 或 Java V7 均可。

规则的数量：分析配置对话框中规则的数量取决于工具安装所在平台。

7. 单击**应用**保存规则配置，然后单击**分析**。分析结果可在“软件分析结果”视图中查看（图 3-20）。

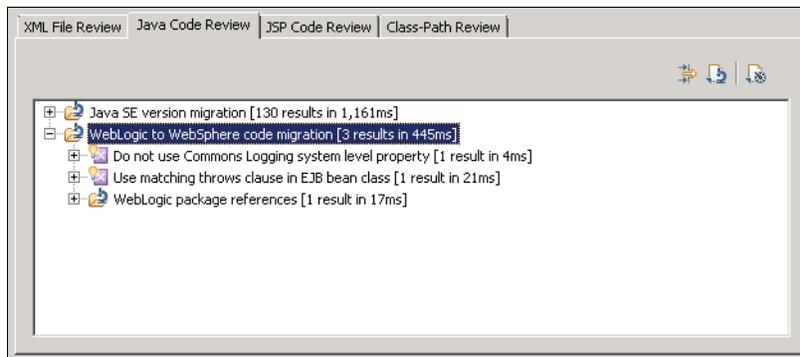


图 3-20 “软件分析结果”视图

右键单击各个结果即可访问各个选项，诸如查看出现问题的源代码或使用提供的修正更正问题。

无结果：若窗口中未显示任何结果，则表示在扫描过程中未发现问题。

针对结果可能执行的操作包括：

- ▶ 查看结果：打开编辑器，其中将会显示触发规则的源文件。问题的原因会突出显示，并在编辑器的左边缘显示违反规则图标。
- ▶ 快速修正：若选择该选项，则会运行相应的转换以修改受影响的 **Java** 代码、**XML** 文件、**JSP** 或清单文件进行修改，确保其能够在 **WebSphere Application Server** 中运行。快速修正可能会直接更改代码或给出完成修正所需的步骤。
- ▶ 快速修正预览：对于支持显示原始代码与应用修正后代码并排对比的规则，可采用该选项。您可以使用该选项来查看更改前的情况。
- ▶ 忽略结果：该选项会将规则从列表中除去，而不更改任何代码。对于 **Java** 文件，会在文件中添加注释，以便在未来的分析运行中不会触发该规则。
- ▶ 全部快速修正：该选项能够解决针对规则所识别的全部问题。
- ▶ 快速修正所有类别：该选项能够对针对规则所属类别识别的问题运行全部快速修正。规则必须在启用“全部快速修正”后，“快速修复全部类别”选项方可运行其快速修正。举例来说，若您针对某个 **Java** 规则选择了该选项，具有“全部快速修复”选项的所有 **Java** 规则即可运行快速修正。

3.5.3 生成报告

Application Migration Tool 也可为您生成报告，以便更好地估算应用程序的迁移范围。报告将汇总过去的分析记录，并提供应用程序代码质量、复杂性和结构的高级输出。在分析运行完毕后，您可以单击“**Software Analyzer 结果**”窗口右侧的图标（即图 3-21 中用红色圆圈突出显示的图标）来生成分析报告。

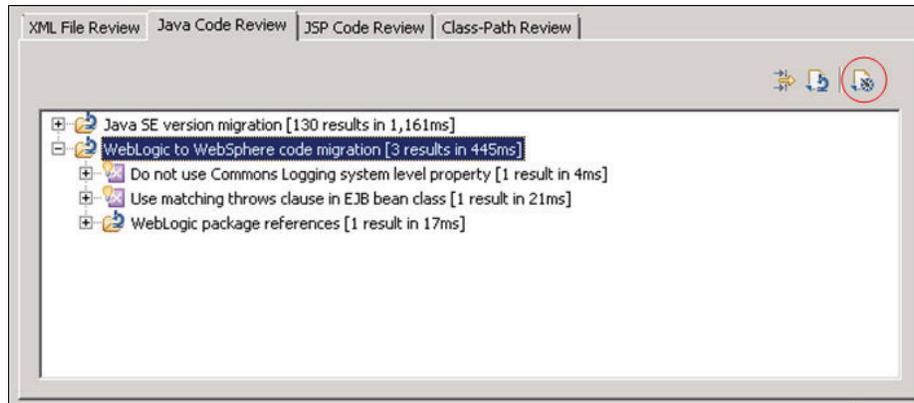


图 3-21 生成报告

单击该图标后，即可生成当前选定结果元素的分析报告。举例来说，对于 Java 代码复查，如图 3-22 所示，共有 4 个选项：

- ▶ Java 代码复查结果
- ▶ Java 代码复查忽略结果
- ▶ Java 代码复查严重性汇总
- ▶ Java 代码复查（按类别）

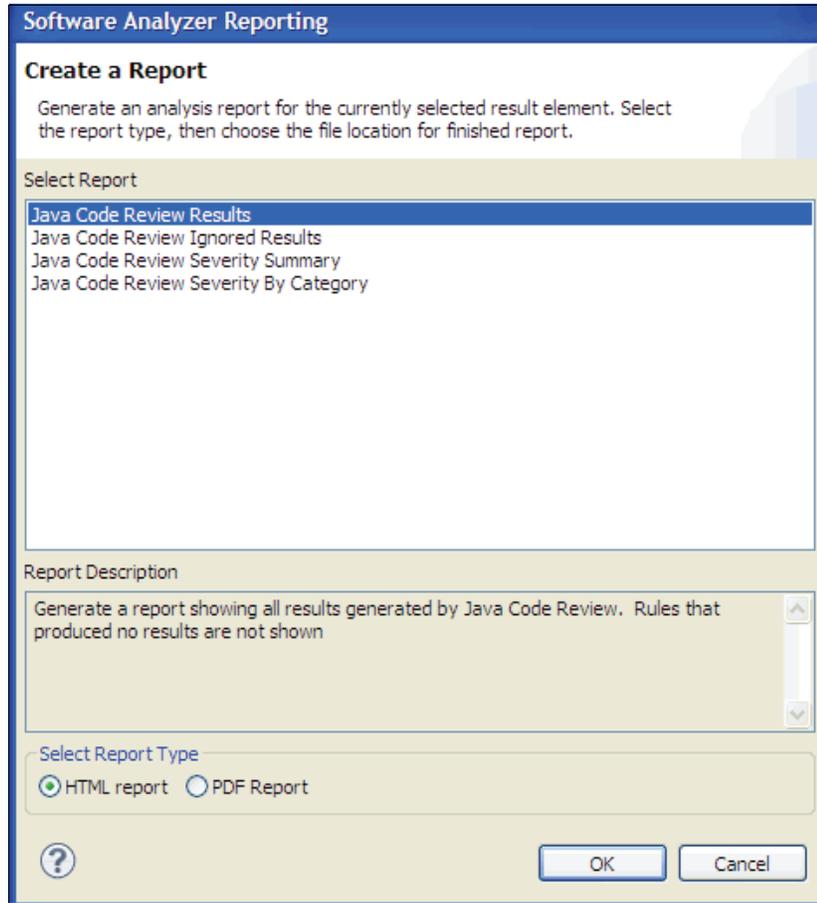


图 3-22 创建报告

报告能够以 HTML 和 PDF 格式生成。生成的文件位于 `workspace/.metadata/.plugins/com.ibm.xtools.analysis.reporting/reports` 目录中。

若要选择报告汇总，请单击生成历史扫描报告。图 3-23 显示的是为分析汇总生成的样板报告。

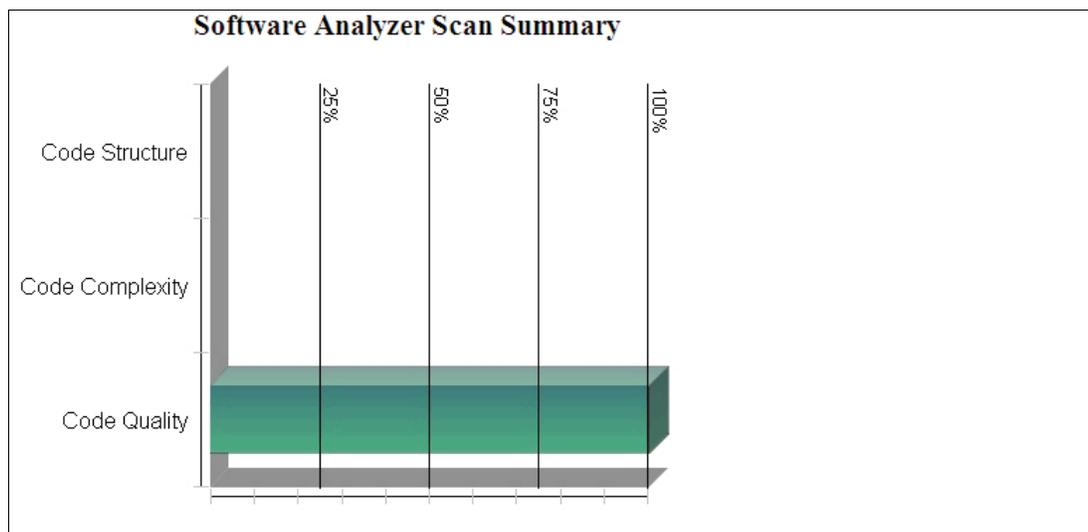


图 3-23 样板报告

3.6 故障诊断

Application Migration Tool 会将错误信息写入两个日志文件：

- ▶ 工作空间日志 (workspace/.metadata/.log) 包含了由 Application Migration Tool 记录的消息以及由 Rational Software Analyzer 或 Eclipse 记录的消息。
- ▶ Application Migration Tool 的服务日志位于 workspace/.metadata/.plugins/com.ibm.ws.appconversion.base 目录中。

该日志包含了全部错误信息，若启用跟踪的话，还包含了详细的跟踪信息。

若要在迁移工具中启用跟踪，可在命令行上设置 `appconversion.trace` 系统变量来启动集成开发环境 (IDE) 或者在 `eclipse.ini` 属性文件中设置该变量：

- ▶ 命令行选项

将该系统变量添加到启动 Eclipse 的命令行中：

```
eclipse.exe -vmargs -Dappconversion.trace=true
```

- ▶ eclipse.ini 选项

将以下选项添加到 `eclipse.ini` 文件中，该文件与 `eclipse.exe` 文件的存储于相同的目录中：

```
-Dappconversion.trace=true
```

提示： 运行跟踪后即可显示该工具。只有在服务需要时，才可打开该工具。



Application Migration Tool - WebSphere Version to Version 的安装和配置

本章介绍了 Application Migration Tool - WebSphere Version to Version 的安装信息；该工具可指导您如何将应用程序迁移至 IBM WebSphere Application Server V8.5。无论您使用的是 Eclipse Marketplace、命令行安装还是 Rational Application Developer V8.5，我们都一一做了详细介绍。

本章所述主题如下：

- ▶ Application Migration Tool - WebSphere Version to Version 概述
- ▶ 在 Eclipse 上安装 Application Migration Tool

4.1 Application Migration Tool - WebSphere Version to Version 概述

Application Migration Tool - WebSphere Version to Version (Application Migration Tool) 是 IBM WebSphere Application Server Migration Toolkit 的一部分。该工具箱是整体迁移战略的一部分。Application Migration Tool 能够通过其扫描功能找出 WebSphere Application Server 不同版本之间可能会影响合规性和应用程序服务器运行时行为的特定结构变化。该工具提供了不仅复查有问题的代码的方法，还提供了相关问题的详细帮助，在某些情况下，还会自动修正代码，确保应用程序能够在 IBM WebSphere Application Server V7、V8 或 V8.5 上运行。

在迁移至较新版本的 WebSphere Application Server 发行版时，会存在多个对代码迁移和开发步骤造成影响的问题。迁移过程可能涉及到 Java 源代码、JavaServer Pages (JSP) 及部署描述符的修改。这些更改包括：

- ▶ 对 Java SE 5、6 和 7 中遇到的 Java 运行时环境 (JRE) 进行更改
- ▶ 删除之前建议不要使用的功能
- ▶ 产品 API 行为方面的更改
- ▶ 基于 Java Enterprise Edition (Java EE) 规范澄清而做出的更改
- ▶ 建议不要使用的功能

迁移过程可能涉及到 Java 源代码、JavaServer Pages (JSP) 及部署描述符的修改。Application Migration Tool 能够协助您执行这些类型的代码更改。Application Migration Tool 能够扫描出在待从 WebSphere Application Server V5.1、V6.0、V6.1 或 V7.0 迁移至 WebSphere Application Server V7.0、V8.0 或 V8.5 的应用程序中的许多已知问题。若有可能，还能够提供快速修正，以更改应用程序代码，从而解决版本之间的更改问题。

对于 WebSphere Application Server 版本迁移，该工具支持从以下版本：

- ▶ WebSphere Application Server V5.1
- ▶ WebSphere Application Server V6.0
- ▶ WebSphere Application Server V6.1
- ▶ WebSphere Application Server V7.0
- ▶ WebSphere Application Server V8.0

迁移至以下版本：

- ▶ WebSphere Application Server V7.0
- ▶ WebSphere Application Server V8.0
- ▶ WebSphere Application Server V8.5

4.1.1 Application Migration Tool V3.5

Application Migration Tool 增加了一些有助于应用程序迁移的新规则，包括：

- ▶ 用于迁移至 WebSphere Application Server V8.5 的新 Java 和 XML 规则
- ▶ 选择使用 Java 7 时标记 Java 7 兼容性变化的新 Java SE 7 规则
- ▶ 新的动态规则集对话框，简化了规则选择且使用方便
- ▶ 新的不建议使用的规则

4.1.2 WebSphere Application Server V8.5 Liberty 概要文件

WebSphere Application Server V8.5 包含有一个新的 Liberty 概要文件服务器，该服务器提供了高度严谨且能够快速启动的动态应用程序服务器运行时环境。Application Migration Tool 能够与 WebSphere Application Server V8.5 Liberty 概要文件开发者工具共存使用，在 Liberty 概要文件环境中运行应用程序。

Application Migration Tool 能够与 IBM WebSphere Application Server V8.5、8.0 和 7.0 开发者工具兼容，因此您能够在单个轻型 Eclipse 集成开发环境 (IDE) 中进行迁移、开发、部署和测试。

有关迁移过程及 Application Migration Tool 概述，请参阅第 29 页上的第 3 章“Application Migration Tool 的安装和配置”。

4.2 在 Eclipse 上安装 Application Migration Tool

Application Migration Tool - WebSphere Version to Version 可按照第 29 页上的第 3 章“Application Migration Tool 的安装和配置”中所述进行安装。以下所列为使用 Eclipse Marketplace 和命令行界面 (CLI) 进行逐步安装。

提示：应先卸载 Application Migration Tool V1.x、V2.0 或 V3.5 测试版，再安装 V3.5。V2.1、V3.0 和 V3.1 可升级至 V3.5。

4.2.1 使用 Eclipse Marketplace 安装 Application Migration Tool

若要使用 Eclipse Marketplace 安装 Application Migration Tool，请完成以下步骤：

1. 单击**帮助** → **Eclipse Marketplace**。
2. 在“搜索”选项卡中，在搜索框中输入 WebSphere 并单击**执行**。

3. 在搜索结果窗口（图 4-1），单击包含 IBM WebSphere Application Server Migration Toolkit 部分旁边的安装按钮。

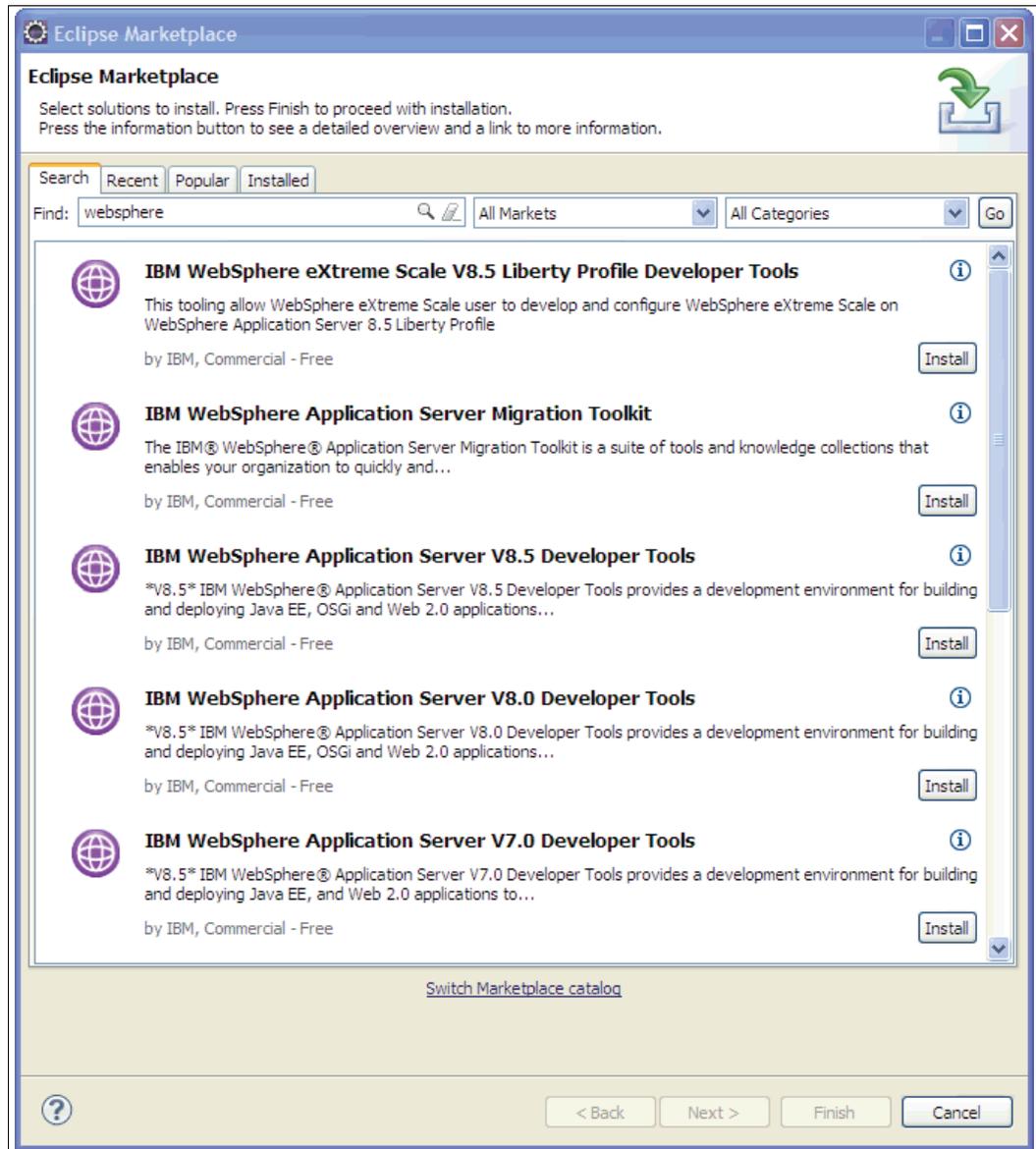


图 4-1 Eclipse Marketplace

4. 选择 **Application Migration Tool** 复选框，这将会同时选择主功能及其他包括的功能，如图 4-2 所示。

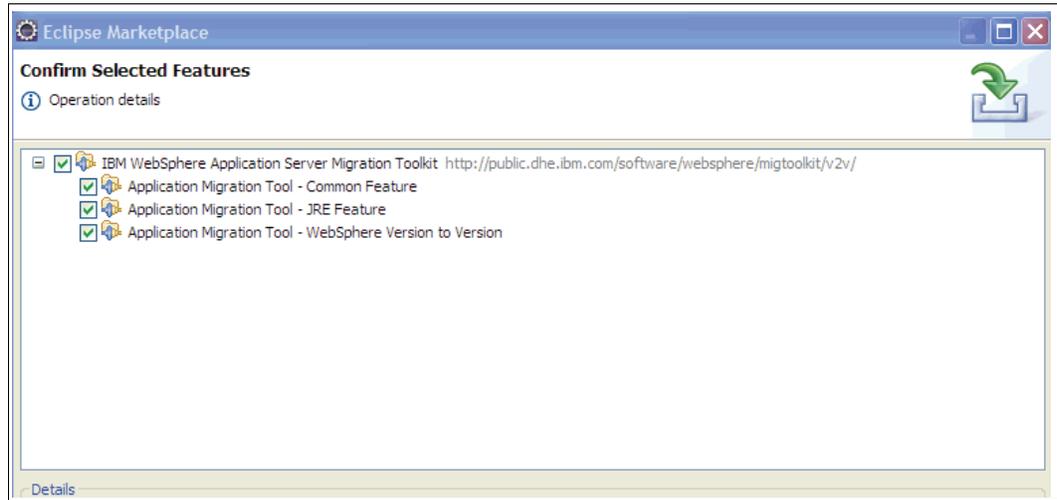


图 4-2 选择功能

5. 接受安装许可协议并开始安装。
6. 安装完成后，重新启动 Eclipse。

4.2.2 使用 CLI 安装 Application Migration Tool

Migration Toolkit 的功能可通过 Eclipse 命令行安装工具安装。示例 4-1 所示为使用命令行安装 Application Migration Tool - WebSphere Version to Version。

示例 4-1 安装

```
C:\Program Files\IBM\SDP\eclipse.exe
-application org.eclipse.equinox.p2.director
-metadataRepository
jar:file:C:/<amtDownloadDir>/Application_Migration_Tool_WebSphere_Version_to_Versi
on_vn.n.n.zip!/
-artifactRepository
jar:file:C:/<amtDownloadDir>/Application_Migration_Tool_WebSphere_Version_to_Versi
on_vn.n.n.zip!/
-installIU com.ibm.ws.appconversion_feature.was2was.feature.group
-nosplash
```

对应的卸载命令如示例 4-2 所示。

示例 4-2 卸载

```
C:\Program Files\IBM\SDP\eclipse.exe
-application org.eclipse.equinox.p2.director
-uninstallIU com.ibm.ws.appconversion_feature.was2was.feature.group
-nosplash
```

4.2.3 使用 Rational Application Developer V8.5 安装 Application Migration Tool

本节介绍了如何使用 Rational Application Developer V8.5 安装并实施 Application Migration Tool。请完成以下步骤：

1. 从 developerWorks® 站点（网址为 http://www.ibm.com/developerworks/websphere/downloads/migration_toolkit.html）下载最新版本的 Application Migration Tool - WebSphere Version to Version 功能部件，并将其保存在本地。
2. 从 Rational Application Developer V8.5 菜单栏中，单击帮助 → 安装新软件。
3. 在“安装”窗口中，单击添加。
4. 在“添加站点”窗口中，输入以下信息（图 4-3）：

名称：输入工具的相应名称，例如 Application Migration Tool Version to Version。

位置：输入所下载压缩文件的位置，例如：

C:\Application_Migration_Tool_WebSphere_Version_to_Version_v3.5.0.zip。

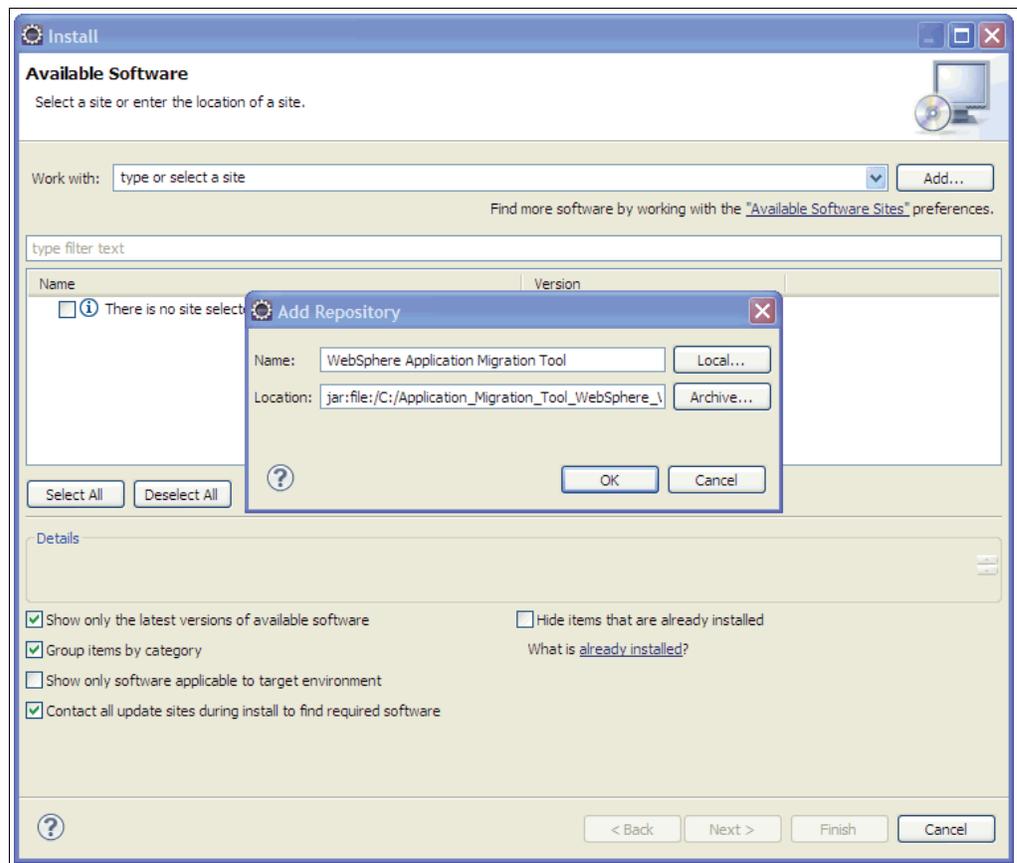


图 4-3 使用 Rational Application Developer V8.5 安装 Migration Toolkit

5. 在“安装”窗口中，选择您想要安装的插件（图 4-4）。单击下一步。

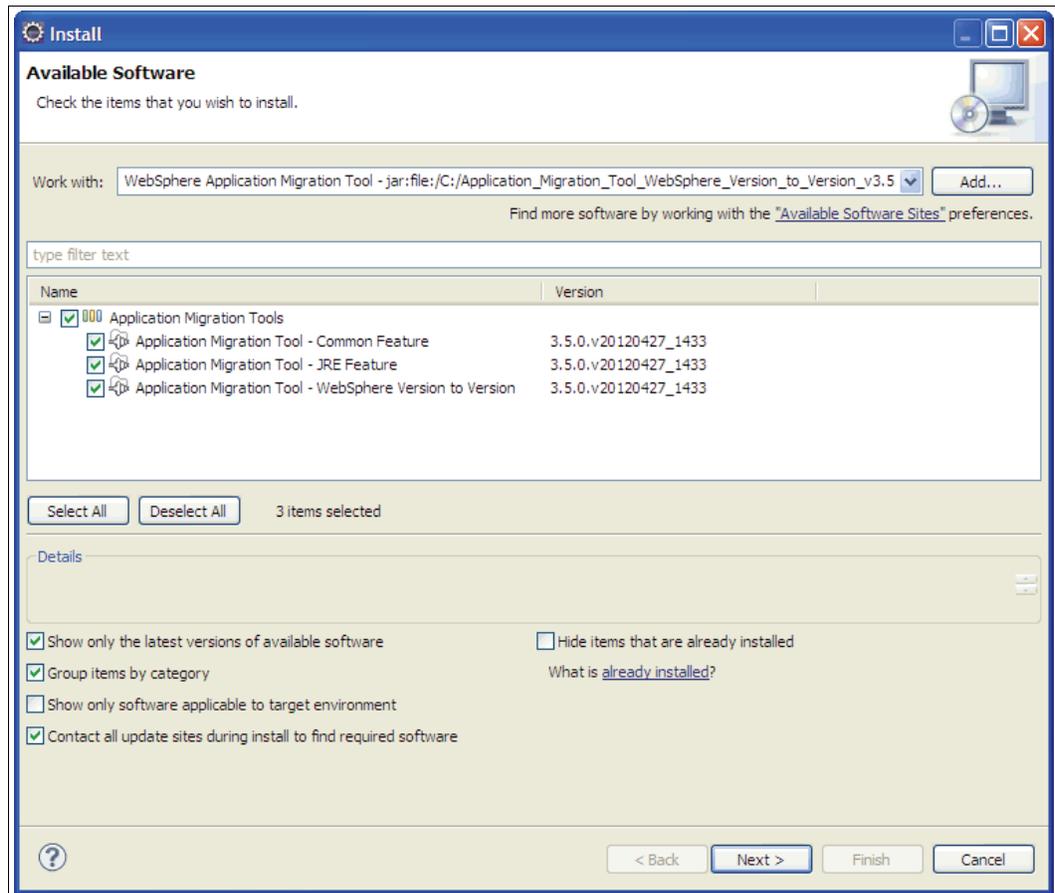


图 4-4 可用的软件窗口

6. 接受许可协议并单击**完成**，即可开始安装。

7. 安装完成后，单击**立即重新启动**，以重新启动 Rational Application Developer。



从较早版本的 WebSphere Application Server 迁移

本章介绍如何从较早版本的 WebSphere Application Server，尤其是从 WebSphere Application Server V6.1、V7.0 或 V8.0 至 V8.5 迁移应用程序。

