



Sun Java™ System

Content Delivery Server 5.1 迁移指南

Sun Microsystems, Inc.
www.sun.com

文件号码: 820-5755-10
2008 年 6 月

请将有关本文档的意见和建议提交至: <http://www.sun.com/sunsurveys/dsc/dsc-feedback.jsp>

版权所有 ©2008 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. 保留所有权利。

对于本文中介绍的产品，Sun Microsystems, Inc. 对其所涉及的技术拥有相关的知识产权。需特别指出的是（但不局限于此），这些知识产权可能包含在 <http://www.sun.com/patents> 中列出的一项或多项美国专利，以及在美国和其他国家/地区申请的一项或多项其他专利或待批专利。

美国政府权利 - 商业用途。政府用户应遵循 Sun Microsystems, Inc. 的标准许可协议，以及 FAR（Federal Acquisition Regulations，即“联邦政府采购法规”）的适用条款及其补充条款。

必须依据许可证条款使用。

本发行版可能包含由第三方开发的内容。

Sun、Sun Microsystems、Sun 徽标、Java、JavaServer Pages、JSP 和 JDK 是 Sun Microsystems, Inc. 在美国和其他国家/地区的商标或注册商标。

UNIX 是 X/Open Company, Ltd. 在美国和其他国家/地区独家许可的注册商标。

Adobe 徽标是 Adobe Systems, Incorporated 的注册商标。

本服务手册所介绍的产品以及所包含的信息受美国出口控制法制约，并应遵守其他国家/地区的进出口法律。严禁将本产品直接或间接地用于核设施、导弹、生化武器或海上核设施，也不能直接或间接地出口给核设施、导弹、生化武器或海上核设施的最终用户。严禁出口或转口到美国禁运的国家/地区以及美国禁止出口清单中所包含的实体，包括但不限于被禁止的个人以及特别指定的国家/地区的公民。

本文档按“原样”提供，对于所有明示或默示的条件、陈述和担保，包括对适销性、适用性或非侵权性的默示保证，均不承担任何责任，除非此免责声明的适用范围在法律上无效。



请回收



Adobe PostScript

目录

前言 xi

1. 迁移 Content Delivery Server 1

迁移过程 1

初始迁移步骤 2

迁移 Content Delivery Server 4

测试迁移 6

2. 配置属性 9

设置配置属性 9

 编辑 workflow 10

样例配置文件 11

3. 转换文件 13

内容管理 API 的变更 13

内容验证 API 的变更 14

订户 API 的变更 15

用户配置文件 API 的变更 20

WAP 网关 API 的变更 23

4. 迁移数据库 25

迁移步骤和标签	25
迁移先决条件	26
迁移术语和变量	27
环境变量设置	27
配置迁移	27
迁移文件标记约定	28
配置文件标记	28
默认范围值	33
配置示例	33
单个服务器部署	34
远程售卖部署	35
记录迁移信息	36
测试内容	36
迁移 CDS_EVENT 表	36
迁移单独目录部署的事件	38
迁移版本的支持设备信息	38
运行迁移	39
迁移单个服务器部署	40
远程售卖部署的迁移准则	43
迁移远程售卖部署	43
迁移远程 Vending Manager	45
更正路径名称差异	50
验证迁移完成情况	51
迁移后修改	52
生成搜索索引	52
5. 数据库迁移故障排除	55
消息	55
错误消息	56

警告消息 60

失败步骤故障排除 64

索引 67

表

表 4-1 配置文件标记 29

代码示例

代码示例 2-1	样例配置 XML 文件	11
代码示例 4-1	单个服务器部署的样例配置文件	34
代码示例 4-2	远程售卖部署的样例配置文件	35

前言

本指南说明了如何将以前版本的 Sun Java™ System Content Delivery Server 成功转换为最新版本。其中介绍了如何通过设置 `properties.xml` 文件来配置 Content Delivery Server 数据库以及将以前版本的 Content Delivery Server 中的信息迁移到最新版本。该文件提供了从以前的发行版到最新的发行版，目录和文件所发生的变更及其对应关系。有关如何安装 Content Delivery Server 最新版本的信息，请参见《Sun Java™ System Content Delivery Server 5.1 安装指南》。