本章所述主题如下：

- ▶ 概念
- ▶ Plants by WebSphere 样例应用程序迁移
- ▶ Plants By WebSphere 的迁移
- ▶ 概率分布样例
- ▶ Web 服务 Axis 2 股票报价样例应用迁移

5.1 概念

对于大部分应用程序来说，从 WebSphere Application Server 的某个版本至较新版本的迁移很容易，我们只要直接把运行应用的应用服务器直接升级就可以了。但是，升级所安装的应用程序服务器的过程仅仅是版本到版本迁移全过程的一小部分。

5.1.1 整体迁移过程

无论是竞争性迁移还是版本到版本迁移，迁移工作都应有正式的项目计划。图 5-1 描述了整体迁移过程。

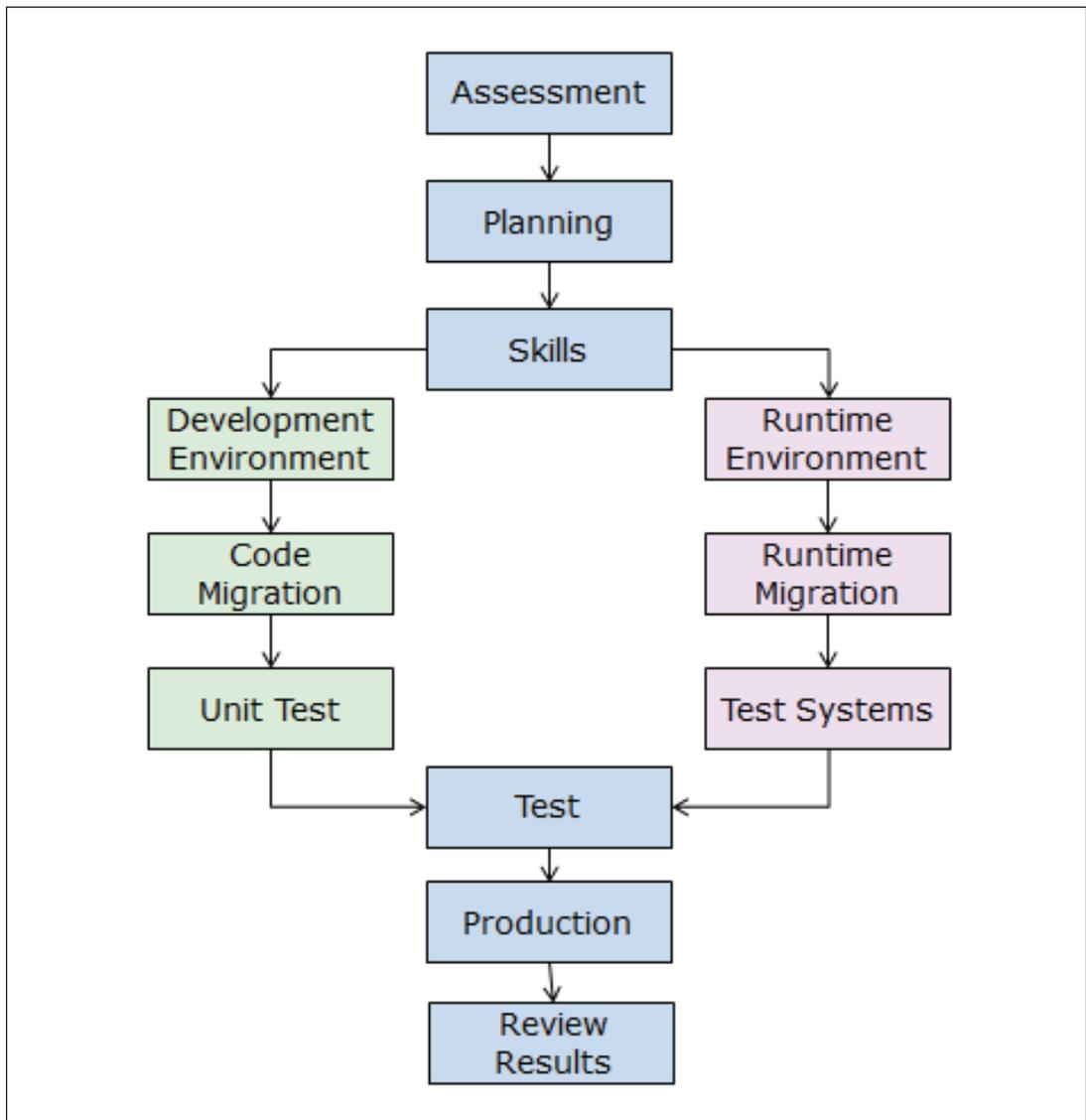


图 5-1 迁移过程

整体迁移过程由几个部分组成：

- ▶ 评估
- ▶ 规划
- ▶ 技能
- ▶ 运行时环境
- ▶ 开发环境
- ▶ 测试、生产和回顾

评估

评估是迁移规划的第一步，确认对迁移规划有影响的所有因素，是一个很重要的步骤。评估是基于各个公司具体情况的要求采集。评估的具体步骤如下：

- ▶ 确认利益相关方并明确一个核心迁移团队（尤其是大型迁移项目）。
- ▶ 确认对开发人员和管理员的培训要求。在该阶段中，还应评估硬件要求并考虑所有可能的升级。应为生产系统和其他用于开发的系统做好硬件升级计划。硬件升级可能是从 32 位系统升级至 64 位系统。
- ▶ 确认 **WebSphere Application Server** 将迁移至何种级别。影响该决策的因素有：对任何特定 **Java EE** 和 **Java SE** 版本的要求、对第三方应用程序或其他 **WebSphere** 产品的任何要求，以及 **WebSphere Application Server** 体系结构或兼容性的变化。在此步骤中，应评估供应商的应用程序是否准备好执行计划迁移至的目标 **WebSphere Application Server** 版本。应为停机容错性和故障转移的拓扑来评估应用程序。然后，应评估应用程序的 **Java EE** 规范变化和 **JRE** 变化。
- ▶ 审核测试操作以确定其是否覆盖全面。自动化测试的数量对确保成功的可重复性也很重要。

规划

迁移规划的建立应将所有评估结果纳入考虑范围。评估的具体步骤如下：

- ▶ 获取硬件升级要求并制定相关计划。
- ▶ 升级任何必备软件，例如操作系统和 **Web** 服务器级别。
- ▶ 列出培训需求。
- ▶ 确认所迁移应用程序的早期使用者，以便将先期的经验应用到与其他团队的进一步迭代中。在您与早期使用者协作时，应确认可用于推动迁移过程的试点项目。
- ▶ 创建应用程序投产策略。根据目前所收集的信息，建立一个执行时间表，并把可用性要求、维护期限和投产封版期限纳入考虑范围。
- ▶ 在试图对生产系统进行迁移之前，应确认可实施的回滚计划。**WebSphere Application Server** 运行时配置迁移工具可以提供一个很好的回滚计划，在您安装新版本的 **WebSphere Application Server** 环境之前不会破坏旧版本的 **WebSphere Application Server** 环境。这样可提高支持回滚计划的便捷性。

技能

有多种途径可用于培训技能，包括正式的课堂培训和在线培训。应考虑的培训相关问题有：新的开发工具、WebSphere Application Server 管理模式的变化、新版本的 WebSphere Application Server 的变化及 Java EE 等新标准。

运行时环境

运行时环境的迁移包含应用程序服务器配置的迁移。我们将介绍这部分迁移过程的相关问题。另外，有关运行时环境的迁移，详见 *WebSphere Application Server V7 Migration Guide*, REDP-4635。

大多数组织有多个运行时环境，包括开发测试、系统测试、性能和预生产环境。迁移期间，需要新旧版本 WAS 服务器并存运行的测试环境，以便维护未迁移的应用程序，同时支持新版本的测试。使用这些测试系统中的某些系统，尤其是最能代表生产环境的系统，测试您的迁移计划也很重要。

开发环境

开发环境或应用程序迁移是版本到版本迁移过程中最耗时的部分之一。

迁移至新版本的 WebSphere Application Server 需要更改集成开发环境 (IDE)。迁移过程中的迭代也非常有用。

一般无需将应用程序更新到您所用版本的 WebSphere Application Server 所支持的最新 J2EE 规范。新版 WAS 服务器仍然为根据早期 J2EE 规范编写的应用程序的运行提供了支持。

迁移过程中，我们可能无需更改应用代码，但这是个理想的状态。如果需要做出更改，必须尽量减少更改，只进行一些必要更改以确保应用程序迁移成功并在新版本中顺利运行。成功并稳定地迁移至 WebSphere Application Server 后，可安全地使用最新版本的新功能。同样地，应适时处理不建议使用的功能。

测试、生产和回顾

应用程序和运行时环境初步迁移成功后，提议的迁移必须经过标准测试周期再投入生产。理想情况下，测试周期应通过一组因维护或业务升级而修改应用程序时所用的标准回归测试完成。让这些测试自动进行是一种最佳实践。

另一个重要方面是监视应用程序的性能。由于 WebSphere Application Server 和 JDK 版本间存在差异，应确保应用程序的性能没有下降。并且需要为生产环境制定一个回滚计划。

此任务完成后，即可进入生产环境迁移的最后一个步骤。

迁移完成后，回顾迁移经验以检查迁移过程中的正确之处、错误之处及经验教训。您可通过这些知识和调整不断提高您的迁移经验。

5.1.2 轻松迁移

大部分应用程序在不同版本的 **WebSphere Application Server** 之间迁移时不会发生问题。现在已尽量减少了不同版本的应用程序之间的更改，以减少会发生问题的应用程序的数量。在迁移跨越的版本越多，不兼容问题数量可能就越大。

5.1.3 已知问题

在不同版本的应用程序服务器间迁移应用程序时可能会产生的问题，可能由一系列已知问题导致。**Application Migration Tool** 中包含了这些已知问题，并使用程序自动化识别您的应用程序所需的更改。**IBM** 还在 **IBM Migration Knowledge Collection** 中提供了一个完整的问题列表，请访问：

<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg27008724>

以下章节介绍了 **WebSphere Application Server V8.5**、**WebSphere Application Server V8**、**WebSphere Application Server V7** 和 **WebSphere Application Server V6.1** 中的主要开发和 **WebSphere API** 的主要变化。

WebSphere Application Server V8.5 中的主要变化

本节介绍了应用程序迁移过程中应考虑的 **WebSphere Application Server V8.5** 中已知的主要问题。

迁移至 Java 7 运行时

以下章节列出了迁移至 **Java 7** 运行时后发生的变化：

► AWT:

- `MouseEvent.getButton()` 方法可返回计划中 0 - 3 范围以外的值。之前，当用户单击按钮或使用滚轮时，`MouseEvent.getButton()` 方法返回 0 - 3 之间的值。为适应带两个滚轮或四五个按钮的新型鼠标，该方法现在可返回从 0 至按钮数量的值。
- 调用 `Windows.setBackground` 可能会导致 `UnsupportedOperationException` 异常。当应用程序在不支持半透明效果的系统中运行时，将不透明背景色应用于帧的应用程序可能会失败。
- 现在 `Toolkit.getPrintJob(Frame, String, Properties)` 将抛出 `NullPointerException`。在先前的发行版中，当您在无图形界面环境中调用 `Toolkit.getPrintJob(Frame, String, Properties)` 时，会抛出 `HeadlessException`，而非上述 `NullPointerException`。
- 现在有多种 `Toolkit` 方法会抛出 `HeadlessException`。例如，现在 `Toolkit.isFrameStateSupported(int)` 和 `Toolkit.loadSystemColors(int[])` 在无图形界面环境中使用时，会抛出 `HeadlessException`。
- `sun.awt.exception.handler` 系统属性被 `Thread.UncaughtExceptionHandler` 类代替。

► 全局化

- 用户语言环境和用户界面语言环境已截然区分。缺省语言环境可单独进行设置，用于两种用途：格式设置用于规定资源的格式，而显示设置用于菜单和对话框。新的 `Locale.getDefault(Locale.Category)` 方法使用 `Locale.Category` 参数。
- 对 **UTF-8** 实现进行了更新，以确保 **Corrigendum** 符合 **Unicode 3.0.1**。之前容许的 **UTF-8** 序列格式有 5 字节和 6 字节格式。这些格式现在均已不再使用。

► IO

`java.io.File.setReadOnly` 和 `setWriteable` 方法有新的行为。不再需要针对目录设置 **DOS** 只读属性。这就意味着如果文件是一个目录，则这些方法失败会返回 `false`。为维护与 `canWrite` 的关系，如果文件是一个目录，`canWrite` 方法将返回 `true`。

► **JDBC**

引入了新的 JDBC 方法，在接口中添加了新方法。有新的方法支持 JDBC 4.1。这些方法包括添加至 `java.sql.Connection`、`java.sql.Driver`、`javax.sql.CommonDataSource` 和 `java.sql.Statement` 接口的方法。

► **JAXP**

- 移除了 `XSLTPProcessorApplet` 类。`XSLTPProcessorApplet` 类是一种应用程序级别工具类，有很多问题，已被移除。
- JAX-WS 服务器遇到 DTD 时，会抛出 SOAP 故障。SOAP Message Construct 是 SOAP 消息的 XML 信息集，不应包含文档类型声明 (DTD) 信息项。

► **Language**

- 之前，当传入值小于 `Thread.MIN_PRIORITY` 时，`ThreadGroup.setMaxPriority` 不会按规定做出相应行为；而是将输入值重置为 `Thread.MIN_PRIORITY`。规范规定忽略小于 `Thread.MIN_PRIORITY` 的值。现在，`ThreadGroup.setMaxPriority` 方法按规定作出行为。
- `java.lang.Character.isLowerCase/isUpperCase` 方法进行了更新，以便符合指定的 Unicode 定义。
- 更新了 `TypeVisitor` 接口。为对此发行版中的语言变化建模，对 `javax.lang.model.*` 进行了几次更新，包括在 `javax.lang.model.type.TypeVisitor` 接口中添加了一个方法。
- 不要在 `java.lang.Throwable` 中将方法定义成 `final`。该设置会影响扩展 `Throwable` 的类。添加了 `addSuppressed` 和 `getSuppressed` 方法。

► **网络**

若您试图在 HTTP 响应代码为 -1 时读取数据，服务器连接将关闭。服务器在没有有效 HTTP 状态行的情况下发送响应时，HTTP 协议处理程序将关闭与该服务器的连接。服务器连接关闭时，若试图在该连接上读取数据，则会导致 `IOException`。

► **Text 文本**

现在，`java.text.BreakIterator.isBoundary(int)` 方法按规定做出相应行为。当偏移超出界限时，该方法将按规定返回 `false`，而不是抛出 `IllegalArgumentException`。

► **Utilites 实用程序**

- 当检测出违反 `Comparable` 约定的 `Comparable` 时，`Arrays` 和 `Collections` 的新行为将抛出 `IllegalArgumentException`。
- 通过构造函数或 `Put` 方法在 `TreeMap` 或 `TreeSet` 中插入 `null` 无效元素，将会抛出 `NullPointerException`。之前，可在空 `TreeMap` 或 `TreeSet` 中插入无效 `null` 空对象和未实现 `Comparable` 接口的对象。多余的对象会导致出现 `NullPointerException` 或 `ClassCastException`。`Collection` 的大部分其他操作也将失败。
- 现在，当“#”标记被指定为转换“s”，且其后的自变量不属于 `Formattable` 实例（包括特例“null”）时，`Formatter.format()` 现在将抛出 `FormatFlagsConversionMismatchException`。

JPA 变化

以下为 JPA 变化:

- ▶ **JPA 级联策略中的变化:** 对于使用级联类型 **PERSIST**、**MERGE** 和 **ALL** 的实体关系, 有一定行为变化。先前的发行版在持续保存与某实体的关系之前, 会检查数据库中是否存在相关实体。此操作会导致将额外的 **Select** 发送至数据库。在当前发行版中, 添加了相关代码, 因此当您与相关实体的无持久状态持续保存进行级联时, 该持续保存 (插入) 无需检查数据库即可直接完成。如果数据库中存在相同实体, 此操作会导致抛出 **EntityExistsException**。要将该行为恢复至先前的发行版, 只需将 **openjpa.Compatibility** 属性 **CheckDatabaseForCascadePersistToDetachedEntity** 的值设为 **true**。
- ▶ 与 **ListAttribute** 相关的 **JPA MetaModel** 代码生成发生了变化。在先前发行版中, **MetaModel** 实现为每个数组生成一个 **ListAttribute**。如果数组没有标注为 **PersistentCollection**, 则该行为是正确的; 如果数组为无标注数组 (如 **byte[]** 和 **char[]**), 则该行为是错误的。对该行为进行了修正, 以确保未存储为 **PersistentCollections** 的数组采用 **SingularAttribute** 而非 **ListAttribute**。我们可以把兼容性属性 **UseListAttributeForArrays** 设置为 **true**, 在 **persistence.xml** 配置文件中设定 **name="openjpa.Compatibility" value="UseListAttributeForArrays=true"**。

WebSphere Application Server V8.0 中的主要变化

本节介绍了应用程序迁移过程中应考虑 WebSphere Application Server V8.0 中的主要已知问题。

迁移至 Java EE 6

WebSphere Application Server V8.0 符合 Java EE 6, 应用程序迁移至 WebSphere Application Server V8.0 时应考虑 Java EE 6 的变化。

EJB 变化

表 5-1 中列出了 必须加以考虑的 EJB 变化。

表 5-1 EJB 变化

接口	详细信息	过去的行为	现在的行为
EJB	Web 模块中的 EJB。	予以忽略	现得到处理。此操作会导致潜在错误。
@ApplicationException	在 EJB 3.1 中, inherit 是一个新的关键词, 缺省为 true 。	inherit=false	inherit=true 。可影响划分子类异常行为。

JAX-WS 变化

表 5-2 中列出了 必须加以考虑的 JAX-WS 变化。

表 5-2 JAX-WS 变化

接口	详细信息	过去的行为	现在的行为
JAX-WS 客户端	一些产生本地异常的调用是无效主机或端口导致的。	针对所调用的方法抛出 WebServiceException 并显示空消息。	调用处理程序的 handleFault 消息时, 会抛出 SOAPFaultException 。
JAX-WS 客户端	WSDL 中的 Policy:Addressing 的处理:	予以忽略。	得到处理。

接口	详细信息	过去的行为	现在的行为
SOAP 消息	如果没有标头，则会发生 getSOAPHeader 和 getSOAPBody 行为。	无错误。	抛出 SOAPException。

JAX_RPC: JAX_RPC 在 Java EE 6 中建议不要使用，而在 WebSphere Application Server 中维持现状。

JCA 增强

表 5-3 中列出了必须加以考虑的 JCA 增强。

表 5-3 JCA 增强

接口	详细信息	过去的行为	现在的行为
SQL 异常错误代码	暂停的数据源没有唯一返回码。	总是返回 0	2147117569

JPA 变化

表 5-4 中列出了必须加以考虑的 JPA 变化。

表 5-4 JPA 变化

接口	详细信息	过去的行为	现在的行为
EntityManager	refresh() 方法传递空值。	无错误	抛出 IllegalArgumentException 异常。
detach(Object entity)	对众多 API 的新方法，JPA 1.x 有一个类似方法：<T> T detach(<T> pc)。	<T> T detach(<T> pc)	从 <T> T detach(<T> pc) 更改为新的 <T> T detachCopy(<T> pc)。

JSF 变化

以下为 JSF 变化：

- ▶ 缺省 JSF 实施从 SUN RI 更改为 MyFaces。MyFaces 为 JSF V2.0。SUN RI 保留为 JSF V1.2 且仅用于与先前版本相兼容。如果要使用 JSF 2.0 的功能，则必须重新编写 JSP 以使用 Facelet。
- ▶ 在 JSF 2.0 中将传入更多异常，而在 JSF 1.2 中则不会出现。

JSP 变化

表 5-5 列出了必须加以考虑的 JSP 变化。

表 5-5 JSP 变化

接口	详细信息	过去的行为	现在的行为
JSP-Property-Groups	is-xml 和 page-encoding 配置选项只应该应用于那些与 URL 模式相匹配的 JSP。	匹配的 JSP 及其包含的 JSP	仅与 JSP 相匹配

OSGi 变化

表 5-6 中列出了必须加以考虑的 OSGi 变化。

表 5-6 OSGi 变化

接口	详细信息	过去的行为	现在的行为
组件组合包 (.cba)	组件服务的可视范围在组件包中定义	组件包对外的可视性	组件包对内的可视性

SIP 变化

表 5-7 中列出了必须加以考虑的 SIP 变化。

表 5-7 SIP 变化

接口	详细信息	过去的行为	现在的行为
SIPFactory	createRequest 和 createAddress(String sipAddress) 将 URI 作为输入或返回参数应包含在尖括号内 (<>)	忽略括号	可正确处理括号

WebSphere 移除

以下为从 WebSphere Application Server V8.5 中移除的功能：

- ▶ 移除了 Apache SOAP 实现，使用标准的 WebServices
- ▶ `com.ibm.websphere.ant.tasks.StopServer.setHost (java.lang.String host)`。
- ▶ `com.ibm.websphere.rsadapter.WSConnectJDBCDataStoreHelper`。
- ▶ `com.ibm.websphere.servlet.error.ServletErrorReport.getCause()`。

WebSphere Application Server V7.0 中的主要变化

本节介绍了应用程序迁移过程中应考虑 WebSphere Application Server V7.0 中的主要已知问题。

迁移至 JRE 6

以下章节介绍了迁移至 JRE 6 的影响：

- ▶ 使用新的语言特性和 JRE 6 的应用程序只能部署到 WebSphere Application Server V7.0 节点。当您编译应用程序时，您可为先前版本的 JRE 目标指定“-source”和“-target”方式。例如，“-source 1.4”和“-target 1.4”。
- ▶ Java 序列化在 JRE 1.4 和先前的发行版之间不兼容。强制 UUID 作为最佳实践。
- ▶ 如果新的类与您应用程序中定义的类相同，Java 新特性中的新类可能会导致不明确的引用。

有关更多信息，请参阅 Java SE 6 网站，网址为：

<http://java.sun.com/javase/6/>

JRE 6 源代码兼容性

除了一些微小的因素之外，JRE 6 通常与 JRE 5 在源代码上面向上兼容：

- ▶ `sun.*` 包中的一些 API 发生了变化。这些 API 不适用于开发人员。开发人员若从 `sun.*` 包中导入，将 *自担风险*。
- ▶ 在某些情况下，`javac` 现在可拒绝之前接受但不正确的程序。
例如：
 - `javac` 正确拒绝了无效的强制类型转换。
 - 未声明为 `public` 的 EJB 业务方法。
 - 映射到不存在的 `Servlet` 的 `FilterMapping`。
- ▶ 调试和概要分析程序接口发生了变化。移除了 Java 虚拟机调试接口 (JVMDI)，禁用了 Java 虚拟机概要分析程序接口 (JVMPDI)。
- ▶ 在 Java SE 6 中，从 `rt.jar` 中移除了非 `Class` 类文件。指定 `-Xbootclasspath:<path to rt.jar>` 和请求所有源文件的 Java 应用程序均会失败，这是因为这些资源现在位于另一个名为 `resources.jar` 的 JAR 文件中。
- ▶ 其他 API 的变化：
 - `Duration` 和 `XMLGregorianCalendar equals()` 方法现在针对 `null` 空对象参数将返回 `false`。
 - `java.beans.EventHandler` 强制有效的自变量。

有关因素的完整清单，请参阅：

<http://java.sun.com/javase/6/webnotes/compatibility.html>

迁移至 Java Platform, Enterprise Edition 5

以下章节介绍了迁移至 Java Platform, Enterprise Edition 5 时必须加以考虑的一些规范变化。本章着重介绍了 WebSphere Application Server V7.0 中升级至 JEE5 的接口中在 J2EE 1.4 内无替代项可用于使用旧版本支持的那些接口。

例如：

- ▶ JSP 从 2.0 升级至 2.1。
- ▶ EJB 升级至 3.0，但 EJB 2.1 应用程序仍在 2.1 级别运行。

有关规范级别选项的完整详细信息，请参阅：

http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.webSphere.base.doc/info/aes/ae/rovr_specs.html

部分 JSP 2.1 变化

下面列出了一部分 JSP 2.1 变化：

- ▶ 新的保留字符：现在 JSP 保留了字符序列 `#{'`。因此，如果 `#{'` 在模板文本中使用或作为基于 1.2 标记库的属性值中的文字，则该序列必须进行转义。
- ▶ 资源注入可能会导致一些 JSP 不再可编译。由于新的资源注入支持，大型 JSP 现在可能会超过 64 K 的限制。在 V7.0.0.11 及更新版本中，可通过在 `ibm-ext-web.xml` 中将 JSP 属性 `“disableResourceInjection”` 设置为 `true` 来关闭此类支持。
- ▶ 不再支持在当前范围内对标记库进行重新定义。若要允许标记库定义，可设置 JSP 属性 `<jsp-attribute name="allowTaglibPrefixRedefinition" value="true"/>` 或将服务器的 Web 容器定制属性设置为 `com.ibm.wsspi.jsp.allowtaglibprefixredefinition=true`。该定制属性会影响到服务器上的所有应用程序。