阅读本文档之前

本指南适用于负责将以前安装的 Content Delivery Server 版本转换为最近安装的版本的系统管理员。

本文档的组织结构

本指南包括以下几章：

第 1 章 说明了将以前版本的 Content Delivery Server 迁移到当前版本的过程。本章将指导您完成初始迁移操作（如迁移配置和署名信息），然后说明如何迁移 Content Delivery Server 数据库。

第 2 章 介绍了属性配置过程并包含一个样例属性配置文件。

第 3 章 列出了从以前安装的 Content Delivery Server 版本到当前版本 API 所发生的变更。

第 4 章 介绍了如何为以前版本的 Content Delivery Server 创建数据库配置文件，以及如何将数据库迁移到当前版本。

第 5 章 提供了 Content Delivery Server 数据库迁移故障排除信息。

Shell 提示符

Shell	提示符
C shell	<i>machine-name%</i>
C shell 超级用户	<i>machine-name#</i>
Bourne shell 和 Korn shell	\$
Bourne shell 和 Korn shell 超级用户	#

印刷约定

字体	含义	示例
AaBbCc123	命令、文件和目录的名称；计算机屏幕输出	编辑 <code>.login</code> 文件。 使用 <code>ls -a</code> 列出所有文件。 % You have mail.
AaBbCc123	用户键入的内容，与计算机屏幕输出的显示不同	% su Password:
<i>AaBbCc123</i>	保留未译的新词或术语以及要强调的词。要使用实名或值替换的命令行变量。	这些称为 <i>class</i> 选项。 要删除文件，请键入 rm filename 。
新词术语强调	新词或术语以及要强调的词。	您 必须 成为超级用户才能执行此操作。
《书名》	书名	阅读《用户指南》的第 6 章。

注 - 字符的显示方式随浏览器设置的不同而有所不同。如果未正确显示字符，请在浏览器中将字符编码更改为 Unicode UTF-8。

相关文档

下表列出了本产品的文档。可以从以下位置获取联机文档：

<http://docs.sun.com/app/docs/prod/cds>

应用	书名	文件号码	格式	位置
门户署名和本地化	《Sun Java™ System Content Delivery Server 5.1 署名和本地化指南》	820-5374-10	PDF HTML	\$(CDS_HOME)/Documentation/branding 和联机文档。
规划系统	《Sun Java™ System Content Delivery Server 5.1 容量规划指南》	820-1939-10	PDF HTML	\$(CDS_HOME)/Documentation/capacity 和联机文档。
使用 Developer Portal	《Sun Java™ System Content Delivery Server 5.1 内容开发者指南》	820-5376-10	PDF HTML	\$(CDS_HOME)/Documentation/devguide 和联机文档。
使用提供的 API	《Sun Java™ System Content Delivery Server 5.1 定制指南》	820-5375-10	PDF HTML	\$(CDS_HOME)/Documentation/customization 和联机文档。
故障排除	《Sun Java™ System Content Delivery Server 5.1 错误消息》	820-5377-10	PDF HTML	\$(CDS_HOME)/Documentation/errmsgs 和联机文档。
安装系统	《Sun Java™ System Content Delivery Server 5.1 安装指南》	820-5378-10	PDF HTML	\$(CDS_HOME)/Documentation/install 和联机文档。
设置和集成现有的基础架构	《Sun Java™ System Content Delivery Server 5.1 集成和配置指南》	820-5379-10	PDF HTML	\$(CDS_HOME)/Documentation/integration 和联机文档。
迁移到最新发行版本	《Sun Java™ System Content Delivery Server 5.1 迁移指南》	820-5755-10	PDF HTML	\$(CDS_HOME)/Documentation/migration 和联机文档。
产品参考信息	《Sun Java™ System Content Delivery Server 5.1 参考手册》	820-5380-10	PDF HTML	\$(CDS_HOME)/Documentation/refman 和联机文档。
监视和管理系统	《Sun Java™ System Content Delivery Server 5.1 系统管理指南》	820-5381-10	PDF HTML	\$(CDS_HOME)/Documentation/system-mgmt 和联机文档。