- ▶ 在定制标签中使用前缀之前，必须有定义该前缀的标记库指令。若要在前面使用前缀，将服务器 Web 容器定制属性设置为 `com.ibm.wsspi.jsp.allowtaglibprefixusebeforedefinition=true`。该定制属性会影响到服务器上的所有应用程序。
- ▶ 不再支持属性在 `jsp:output` 元素中多次出现。若要支持该配置，将服务器 Web 容器定制属性设置为 `com.ibm.wsspi.jsp.allowjspoutputelementmismatch=true`。该定制属性会影响到服务器上的所有应用程序。

V6.1 Feature Pack 功能包的差异

本节介绍了 V6.1 和 V7.0 功能包间的主要差异。

V6.1 的 EJB 3.0 功能包

下面列出了部分差异：

- ▶ EJB 3.0 绑定文件错误：一些应用程序可能无法在 WebSphere Application Server V7.0 中启动，这是因为现在对 EJB 3.0 绑定文件中使用的名称进行了唯一性检查。
- ▶ 隐式使用运行时 JAR 文件：当部署企业 bean (EJB) 应用程序时，可能会抛出运行时 `ClassNotFoundException` 异常，这是因为类路径条目 Java 归档 (JAR) 文件未导出或未发布，并使用了一个或多个运行时 JAR 文件。该消息只是一个警告。
- ▶ WebSphere Application Server V7.0 不支持在 EJB 3.0 级别模块中使用 bean 管理的持久性 (BMP) 和容器管理的持久性 (CMP) 实体 bean。在 EJB 3.0 功能包中支持 BMP 实体 bean。

V6.1 的 Web 服务功能包

下面列出了部分差异：

- ▶ 在 WebSphere Application Server V7.0 中，仅在 Java EE 5 或更高版本的模块中支持 JAX-WS 注释。
- ▶ 在 WebSphere Application Server V7.0 中，JAXB 作为 JRE 6 的一部分提供。V7.0 中的工厂实现有别于 V6.1 中的工厂实现。

Web 服务和 EJB 功能包

若要保持与 V6.1 版功能包的兼容性，您必须启用下列属性中的一个属性，以便在应用程序安装和服务器启动期间请求扫描：

- ▶ `UseWSFEP61ScanPolicy` 属性（适用于 Web 服务功能包）
- ▶ `UseEJB61FEPScanPolicy` 属性（适用于 EJB 3.0 功能包）

JSF 1.2 影响

本节介绍了 JSF 1.2 的差异：

- ▶ 有一些修正工具可解决先前 JSF 和非 JSF 标记间出现内容交织的问题。
- ▶ `ViewHandler` 扩展器将受到影响。一些定制脚本和一些第三方包必须支持 JSF 1.2，例如 `Tiles`。其他示例还包括 `Tomahawk`、`Trinidad`、`IceFaces`、`Facelets` 等。
- ▶ JSF Portlet 报告内存消耗更高，导致会话量更大。
- ▶ 对于 JSF 1.0 应用程序，移除 Rational Application Developer 放入应用程序中的 JSF 1.0 JAR 文件 (`jsf-impl.jar` 和 `jsf-api.jar`)。JSF 现在是 Java EE 的一个部分，是运行时相关的一流技术。

扫描注释

JEE5 引入了注释支持。此支持需要在安装应用程序时扫描 Java 注释，这一新步骤可能需要相当长的一段时间。Web 模块不仅包括直接打包在 WAR 文件内的类（WEB-INF/classes 目录下），还包括打包在 WAR 文件中的 JAR 文件内的类。

对于包含有 Java Platform, Enterprise Edition 5 内容的应用程序，将此类模块标记为 JEE5 级别即可对应用程序进行优化。若已知模块中不包含注释，可使用“metadata-complete”标记来提高扫描效率。此外，可对应用程序的结构进行重新调整，将实用程序 JAR 文件（已知不包含注释信息的文件）放置于共享库或 EAR 文件的根目录中。这样，不会对这些文件进行注释扫描。

有关更多信息，请参阅：

http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/urun_rapp_metadata_lockdd.html

在 WebSphere Application Server V7.0.0.5 及随后版本中提供有可配置过滤功能，能够识别忽略注释处理的模块或 Java 包。

有关更多信息，请参阅：

<http://www.ibm.com/support/docview.wss?uid=swg1PK87053>

通过注释，更多的文件类型能够作为扫描的一部分加以处理，例如在 WebSphere Application Server V7.0 之前被忽略的加密压缩文件。通过以下步骤，可在 WebSphere Application Server V7.0.0.7 中避免某些问题：

1. 登录部署服务器的管理控制台并单击**应用程序服务器** → <部署服务器名称> → **Java 和进程管理** → **进程定义** → **控制** → **Java 虚拟机** → **定制属性**。将 `com.ibm.websphere.application.migration.disabled` 添加到定制属性列表，其值设为 `true`，然后保存配置。
2. 导航至 `<profile_home>/properties/wsadmin.properties` 文件并编辑该文件，添加 `com.ibm.websphere.application.migration.disabled=true`。

有关更多信息，请参阅：

<http://www-01.ibm.com/support/docview.wss?uid=swg1PK92880>

WebSphere API 迁移详情

以下章节对已移除的功能进行了详细介绍：

- ▶ 移除了使用通用 JDBC 驱动程序的 Derby Network Server Provider。使用 Derby Client 的 Derby Network Server 作为替代服务器使用。
- ▶ 移除了对基于 DB2 CLI 的 2 类 JDBC 驱动程序及基于 DB2 CLI 的 2 类 JDBC 驱动程序 (XA) 的支持，应将 DB2 通用 JDBC 驱动程序作为替代驱动程序使用。
- ▶ 移除了 WebSphere Connect JDBC 驱动程序、Microsoft SQL Server 2000 Driver for JDBC 以及 WebSphere SequeLink JDBC Driver for Microsoft SQL Server。您可以使用 DataDirect Connect JDBC 驱动程序或 Microsoft SQL Server 2005 JDBC 驱动程序。您可以使用 `WebSphereConnectJDBCdriverConversion` 命令来转换 WebSphere Connect JDBC 驱动程序中的的数据源。
- ▶ 移除了 Integrated Cryptographic Services Facility (ICSF) 认证机制。您可以使用轻量级第三方认证 (LTPA) 机制。

表 5-8 说明了对安全定制属性的支持。

表 5-8 对安全定制属性的支持

旧属性	新属性
com.ibm.security.SAF.unauthenticatedId	com.ibm.security.SAF.unauthenticated
com.ibm.security.SAF.useEJBROLEAuthz	com.ibm.security.SAF.authorization
com.ibm.security.SAF.useEJBROLEDelegation	com.ibm.security.SAF.delegation

以下所列为部分已移除的接口：

- ▶ com.ibm.websphere.servlet.filter 包中的所有类（包括 ChainedRequest、ChainedResponse、ChainerServlet 和 ServletChain）均已移除。您可以使用 javax.servlet.filter 类。
- ▶ 已移除 Web 服务网关定制 API。您可以使用 JAX-RPC 处理程序和服务集成总线调解的组合来代替现有的过滤器。

表 5-9 说明了已移除的其他类。

表 5-9 已移除的其他类

旧类	新类
com.ibm.websphere.servlet.session.UserTransactionWrapper	将 UserTransaction 直接存储在 HTTP 会话中。
com.ibm.websphere.rsadapter.DataDirectDataStoreHelper	com.ibm.websphere.rsadapter.ConnectJDBCDataStoreHelper.
com.ibm.websphere.rsadapter.MSSQLDataStoreHelper	com.ibm.websphere.rsadapter.MicrosoftSQLDataStoreHelper.

V6.1 中的主要问题

本节介绍了应用程序迁移过程中应考虑的 WebSphere Application Server V6.1 中的主要已知问题。

迁移至 JRE 5

以下所列为迁移至 JRE 5 时的主要注意事项：

- ▶ 使用新的语言特性和 JRE 5 的应用程序只能部署到 WebSphere Application Server V7.0 节点。当您编译应用程序时，您可为先前版本的 JDK 目标指定“-source”和“-target”方式。例如，“-source 1.4”和“-target 1.4”。
- ▶ 序列化在 JRE 1.4 和先前的发行版之间不兼容。
- ▶ 如果新的类与应用程序中定义的类相同，Java 新特性中的新类可能会导致不明确的引用。

JRE 5 源代码兼容性

下节列出了 JRE5 源代码兼容性方面的主要注意事项：

- ▶ JRE 5 通常与 JRE 1.4.2 在源代码方面向前兼容，但已更改的 sun.* 包中的某些 API 除外。这些 API 不适用于开发人员。开发人员若从 sun.* 包中导入，将自担风险。
- ▶ 名为 enum 的变量现在是一个 Java 语言的关键字，某些由 IBM Web 服务生成的代码可将 enum 用作包名称。若应用程序使用 enum，则除非在 javac 命令上指定 -source 1.4，否则它在 JDK 5 将不再能够编译。
- ▶ 大多数使用通用类、构造函数、方法和字段的源代码将继续在 JDK 5 中编译，但也有一些并非如此。

- ▶ 在您导入 `java.net.Proxy.*`、`java.lang.reflect.*` 或 `java.util.Queue` 类时可能会出现错误，这是因为在 **Java 2 Platform, Standard Edition (J2SE) V5** 中存在的新类与其他包名称相冲突，诸如 `javax.jms.Queue`。当引用某个类名称时，将会提供所必需的全部信息，包括基本名称及相应的扩展名，即可解决这一问题。
- ▶ 有关 JAXP 的变化，请参阅：
http://java.sun.com/j2se/1.5.0/docs/guide/xml/jaxp/JAXPCompatibility_150.html
- ▶ 在 JDBC 5.0 中，针对时间戳记调用 `compareTo` 来对比 `java.sql.Timestamp` 和 `java.util.Date` 时会抛出 `ClassCastException`。
- ▶ `BigDecimal toString()` 方法的行为与之前版本也有不同。J2SE 5.0 在 `BigDecimal` 中增加了 `toPlainString()`，这种方法与之前版本中的 `toString()` 方法非常相似。
- ▶ 不建议直接使用 XML 和 XSL 解析器的私人实现。可使用现有的类装入器支持来使用应用程序类路径，以此来代替 Java 虚拟机引导程序类路径。

JRE 5 VerboseGC 输出

GC 输出格式会随所用垃圾回收策略的不同而不同。表 5-10 给出了 `MustGather` 信息链接。

表 5-10 `MustGather` 表

标题	描述	URL
MustGather: 分析 -Xgcpolicy:gencon 的详细垃圾回收输出	gencon 垃圾回收策略的详细垃圾回收输出格式	http://www-01.ibm.com/support/docview.wss?uid=swg2122486
MustGather: 分析 -Xgcpolicy:optavgpaue 详细垃圾回收输出	optavgpause 垃圾回收策略的详细垃圾回收输出格式	http://www-01.ibm.com/support/docview.wss?rs=727&uid=swg2122459
MustGather: 分析 -Xgcpolicy:optthruput 的详细垃圾回收输出	optthruput GC 策略的详细垃圾回收输出格式	http://www-01.ibm.com/support/docview.wss?rs=727&uid=swg2122443

JRE 5 和 JSSE 2

自 JDK V1.4 起，`com.ibm.net.ssl` 包已弃用，WebSphere Application Server V6.1 中也移除了该包。同时使用 `javax.net.ssl` 包代替。与 `KeyManager`、`TrustManager`、`X509KeyManager` 和 `X509TrustManager` 等创建和配置安全套接字工厂相关的类也受到了影响。有关更多详情，请转至 <http://www.ibm.com/developerworks/java/jdk/security/50/> 并搜索 IBMJSSE2 指南及样板。

JDK 5 功能的使用及 JSP

由于 JSP 的默认运行时编译器设置为 JDK 1.3，因此使用 JDK 1.5 特定代码 JSP 会在编译过程中显示错误消息。对于此点，可通过两种方法来解决：

- ▶ 在安装应用程序时指定 JDK 源代码级别。
- ▶ 在 Rational Application Developer 中，转至 **Web 扩展** 选项卡。在“JSP 属性”表中，添加 `jdkSourceLevel` 为名称，其值设为 1.5。通过这些处理会在 `Web Project/WEB-INF/ibm-web-ext.xmi` 中添加一个条目，举例来说，`<jspAttributes xmi:id="JSPAttribute_1" name="jdkSourceLevel" value="15"/>`。

WebSphere 中移除的 API

自 V5.1 起，弃用了公共连接器框架 (CCF)。同时使用 Java2Connectors(J2C) 代替。WebSphere Application Server V6.1 弃用了 Cloudscape V5.1 并使用 Cloudscape V10.1 代替。这一变化可能会影响到 JDBC 驱动程序配置，因此必须予以修改，此外由于 Cloudscape DB 不兼容，也必须予以迁移。

WebSphere Application Server V6.1 中弃用了 Mozilla Rhino JavaScript 和 JDOM。此外，还移除了安全属性 `com.ibm.websphere.security.CustomRegistry`。同时使用 `com.ibm.websphere.security.UserRegistry` 代替。

有关更多信息，请参阅：

http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/topic/com.ibm.websphere.nd.doc/info/ae/ae/rmig_deprecationlist.html

EJB setRollbackOnly() 的变化

应注意其中包含两种 EJB，分别为 EJB1 和 EJB2，它们运行于同一个 WebSphere Application Server 中。EJB1:method1 和 EJB2:method2 均具有容器管理的事务，通过“TX_REQUIRED”属性来定义。HTTP 请求调用事务之外的 EJB1:method1（EJB 容器将会为 EJB1 创建事务），而 EJB1:method1 会调用 EJB2:method2。（EJB2:method2 将运行于同一事务下）。当 EJB2:method2 调用 `setRollbackOnly()` 时，EJB2:method2 会正常返回，而 HTTP 请求则会收到 `RemoteException org.omg.CORBA.TRANSACTION_ROLLEDBACK`。不过在 WebSphere Application Server V6.0 及更早版本中，HTTP 会收到 EJB1:method1 的业务结果，而非 `RemoteException`。在各种情况下，事务均会回滚，但在 WebSphere Application Server V6.1 中，HTTP 请求不会收到由 EJB1:method1 返回的业务结果。

Solaris 10 x86_64 上的 JNI

Solaris 10 x86_64 上的 WebSphere Application Server V6.0 作为 32 位应用程序构建（32 位 Java 和 32 位 JNI）。客户可能使用其自有的应用程序，此类应用程序会依赖于不属于 WebSphere Application Server 一部分的 JNI，例如：特殊的数据库驱动程序。若要在该环境中运行，JNI 就必须为 32 位 JNI。Solaris 10 x86_64 上的 WebSphere Application Server V6.1 作为 64 位应用程序构建（64 位 Java 和 64 位 JNI）。若 V6.0 中的某些应用程序使用不属于 WebSphere Application Server 一部分的 32 位 JNI，这些应用程序就无法在 WebSphere Application Server V6.1 中使用。64 位 JVM 无法加载 32 位 JNI。除非将 JNI 重新构建为 64 位，否则就应与该 JNI 的供应商联系，以获得该 JNI 的 64 位版本。

有关更多信息，请参阅：

<http://www-1.ibm.com/support/docview.wss?uid=swg2700752>

5.2 Plants by WebSphere 样例应用程序迁移

Plants by WebSphere 是由 WebSphere Application Server 样例库提供的一个应用程序。正如其名称所示，该应用程序使用专门销售设备、木料及附件的在线商店展示了 Java Platform, Enterprise Edition 的多个功能。

通过 Plants by WebSphere 店面，客户能够开立帐户、浏览要购买项目、查看产品详情及下达订单。Plants by WebSphere 应用程序使用容器管理的持久性 (CMP)、容器管理的关系 (CMR)、无状态会话 bean、有状态会话 bean、JSP 页面及 servlet。

该应用程序仅用于展示 WebSphere Application Server 本身的功能。不过，其在很多测试和开发方案中均非常有用。Plants by WebSphere 的源代码通过 WebSphere Application Server 分发，也正因为如此，您可以随意对其进行修改。

在 WebSphere Application Server 5.0 以上的各个版本中均包含有 Plants by WebSphere。尽管本章所述适用于多个版本的 WebSphere Application Server，但在本章中假定所安装版本为 WebSphere Application Server V6.1，因此若使用较早的版本，所提供的目录路径和应用程序命令可能有误。

源代码的目录位置为 <WAS_DIR>\samples\bin\PlantsByWebSphere。该代码可在任何开发环境中修改。不过，利用由支持 J2EE 应用程序的 IDE 提供的功能可能更为方便。

举例来说，Eclipse V3.2 及更高版本能够与 WebSphere Application Server Developer Tool 很好的兼容，这是因为在您提供与 WebSphere Application Server 的无缝集成时能够轻松地导入源代码。有关该环境的更多信息，请参阅 Eclipse Foundation Webtools 网站或 WebSphere 开发者工具网站。若要下载该工具，请转至：

https://www.ibm.com/developerworks/mydeveloperworks/blogs/wasdev/entry/download_wdt?lang=en

5.2.1 应用程序概述

Plants by WebSphere 样例采用了以下技术：

- ▶ 容器管理的持久性 (CMP) 实体 bean
- ▶ Bean 管理的持久性 (BMP) 实体 bean
- ▶ 无状态会话 bean
- ▶ 有状态会话 bean
- ▶ Servlet
- ▶ JavaServer Pages (JSP) 文件和 HTML
- ▶ 容器管理的关系 (CMR)
- ▶ Java 2, Enterprise Edition (J2EE) 安全
- ▶ Java API for XML-based remote procedure call (JAX-RPC)

Plants by WebSphere 应用程序由一系列 JSP 页面和 HTML 页面构成。这些页面使用以下 servlet 进行访问：AccountServlet、ShoppingServlet、ImageServlet 及 AdminServlet。servlet 使用各种企业 bean 业务方法，而这些业务方法会在需要时访问数据库中的数据。总的来说，无状态会话 bean 用于与实体 bean 的交互，以减少事务的数量。

图 5-2 给出了该应用程序的高级体系结构。

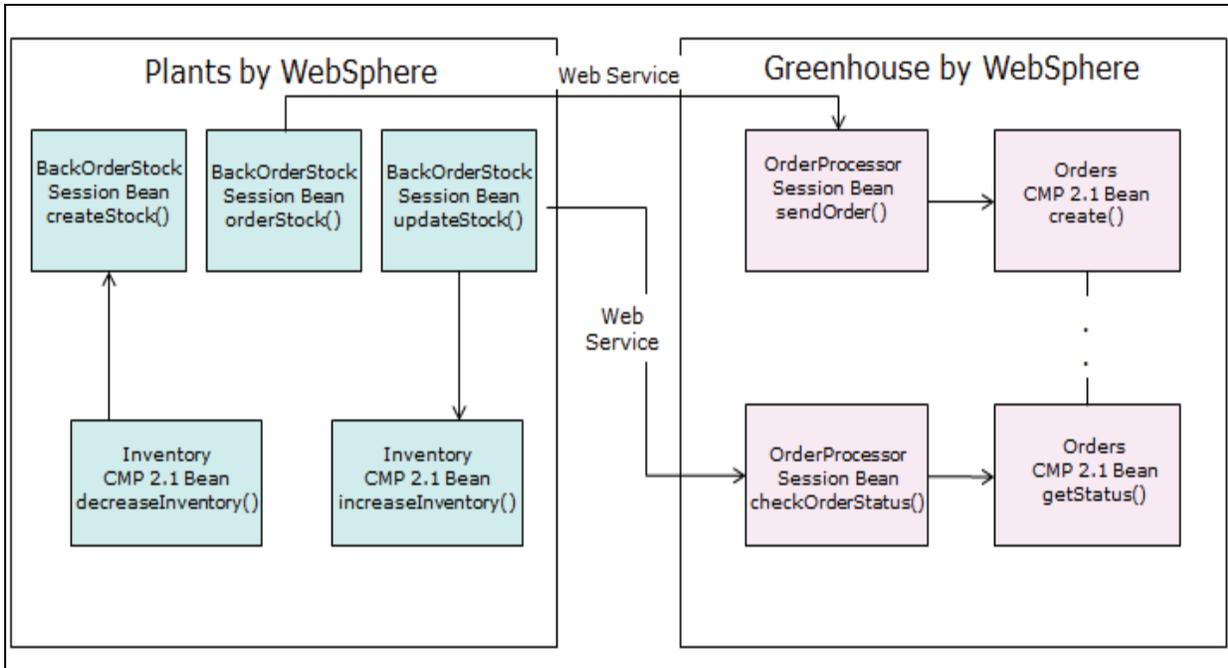


图 5-2 Plants by WebSphere 应用程序高级体系结构

Servlet

本节介绍了 Plants by WebSphere 应用程序所用 servlet 的相关信息。

► ShoppingServlet

ShoppingServlet servlet 能够处理浏览器和企业 bean 之间的大多数交互。此类交互包括库存、购物车及订单功能。浏览购物种类时会使用 **Catalog** 会话 bean，用于查找并显示全部的相关库存对象。某个物品详细信息的显示也可使用 **Catalog** 会话 bean 从 **Inventory CMP** 实体 bean 中获取信息。

将某个项目添加到购物车时会创建 **ShoppingCart** 有状态会话 bean。**Catalog** 会话 bean 能够获取存货数据并将项目置于 **ShoppingCart bean** 之中。若要查看和更新购物车，可通过使用 **ShoppingCart** 有状态会话 bean 来完成。在输入开票和发货信息后，**ShoppingCart bean** 将创建一个 **Order CMP** 实体 bean。一旦完成结账程序，将会创建 **Mailer** 无状态会话 bean，以使用 **Java Mail API** 发送用于确认的电子邮件。

► ImageServlet

ImageServlet servlet 用于获取并将产品图像置于数据库之中。该 servlet 会从 **JSP** 页面中获取图像并从数据库中获得 **HTML** 页面，转而通过 **HTTP** 响应为浏览器提供服务。

► AdminServlet

AdminServlet servlet 会处理来自用户（浏览器）库存后台的订单、数据库重新填充及供应商配置的请求。在客户结账程序过程中，会对存货数量进行检查。当存货数量达到最低阈值时，**Inventory bean** 将创建 **BackOrder CMP** 实体 bean。管理员可在 **Plants by WebSphere** 管理页面中处理该后台订单。

在处理来自 **supplierconfig.jsp** 页面的请求时，**AdminServlet sevlet** 会创建 **Suppliers** 无状态会话 bean，用于将请求传递至 **Supplier CMP 2.0 bean**。**Supplier** 配置信息包括姓名、地址、电话及 **Web** 服务的 **Web** 地址。后台订单请求将通过该 **Web** 地址上的 **Web** 服务发送至 **Supplier**。

在处理来自 **help.jsp** 页面的重新填充请求时，**AdminServlet servlet** 将会创建一个名为 **ResetDB** 的 bean 管理的持久性 (**BMP**) bean。该 bean 会删除全部的数据库表格并使用来自 **pbw.properties** 属性文件的初始数据值重新填充这些表格。

企业 bean

本节介绍了 Plants by Websphere 应用程序所用企业 bean 的相关信息。

► Catalog

Catalog 是一个无状态会话 bean。该 bean 是 Inventory 容器管理的持久性 (CMP) 实体 bean 的主要访问 bean。无状态会话 bean 通常会访问实体 bean 数据，同时限制所用事务的数量。Catalog 会话 bean 具有从一个或多个 Inventory bean 获取信息的业务方法。此类方法包括存货项目的添加和移除。Catalog 会话 bean 同时还具有修改现有 Inventory bean 的方法。

► Customer

Customer 是一个 2.0 CMP 实体 bean。该 bean 包含并管理客户所需的帐户数据。Customer 实体 bean 具有创建、查找并更新客户信息以及验证密码并获取 Customer 实体 bean 中的字段的相应方法。

► Inventory

Inventory 是一个 2.0 CMP 实体 bean。该实体 bean 包含并管理库存物品数据。其中可用的方法包括查找、创建、获取和设置数据字段。

► Login

Login 是一个无状态会话 bean。该会话 bean 能够与 Customer CMP 实体 bean 进行交互，以创建并更新客户帐户。

► Mailer

Mailer 是一个无状态会话 bean。该会话 bean 能够使用 JavaMail API 创建并发送用于确认订单的电子邮件。

► Order

Order 是一个 2.0 CMP 实体 bean。该会话 bean 包含和管理订单数据。其中可用的方法包括查找、创建和获取数据字段。

► OrderItem

OrderItem 是一个 2.0 CMP 实体 bean。该实体 bean 包含并管理单个订单物品。其中可用的方法包括查找、创建和获取数据字段。

► ReportGenerator

ReportGenerator 是一个无状态会话 bean。该会话 bean 能够根据在订单中找到的信息生成报告。ReportGenerator 会话 bean 能够与 Order CMP 实体 bean 进行交互。该报告由设定日期范围内的最畅销物品及设定日期范围内的最畅销邮编构成。

► ShoppingCart

ShoppingCart 是一个有状态会话 bean。该会话 bean 用于在整个 HTTP 会话过程中维护待购买的物品列表。ShoppingCart 会话 bean 具有添加、移除和更新物品的业务方法。ShoppingCart 会话 bean 还具有在客户将要完成购买时创建 Order CMP 实体 bean 的方法。

► BackOrder

BackOrder 是一个 2.0 CMP 实体 bean。该实体 bean 包含和管理后台订单数据。其中可用的方法包括查找、创建和获取数据字段。

► BackOrderStock

BackOrderStock 是一个无状态会话 bean。该会话 bean 能够与 BackOrder CMP bean 进行交互，以创建并处理后台订单物品。

► Supplier

Supplier 是一个 2.0 CMP 实体 bean。该实体 bean 包含和处理供应商配置数据。其中可用的方法包括查找、创建和获取数据字段。

- ▶ **Suppliers**

Suppliers 是一个无状态会话 bean。该会话 bean 能够与 Supplier CMP bean 进行交互，以创建和处理 Supplier 配置信息。

- ▶ **ResetDB**

ResetDB 是一个 BMP 实体 bean。该实体 bean 能够删除数据库表中的所有行。

5.2.2 修改和重建 Plants by WebSphere 样例

WebSphere Application Server V6.1 随附的样例应用程序存储于以下目录中。确保将此类样例应用程序作为 WebSphere Application Server V6.1 安装的一部分予以安装。

- ▶ `<profile_root>/samples/src/PlantsByWebSphere`: 包含有 Plants by WebSphere 样例应用程序所用的样例源代码。
- ▶ `<profile_root>/samples/lib/PlantsByWebSphere`: 包含 Plants by WebSphere 样例应用程序所用的 Web 归档 (WAR) 文件、Java 归档 (JAR) 文件及最终 `PlantsByWebSphere.ear` 文件。
- ▶ `<profile_root>/samples/javadoc/PlantsByWebSphere`: 包含构建 Plants by WebSphere 样例应用程序时生成的 Javadoc 文档。
- ▶ `<profile_root>/samples/bin/PlantsByWebSphere`: 包含用于配置和安装 Plants By WebSphere 应用程序的 `wsadmin jac1` 脚本。

z/OS 客户: 由于样例应用程序并非在 z/OS 平台上构建，因此该平台上未提供样例应用程序的源代码树。

在产品安装过程中若选择安装样例应用程序，则 Plants by WebSphere 样例应用程序将会与 WebSphere Application Server 一同安装。Plants by WebSphere 样板应用程序在使用之前，无需予以构建。以下各节说明了重新构建样板应用程序的步骤以及有关修改的更多信息。

修改后的样例应用程序: WebSphere Application Server V6.1 的 Plants By WebSphere 样例应用程序已经过修改，确保其适用于演示迁移过程。您可以按照第 123 页上的附录 B “附加材料”中所述的指示下载此类修改最少的应用程序。

运行样例构建脚本

若要运行用于重新构建样例的样例构建脚本，请完成以下步骤：

1. 打开命令提示符。
2. 在命令行中，将目录切换至 `<profile_root>/samples/src/PlantsByWebSphere` 目录。
3. 运行示例 5-1 中所示的命令，以运行样例构建脚本。

示例 5-1 样例应用程序构建命令

在 Windows 开发平台上，输入以下命令：

```
<profile_root>\samples\bin\PlantsByWebSphere\buildplantsby.bat
```

在 UNIX 和 Linux 开发平台上，输入以下命令：

```
<profile_root>/samples/bin/PlantsByWebSphere/buildplantsby.sh
```

至此，样例构建完成。

构建过程: 在构建过程中，仅会对有变化的类文件进行重新构建。

5.2.3 删除及重新安装 PlantsByWebSphere.ear 文件

缺省情况下, Plants by WebSphere 样例会安装至 `profile_root/installedApps/<host name>` 目录, 其中 `<host name>` 是指安装应用程序之主机的名称。若要重新安装样例, 请完成以下步骤:

1. 停止应用程序服务器。
2. 运行以下示例中所述的命令, 以删除 Plants by WebSphere 样例应用程序 EAR 文件。

示例 5-2 给出了可在 Windows 操作系统上运行的命令。

示例 5-2 在 Windows 上卸载 PlantsByWebSphere 样例应用程序的命令

```
cd <profile_root>\bin
    setupCmdLine
```

其中 `<profile_root>` 表示概要文件安装目录。

```
wsadmin -conntype none -c "$AdminApp uninstall PlantsByWebSphere"
```

注意: 必须在 PATH 中设置 `install_root/bin`

示例 5-3 给出了可在 Linux 操作系统上运行的命令。

示例 5-3 在 Linux 上卸载 Plants By WebSphere 样例应用程序的命令

```
cd <profile_root>/bin
    ./setupCmdLine.sh
```

其中 `<profile_root>` 表示概要文件安装目录。

```
wsadmin.sh -conntype none -c "\$AdminApp uninstall PlantsByWebSphere"
```

3. 要重新安装 Plants by WebSphere 样例应用程序, 请完成以下步骤:
 - a. 将目录切换为 `profile_root/samples/lib/PlantsByWebSphere` 目录。
 - b. 找到 `PlantsByWebSphere.ear` 文件。
 - c. 运行以下示例中给出的命令。

示例 5-4 给出了在 Windows 操作系统上安装 Plants By WebSphere 样例应用程序的命令。

示例 5-4 在 Windows 上安装 Plants By WebSphere 样例应用程序的命令

```
install_root/bin/wsadmin -conntype none -c "$AdminApp install
PlantsByWebSphere.ear
    {-appname PlantsByWebSphere -installdir
$(APP_INSTALL_ROOT)/$(CELL) -usedefaultbindings -node <node> -deployejb
    -deployejb.dbtype DERBY_V10}"
```

其中 `<node>` 是安装样例应用程序之节点的名称。

示例 5-5 给出了在 Linux 操作系统上安装 Plants By WebSphere 样例应用程序的命令。

示例 5-5 在 Linux 上安装 Plants By WebSphere 样例应用程序的命令

```
install_root/bin/wsadmin.sh -conntype none -c "\$AdminApp install
PlantsByWebSphere.ear
    {-appname PlantsByWebSphere -installdir
\\$(APP_INSTALL_ROOT)/\\$(CELL) -usedefaultbindings -node <node> -deployejb
    -deployejb.dbtype DERBY_V10}"
```

4. 启动应用程序服务器。
5. 访问以下 URL 来运行应用程序：
`http://localhost:9081/PlantsByWebSphere`

其中：

- localhost 应使用 WebSphere Application Server 已在其上启动且正在运行的系统的主机名或 IP 地址代替。
- 9081 应用运行应用程序的端口号代替。

至此，重新构建的 PlantsByWebSphere.ear 文件已经部署至您的 WebSphere Application Server。

5.2.4 从管理控制台安装 Plants By WebSphere

本节介绍了使用管理控制台安装 Plants By WebSphere 样板应用程序的安装过程。

请完成以下步骤：

1. 登录管理控制台并浏览到应用程序 → 安装新的应用程序。
2. 浏览到重新构建的 PlantsByWebSphere.ear 文件所在的位置。
3. 提供应用程序名称、位置和其他详细信息（图 5-3）。

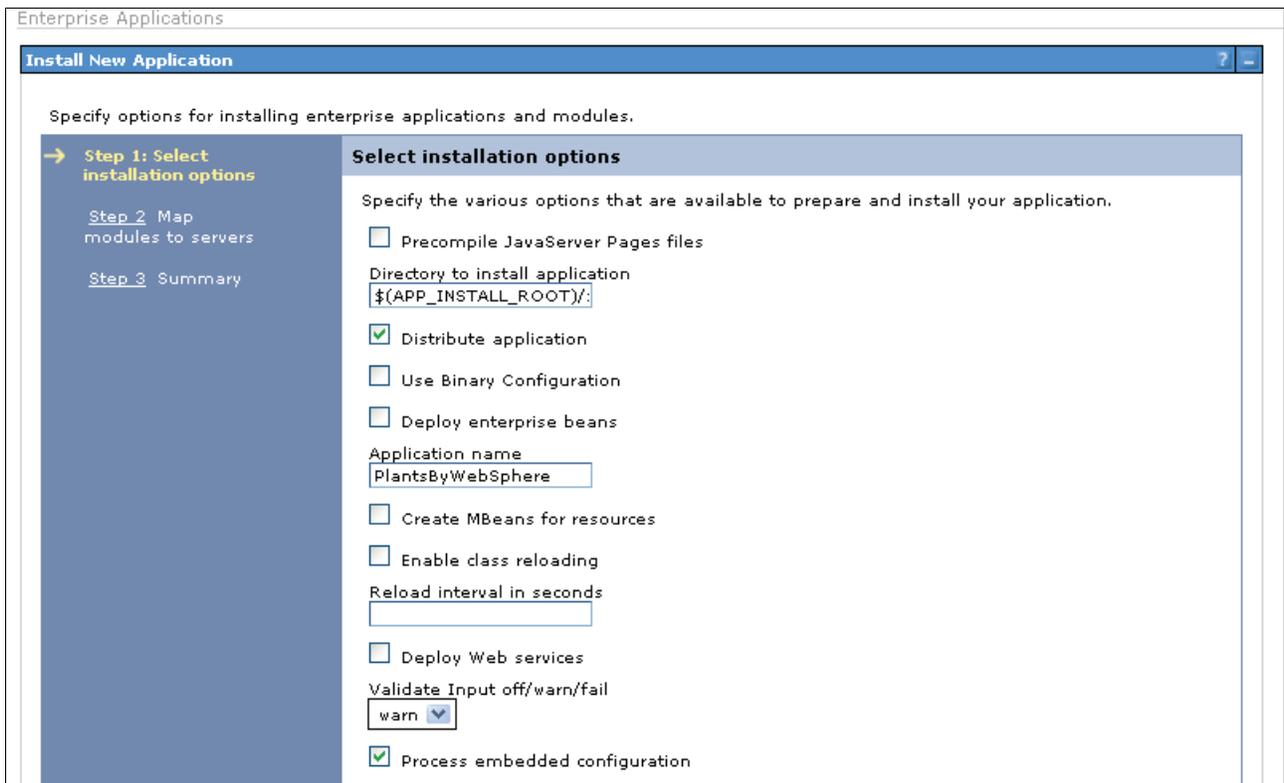


图 5-3 步骤 1：选择安装选项

4. 将应用程序模块映射到目标应用程序服务器（图 5-4）。

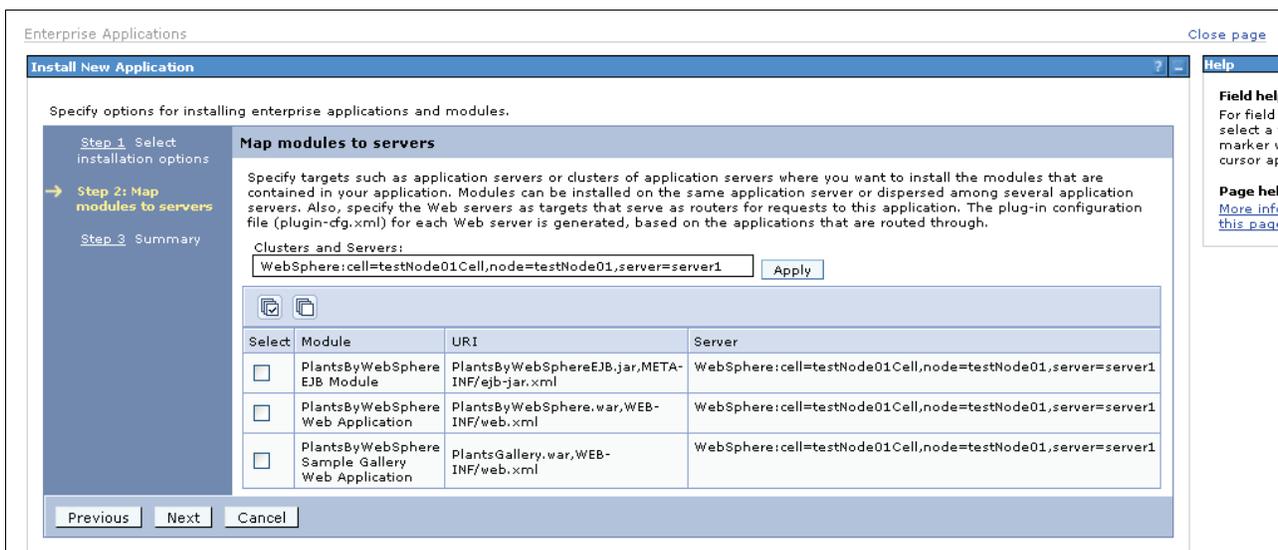


图 5-4 步骤 2: 将模块映射到服务器

5. 检查摘要，完成安装。

电子邮件资源的可选配置

若要发送订单确认电子邮件，您必须为 Plants by WebSphere 样板应用程序配置邮件会话。若不要发送订单确认电子邮件，请跳过本节。

若要配置邮件资源，请完成以下步骤：

1. 在管理控制台中，单击**资源** → **邮件提供程序** → （将范围设置为节点 **<node>**） → **内置邮件提供程序** → **邮件会话**。
2. 单击**新建**创建邮件会话。
3. 在“名称”字段输入 Plants By WebSphere 邮件会话。
4. 在“JNDI 名称”字段输入 mail/PlantsByWebSphere。
5. 在“邮件传输主机”字段输入一个值，如 yourcompany.com。
6. 在“发送邮件地址”字段输入一个值，如 userid@yourcompany.com。
7. 单击**确定**。
8. 单击**保存**（位于管理控制台主面板顶部）。
9. 在“保存至主配置”窗口中单击**保存**。
10. 重新启动 WebSphere Application Server，以应用所有配置更改。

Plants by WebSphere 样例应用程序的邮件资源配置现已完成。

查找数据库表

本样例应用程序的数据库表有 CUSTOMER、INVENTORY、ORDERINFO、ORDERITEM、IDGENERATOR、BACKORDER 和 SUPPLIER。这些表建在 PLANTSDB 数据库中。PLANTSDB 数据库在 PlantsByWebSphere.ear 文件中，位于 profile_root/installedApps/cellname/PlantsByWebSphere.ear/Database/PLANTSDB 目录中，其中单元名称为安装应用程序安装的单元的名称。

5.2.5 运行样例应用程序

启动应用程序并通过访问以下 URL 进入 Plans By WebSphere:

<http://localhost:9081/PlantsByWebSphere>

其中:

- ▶ localhost 应使用 WebSphere Application Server 已在其上启动且正在运行的系统的主机名或 IP 地址代替。
- ▶ 9081 应用运行应用程序的端口号代替。

主索引页将会打开 (图 5-5)。

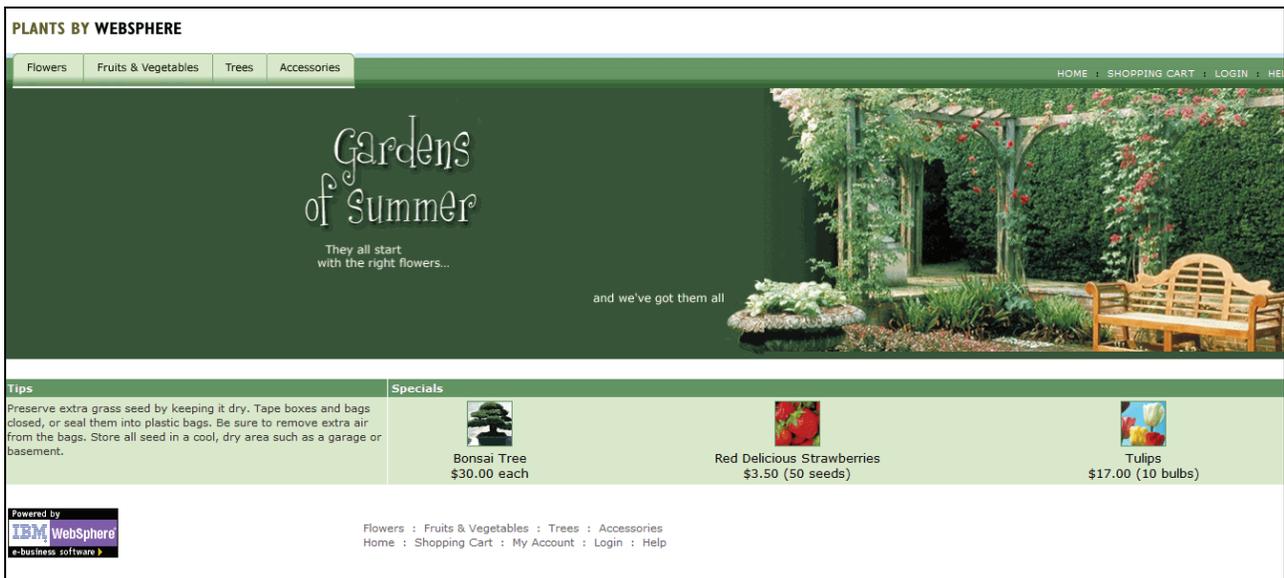


图 5-5 Plants By WebSphere 样例应用程序的索引页

在“登录”窗口上单击注册即可创建 Plants by WebSphere 样例应用程序的唯一电子邮件地址和密码 (图 5-6)。

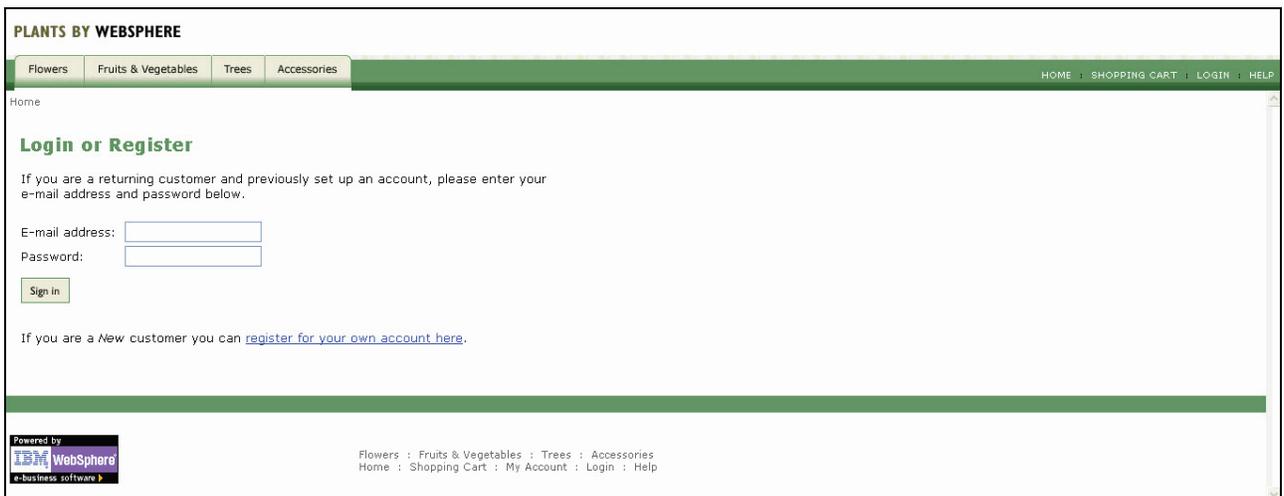


图 5-6 Plants By WebSphere 样板应用程序的登录/注册页

登录帐户后，客户可开始购物并添加物品到购物车（图 5-7）。

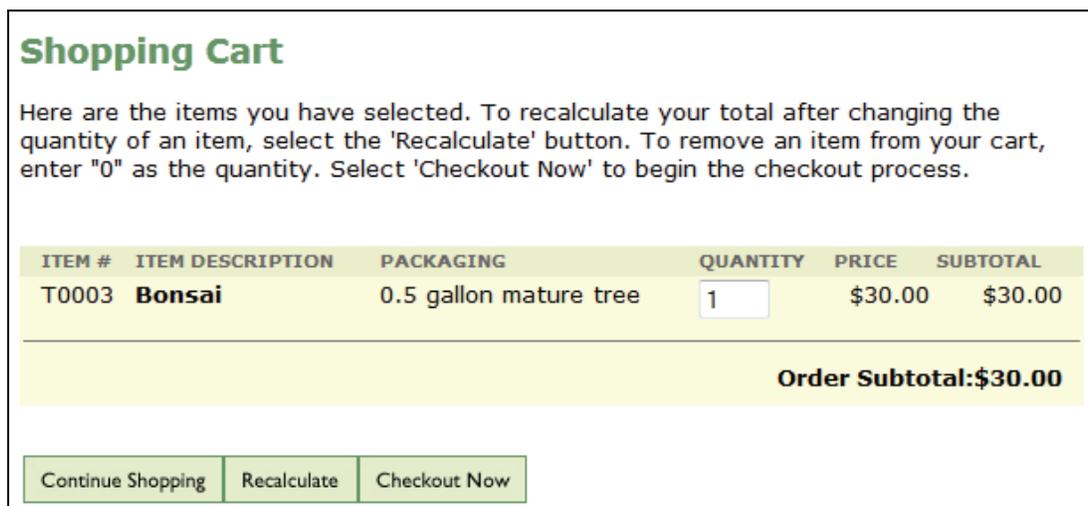


图 5-7 Plants By WebSphere 的购物车

将所需物品添加到购物车后，客户提供付款方式和送货地址来确认订单并进行结帐。最终显示订单成功信息，提供订单号和所购买物品的预计到货日期。

5.3 Plants By WebSphere 的迁移

本章节详细介绍了如何用 WebSphere Application Migration Toolkit - WebSphere Version to Version 迁移对 Plants By WebSphere 样例应用程序进行迁移的过程。

5.3.1 导出 PlantsByWebSphere.ear 以及将其导入 Eclipse 以进行分析

本节详细介绍了如何导出 PlantsByWebSphere.ear 文件并将其导入 Eclipse。

导出 PlantsByWebSphere.ear 文件

请完成以下步骤：

1. 登录管理控制台并浏览到**企业应用程序**。
2. 选择 **PlantsByWebSphere** 应用程序。
3. 单击**导出**和**下载** PlantsByWebSphere.ear 文件（图 5-8）。

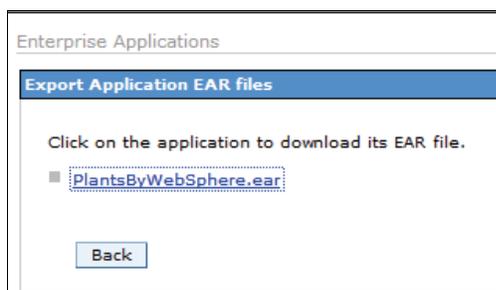


图 5-8 从 WebSphere Application Server 管理控制台中导出 PlantsByWebSphere.ear 文件

4. 或者，您也可以从 <profile_root>/samples/lib/PlantsByWebSphere> 中找到 EAR 文件。

导入 PlantsByWebSphere.ear 文件

将 EAR 文件和源代码一起导入 Eclipse Java EE IDE 中，开始 Plants By WebSphere 应用程序的迁移。

注意： Application Migration Tool 需要有源代码才能执行 Java 代码分析。

请完成以下步骤：

1. 从 Eclipse Java EE Enterprise Explorer 透视图中，单击文件 → 导入 → **EAR 文件**（在 Java EE 中），然后单击**下一步**。您将导入基于选定资源的企业应用程序项目，如图 5-9 所示。

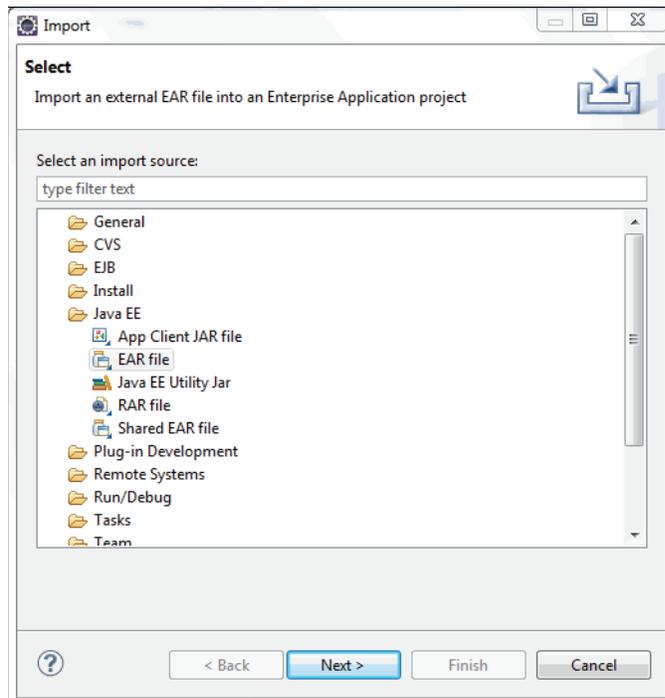


图 5-9 导入 EAR 文件

- 单击**浏览**，然后选择 PlantsByWebSphere.ear 文件。在本例中，该文件位于 C:\test\PlantsByWebSphere.ear（图 5-10）。

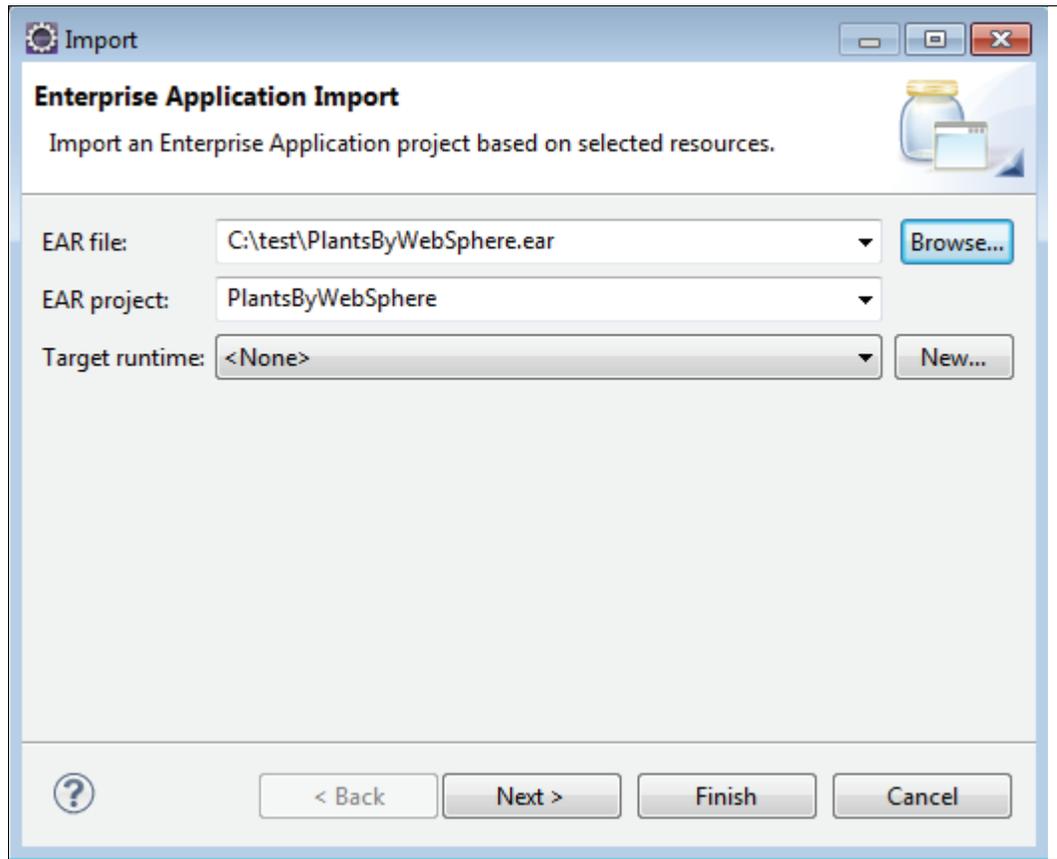


图 5-10 选择要导入的 EAR 文件

3. 在“Ear 模块和实用程序 JAR 项目”窗口中，确定三个模块已全部选定（图 5-11），然后单击完成。

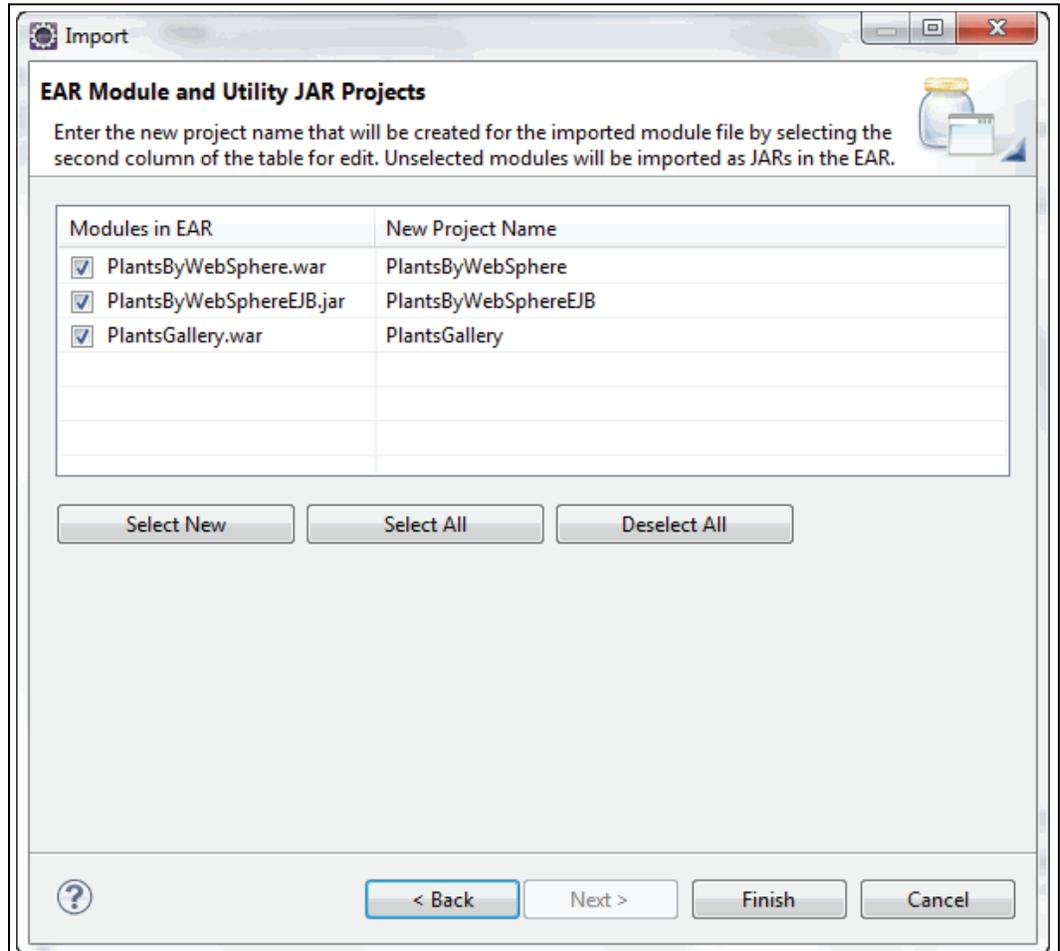


图 5-11 应用程序导入过程中的 EAR 模块和实用程序 JAR 项目

5.3.2 配置和运行 Application Migration Tool - WebSphere Version to Version

在本章节中，我们将配置 Application Migration Tool 分析功能以扫描 WebSphere Application Server V6.1 中 Plants By WebSphere 应用程序，并提供为确保应用程序可在 WebSphere Application Server V8.5 中运行而必须予以修正的相关信息。