文档、支持和培训

Sun 服务	URL
文档	http://www.sun.com/documentation/
支持	http://www.sun.com/support/
培训	http://www.sun.com/training/

第三方 Web 站点

Sun 对本文档中提到的第三方 Web 站点的可用性不承担任何责任。对于此类站点或资源中的（或通过它们获得的）任何内容、广告、产品或其他资料，Sun 并不表示认可，也不承担任何责任。对于因使用或依靠此类站点或资源中的（或通过它们获得的）任何内容、产品或服务而造成的或连带产生的实际或名义损害或损失，Sun 概不负责，也不承担任何责任。

Sun 欢迎您提出意见

Sun 致力于提高其文档的质量，并十分乐意收到您的意见和建议。可通过以下方式提交您的意见：访问 <http://docs.sun.com>，然后单击“发送意见”（Send Comments）。

请在反馈中提供文档的标题和文件号码：

《Sun Java™ System Content Delivery Server 5.1 迁移指南》，文件号码 820-5755-10

第1章

迁移 Content Delivery Server

本章指导您完成将 Content Delivery Server 版本 5.0 转换为 Content Delivery Server 版本 5.1 的过程中所涉及的步骤，并指出了从版本 5.0 PU1 迁移到 5.1 的特定步骤（该迁移过程与从 5.0 到 5.1 的迁移过程的不同之处）。

本章包括以下主题：

- [迁移过程](#)
- [初始迁移步骤](#)
- [迁移 Content Delivery Server](#)
- [测试迁移](#)

注 – 本文档未介绍订户 API 的用法。如果要使用用户接口和 Subscriber Portal API，则必须执行与本文档介绍的过程不同的迁移过程以满足相应的要求。

迁移过程

迁移过程包括以下阶段：

1. 执行初始迁移步骤，例如，集成属性以及迁移署名和本地化信息
2. 迁移 Catalog Manager
3. 迁移 Vending Manager
4. 启动新版本的 Catalog Manager 和 Vending Manager
5. 更新 Catalog Manager 的 Vending Manager Server 帐户信息。

启动新版本的 Catalog Manager 后，您可能希望执行某些测试以确保迁移成功完成。有关详细信息，请参见第 6 页上的“测试迁移”。

迁移过程中，不必关闭整个 Content Delivery Server 系统。遵循上文中概述的过程，您可以减少系统的总关闭时间。迁移 Catalog Manager 数据库时，Catalog Manager 将处于只读模式，并限制对 Developer Portal 和 Catalog Manager 的访问。

Vending Manager 可以继续运行，以便为客户提供服务。完成 Catalog Manager 迁移后，可以并行或逐一迁移 Vending Manager。请使用最有效的策略来满足您的需求。

注 – 迁移 Content Delivery Server 的具体细节因用户而异。请注意，某些步骤可能不适用于您的特定情况。另外，完成每个阶段所需的时间取决于部署配置以及涉及的组件数量和类型。

有关迁移 Content Delivery Server 的详细过程，请参见第 4 章。

在本章的其余部分，术语“旧版本”是指 Content Delivery Server 版本 5.0 (或 5.0 PU1，如果适用)。术语“新版本”是指 Content Delivery Server 版本 5.1。如果过程专门针对 5.0 PU1，则会明确指出该版本。

初始迁移步骤

迁移 Catalog Manager 数据库之前，必须先执行以下高级初始迁移步骤：

1. 应用备份安全过程。

例如，备份 Content Delivery Server 旧版本的数据库，并将 Content Delivery Server 旧版本的安装目录复制到备份目录。