必须安装 Application Migration Tool 以查找与 Java 6 或 Java 7 代码相关的问题。但是，Application Migration Tool 也可进行全面的 Java 代码复查。该工具为标记为违反迁移规则的问题提供了快速修正（如可用）。

快速修正：该工具并未对所有可能的情景提供快速修正。当出现了违反规则的情况而又没有可用的快速修正时，规则图标会包括一个亮起的灯泡图标，在上下文菜单中没有快速修正选项。

请完成以下步骤：

1. 从运行菜单中选择分析，显示 Application Migration Tool 主配置对话框（图 5-12）。

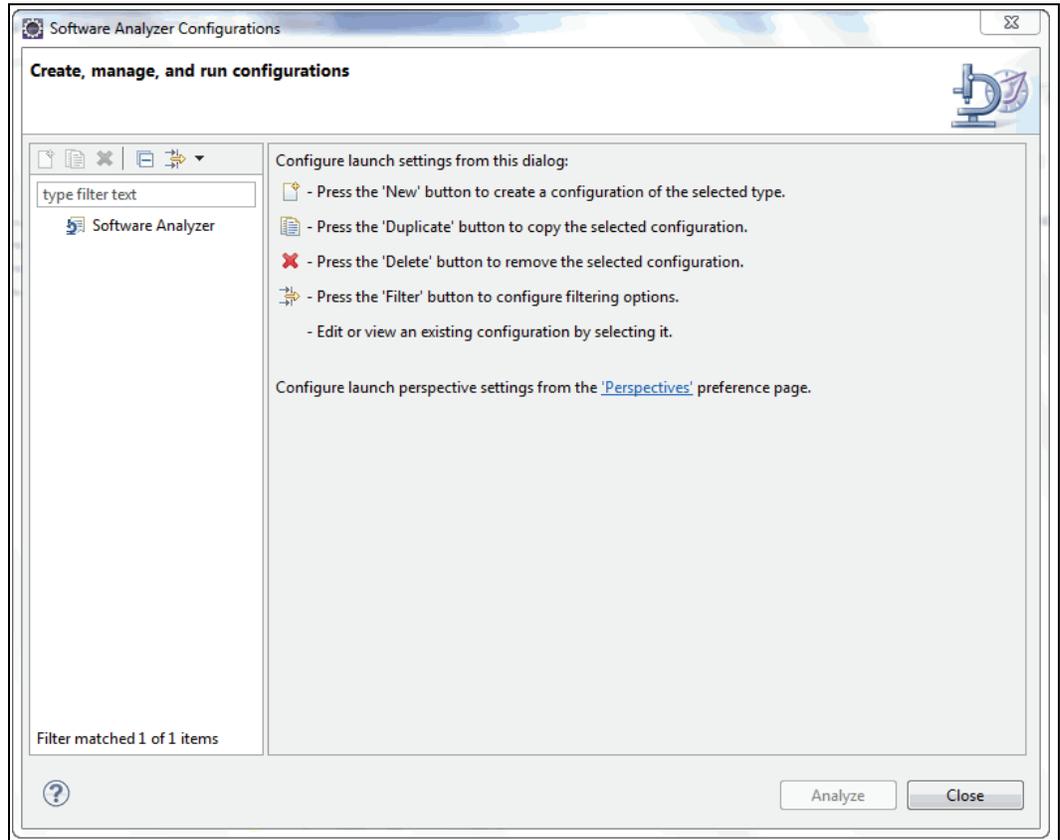


图 5-12 Software Analyzer 配置

2. 右键单击 **Software Analyzer** 并选择**新建**选项在 **Software Analyzer** 下创建一个 **New_configuration**。您也可以突出显示 **Software Analyzer** 并单击工具栏中的“新建”图标（第一个带 + 符号志的图标）。右侧更改为显示可用的基本配置选项的说明。
3. 在“名称”字段，输入配置名称 **PlantsVersiontoVersion**。

4. 在“范围”选项卡中，让**分析整个工作空间**单选按钮保留选中状态（图 5-13）。

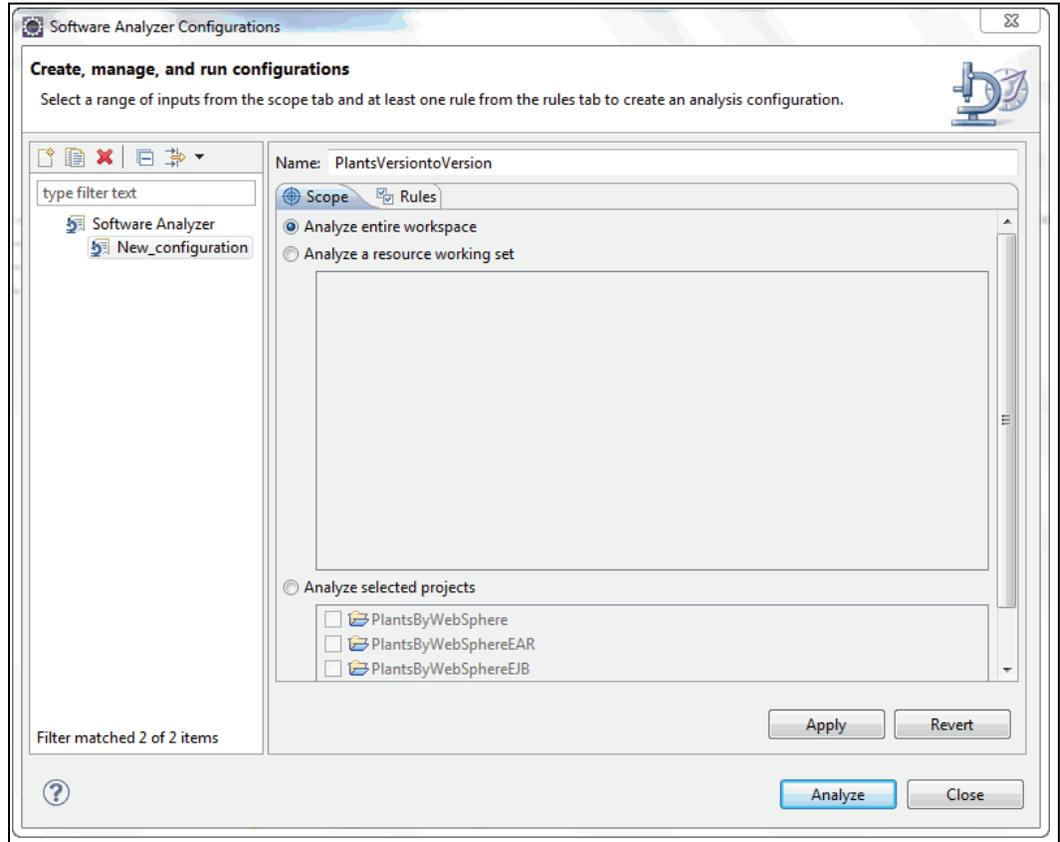


图 5-13 设置新的 *Software Analyzer* 配置

5. 单击**规则**选项卡。

6. 从“规则集”下拉菜单中选择 **WebSphere Application Server 版本迁移**，然后单击**设置**（图 5-14）。

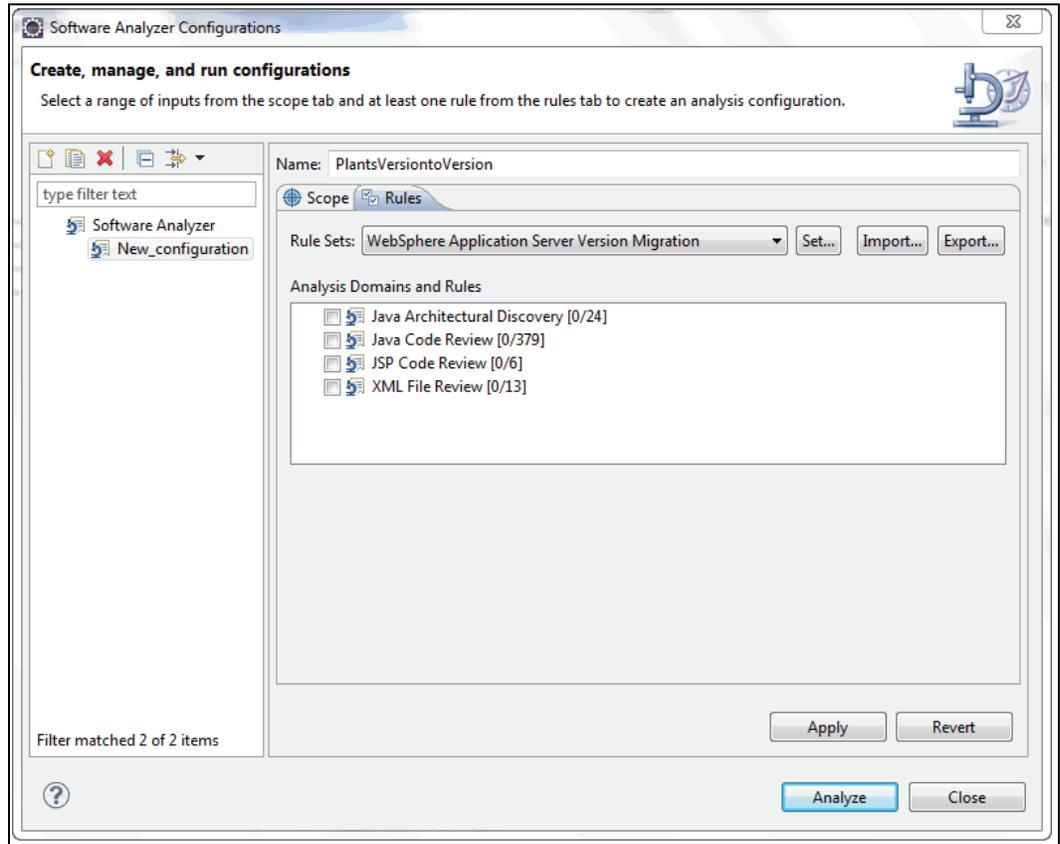


图 5-14 设置 Software Analyzer 配置的规则

7. 将看到“规则集配置”对话框（图 5-15）。将“来源应用程序服务器”设置为 **WebSphere Application Server V6.1**，将“目标应用程序服务器”设置为 **WebSphere Application Server V8.5**。来源 **Java 版本**。可选择目标 **Java 版本** 类别，如图 5-15 所示。单击**确定**。

规则： 缺省情况下，在用户选择所需规则然后单击**设置**之前，没有规则处于选中状态。设置配置后，配置可重复使用。

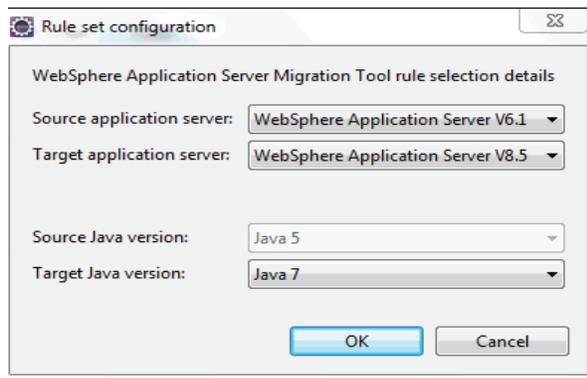


图 5-15 规则集配置对话框

8. Application Migration Tool 现已配置完毕，并且规则设置成功。您可以对 Plants By WebSphere 样例应用程序运行分析。您可以展开任何分析域查看选择了什么规则（图 5-16）。

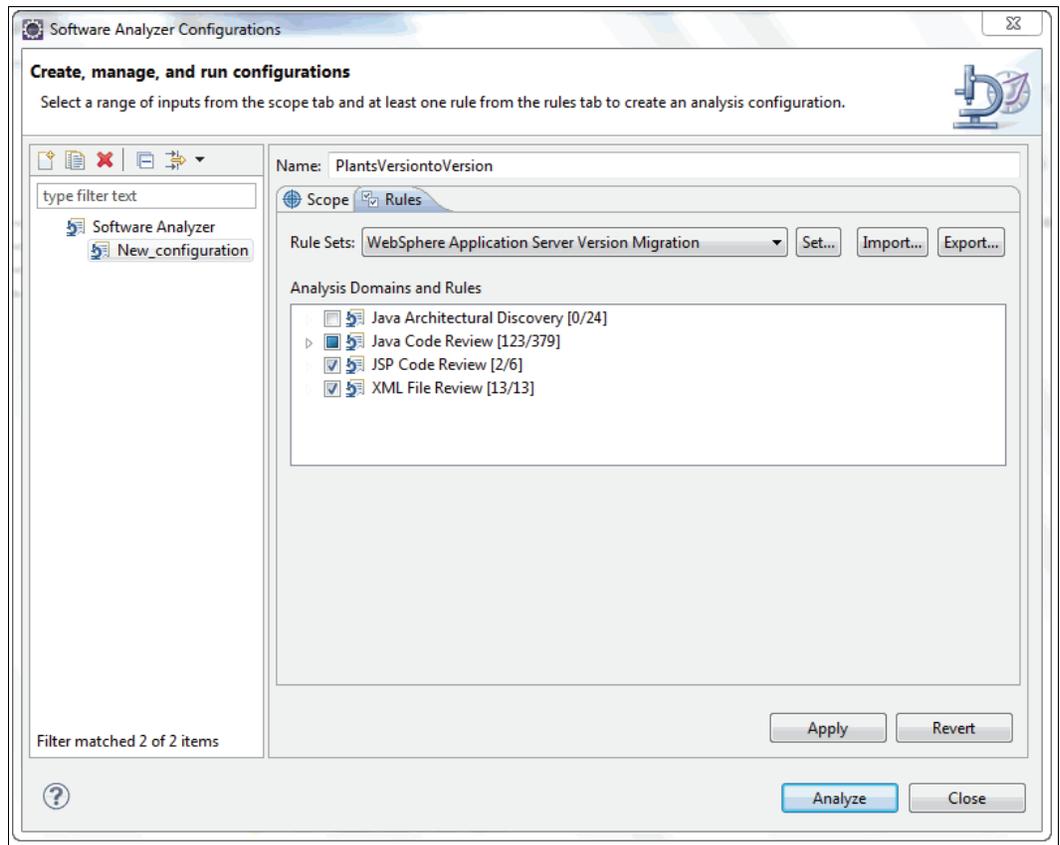


图 5-16 设置完毕的 Software Analyzer 配置

9. 单击**分析**来对工作空间中的所有项目进行分析。Software Analyzer 视图在窗口的底部打开。视图中有三个选项卡。根据之前选择的规则集，每个选项卡对应一个选定的分析域。
- “Java 代码复查”选项卡中列出了 Java 代码相关问题。
 - “XML 文件复查”选项卡中列出了部署描述符的相关问题。
 - “JSP 代码复查”选项卡中列出了在 JSP 文件中发现的通常与 Web 项目有关的问题。
- 分析结果显示在图 5-17 中。

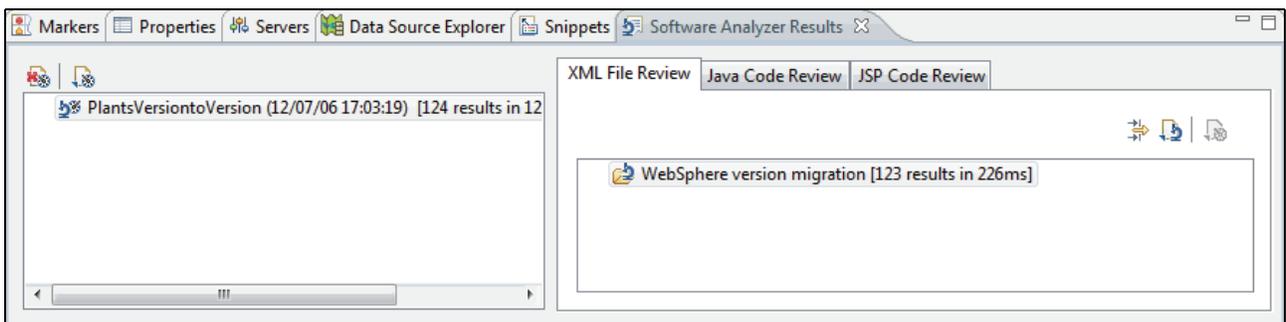


图 5-17 Software Analyzer 结果

10. 单击 **Java 代码复查** 选项卡。

可以看到，Application Migration Tool 检测到不同版本的 Java 之间的兼容性问题（图 5-18）。

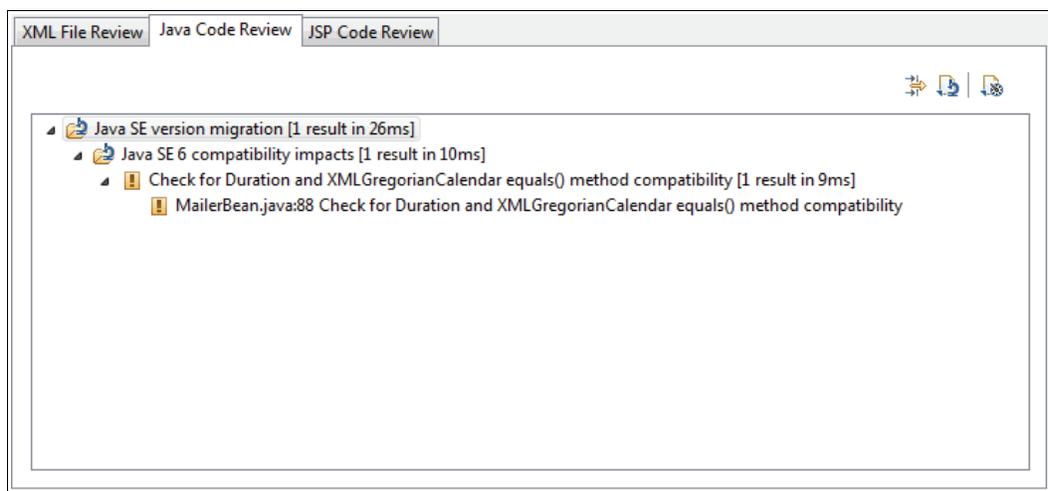


图 5-18 Java 代码复查

11. 单击 **XML 文件复查** 选项卡。

该工具扫描了项目中的 XML 文件，并按类别 / 违反规则标记出了潜在的部署描述符问题。您可以展开每个规则查看有问题的文件及其所属项目（图 5-19）。

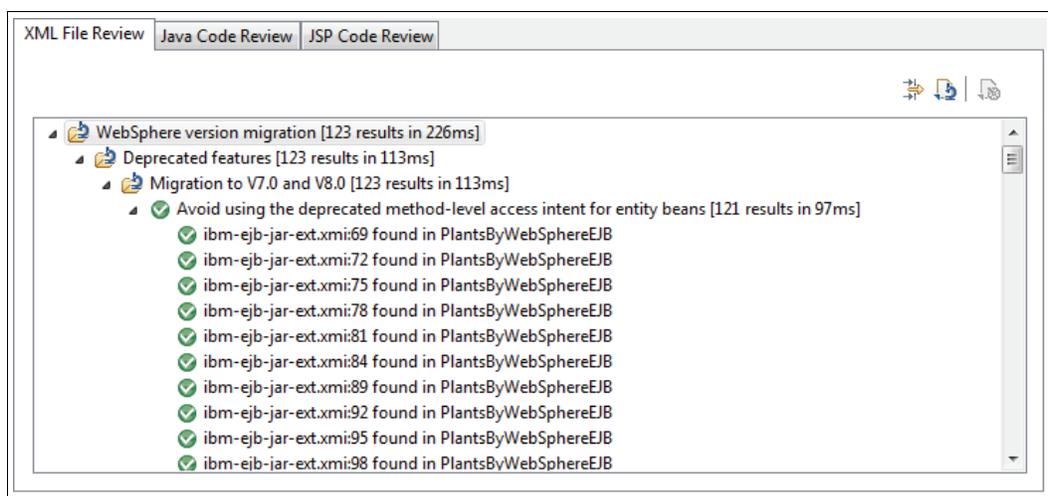


图 5-19 XML 文件复查

您可以右键单击某规则下的标记文件，查看菜单中的可用选项列表。例如，在“Java 代码复查”选项卡中，展开“Java SE 版本迁移”并右键单击列表中的任意文件。然后选择**查看结果**。有问题的 Java 文件便会打开，光标位于有问题的代码行。

5.3.3 修正由 Application Migration Tool 报告的错误。

本章节介绍了修正由 Application Migration Tool 报告的错误的过程。

修正在“Java 代码复查”选项卡下报告的错误

展开“Java 代码复查”部分，显示关于 Java SE 6 兼容性影响结果的详细信息。右键单击各个结果即可访问各个选项，诸如查看出现问题的源代码或使用提供的修正来更正问题。

若要修正在“Java 代码复查”选项卡下报告的错误，请完成以下步骤：

1. 右键单击某个结果并选择**查看结果**。
2. 一个编辑器将会打开，并显示触发规则的源文件。问题的原因会突出显示，并在编辑器的左边缘显示违反规则图标。（图 5-20）

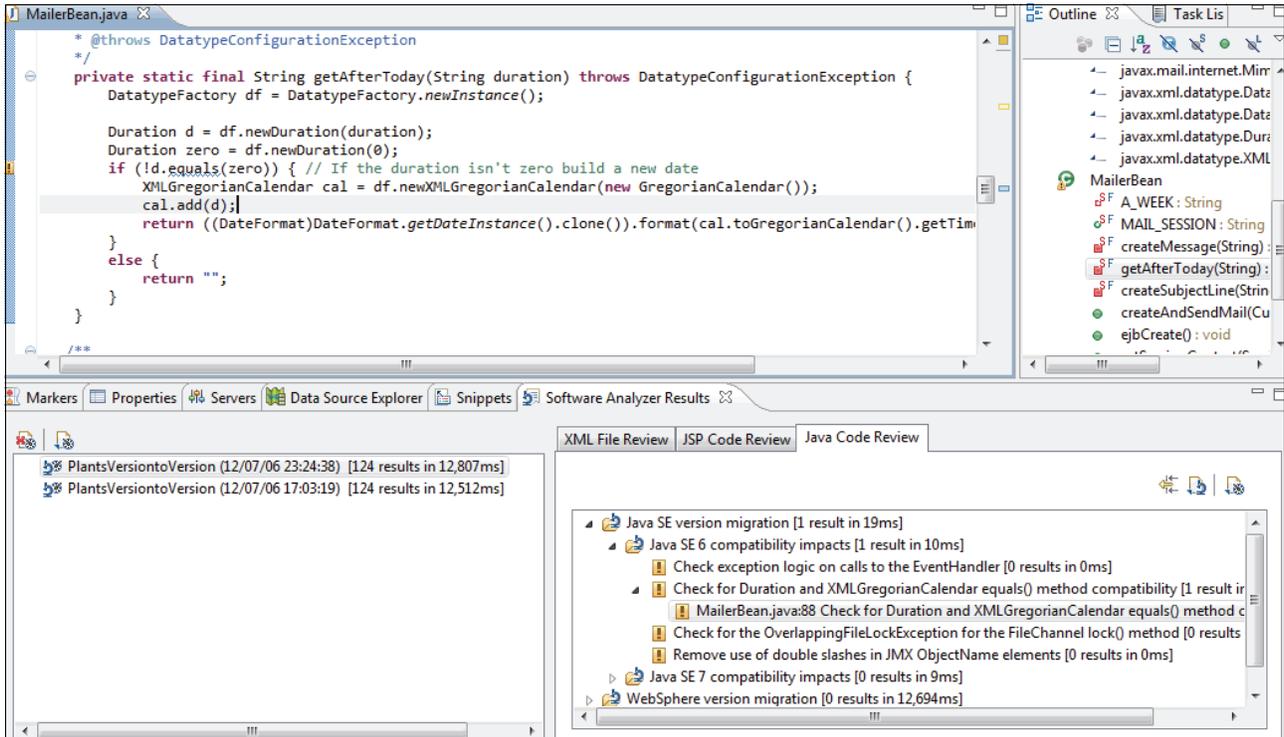


图 5-20 查看 Java 代码复查结果

此处有问题的代码显示在示例 5-6 中。

示例 5-6 待分析的 Java 代码

```
Duration d = df.newDuration(duration);
Duration zero = df.newDuration(0);
if (!d.equals(zero))
```

这个规则标记了 equals(Object param) 方法在 javax.xml.datatype.Duration 或 javax.xml.datatype.XMLGregorianCalendar 中的使用。如果传递的参数为空值，则 Java 6 现在将返回 false。先前版本的 Java 会抛出 NullPointerException。检查应用程序逻辑，查看测试代码时是否返回 false 而不是抛出 NullPointerException。

在示例 5-7 中，标记了 super.equals(someObject) 调用。

示例 5-7 导致标记的 Duration 类 equals 方法的使用

```
public MyClass extends Duration{
private void doX(){
boolean b = super.equals(someObject);
}
}
```

在示例 5-8 中，标记了 c.equals(someObject) 调用。

示例 5-8 导航标记的 Duration 类 equals 方法的使用

```
public MyClass {
private void doX(){
XMLGregorianCalendar c = getCalendar();
boolean b = c.equals(someObject);
}
}
```