2. 按照《Sun Java™ System Content Delivery Server 5.1 安装指南》中的说明，安装 Content Delivery Server 新版本以及默认数据库。

迁移过程的数据库初始化与用于默认部署的数据库初始化不同。cdsi db init 命令可创建空数据库和表。它随后使用默认数据填充某些数据库表。迁移过程要求使用空白的数据库，该数据库绝对不能包含任何表数据，并且表之间不能设置任何外键约束。要达到此要求，您需要使用以下两个命令创建数据库：

```
$cdsi db users
$cdsi db schemas_noc [-conf {dbconf}] [-cs] [-vs {vending}]
```

第一个命令在数据库服务器上创建用户。第二个命令创建所有表，但不创建任何表间约束。

注 – cdsi db users 命令彻底删除与 DBConf.xml 文件中指定的数据库用户关联的所有信息，因此，如果已使用 cdsi db init 创建了一个空数据库，则通过运行 cdsi db users 重新开始创建。

注 - `cdsi help` 未介绍 `cdsi db schemas_noc` 命令。

3. 配置新部署以连接到基本集成组件。

例如，此类组件可以是无线访问协议 (Wireless Access Protocol, WAP)、推送代理网关 (Push Proxy Gateway, PPG)、轻量目录访问协议 (Lightweight Directory Access Protocol, LDAP)、短消息服务中心 (Short Message Service Center, SMSC)、发送者和接收者以及通知和电子邮件服务器。

此集成迁移步骤包括修改定制适配器，以便将它们与 Content Delivery Server 最新版本结合使用。通常，需要迁移以下适配器：

- 记帐管理器适配器
- 后付费适配器
- 订户适配器

将为每个适配器提供一个默认实现实例。

配置属性迁移文件

`$CDS_HOME/cfg/migration/props_migration_5.0_5.1.xml`，以便将 Content Delivery Server 旧版本中的属性和现有本地化数据迁移到新版本。必须迁移所有新的本地化信息。

如果要从版本 5.0 PU1 进行迁移，应配置属性迁移文件

`$CDS_HOME/cfg/migration/props_migration_5.0PU1_5.1.xml`。

4. 运行 `cdspm migrateprops -f props_filename` 命令以迁移已配置的属性。

有关配置和运行属性迁移文件的详细信息，请参见第 2 章。有关目录和文件结构变更的信息，请参见第 3 章。

5. 将所有署名信息迁移到新的 Content Delivery Server 部署。

必须手动迁移所有署名信息，以便保留 Content Delivery Server 版本 5.0（或 5.0 PU1）中的署名信息。进行实时迁移之前，请先在单独的线程中运行和测试署名，以便了解署名在 Content Delivery Server 版本 5.1 中的部署情况。署名迁移可能需要很长时间，具体取决于定制的复杂程度和要署名的站点数量。

署名迁移可能包括（但不限于）以下操作：

- 迁移图像
- 迁移使用 JavaServer Pages™ (JSP™) 技术创建的页面
- 迁移本地化和动态标题
- 迁移样式表
- 创建设备样式表
- 迁移目录和文件
- 迁移特定于运营商的定制

注 – 切记添加 Content Delivery Server 旧版本中不包含的任何所需的新信息（如库）。还必须重置默认设备、默认售卖计划、开发者计划的值以及其他此类管理信息。

此步骤假定您熟悉 Content Delivery Server 署名和定制过程。有关详细信息，请参见《Sun Java™ System Content Delivery Server 5.1 署名指南》和《Sun Java™ System Content Delivery Server 5.1 定制指南》。

6. 测试核心 Content Delivery Server 功能，其中包括旧服务和新服务、所有已署名的门户以及基本集成组件。

有关详细信息，请参见第 6 页上的“测试迁移”。

7. 部署任何定制集成适配器。

此步骤假定在迁移阶段之前，所有定制适配器均符合 Content Delivery Server 的要求，或者已对这些适配器进行了扩展以便与 Content Delivery Server 结合使用。

8. 测试完全集成的 Content Delivery Server 部署。

请注意，必须按照规范对服务器进行全面配置，包括在执行迁移之前配置工作流。有关如何编辑 SubmissionVerifiedWorkflows.xml 文件以满足部署需求的信息，请参见《Sun Java™ System Content Delivery Server 5.1 安装指南》。

尝试在开发或实施环境中执行初始迁移测试，以确定迁移 Content Delivery Server 配置所需的大致时间。

迁移 Content Delivery Server

即使使用的是同一数据库服务器，也要创建新的数据库，这是因为 Content Delivery Server 新版本的结构可能与旧版本中使用的结构不同。

注 – 在此迁移阶段，Vending Manager 可以继续运行并为客户提供服务。

在 Catalog Manager 数据库迁移过程中，必须满足以下要求：

- 以只读模式运行 Catalog Manager，直至迁移过程结束。
- 不能对 Catalog Manager 数据库中的内容进行任何修改。任何修改操作都会破坏数据迁移，并使 Content Delivery Server 数据库的新旧版本之间产生差异。
- 此时，无法向 Catalog Manager 提交任何新内容。

但请注意，对于 Vending Manager，可以任意储存和取消储存内容。

根据其从属关系指定迁移的开始时间，以设置预期完成时间并确定停止 Catalog Manager 的时间。

此阶段包括以下高级步骤：

1. 停止运行在初始阶段部署的 **Content Delivery Server** 新版本的所有管理器和服务器。
2. 将 **Catalog Manager** 置于只读模式以限制对 **Catalog Manager** 数据库的访问。
只读表示不能对 **Catalog Manager** 数据库进行任何修改，即，开发者不能提交任何内容，管理员不能进行任何修改。**Catalog Manager** 和 **Developer Portal** 均被视为已冻结。不要进行任何影响数据库的更改。以下操作可帮助您防止更改数据库：
 - 禁用开发者帐户。
 - 仅保留一个 **Catalog Manager** 管理员帐户，禁用所有其他 **Catalog Manager** 管理员帐户。
 - 如果需要，请重新启动 **Catalog Manager** 和 **Developer Portal** 以丢弃所有登录的现有用户。
3. 将 **Catalog Manager** 数据库迁移到 **Content Delivery Server** 新版本。
有关迁移数据库的详细步骤，请参见第 4 章。
4. (可选) 运行迁移验证脚本。
在执行每个迁移步骤后，验证数据库迁移。有关运行该脚本的信息，请参见第 4 章中的第 51 页上的“验证迁移完成情况”。
5. 停止在 **Content Delivery Server** 旧版本上运行的 **Catalog Manager** 和 **Vending Manager**。
 - a. 如果 **Catalog Manager** 和 **Vending Manager** 位于相同的服务器上，则关闭这两个管理器。
 - b. 如果 **Catalog Manager** 和 **Vending Manager** 位于不同的服务器上，则关闭 **Catalog Manager**。
Vending Manager 可以继续运行，但必须停止售卖服务（例如，事件、消息传送、后付费和监视服务）。
6. 将 **Vending Manager** 数据库迁移到 **Content Delivery Server** 新版本。
Vending Manager 迁移过程中，**Vending Manager** 或其关联的 **Subscriber Portal** 上不会发生任何活动。
7. (可选) 运行迁移验证步骤。
8. 在 **Content Delivery Server** 新版本中启动管理器。
 - a. 如果 **Catalog Manager** 和 **Vending Manager** 位于相同的服务器上，则在 **Content Delivery Server** 新版本中启动这两个管理器。
 - b. 如果 **Catalog Manager** 和 **Vending Manager** 位于不同的服务器上，则只能在 **Content Delivery Server** 新版本中运行 **Catalog Manager** 之后启动 **Vending Manager**。

9. 关闭仍在 Content Delivery Server 旧版本上运行的所有其他 Vending Manager，并停止它们的所有服务。
10. 将所有其他 Vending Manager 数据库迁移到 Content Delivery Server 新版本。
11. (可选) 运行迁移测试。
12. (可选) 将特定内容类型设置为只读。