由于代码不需要修正，清除“Java 代码复查”结果时可忽略结果并重新运行分析结果（图 5-21）。

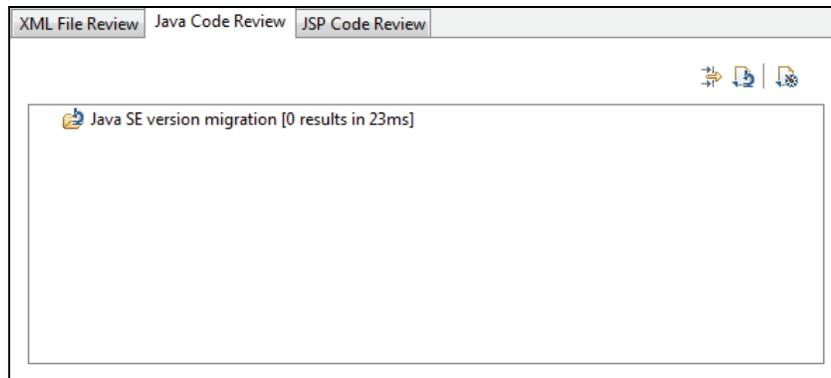


图 5-21 清除 Java 代码复查结果

修正在 XML 文件复查选项卡下报告的错误

根据报告结果，您应避免使用不建议使用的实体 bean 方法级别的访问意向 (access intent)。该规则标记了实体 bean 方法级别访问意向配置。建议不要使用方法级别访问意向，因为它可能会导致死锁等数据访问问题。若要修正重新配置实体 bean，请使用 bean 级别访问意向，如图 5-22 所示。

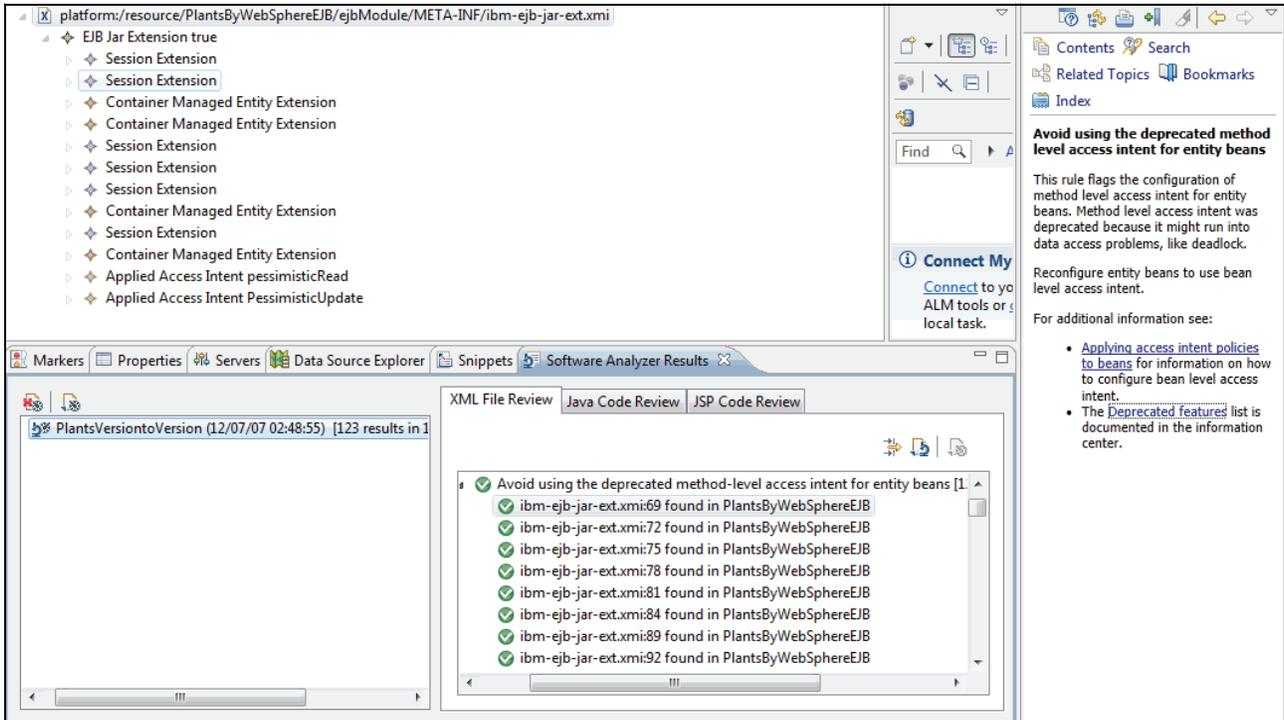


图 5-22 修正 XML 文件复查结果

5.3.4 Application Migration Tool 未检测到的其他迁移问题

虽然 Application Migration Tool 可检测很多迁移问题，但还有一些 Application Migration Tool 目前不能检测到的迁移问题：

- ▶ 最初为 WebSphere Application Server V5.1 之前的版本编写的应用程序的迁移问题
- ▶ 作为代码片段包含在 JSP 中的 Java 代码迁移问题
- ▶ 通过使用 Java 反射隐藏的 Java 代码迁移问题

如果您怀疑正在迁移的应用程序有上述任意一种问题的代码，您必须实施一个计划来管理未检测到迁移问题的风险，例如，对受影响的应用程序或应用程序部分实施手动代码复查计划。

实际上，Plants By WebSphere 应用程序在其中一个 JSP 文件中有一个此类问题。error.jsp 文件含有一部分 Java 代码，如示例 5-9 所示。这部分代码使用 Java 反射在 WebSphere Application Server 提供的 ServletException 中调用 getStackTrace 方法。该方法在 WebSphere Application Server V5.1 作了修改，以避免因 Java 1.4 的变化与 Throwable 类发生冲突。这一特定变化可能会导致应用程序迁移工具未检测到三种潜在的应用程序迁移问题。

示例 5-9 包含在 *PlantsByWebSphere* 中的 *error.jsp* 文件代码片段

```
//Using reflection here so that if the class com.ibm.websphere.servlet.error.ServletErrorReport
//does not exist at compile time there will not be a problem
//if this class does not exist we will just use the attributes specified by Servlet
2.2

Class myClass = Class.forName("com.ibm.websphere.servlet.error.ServletErrorReport");
Method myMethod = myClass.getMethod("getErrorCode", null);
Object o = myMethod.invoke(myReport, null);
status_code = ((Integer) o).intValue();

myMethod = myClass.getMethod("getMessage", null);
o = myMethod.invoke(myReport, null);
message = (java.lang.String) o;

myMethod = myClass.getMethod("getStackTrace", null);
o = myMethod.invoke(myReport, null);
exception_info = (java.lang.String) o;
needInfo = 0;
method = "Using attribute of type com.ibm.websphere.servlet.error.ServletErrorReport
to get information.";
```

该问题的修正可直接进行；将对 `getStackTrace` 的引用更改为替代方法 `getStackTraceAsString`。

其他未检测到的问题： 关于工具箱未检测到的其他问题，请参阅：
<http://www-01.ibm.com/support/docview.wss?rs=180&uid=swg27008724>

5.3.5 导出经修正的 **Plants By WebSphere** 应用程序并在 **WebSphere Application Server V8.5** 上运行该应用程序

本章节介绍了如何导出已迁移的 **PlantsByWebSphere**。请完成以下步骤：

1. 在 Enterprise Explorer 透视图，从 Eclipse IDE 中右键单击 **PlantsByWebSphereEAR** 项目。
2. 选择**导出**。
3. 选择需要导出 **PlantsByWebSphere.ear** 文件的目标目录并保存应用程序。
4. 启动 **WebSphere Application Server V8.5** 服务器。
5. 登录管理控制台并浏览到**应用程序 → 安装新的应用程序**。
6. 浏览到重新构建的 **PlantsByWebSphere.ear** 文件所在的位置。
7. 提供应用程序名称、位置和其他详细信息。
8. 将应用程序模块映射到目标应用程序服务器。
9. 检查摘要，保存安装。
10. 启动应用程序，访问下列地址进入 **PlantsByWebSphere** :
`http://localhost:9081/PlantsByWebSphere/`

有关在 **WebSphere Application Server** 上安装并运行 **PlantsByWebSphere.ear** 文件的详细分步步骤，请参阅第 79 页上的 5.2.4 “从管理控制台安装 **Plants By WebSphere**” 和第 81 页上的 5.2.5 “运行样例应用程序”。

5.4 概率分布样例

概率分布样例是一个小 JSF 应用程序，您可用各种随机数分布进行一些简单的实验。该应用程序使用作为 Rational Application Developer 的一部分提供的 JSF 组件库 (JWL)。此开发类库仅仅支持使用 JavaServer Pages (JSP) 技术构建的 JSF 页面。该库的最新版本 (V3.1.6 或更高版本) 与 JSF 2.0 规范兼容，可在 WebSphere Application Server V8.0 和 V8 中运行。但是，更新应用程序以使用新的 JSF 2.0 功能时，必须弃用该库。

下载样例应用程序： 有关如何下载本章所用的样板应用程序的详细信息，请参阅第 123 页上的附录 B “附加材料”。

5.4.1 将应用程序导入 Rational Application Developer V8.5 中

使用 Rational Application Developer 在两个版本的 WebSphere Application Server (V6.1 和 V8.5) 中安装应用程序。若要开始迁移过程，请完成以下步骤将应用程序导入 Rational Application Developer 工作空间中：

1. 单击**文件** → **导入 ...**，然后单击 **Java EE** → **EAR 文件**，再单击**下一步**。
2. 单击**浏览**，然后选择 blackswan2.ear 文件。
3. 由于 EAR 文件的文件名与包含在 EAR 文件中的 WAR 文件名一样，Rational Application Developer 发出项目名称命名冲突的警告。若要解决该冲突，请将 EAR 项目的名称更改为 blackswan2ear。
4. 由于您正在将导入的 EAR 文件安装到 WebSphere Application Server V6.1 中，现在请通过单击**新建 ...** 设置运行时。在“新建服务器运行时环境”向导中，选择 **WebSphere Application Server 6.1** 并单击**下一步**。单击**浏览**，选择 WebSphere Application Server V6.1 的安装目录，然后单击**完成**。
5. 返回“企业应用程序导入”向导，单击**完成**。

5.4.2 在 WebSphere Application Server V6.1 上安装应用程序

现在，您可以从 Rational Application Developer 中将应用程序直接安装到您的 WebSphere Application Server V6.1 服务器上。您必须将服务器添加到 Rational Application Developer 中，然后在该服务器中运行应用程序。若要完成此任务（假设您已经在 WebSphere Application Server V6.1 中创建了应用程序服务器概要文件），请完成以下步骤：

1. 在 Rational Application Developer 的“服务器”视图中右键单击，然后选择**新建** → **服务器**。
2. 选择 **WebSphere Application Server V6.1** 并单击**下一步**。
3. 选择您要使用的概要文件名称（我们使用缺省的 AppSrv01），如果您的服务器启用了安全性，请提供管理员标识和密码，再单击**下一步**。
4. 由于我们要在此服务器上运行 blackswan2 应用程序，因此选择 **blackswan2ear** 项目，单击**添加**，然后单击**完成**。

5.4.3 在 WebSphere Application Server V6.1 上测试应用程序

在 Rational Application Developer 中定义服务器并安装应用程序后，可以开始测试应用程序。请完成以下步骤：

1. 展开 **BlackSwan2** 项目并右键单击 `index.html`（在 `WebContent` 文件夹中）。选择 **运行方式** → **在服务器上运行**。
2. 确定 **WebSphere Application Server V6.1** 已选定，然后单击 **完成**。
3. 如果没有启动应用程序服务器，Rational Application Developer 将先启动服务器再启动应用程序，然后打开应用程序的 `index.html` 页面。现在，您可以通过选择分布并输入样本大小，测试应用程序的功能。单击 **提交** 后，相关分布为样本分布，并显示所收集样本的图形。

5.4.4 运行 Application Migration Tool - WebSphere Version to Version

Application Migration Tool - WebSphere Version to Version 可用于识别潜在的迁移问题。若要针对该概率分布样例执行该任务，请完成以下步骤：

1. 右键单击 `stock_quote_example_axis2` 项目并选择 **Software Analyzer** → **Software Analyzer 配置 ...**。
2. 单击 **新建启动配置**。
3. 单击 **规则选项卡**，然后使用“规则集：”下拉菜单选择 **WebSphere Application Server 版本迁移规则集**，再单击 **设置 ...**。
4. 在“规则集”配置窗口中，将“来源应用程序服务器”设置为 **WebSphere Application Server V6.1**，将“目标应用程序服务器”设置为 **WebSphere Application Server V8.5**（图 5-23），然后单击 **下一步**。

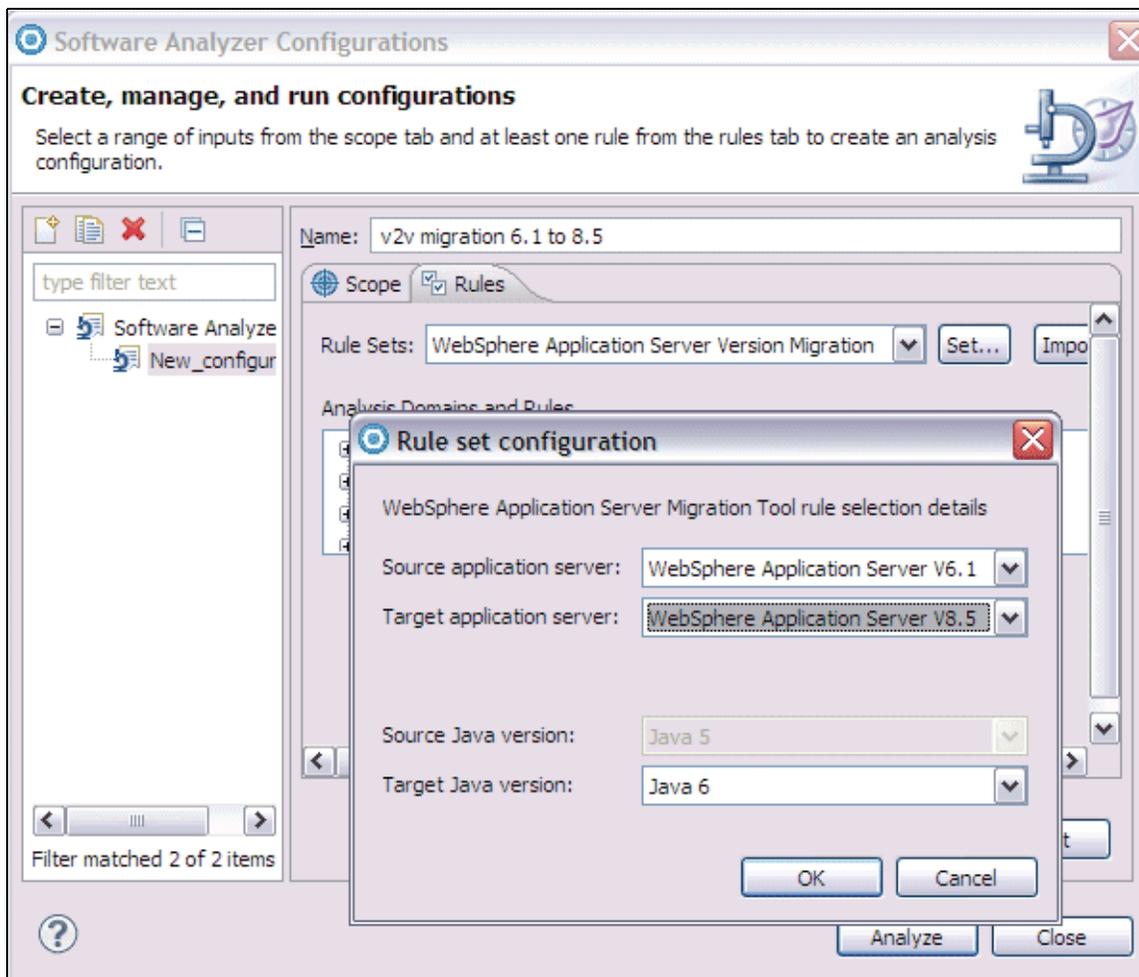


图 5-23 为概率分布样板应用程序配置 Application Migration Toolkit

5. 单击**分析**运行分析。

在此应用程序中，报告了三个潜在问题：

- ▶ 出现两处“避免使用不建议的 IBM 部署描述符扩展重新装入属性”。
- ▶ 出现一处“检查 JavaServer Faces (JSF) 应用程序行为变化”。

5.4.5 解决发现的迁移问题

解决报告的迁移问题是应用程序迁移过程的一个重要步骤。虽然修正问题并不能确保应用程序在迁移完成后能正常工作，但 Application Migration Tool 识别出的问题可能是该应用程序在较高版本的应用程序服务器中导致停止工作的严重问题。

解决不建议使用的重新装入属性

对于两处“避免使用不建议使用的 IBM 部署描述符扩展重新装入属性”的修正，可直接修订。双击各个条目，即可看到检测为潜在问题的 XML 属性。检查示例中的 `reloadInterval` 和 `reloadingEnabled` 属性后，确定这两个属性是应用程序不需要的属性，可以移除（这两个属性最初是在应用程序首次创建时由 Rational Application Developer 添加的）。

检查 JSF 应用程序的行为变化

剩下的问题是“检查 JavaServer Faces (JSF) 应用程序行为变化”的结果。这个问题让您复查 JSF 的应用程序使用以确认其可在 JSF 2.0 运行时中运行。该错误要求您复查 JSF 的使用，其中包括检查应用程序正在使用的所有库以确认它们是否可在 JSF 2.0 中运行。

对于示例应用程序，它使用的是 Rational Application Developer 提供的 JWL。查阅 WebSphere Application Server V8.5 信息中心（网址为 http://www14.software.ibm.com/webapp/wsbroker/redirect?version=phil&product=was-nd-mp&topic=cweb_jsfmigrate）中 JavaServer Faces Migration 处的资料后，我们必须检查应用程序所使用的 JWL 版本是否受支持。JWL 以一组 JAR 文件的形式提供，这些文件位于 Web 内容的 WEB-INF/lib 目录中。JWL 的版本号可在 jsf-ibm.jar 文件的清单中找到。若要查看此版本号，请完成以下步骤：

1. 右键单击 JAR 文件，然后单击**导出 ...**。
2. 展开**常规**，选择**文件系统**，然后单击**下一步**。
3. 在“至目录”输入字段中输入一个临时目录，然后单击**完成**。
4. 在命令行中，让该临时目录成为当前目录，然后输入以下命令：

```
jar -xvf jsf-ibm.jar META-INF/MANIFEST.MF
```

我们假设在您的路径中有 JDK。如果没有，则您可使用包含在您的 WebSphere Application Server 安装文件中的 JDK 内的 jar.exe。

5. 将该文件从 JAR 文件中解压缩后，您可用编辑器检查 manifest.mf 文件。版本号在以 Implementation-Version: 开头的行中列出。有关概率分布样本中的 manifest.mf 文件的具体内容，请参阅示例 5-10。

示例 5-10 概率分布样例中 jsf-ibm.jar 的 manifest.mf 文件

```
Manifest-Version:1.0
Ant-Version:Apache Ant 1.8.2
Created-By:IBM Corporation
Specification-Title:JavaServer Faces
Implementation-Title:JSF Widget Library (JWL) - Base
Implementation-Version:JWL v3_1_13
Implementation-Vendor:IBM
Build-Version:20120425.1128
Build-Date:April 25 2012
Copyright-Info:Copyright (c) 2003,2008, International Business Machines
Corporation.All Rights Reserved.
```

在概率分布样例中使用的 JWL 的版本为 V3.1.13，支持在 JSF 2.0 运行时中运行。如果为 V3.1.6 之前的版本，您必须将应用程序中使用的 JWL 版本替换为 Rational Application Developer V8.5 提供的 JWL 最新版本。

如果确认应用程序可在 JSF 2.0 中运行，您可返回 Software Analyzer 结果页面，右键单击**检查 JavaServer Faces (JSF) 应用程序行为变化**问题，然后单击**忽略结果**将该条目从 Software Analyzer 结果中移除。

5.4.6 在 Rational Application Developer 中准备应用程序在 WebSphere Application Server V8.5 上安装

Rational Application Developer 支持将同一个应用程序部署到多个服务器运行时。但是，要支持此类部署，组成应用程序的项目需有正确的项目特性 (Project Facets)。我们的应用程序以 WebSphere Application Server V6.1 为目标，因此我们必须更改项目特性和目标运行时，以支持在 V8.5 版的部署。要完成此任务，请完成以下步骤：

1. 右键单击 **BlackSwan2** 项目并选择**首选项**。
2. 选择属性列表中的**项目特性视图 (Project Facets)**。
3. 清除 **WebSphere Web (扩展)** 特性并单击**应用**。该面在 WebSphere 扩展的部署描述符编辑器中添加了对部署描述符的支持。
4. 在属性列表中选择**目标运行时**视图。
5. 如果在运行时列表中没有显示 WebSphere Application Server V8.5，则单击**新建 ...**，完成以下步骤（我们假设您已经在 WebSphere Application Server V8.5 安装中定义了应用程序服务器概要文件）：
 - a. 选择 **WebSphere Application Server V8.5**（在“新建服务器运行时环境”向导中选择），然后单击**下一步**。
 - b. 单击**浏览 ...**，选择您的 WebSphere Application Server V8.5 安装的根目录，然后单击**完成**。
6. 在“目标运行时”列表中选择 **WebSphere Application Server V8.5** 复选框，然后单击**确定**。

在 blackswan2ear 项目中，必须重复进行这些步骤。

5.4.7 在 WebSphere Application Server V8.5 上安装应用程序

与 WebSphere Application Server V6.1 的安装一样，您需定义一个 V8.5 服务器并在其上添加应用程序。要完成此任务，请完成以下步骤：

1. 在 Rational Application Developer 的“服务器”视图中右键单击，然后选择**新建 → 服务器**。
2. 选择 **WebSphere Application Server V8.51** 并单击**下一步**。
3. 选择您要使用的概要文件名称（我们使用缺省的 AppSrv01），如果您的服务器启用了安全性，请提供管理员标识和密码，再单击**下一步**。
4. 由于我们要在此服务器上运行 blackswan2 应用程序，因此选择 **blackswan2ear** 项目，单击**添加**，然后单击**完成**。

5.4.8 在 WebSphere Application Server V8.5 上测试应用程序

在 Rational Application Developer 中定义服务器并安装应用程序后，可以通过完成以下步骤测试应用程序：

1. 展开 **BlackSwan2** 项目，右键单击 index.html 文件（在 WebContent 文件中），然后选择**运行方式 → 在服务器上运行**。
2. 检查 WebSphere Application Server V8.5 已选定，然后单击**完成**。
3. 如果没有启动应用程序服务器，Rational Application Developer 将先启动服务器再启动应用程序，然后打开应用程序的 index.html 页面。在这里，您可以通过选择分布并输入样本大小，测试应用程序的功能。单击“提交”后，相关分布为样本分布，并显示所收集样本的图形。

5.4.9 迁移后升级应用程序的其他选项

迁移概率分布样例后，有多个选项可对其进行修改。

让应用程序保持不变

一个选项是让应用程序保持不变，或至少将应用程序的更改限制为不需要移除 JWL 库的更改。

此选项适用于在可预见的未来不需要大规模增强或修改的应用程序。

将应用程序更改限制为 JWL 支持的更改

如果启用了使用限制，则在 JSF 2.0 中支持 JSF 窗口小部件库。主要限制是应用程序不能使用 Facelets Web 模板系统。通常，这种限制对这个应用程序而言无足轻重。您通常会在新的 JSF 2.0 应用程序中使用 Facelets，但一般不会将在同一个应用程序中混合使用 Facelets 和非 Facelets JSF 代码。因此，您可保持和增强该应用程序。

移除对 JSF 窗口小部件库的依赖关系

不建议使用 JWL，且它不支持 JSF 2.0 的所有功能（尤其是，使用 Facelets 的所有 JSF 2.0 应用程序也都不能使用 JSF 窗口小部件库）。如果您要更新应用程序以使用添加至 JSF 2.0 的功能，您必须移除 JSF 窗口小部件库并用另一个提供有类似功能的窗口小部件库代替。在示例中，我们使用 JSF 窗口小部件库就从抽样概率分布样本组中抽出的样本组的各个数值出现的次数制作了一个条形图。

本书编著之时，没有替代的 JSF 窗口小部件库，但一些第三方 JSF 窗口小部件库中包含有图表组件。我们需要对这些替代库进行评估，用这些替代库中的某个库中的图表替代我们的条形图，或编写我们自己的 JSF 窗口小部件实现图表功能。

5.5 Web 服务 Axis 2 股票报价样例应用迁移

我们迁移的第三个应用程序是基于 Axis 2 Web 服务引擎的 Web 服务应用程序。Web 服务是一种允许应用程序使用互联网标准和协议进行调用的技术。这种应用程序划分为多层，一般至少有两层：客户机和服务器。这两者之间使用 SOAP 1.1 或 1.2 协议定义的 XML 消息进行通信。

WebSphere Application Server V6.1 为 JAX-RPC 和 JAX-WS 协议提供支持，并提供更多功能，诸如 UDDI 注册中心和称为 WSIF 的 SOAP 协议扩展。

WebSphere Application Server 还支持多个在基础 Web 服务协议之上构建的协议，诸如 WS-Notification 和 WS-ReliableMessaging。

所有这些功能都由应用程序服务器提供的 Web 服务引擎提供。但是，在某些情况下，您必须使用 Axis 2 等第三方 Web 服务引擎。例如，如果您要在多个应用服务器中部署使用同一个 WebServices 服务运行时的应用程序，如 WebSphere Application Server、JBoss 和 WebLogic，您将使用第三方引擎。

IBM 支持允许第三方 JAX-WS 运行时在 WebSphere Application Server 中运行，并确保成功部署使用此类运行时的应用程序。但是，IBM 将不会提供官方技术支持来解决，由于 Java 文件中类冲突和在异常堆栈跟踪日志中显示是第三方类库引起的问题。此外，使用第三方 Web 服务引擎将阻碍应用程序服务器将第三方引擎提供的 Web 服务视作事实上的 Web 服务，进而阻碍 WS-Security、WS-RM 和 WS-Transactions 等应用程序级别策略集的使用，并阻碍使用 WSDM 或 JNDI 查找来检索 JAX-WS 服务或端口实例。

5.5.1 概述

本章节例举了一个使用第三方 Web 服务引擎（即 Axis 2 Web 服务引擎）的应用程序。这个应用程序很简单（实际上，该应用程序只提供了两项 Web 服务），但说明了此类应用程序迁移的潜在问题。

该应用程序提供了一项股票报价服务，客户提供所需股票价格的符号作为输入，即可查询股票价格。您可以使用 Axis 对象模型 (AXIOM) 允许任意 XML 片段在 Web 服务消息中传递和返回。在此例中，简单的 Plain Old Java 对象 (POJO) 可用作 Web 服务的实现以及 Web 服务的定义。这个应用程序的开发者选择对 Web 服务使用 AXIOM 样式。

5.5.2 在 WebSphere Application Server V6.1 上安装

StockQuoteAxis2Example 应用程序包含了 Axis 2 运行时和 StockQuote Web 服务的 WAR 文件。Web 服务以 WAR 文件形式在 WAR 文件的服务目录中提供。在示例中，我们完成以下步骤通过 WebSphere Application Server V6.1 的应用程序控制台安装应用程序：