迁移过程中，某些内容类型可能不包含可以将该内容类型设置为可编辑或只读的属性。要确保某些内容类型不可编辑，在所有 Catalog Manager 和 Vending Manager 的迁移和迁移验证过程完成后，请执行以下步骤：

a. 确保停止了所有部署。

b. 运行以下 SQL 命令：

```
update content_type set editable=0 where name in
('bundle', 'midlet', 'text', 'iappli', 'streaming_video',
'streaming_audio');
```

必须对目录数据库（带有 _PS_APP 后缀）和所有相应的售卖数据库（带有 _VS_APP 后缀）运行 SQL 命令。

部署 Content Delivery Server 新版本后，SQL 命令中指定的内容类型将被视为只读。

13. 在 Content Delivery Server 新版本中启动其他 Vending Manager。

现在，可以启动 Vending Manager 服务。还可以停止可能已运行的任何服务中断计数器。现在，您正在运行 Content Delivery Server 版本 5.1。不要尝试重新使用 5.0（或 5.0 PU1）版服务器，因为对该数据库所做的任何更改都不会反映在新数据库中。

测试迁移

为确保 Content Delivery Server 迁移成功完成，必须对以下迁移部分进行全面的测试：

- 数据库和属性配置
- 署名
- 单个服务器和服务迁移
- 集成

最后，在迁移完每个列出的部分后，测试完全集成和署名的 Content Delivery Server 版本。

迁移测试包括（但不限于）以下部分：

- 生产客户数据导入（专用网络和降低的访问量）
- 署名和设备用户接口迁移
- 数据库和属性迁移脚本
- 批处理、实时记帐和用户配置文件的集成迁移测试
- 迁移测试计划制订（递进测试用例）

有关迁移验证的信息，请参见第 4 章中的第 51 页上的“验证迁移完成情况”。

第2章

配置属性

本章说明了将 Content Delivery Server 版本 5.0 转换为 Content Delivery Server 版本 5.1 时需要创建和设置的属性，并明确指出了从版本 5.0 PU1 迁移到版本 5.1 的特定步骤。

本章包括以下主题：

- [设置配置属性](#)
 - [样例配置文件](#)
-

设置配置属性

1. 按正常方式配置新版本的 Content Delivery Server，其中包括设置 `.cfg` 配置文件。

在 `.cfg` 配置文件中，检查旧版本和新版本中的 Vending Manger 服务器帐户名称属性是否相同。如果不相同，完成属性迁移后，请在配置 (`conf`) 目录中手动设置为新版本指定的 Vending Manger 服务器帐户名称。

2. 运行以下部署命令：

```
cdsi deploy
```

3. 打开属性迁移文件 `$CDS_HOME/cfg/migration/props_migration_5.0_5.1.xml` 并对其进行以下编辑（如果从版本 5.0 PU1 迁移，则编辑文件 `$CDS_HOME/cfg/migration/props_migration_5.0PU1_5.1.xml`）：

- a. 替换 [[CDS New Home]] 以将 new-cds-home 设置为 **Content Delivery Server 新版本 (5.1) 的安装目录**，并替换 [[CDS Template Home]] 以将 template-home 设置为 **Content Delivery Server 旧版本 (5.0 或 5.0 PU1, 如果适用) 的安装目录**。

旧版本的安装包含 Content Delivery Server 旧版本的配置和本地化文件模板。如果安装不包含这些模板，请应用可从支持部门获取的最新修补程序。必须提供绝对路径。