1. 在左侧窗格任务列表中单击**安装新的应用程序**。
2. 单击**浏览 ...** 并选择 WAR 文件。
3. 在“上下文根”字段输入“stockquote”（图 5-24）。

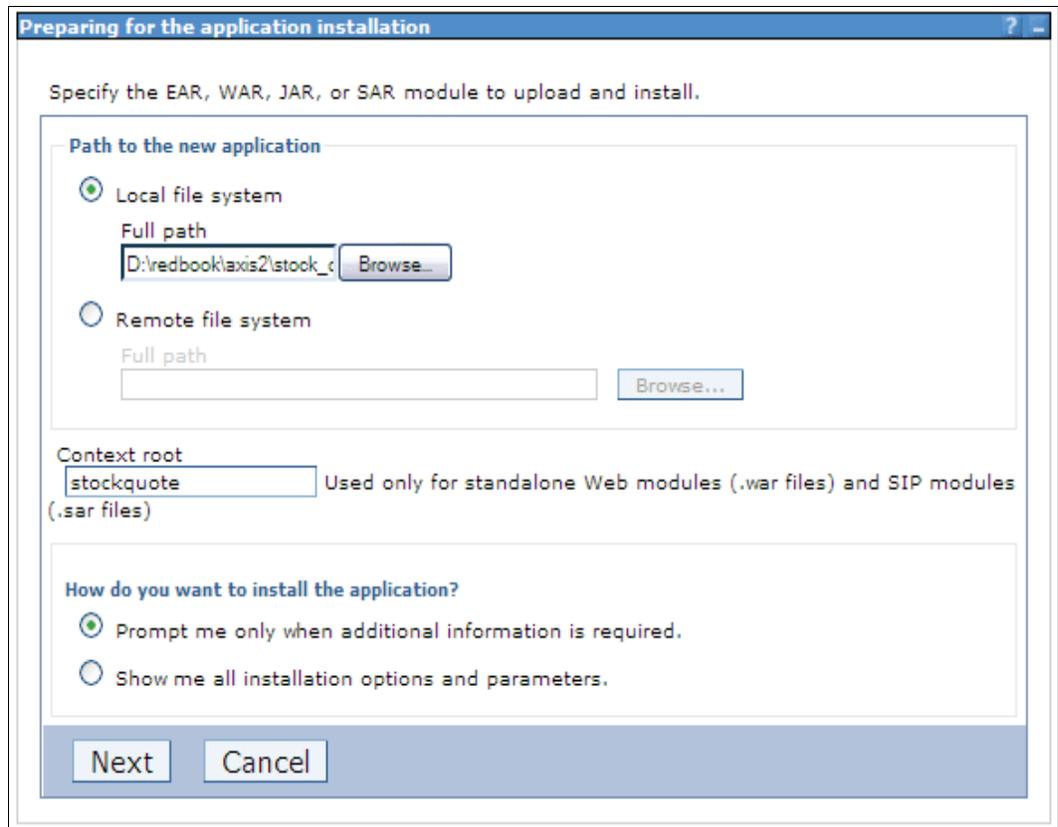


图 5-24 在 WebSphere Application Server V6.1 上安装 Axis 2 应用程序

4. 单击**下一步**。
5. 对于“步骤 1. 选择安装选项”，单击**下一步**。
6. 对于“步骤 2. 将模块映射至服务器”，单击**下一步**。
7. 对于“步骤 3. 为 Web 模块映射虚拟主机”，单击**下一步**。

8. 对于“步骤 4. 为 Web 模块映射上下文根”，单击**下一步**。
9. 对于“步骤 5. 摘要”，单击**完成**。
10. 应用程序安装过程完成后，单击**保存**以将服务器配置更改保存到主配置。

应用程序现已安装完成，必须启动应用程序以对其进行测试。要完成此任务，请完成以下步骤：

11. 在左侧窗格任务列表中单击**企业应用程序**。
12. 选择 stock_quote_example_axis2_war 条目，然后单击**启动**。

5.5.3 在 WebSphere Application Server V6.1 上测试

若要测试 Web 服务是否可运行，请调用应用程序提供的测试客户机程序。由于客户机程序需要在类路径上有 axis2 JAR 文件，从 <http://axis.apache.org/axis2/java/core/download.cgi> 下载 axis2 二进制分发并解压缩下载的压缩文件。然后，将客户机测试程序导入 Eclipse 并使用所下载 axis2 分发的 lib 目录中的 JAR 文件更新项目类路径。

对于具有良好定义的 Web 服务的应用程序，可使用 WebSphere Developer Tool（和在 Rational Application Developer 中）中提供的通用服务客户机或 Web 服务浏览器与 Web 服务直接交互。股票报价应用程序使用 AXIOM 并依赖 axis2 Web 服务引擎的功能生成基于实现签名的 Web 服务定义。但是，此操作会生成不适用于这些测试客户机中任一个使用的定义，因此您不能使用这些测试客户机。幸运的是，该应用程序提供了一组使用 JUnit 的测试，您可用来测试 Web 服务的功能。

测试的第一阶段是验证应用程序使用的 axis2 Web 服务引擎是否正确安装。该引擎以 Web 应用程序的形式提供，并且引擎开发者提供了一组 Web 页面来验证和检查引擎状态。我们可以使用这些页面检查引擎的基本安装是否成功。

若要进行此验证，使用 Web 浏览器导航至 Web 服务引擎的主页。在我们的测试系统中，该页面位于 <http://localhost:9082/stockquote>（端口号取决于应用程序概要文件中定义的 WC_defaulthost 端口）。此操作会显示 axis2 Web 服务引擎的主页。单击**验证**链接，运行 axis2 引擎提供的验证检查。最后，我们导航到“服务”页面验证我们的 StockQuote 服务是否可用。

下一个阶段是测试实际 Web 服务。此操作要求生成实际 Web 服务请求并提交至 Web 服务。在示例中，我们使用现有的一组 JUnit 测试来验证应用程序的功能。若要验证功能，请完成以下步骤：

1. 启动 Eclipse 并单击**文件** → **导入 ...**。
2. 选择**工作空间中的现有项目**，然后选择**下一步**。
3. 选择**归档**并单击**浏览 ...**。
4. 查找包含 StockQuoteClient UnitTest 的压缩文件并单击**完成**。
5. 创建 StockQuoteClientSideUnitTest 项目后，右键单击该项目并选择**构建路径** → **配置构建路径**。
6. 单击**库**选项卡，然后单击**添加外部 Jar**。选择 axis2 二进制分发 lib 目录中的所有 JAR 文件。单击**确定**。

7. 可能需要更新 `StockQuoteClientSideUnitTest`，以提供正确的 Web 服务端点。要完成此任务，请完成以下步骤：
 - a. 展开 **StockQuoteClient** 项目，然后展开 **src** 文件夹。展开股票项目后，双击 `StockQuoteClient.java` 文件。
 - b. 更新 `STOCK_SERVICE_EPR` 常量以获取端点的 URL。您很可能必须更新端口号以匹配您的 WebSphere Application Server V6.1 服务器的端口号。

在示例中，我们基于我们的服务器端口号和应用程序服务器上安装应用程序时提供的上下文根，使用 `http://localhost:9082/stockquote/services/StockQuoteService`。
8. `StockQuoteClient` 由 Eclipse 编译完成并确认无错误后，您可以开始运行测试程序。要完成此任务，请完成以下步骤：
 - a. 展开 **StockQuoteClient** 项目，然后展开 **src** 文件夹。展开股票项目后，右键单击 `StockQuoteClient` 类。
 - b. 单击运行方式 → **JUnit 测试**。

所有 JUnit 测试完成后显示绿色的条，说明 Web 服务成功通过这组测试。

5.5.4 将应用程序源代码导入到 Eclipse

Stock Quote 以包含 axis2 运行时的 WAR 文件方式提供。若要将该文件导入到 Eclipse，请完成以下步骤：

1. 单击文件 → 导入 ...，单击 **Web** → **WAR 文件**，然后单击下一步。
2. 单击浏览，然后查找 WAR 文件。
3. 将目标运行时更改为 **WebSphere Application Server V8.5**。如果您没有为在 **WebSphere Application Server V8.5** 中运行的服务器创建目标运行时，可加以创建。要完成此任务，请完成以下步骤：
 - a. 单击**新建**。
 - a. 打开 IBM 文件夹，选择 **WebSphere Application Server V8.5**，然后单击**下一步**。
 - b. 单击**浏览 ...**，查找您的 WebSphere Application Server V8.5 安装的根目录。单击**完成**返回“WAR 导入”向导。
4. 清除**将项目添加至 EAR**复选框（如果该选项为选定）。
5. 单击**完成**。

5.5.5 运行 Application Migration Tool - WebSphere Version to Version

完成示例中的所有应用程序迁移后，我们即可运行 Application Migration Tool 识别潜在的应用程序代码迁移问题。若要完成此任务，我们需创建并运行一个新的 Software Analyzer 配置。（我们假设工具已安装完毕。）

请完成以下步骤：

1. 右键单击 `stock_quote_example_axis2` 项目并选择 **Software Analyzer** → **Software Analyzer 配置 ...**。
2. 单击**新建启动**。
3. 单击**规则选项卡**，然后使用“规则集：”下拉菜单选择“WebSphere Application Server 版本迁移”规则集。单击**设置 ...**。

- 在“规则集”配置窗口中，将来“源应用程序服务器”设置为 WebSphere Application Server V6.1，将“目标应用程序服务器”设置为 WebSphere Application Server V8.5，如图 5-25 所示。单击**确定**。

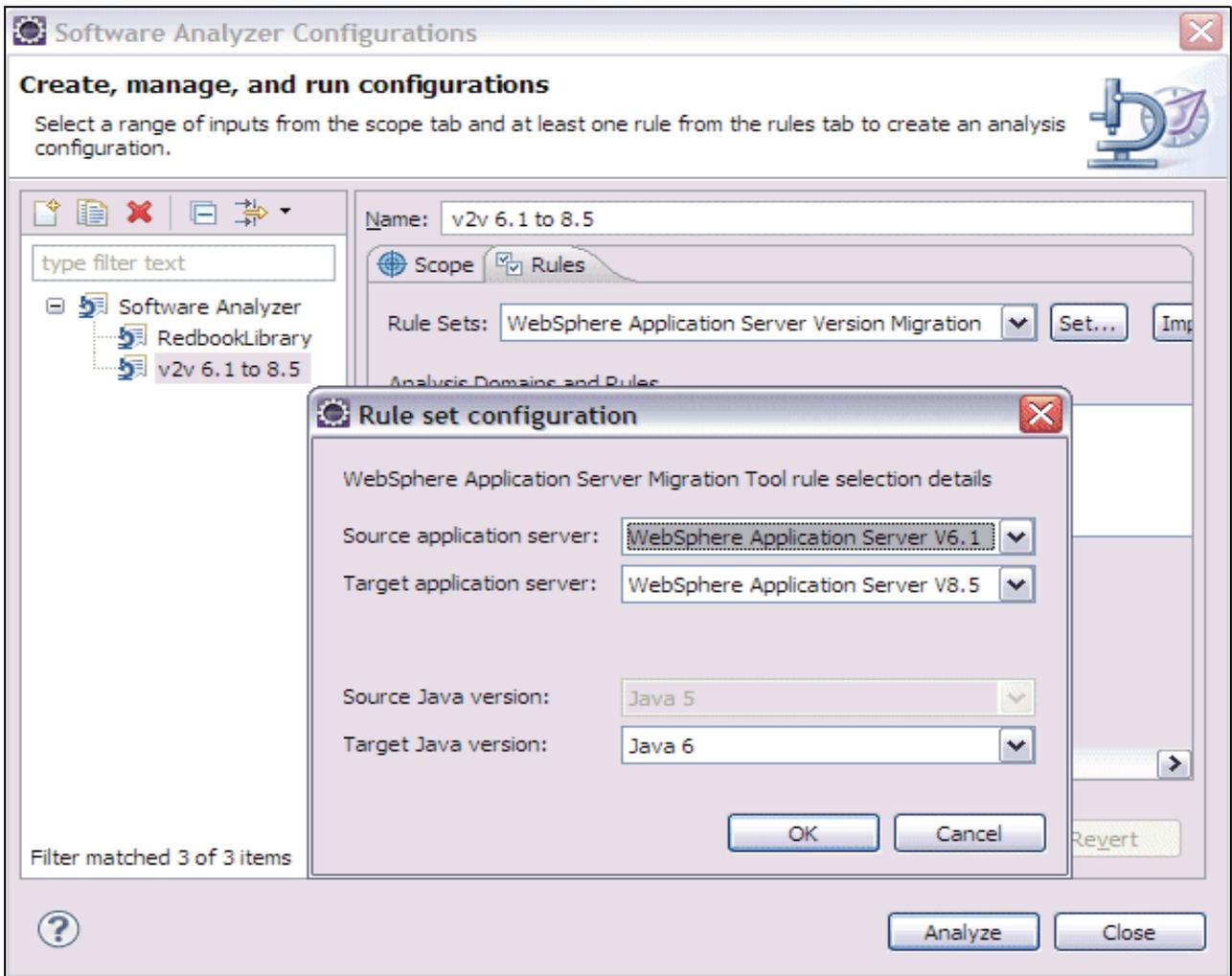


图 5-25 为 Stock Quote Axis2 应用程序配置 Application Migration Toolkit

- 单击**分析**分析工作空间是否存在潜在的迁移问题。

分析报告未检测到任何问题。但是，Application Migration Toolkit V3.5 没有对 axis2 等第三方 Web 服务引擎进行检查，也没有报告称需对该服务引擎的配置进行一些小变更以便应用程序在 WebSphere Application Server V8.5 中顺利运行。

5.5.6 准备在 WebSphere Application Server V8.5 上进行安装

若要在 WebSphere Application V8.5 中安装使用 axis2 Web 服务引擎的应用程序，需对该应用程序进行一些小变更。

更新 axis2.xml 文件以指定 EnableChildFirstClassLoading

从 V1.5.5 的 axis2 Web 服务引擎开始，对 axis2 配置添加了一个新的参数。这个参数控制 axis2 Web 服务引擎创建的类装入器的行为方式。为使 axis2 Web 服务引擎在 WebSphere Application Server V8.5 中正确运行，必须将该参数从缺省值 false 更改为 true。若要完成此任务，请展开 stock_quote_example_axis2 项目并在 WebContent 的 WEB-INF/conf 文件夹中查找 axis2.xml 文件。将 EnableChildFirstClassLoading 属性更新为 true，如示例 5-11 所示。

示例 5-11 axis2.xml 文件中的 EnableChildFirstClassLoading 属性

```
<parametername="EnableChildFirstClassLoading">>false</parameter>
```

禁用针对应用程序的 Web 服务注释扫描功能

在缺省情况下，WebSphere Application Server 对所安装的应用程序进行注释扫描以确定这些应用程序提供的 Web 服务。若您使用的是第三方 Web 服务引擎，则这个扫描功能是多余的，还可能会引发问题。若要禁用应用程序注释扫描功能，更新应用程序 WAR 文件的 MANIFEST.MF 文件以添加 DisableIBMJAXWSEngine 属性，如示例 5-12 所示。

示例 5-12 stock_quote_example_axis2 项目 WebContent 中的 META-INF/MANIFEST.MF 文件

```
Manifest-Version:1.0
DisableIBMJAXWSEngine:true
Archiver-Version:Plexus Archiver
Created-By:Apache Maven
Built-By:sagara
Build-Jdk:1.6.0_22
```

禁用注释扫描：修改 META-INF/MANIFEST.MF 文件的替代方法是，您可以在管理控制台设置 JVM 属性以关闭整个 WebSphere Application Server 实例的注释扫描功能。若要完成此任务，请单击 **应用程序服务器** → **服务器** → **流程定义** → **Java 虚拟机**，然后将下面一行添加至“通用 JVM 自变量”字段：

```
-Dcom.ibm.websphere.webservices
```

导出可在 V8.5 上安装的 WAR 文件

由于应用程序已打包成一个简单的 WAR 文件，应用程序不能直接从 Eclipse 安装到应用程序服务器上。必须将其以 WAR 文件导出。要完成此任务，请完成以下步骤：

1. 右键单击 **stock_quote_example_axis2** 项目，然后单击 **导出 ...**。
2. 为导出的 WAR 文件提供一个文件名。所指定的名称和位置不应与安装在 WebSphere Application Server V6.1 上的原始 WAR 文件的相同，以便您保留原始 WAR 文件，在将应用程序重新导入到 Eclipse 时使用。在示例中，我们将文件命名为 **stock_quote_example_axis2_85.war**。
3. 单击 **完成**。

5.5.7 在 WebSphere Application Server V8.5 上安装

在示例中，我们使用管理控制台将应用程序安装到 WebSphere Application Server V8.5 上：

1. 在左侧窗格任务列表中单击**新建应用程序**。
2. 单击**新建企业应用程序**链接。
3. 单击**浏览**，然后选择 WAR 文件。
4. 单击**下一步**。
5. 在“您要如何安装应用程序？”窗口中单击**下一步**。
6. 对于“步骤 1. 选择安装选项”，单击**下一步**。
7. 对于“步骤 2. 将模块映射至服务器”，单击**下一步**。
8. 对于“步骤 3. 为 Web 模块映射虚拟主机”，单击**下一步**。
9. 对于“步骤 4. 为 Web 模块映射上下文根”，对上下文根输入 `/stockquote`（此条目与在 V6.1 上安装时指定的上下文根匹配），然后单击**下一步**。
10. 对于“步骤 5. 摘要”，单击**完成**。
11. 应用程序安装过程完成后，单击**保存**以将服务器配置更改保存到主配置。

5.5.8 配置 WebSphere Application Server V8.5 以支持应用程序

若要解决 axis2 Web 服务引擎与 WebSphere Application Server V8.5 运行时内部使用的代码之间的冲突，我们必须将使用 axis2 Web 服务引擎的应用程序配置为父类最后类加载策略（Parent Last）。

若要更新应用程序的配置，请完成以下步骤：

1. 在 WAS 管理控制台 Web 界面左侧窗格任务列表中单击 **WebSphere 企业应用程序**（您可能需要展开**应用程序类型**才能找到该任务）。
2. 单击 `stock_quote_example_axis2_85_war`，然后单击**管理模块**和 **Apache-Axis2**。

3. 使用“类装入器顺序”下拉菜单，选择按本地类装入器优先（父类最后）装入的类，如图 5-26 所示，然后单击确定。

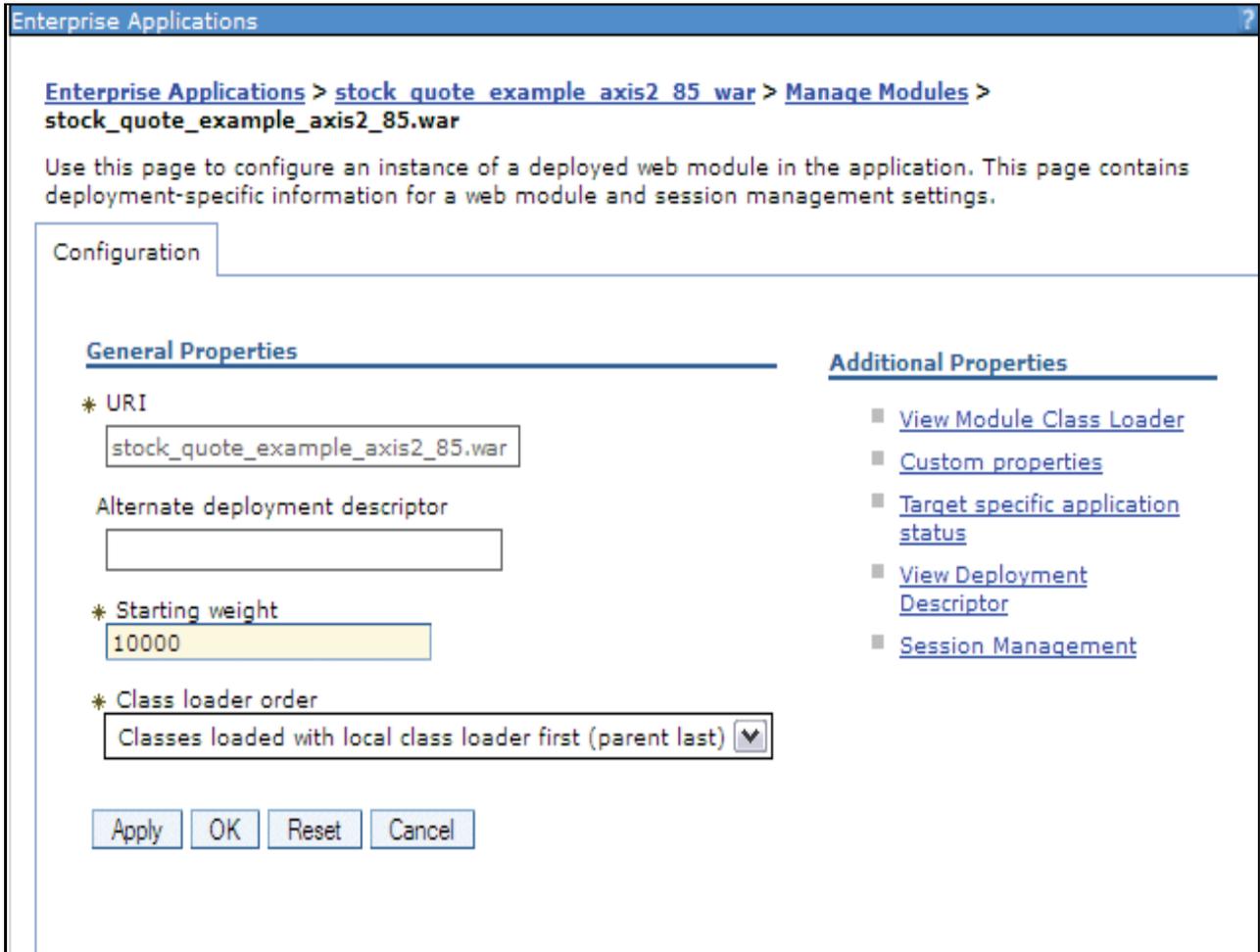


图 5-26 针对类装入器顺序选择父类最后选项

5.5.9 在 WebSphere Application Server V8.5 上测试应用程序

应用程序现已安装并配置完成，必须启动应用程序以对其进行测试。要完成此任务，请完成以下步骤：

1. 在 WAS 管理控制台 Web 界面左侧窗格任务列表中单击 **WebSphere 企业应用程序**（您可能需要展开**应用程序类型**才能找到该任务）。
2. 选择 `stock_quote_example_axis2_85_war` 条目，然后单击**启动**。

应用程序启动后，您可重复使用 `StockQuoteClientSideUnitTest` 对在 WebSphere Application Server V8.5 上运行的应用程序进行测试。对于在 WebSphere Application Server V8.5 中的应用程序测试，测试实例必须更新以提供 Web 服务 HTTP 端点的 URL。要完成此任务，请完成以下步骤：

1. 展开 **StockQuoteClient** 项目，然后展开 **src** 文件夹。展开项目后，双击 `StockQuoteClient.java` 文件。
2. 必须更新 `STOCK_SERVICE_EPR` 常量以提供端点的 URL。您很可能必须更新端口号以匹配您的 WebSphere Application Server V8.5 服务器的端口号。

在示例中，我们基于我们的 V8.5 服务器端口号和应用程序服务器上安装应用程序时提供的上下文根，使用 `http://localhost:9083/stockquote/services/StockQuoteService`。

所有 JUnit 测试完成后显示绿色的条，说明 Web 服务成功通过这组测试。

5.5.10 Web 服务 Axis 2 股票报价应用程序迁移的摘要

通常，在不同版本的应用程序服务器之间的应用程序迁移可直接进行，但对于某些应用程序，如使用第三方 Web 服务引擎的应用程序，迁移过程较为复杂。本章节说明，如谨慎对待，此类应用程序的迁移还是很容易操作的。



A

迁移调查问卷

本附录包含多套问题，您可以使用这些问题来根据您能预见的独特挑战来设计您自己的调查问卷。本附录并非要用作最终调查问卷，而是用作了解如何设计迁移调查问卷的基础。

本附录中所述主题如下：

- ▶ 业务要求
- ▶ 一般信息
- ▶ 应用程序体系结构
- ▶ 代码
- ▶ 开发迁移调查问卷
- ▶ 运行时迁移调查问卷
- ▶ 测试迁移调查问卷

业务要求

业务要求相关问题如下所列：

1. 促使业务做出该变化的动因是什么？
2. 应用程序是否为业务关键型？
3. **WebSphere Application Server** 的目标版本是什么？
4. 此次迁移涉及到哪些其他 **WebSphere** 产品？
举例来说，可能包括 **WebSphere Portal Server** 和 **WebSphere Edge Server**。
5. 此次应用程序迁移的截止日期是何时？
6. 若在上述截止日期仍未能实施变化，业务将会面临怎样的风险？
7. 若应用程序未运行，业务将会面临怎样的风险？
8. 此次迁移是否为新应用程序发行版 / 版本的一部分？
9. 请描述迁移的各个重要里程碑。包括开发日期、**QA** 日期等等。
10. 您认为什么是成功的迁移？

一般信息

一般信息相关问题如下所列：

1. 应用程序的名称是什么？
2. 用于访问应用程序的 **URL** 是什么？
3. 应用程序的当前版本是多少？
4. 应用程序的受众有哪些：内部人员、外部人员或两者皆是？
5. 应用程序的主要功能是什么？
6. 该应用程序能够提供什么服务？
7. 应用程序的用户有哪些？
8. 请描述此次迁移中需要升级的各个组成部分？
这些升级可能包括的内容诸如平台基础结构（操作系统或相关堆栈组件）或开发 **Java EE** 所提供功能等。
9. 迁移是否能分阶段进行？
10. 应用程序是否能够划分为多个较小部分分别独立升级？
数据库、**JSP**、**EJB** 等的分阶段迁移即为增量迁移，有利于质量控制及隔离非预期问题。
11. 您计划多久对您的应用程序代码进行一次独立的审核？

应用程序体系结构

一般应用程序体系结构相关问题如下所列：

1. 应用程序体系结构是否符合 **Java EE** 所规定的模型 2 体系结构？
2. 请描述应用程序对已确定设计模式的使用情况。
3. 举例来说，应用程序是否使用了外观 (**Facade**)、命令 (**Command**)、战略 (**Strategy**) 或代理 (**Proxy**) 等著名模式？

依赖关系

依赖关系相关问题如下所列：

1. 类包之间是否存在依赖关系循环？

当引用循环涉及的两种或多种类型相互之间紧密绑定在一起时，依赖关系循环便会产生。依赖关系循环并非总是坏现象。在两种或多种类型之间通常需要循环。然而，当类包或模块之间存在依赖关系循环时，即会成为问题。这种问题的危险在于其会导致这些类包或模块之间出现耦合，进而通常会导致代码更不可靠且难以更改。该问题的要点在于了解类包之间是否存在此类循环。

2. Java EE 模块之间是否存在依赖关系循环？

Java EE 模块之间的常见依赖关系循环是指某个 EJB 模块引用了某个 Java 实用程序模块中的帮助类，而该模块紧接着又引用了该 EJB 模块。举例来说，此类帮助类一般被用作查找 EJB Home 的方便方法。

3. 请描述应用程序中的任何类加载依赖关系。

类加载问题以及与库的依赖关系在旧版本的 Java EE 型应用程序中较为常见。早期版本的规范中并未清晰说明类加载如何发生以及必须如何访问库类。

持久性

与持久性相关的问题只有一个：

1. 如何实现应用程序数据的持久性？

该问题旨在了解如何持久保存应用程序数据。一般来说，应用程序数据存储在数据库中，应用程序的某个层负责将这些数据从数据库移至 Java 对象，经过处理后再移回到数据库中。持久性可使用 EJB CMP 实体 bean、第三方产品（例如 TopLink for Java）或者使用数据访问对象 (DAO) 模式的定制实现来实现。持久性也可在整个应用程序代码中采用一种*特别*的方式使用 JDBC 语句来实现。

国家语言

国家语言相关问题如下所列：

1. 应用程序支持哪些国家语言？

2. 应用程序如何确定语言？

应用程序可以利用 HTTP 请求头中的语言属性，可能还可以利用由用户结合使用 cookie 及 HTTP 会话数据提供的概要文件信息。

3. 应用程序如何存储语言数据？

代码

本节包含与代码相关的问题。这些问题只需要大致的答案。不过，答案越准确，结果就越准确。

Java

Java 相关问题如下所列：

1. 应用程序中定义了多少个客户开发的类？

客户化类是指由您的开发团队创建的类。客户化类不包括属于标准 Java 库、中间件或第三方所提供库的一部分的类。此外，客户化类也不包括由容器生成并用以在运行时为您的应用程序提供支持的类。

2. 应用程序中有多少个客户化类声明为抽象？
3. 应用程序中有多少个客户化接口？

客户化接口是指由您的开发团队创建的接口。客户化接口不包括属于标准 Java 库、中间件或第三方所提供库的一部分的接口。

4. 应用程序中有多少行代码？

请指明有用代码行的数量（此值不包括注释行）。

EJB 使用情况

EJB 使用情况相关问题如下所列：

1. 使用的是哪个版本的 EJB 规范？
2. 是否应用程序中的所有企业 bean 都严格符合此版本的规范？
3. 应用程序中有多少个无状态会话 bean？
4. 应用程序中有多少个有状态会话 bean？
5. 应用程序中有多少个 CMP 实体 bean？
6. 应用程序中有多少个 BMP 实体 bean？
7. 应用程序中有多少个消息驱动的 bean？

Servlet 和 JSP

Servlet 和 JSP 相关问题如下所列：

1. 使用的是哪个版本的 JSP 规范？
2. 是否应用程序中的所有 JSP 文件都严格符合此版本的规范？
3. 应用程序中有多少个 JSP 文件？
4. 您的 JSP 文件是否包含 Java 代码？

请列出应用程序所用的第三方标记库 (taglib)。若有可能，请包括版本号。

5. 请描述您为应用程序构建的任何定制标记库 (taglib)。
6. 使用的是哪个版本的 servlet 规范？
7. 是否应用程序中的所有 servlet 都严格符合此版本的规范？
8. 应用程序中有多少个 servlet？

9. 大约有多少数据在 `HttpSession` 中维护？

此信息可能难以估算，特别是在会话数据的大小变化较大时。若要在某个特定时刻准确地度量会话，可能的方法是创建一个使用 `ObjectOutputStream` 的 `Servlet`，以将 `HttpSession` 的内容转储到文件中。根据文件的大小即能较准确地估算出会话大小（但并非对会话数据量的精确度量）。

10. 是否所有存储在 `HttpSession` 中的对象都可序列化？

应用程序服务器使用序列化实现服务器 (JVM) 之间的数据传输。若某个对象实现 `java.io.Serializable` 接口，即表示该对象可序列化。

只有全部对象均可序列化，方可以“是”来回答该问题，此处所述的全部对象也包括通过引用其他对象间接包含在 `HttpSession` 中的对象。

11. 请描述可能妨碍应用程序存储在集群内服务器之间的 `HttpSession` 中之数据共享的任何问题。

Web 服务

Web 服务相关问题如下所列：

1. 请列举应用程序访问的外部 Web 服务。

我们所谓的外部是指应用程序外部，包括由其他应用程序运行提供的任何 Web 服务，与其运行语言和所在平台无关。若有可能，请描述所用的各个 Web 服务并记下据您所知可能会影响迁移的任何潜在问题。

2. 请列举应用程序访问的内部 Web 服务。

内部 Web 服务是属于应用程序一部分的 Web 服务。举例来说，应用程序可能会使用某个 Web 服务来实现 `Servlet` 层和 `EJB` 层之间的外观 (Facade) 模式。

3. 请列举应用程序实现的 Web 服务。

对于此问题，请列出应用程序为内部或外部客户机提供的 Web 服务。若有可能，请描述各个 Web 服务所提供的功能。

4. 应用程序所实现的 Web 服务是否符合 WS-I？若符合，请指明 WS-I 版本。

数据库访问

数据库访问相关问题如下所列：

1. 使用的是哪个版本的 JDBC 规范？
2. JDBC 的使用是否严格符合此版本的规范？
3. 应用程序进行了多少次不重复的 JDBC 查询？
4. 应用程序使用了哪个数据库管理器？
5. JDBC 查询是否使用了供应商特定的 SQL 扩展？
6. 应用程序是否直接使用 `DriverManager`？
7. 应用程序是否通过数据源使用 J2EE 连接池？
8. 请描述所用的任何第三方连接池机制。
9. 举例来说，应用程序是否使用了 ORACLE 连接池？
10. 是否所有 JDBC 连接都已显式关闭？
11. 是否所有 JDBC 语句都已显式关闭？
12. 是否所有 JDBC 资源都已在 `finally` 子句中关闭？

异常处理程序 `finally` 块中包含的代码会运行（无论是否发生异常）。通过在 `finally` 子句中关闭这些共享资源，可确保这些资源保持可用。

JMS

JMS 相关问题如下所列：

1. 使用的是哪个版本的 JMS 规范？
2. JMS 的使用是否严格符合此版本的规范？
3. 应用程序使用了多少个队列？
4. 应用程序使用了多少个主题？
5. 通过 JMS 传输的是何种数据？
举例来说，纯文本、XML 或序列化对象。
6. 应用程序是否写入 JMS 队列？
7. 应用程序是否从 JMS 中读取？
8. 应用程序是否从 JMS 中进行分块读取？
9. 应用程序是否从会话或实体 bean 中进行从 JMS 中分块读取？

JNDI 命名

EJB 命名相关问题如下所列：

1. 应用程序是否使用 JNDI 而不是 EJB Home 来访问资源？
2. 应用程序的部署描述符是否包含本地 JNDI 资源引用？
有关 JNDI 资源引用的部分位于 web.xml 和 ejb-jar.xml 中。
3. 请描述应用程序中进行的典型 JNDI 访问。若有可能，请提供一个代码片段。
4. 是否所有 JNDI 资源 (InitialContexts) 都已在 finally 子句中关闭？
异常处理程序的 finally 块中包含的代码会运行（无论是否发生异常）。通过在 finally 子句中关闭这些共享资源，可确保这些资源保持可用。

应用程序跟踪和日志

应用程序跟踪和日志相关问题如下所列：

1. 应用程序使用何种技术来提供日志？
举例来说，应用程序可能会使用 Log4J 等第三方产品。或者，应用程序可能会将消息写至标准输出 (System.out)。
2. 整个应用程序是否使用了统一的日志？

Struts

若应用程序未使用 Struts，请跳过本节。Struts 相关问题如下所列：

1. 使用的是哪个版本的 Struts？
2. 应用程序中有多少个定制 ActionForm？
3. 应用程序中有多少个定制 ActionHandler？
4. 请描述您对 Struts 做出的任何修改。
5. Struts 属于一种开放源框架，可由组织予以定制，以提供扩展功能。若您对 Struts 进行了扩展，请对该扩展予以描述。

事务

事务相关问题如下所列：

1. 事务管理的方式是隐式还是显式？
2. 应用程序是否需要两阶段落实 (2PC)？
3. 与应用程序交互的是何种事务系统？

线程

本节所述仅涉及由应用程序直接操纵的线程。本节不涉及由应用程序服务器管理的线程。

线程相关问题如下所列：

1. 应用程序是否创建线程？
2. 应用程序是否创建守护线程？
3. 应用程序是否使用 **stop()** 方法来显式停止线程？
4. 请列出应用程序所用的任何第三方线程池库。若有可能，请包括版本号。
5. 请描述应用程序所用的任何定制线程池库。
6. 请描述应用程序如何使用线程。

此问题仅涉及应用程序如何显式使用线程。举例来说，应用程序是否创建线程以并行执行工作或为各种服务提供超时？若为后者，请指出线程为何种服务提供超时。

套接字

若您的应用程序未直接使用套接字，请跳过本节。本节所述仅涉及由应用程序直接创建和使用的套接字；不涉及由应用程序服务器管理的套接字。

套接字相关问题如下所列：

1. 请描述应用程序如何直接使用套接字。
2. 应用程序是否创建任何守护程序套接字？
3. 应用程序使用安全套接字 (SSL)？

XML

XML 相关问题如下所列：

1. 应用程序使用何种 XML 库？请包括版本信息。
2. 应用程序使用何种 XSLT 库？请包括版本信息。

开发迁移调查问卷

以下问题与迁移的开发环境有关。

工作站配置

工作站配置相关问题如下所列：

1. 使用何种硬件？
2. 使用的是哪个操作系统及版本？
3. 所安装的 **RAM** 是多少？

集成开发环境

集成开发环境相关问题如下所列：

1. 使用何种 **Java** 集成开发环境 (**IDE**)？
2. 使用了何种源代码管理 (**SCM**) 工具？
3. 源目录如何组织？
4. 使用哪些工具进行单元测试？
5. 在开发过程中使用了哪些其他工具？
6. 本次迁移工作中是否会采用 **WebSphere** 开发工具？
7. 本次迁移采用了开发工具？
 - **WebSphere Studio Application Developer**
 - **WebSphere Studio Application Developer Integration Edition**
 - **Rational Application Developer**
 - 其他

开发测试配置

开发测试配置相关问题如下所列：

1. 开发团队是否有专用开发测试硬件？
2. 使用何种硬件？
3. 使用的是哪个操作系统及版本？
4. 所安装的 **RAM** 是多少？
5. 使用何种应用服务器及版本进行开发测试？
6. 是否使用其他软件进行开发测试？

软件开发技能

软件开发技能相关问题如下所列：

1. 负责应用程序构建的团队由多少名软件开发人员组成？
2. 在开发人员中，有多少名开发人员具有体系结构级别技能？
3. 有多少名开发人员可投入到本次迁移中？
4. 开发人员对 **Java EE** 的熟悉程度如何？
 - 他们均熟练掌握全面的 **Java EE** 知识。
 - 他们熟悉构成 **Java EE** 的各种技术（例如 **servlet** 及 **EJB**）。
 - 他们不熟悉 **Java EE**。
5. 开发人员是否具有使用 **Eclipse** 进行软件开发的经验？
6. 开发人员是否具有使用 **RAD** 进行软件开发的经验？
7. 对开发团队是否有具体的培训要求？

开发方法论

开发方法论相关问题如下所列：

1. 使用何种开发方法论？
2. 典型的开发周期是多长时间？
3. 交付应用程序内部发行版的频率如何？
这些交付包括用于内部测试或用户验收测试的发行版。
4. 交付应用程序外部发行版的频率如何？
5. 将新版本的应用程序交付给生产环境的频率如何？

构建和包装

构建和包装相关问题如下所列：

1. 构建应用程序时使用了哪些工具？
示例：**Apache Ant (Ant)**、**make** 或无。若有可能，请提供版本信息。
2. 交付给运行时环境的应用程序包装成何种形式？应用程序是包装成 **EAR**、**WAR** 还是 **JAR** 文件？或者，是否以其他形式交付应用程序？
3. 如何包装交付给运行时环境的应用程序？
4. 将应用程序交付给运行时环境时实施了哪些机制？包装的应用程序是通过 **FTP** 交付还是采用其他传输机制？
5. 交付应用程序时采用了何种流程？
6. 静态内容如何包装并交付给运行时环境？
7. 包装的应用程序中是否随附安装和配置脚本？

Ant

若您在构建应用程序时未使用 **Ant**，请跳过本节。

1. 使用的是哪个版本的 **Ant**？
2. 构建过程是否使用了一个或多个 **Ant** 脚本？
 - 1 个
 - 多个
3. 在构建过程中使用了哪些可选的 **Ant** 任务？

运行时迁移调查问卷

以下问题与运行时迁移有关。

概述

概述相关问题如下所列：

1. 是否会迁移现有的运行时基础结构？
2. 迁移后是否支持现有的运行时基础结构？
若您预计仅迁移当前的应用程序并保留原有的基础架构继续运行，请回答“是”。
3. 运行时基础结构是否包括专用生产前模拟环境？
生产前环境可能称为质量保证环境。无论名称为何，在将此类应用程序迁移至生产环境之前，我们可以在此环境中测试应用程序的整体质量。
4. 运行时基础结构是否包括专用性能测试环境？
请描述运行时基础结构中所包含的各个单独运行时间环境（生产、生产前、系统测试、开发测试、性能测试等）。

当前硬件

当前硬件相关问题如下所列：

1. 请描述用于运行当前应用程序服务器的硬件。
如果您的环境为异构环境，请描述每个硬件配置。请指出每个配置的数量。此外，还请指出所用的操作系统类型及版本。
2. 当前有多少台生产计算机安装了生产应用程序？
请描述在生产环境中所用的任何其他硬件。请包括防火墙、IP sprayer、Web 服务器、集群管理器、数据库服务器、入侵检测系统等的所有详细信息。请指出在这些系统中使用的软件（若有），包括操作系统及版本。
3. 请描述生产前模拟环境的配置与生产环境的配置有何不同。
4. 生产前环境配置是否与生产环境配置复杂程度相当，只是规模较小？生产前环境中是否遗漏了关键要素？
5. 请描述您计划用于运行迁移后应用程序的硬件。
若您的环境为异构环境，请描述每个硬件配置。请指出每个配置的数量。此外，还请指出所用的操作系统类型及版本。
6. 使用了多少台生产计算机来安装生产应用程序？
请描述在生产环境中所用的任何其他硬件。请包括防火墙、IP sprayer、Web 服务器、集群管理器、数据库、入侵检测系统等的所有详细信息。请指出在这些系统中使用的软件（若有），包括操作系统及版本。

软件

软件相关问题如下所列：

1. 您的生产主机使用的是哪个版本的 **Java** 运行时间环境 (**JRE**)？
2. 请提供 **JRE** 提供商的名称，可能为 **Sun**、**IBM** 或其他 **JRE** 供应商。
3. 请描述在应用程序服务器所在的相同生产硬件上安装并使用的其他软件。
4. 若应用程序使用连接服务来访问后台主机（例如 **CICS**），请描述主机桥接网络的基础结构。

HTTP 服务器

HTTP 服务器相关问题如下所列：

1. 使用何种 HTTP 服务器？
2. 是否将 HTTP 服务器配置为与特定应用程序服务器实例之间实现粘连性？
3. 有多少台服务器托管 Web 服务器？

IBM WebSphere Application Server 几乎能够与目前市面上的所有 Web 服务器共用。请务必了解将如何使用当前 Web 服务器，并了解若 **WebSphere Application Server** 在生产切换期间继续使用相同的 Web 服务器，将会产生何种影响。

4. 请描述由 HTTP 服务器提供的任何其他服务，包括超出为静态内容提供服务及应付应用程序服务器所需服务范围的其他服务。举例来说，HTTP 服务器是否使用 **CGI/Perl** 提供了附加功能？
5. 请描述安全网络传播的基础结构。
6. 在 Web 服务器上启用 **HTTPS** 是将其从浏览器连接 Web 服务器端的一种方法。企业也可使用硬件加速器来提高 **SSL** 加密和解密过程的速度。若是如此，硬件加速器如何与 Web 服务器集成？
7. 从 HTTP 服务器插件与应用程序服务器进行的通信在当前环境中是否安全？

网络边缘服务器 (Edge Component)

网络边缘服务器相关问题如下所列：

1. 使用了哪些负载均衡技术？
2. 是否将负载均衡器配置为与特定应用服务器实例之间实现粘连性？
3. 请描述如何为负载均衡器提供故障转移。

可用性

可用性相关问题如下所列：

1. 请描述与应用程序的可用性相关的现有服务级别协议，此类协议决定可用性及切换至 **WebSphere Application Server** 正常运行服务器的灵活程度。
2. 请描述如何在当前环境中提供集群。请阐释您的集群环境。
3. 请描述如何在生产环境中发布应用版本？
若不同应用程序采用不同过程，请分别描述每个过程。
4. 当某个节点或服务器出现故障或停机时，当前采用何种备用策略？
请详细阐释您的故障转移策略。

应用问题

应用相关问题如下所列：

1. 旧版本的应用程序是否需要与迁移后的应用程序共存？
2. 迁移是否涵盖生产服务器上的所有应用程序？若否，是否也制定了迁移这些应用程序的计划？

共存非常重要，在将公用框架和应用程序开发模型迁移至新平台时尤其如此。在移至新平台时，可能存在必须维护的应用程序依赖关系。
3. 迁移过程中是否计划对生产计算机上的任何其他软件进行更新？

在迁移期间，生产环境的迁移是重要任务，而风险的迁移也至为关键。硬件和软件更新仅在迁移过程中有必要时才会进行。此类更新应予以排除，以确保将变数降至最低程度。

管理

管理相关问题如下所列：

1. 您有多少名管理员负责为生产环境提供支持？
2. 您是否有专门负责生产基础结构的管理员？
3. 请描述生产环境管理员的技能等级和经验。
4. 您是如何部署应用程序的？
5. 生产管理员是否使用部署脚本来完成这一任务？该任务是否涉及定制脚本？请对该过程予以阐释。
6. 请描述用于管理应用程序服务器的任何定制脚本。
7. 您在管理当前设置时是否使用了任何定制脚本？若是，其用途为何？
8. 是否设置了生产环境，以便于进行日志轮替和文件系统维护？
9. 日志文件可能会变得很大并占用生产主机上的大量可用空间。是否制定了相应程序或过程来处理文件系统维护？

安全性

安全性相关问题如下所列：

1. 安全性在您的生产运行时基础结构中重要程度如何？
 - 至关重要
 - 非常重要
 - 重要
 - 不重要
2. 请描述如何管理应用程序服务器的内部安全性。

在生产环境中，有多个角色适用于安全性保障，包括根管理员、机器维护操作员、应用程序部署人员和开发人员。请按这些角色阐释如何管理应用程序服务器硬件和软件的访问。
3. 如何强制执行内部安全性策略？
4. 如何验证用户的身份？
5. 如何管理用户权限？
6. 请描述您用于用户身份验证和授权的特定于应用程序的服务。
7. 请描述您用于用户身份验证和授权的定制服务。
8. 请描述所用的任何第三方安全性机制，可包括 **Netegrity SiteMinder** 等产品。

9. 请描述生产运行环境如何提供（或参与）单点登录解决方案。请描述所实施的任何网络安全措施。
10. 在您的配置中，是否在各层之间安置防火墙？若是，如何配置？配置在何处？

测试迁移调查问卷

以下问题与测试环境有关。

硬件

所用硬件相关问题如下所列：

1. 是否有专用硬件专用于用户验收测试？
2. 是否有专用硬件专用于质量保证测试？
3. 是否有专用硬件专用于性能测试？
4. 请描述测试硬件配置与生产硬件配置有何不同。

实践和工具

所用实践和工具相关问题如下所列：

1. 进行了哪些测试？何时进行？
2. 回归测试周期的典型时长是多少？
3. 应用程序进行回归测试的频率如何？
4. 用户验收测试周期的典型时长是多少？
5. 应用程序进行用户验收测试的频率如何？
6. 此迁移工作是否需要用户验收测试？
7. 质量保证测试周期的典型时长是多少？
8. 应用程序进行质量保证测试的频率如何？
9. 请描述将应用程序从开发环境移至生产环境过程中所经历的测试阶段。
10. 您如何测试应用程序是否符合 J2EE 规范？
11. 您是否制定了一整套应用程序开发最佳实践？这些实践包括编码编写标准、代码维护和性能最佳实践。
12. 如何强制执行应用程序开发最佳实践？
13. 您使用哪些工具对应用程序代码进行单元测试？
14. 您使用哪些工具对应用程序代码进行功能测试？
15. 您使用哪些工具对应用程序代码进行回归测试？
16. 您使用哪些工具对应用程序代码进行性能测试？
17. 测试程序是否包含其代码依赖于源平台的任何工具？



附加材料

本文引用了一些附加材料，下面各节说明如何从互联网下载这些附加材料。

查找 Web 材料

与本文有关的 Web 材料，可通过互联网从 IBM Redbooks Web 服务器下载其软拷贝。通过 Web 浏览器访问以下地址：

<ftp://www.redbooks.ibm.com/redbooks/SG248048>

或者，您也可以转至 IBM Redbooks 网站，网址为：

ibm.com/redbooks

选择 **Additional materials**（附加材料）并打开与 IBM Redpaper™ 表格编号 REDP 相对应的目录。

使用 Web 材料

本文随附的附加 Web 材料包括以下文件：

文件名	描述
WASV2V_migration.zip	在第 51 页上的第 4 章“Application Migration Tool - WebSphere Version to Version 的安装和配置”中使用的配置和应用程序文件

下载 Web 材料的系统要求

Web 材料的系统配置要求如下所述：

硬盘空间：	至少 20 MB
操作系统：	Linux 或 Windows

下载并解压缩 **Web** 材料

在您的工作站中创建一个子目录（文件夹），然后将 **Web** 材料压缩文件的内容解压缩到该文件夹中。

相关出版物

本节所列的出版物适于在对本文所涵盖的主题进行更为详细的讨论时使用。

IBM Redbooks

以下 IBM Redbooks 出版物提供了有关本文档中所述主题的更多信息。该列表中所引用的某些出版物可能仅提供软拷贝。

Experience JEE Using Rational Application Developer V7.5, SG24-7827

Getting Started with WebSphere Application Server Feature Pack for Service Component Architecture, REDP-4633

Oracle to DB2 Conversion Guide: Compatibility Made Easy, SG24-7736

Rational Application Developer for WebSphere Software V8 Programming Guide, SG24-7835

WebSphere Application Server V6.1: Classloader Problem Determination, REDP-4307

WebSphere Application Server V7: Competitive Migration Guide, SG24-7870

WebSphere Application Server V8.5 Concepts, Planning, and Design Guide, SG24-8022

您可以在以下所列网站搜索、查看、下载或订购这些文档和其他 Redbook、Redpapers、Web 文档、草稿及其他材料：

ibm.com/redbooks

其他出版物

该出版物也可用作更多详细信息的来源：

将 Microsoft Access 2000 数据库迁移至 IBM DB2 通用数据库 7.2 (Migrating a Microsoft Access 2000 Database to IBM DB2 Universal Database 7.2)

在线资源

下列网站也可用作更多详细信息的来源：

Apache Tomcat 7.0.27 下载网站

<http://tomcat.apache.org/download-70.cgi>

Apache Tomcat 7.0.27: 安装、配置和管理说明 (Instructions for installing, configuring, and managing)

<http://tomcat.apache.org/tomcat-7.0-doc/index.html>

Hibernate Validator 分发下载页面

<http://www.hibernate.org/subprojects/validator/download>

有关针对 JPA 2.0 的类型安全查询的 IBM developerWorks 文章

<http://www.ibm.com/developerworks/java/library/j-typesafejpa>

IBM Education Assistant - IBM WebSphere Application Server Migration Toolkit V3.5

<http://publib.boulder.ibm.com/infocenter/ieduasst/v1r1m0/topic/com.ibm.iea.wasmt/wasmt/3.5/Overview.html>

IBM Passport Advantage

<http://www-01.ibm.com/software/support/probsub.html>

IBM WebSphere Developer Services - 如何提出 PMR

<http://www.ibm.com/isv/tech/remoteEmail/entryForm.jsp>

Java EE 规范

<http://www.oracle.com/java>

符合 Java 规范请求 (JSR) 286 的 portlet

<http://jcp.org/en/jsr/detail?id=286>

JBoss 下载网站

<http://www.jboss.org/jbossas/downloads/>

迁移工具包相关的在线信息

<http://www.ibm.com/developerworks/websphere/downloads/migtoolkit/index.html>

OSGi 应用程序

<http://www.osgi.org/About/WhatIsOSGi>

Oracle Java 应用程序验证套件 (AVK)

http://java.sun.com/j2ee/verified/avk_enterprise.html

Oracle WebLogic Server 10.3.6 安装说明

http://www.oracle.com/technology/software/products/ias/htdocs/wls_main.html

Quickstarts 分发下载页面

<http://download.jboss.org/jbossas/7.1/jboss-as-7.1.1.CR2/jboss-as-quickstarts-7.1.1.CR2-dist.zip>

Shrinkwrap 库

<http://www.jboss.org/shrinkwrap>

SLF4J 分发下载页面

<http://www.slf4j.org/download.html>

Spring Framework 简介

http://www14.software.ibm.com/webapp/wsbroker/redirect?version=phil&product=was-nd-mp&topic=cspr_intro

Using Spring and Hibernate with WebSphere Application Server 文章

http://www.ibm.com/developerworks/websphere/techjournal/0609_alcott/0609_alcott.html

WebSphere Application ServerBase V8.5

<http://www.ibm.com/software/webservers/appserv/was/>

WebSphere Application ServerExpress V8.5

<http://www.ibm.com/software/webservers/appserv/express/>

WebSphere Application Server for Developers V8.5

<http://www.ibm.com/software/webservers/appserv/developer/index.html>

WebSphere Application Server 信息中心

<http://publib.boulder.ibm.com/infocenter/iisinfsv/v8r5/index.jsp>

WebSphere Application Server Network Deployment V8.5

<http://www.ibm.com/software/webservers/appserv/was/network/>

WebSphere Application Server z/OS? V8.5

http://www.ibm.com/software/webservers/appserv/zos_os390/

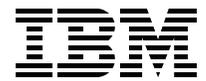
IBM 提供的帮助

IBM 支持与下载

ibm.com/support

IBM Global Services

ibm.com/services



WebSphere Application Server V8.5 V2V 迁移指南



中国销售及 PLM 团队请求将 WAS 8.5 迁移指南的“部分”内容（仅包括 V2V 章节及部分概述章节，不包括竞争性迁移章节）以及由 Tiger Team 编写以供迁移研讨会使用的 4 个 PowerPoint 演示文档翻译为简体中文。今年，我们将在中国大规模推广 WAS 的 V2V 迁移，并且大部分客户都需要经过本地化的文档。

国际技术 支持组织

根据实践经验撰写技术
信息

IBM Redbooks 由 IBM 国际技术支持组织开发。来自世界各地的 IBM、客户和合作伙伴专家会根据现实中的方案及时撰写技术信息。提供一些有针对性的建议来帮助您在您的环境中更高效地实现 IT 解决方案。

有关更多信息，
请访问：
ibm.com/redbooks