模板必须合并修补程序提供的与 template-home 所指向的文件相对应的所有文件和属性。在运行迁移之前，这些模板必须包含属性迁移所需的所有属性。

- b. 替换 [[CDS Old Home]] 以将 old-cds-home 设置为 **Content Delivery Server 旧版本 (5.0 或 5.0 PU1, 如果适用) 的安装目录**。
必须提供绝对路径。
- c. 替换 [[CDS New Deployment Name]] 以将 new-name 的所有实例设置为 **Content Delivery Server 新版本的部署名称**。
- d. 替换 [[CDS Old Deployment Name]] 以将 old-name 的所有实例设置为 **Content Delivery Server 旧版本的部署名称**。
- e. 替换 [[CDS Old Distribution Name]] 以将 template-name 的所有实例设置为 **cds**。

此发行版包含 Content Delivery Server 旧版本的配置和本地化文件模板。

4. 针对 `$CDS_HOME/cfg/migration/props_migration_5.0_5.1.xml` (或 `$CDS_HOME/cfg/migration/props_migration_5.0PU1_5.1.xml` 文件, 如果适用) 运行 `cdspm migrateprops -f props_filename` 命令。

下一节中的样例属性配置文件提供了如何设置这些属性的示例。

编辑 workflow

在设置配置属性后，您需要修改 `SubmissionVerifiedWorkflow.xml` 文件以便部署 Content Delivery Server。有关如何编辑该文件以满足部署需求的信息，请参见《Sun Java™ System Content Delivery Server 5.1 安装指南》。如果选择创建自己的 `SubmissionVerifiedWorkflow.xml` 文件，请参见《Sun Java™ System Content Delivery Server 5.1 集成指南》。

样例配置文件

可以使用以下样例属性配置文件来配置 Content Delivery Server 属性。

代码示例 2-1 样例配置 XML 文件

```
<props
  new-cds-home="/home/cds_new_version"
  old-cds-home="/home/cds_old_version"
  template-home="/home/cds_old_version">
  <path
    new-name="deployment/cds/conf"
    old-name="deployment/cds/conf"
    template-name="dist/cds/conf/templates"
    action="merge">
    <file
      new-name="ConfFile.properties"
      old-name="ConfFile.properties"
      action="merge"/>
    </path>
  <path
    new-name="deployment/cds/localization"
    old-name="deployment/cds/localization"
    template-name="dist/cds/localization"
    action="merge">
    <file
      new-name="NewL10nFile.properties"
      old-name="OldL10nFile.properties"
      action="merge"/>
    </path>
  </props>
```

注 – 在本示例中，"old_version" 是指版本 5.0 或版本 5.0 PU1（视您的具体情况而定）。

转换文件

本章介绍了 Content Delivery Server 从版本 5.0 到 5.1，其 API 以及桌面和设备 Subscriber Portal 所发生的变更。

相关变更项目下面说明了从版本 5.0 PU1 到 5.1 所发生的变更。

在版本 5.1 中，以下 Content Delivery Server API 发生了转换：

- [内容管理 API 的变更](#)
- [内容验证 API 的变更](#)
- [订户 API 的变更](#)
- [用户配置文件 API 的变更](#)
- [WAP 网关 API 的变更](#)

内容管理 API 的变更

从版本 5.0 到 5.1，新版本中的内容管理 API 所发生的变更和添加的内容如下所示：

- 向内容管理 API 中添加了以下类：
 - ContentPreview.java
 - MimeType.java
 - Model.java
- 添加了 Popularity.java 接口。
此新接口用于计算内容项的相对受欢迎程度值。
- ContentConstants.java 具有以下新字段：
 - `public static final java.lang.String KEY_CONTENT_PREVIEWS`
 - `public static final java.lang.String TARGET_INTERFACE_WAP`

- `public static final java.lang.String TARGET_INTERFACE_WEB`
- `ContentInfo.java` 具有以下新方法:
 - `public void setValue(java.lang.String key, java.lang.Object value)`
 - `public java.lang.Object getValue(java.lang.String key)`
- 从 `ContentInfo.java` 中删除了以下方法:
 - `protected void set(java.lang.String key, java.lang.Object value)`。

内容验证 API 的变更

新版本中的内容验证 API 所发生的变更和添加的内容如下所示:

- 向 `ValidationContent.java` 中添加了以下字段:
 - `public static java.lang.String KEY_AUTO_PUBLISH`
- 向 `ValidationContent.java` 中添加了以下方法:
 - `getCategories(java.lang.String delimiter)`
 - `getContentPreviews()`
 - `getContentType()`
 - `getDeveloperContentId()`
 - `getDeveloperName()`
 - `getDeveloperPlans()`
 - `getDeveloperUsername()`
 - `getDisplayName()`
 - `getEditionName()`
 - `getLocationType()`
 - `getMatchedModels()`
- 在 `Validationcontent.java` 中, 发生了以下参数变更:
 - `ValidationContent` 构造函数的 `java.util.Map` 变更为 `java.util.Map<java.lang.String,byte[]>`。
 - `setCapabilityMap` 方法的 `java.util.Map` 变更为 `java.util.Map<java.lang.String,java.lang.String[]>`。
 - `setMimeBytesMap` 方法的 `java.util.Map` 变更为 `java.util.Map<java.lang.String,byte[]>`。
- 在 `Validationcontent.java` 中, 发生了以下返回类型变更:

- `getCapabilityMap` 方法的 `java.util.Map` 变更为 `java.util.Map<java.lang.String,java.lang.String[]>`。
- `getMimeBytesMap` 方法的 `java.util.Map` 变更为 `java.util.Map<java.lang.String,byte[]>`。

订户 API 的变更

新版本中的订户 API 所发生的变更和添加的内容如下所示：

- 向订户 API 中添加了以下接口：
 - `IContentPreview`
 - `ICustomField`
 - `IMimeType`
 - `IStreamingService`
- 添加了 `ApiContextFactory.java` 的以下方法：
 - `createAnonymousApiContext(java.lang.String localeCode, long modelId, boolean expires)`
 - `isAuthKeyValid(java.lang.String authKey)`
 - `removeApiContext(java.lang.String authKey)`
- 添加了 `ApiUtil.java` 的以下方法：
`isContentExpired(long contentId)`
- 添加了 `CDSException.java` 的以下字段：
 - `public static final java.lang.String CDS_EX_DOWNLOAD_BEFORE_START_DATE_NOT_ALLOWED`
在版本 5.0 PU1 中添加了该字段。
 - `public static final java.lang.String CDS_EX_INVALID_AUTH_KEY`
- 删除了 `CDSException.java` 的以下字段：
`public static final java.lang.String CDS_EX_INVALID_URL_TICKET`
- 向 `IApiContext.java` 中添加了以下方法：`getAuthKey()`
- 向 `ICategory.java` 中添加了以下方法：`getSortingQuery()`
- 在 `ICategoryService.java` 中，以下方法的参数 `java.util.Collection` 变更为 `java.util.Collection<ICategory>`：
 - `hideCategory()`
 - `moveCategoryDown()`
 - `moveCategoryUp()`
 - `showCategory()`

- 在 `ICategoryService.java` 中, 以下方法的返回类型 `java.util.Collection` 变更为 `java.util.Collection<ICategory>`:
 - `getNotEmptySubCategories()`
 - `getSubCategories()`
 - `hideCategory()`
 - `moveCategoryDown()`
 - `moveCategoryUp()`
 - `showCategory()`
- 添加了 `IContentDetails.java` 的以下方法:
 - `getUserGuideBinary()`
 - `getSmallIconBinary()`
 - `java.util.Collection<IContentPreview> getPreviews(int target)`
 - `java.util.Collection<IContentPreview> getPreviews()`
 - `getLargeIconBinary()`
 - `getDeviceIconBinary()`
- `IContentDetails.java` 的以下方法已过时:
 - `getPreviewUrl()`
改用 `getPreviews(int)`。
 - `getScreenShot1Url()`
改用 `getPreviews(int)`。
 - `getScreenShot2Url()`
改用 `getPreviews(int)`。
 - `isPreviewEnabled()`
- 删除了 `IContentDetails.java` 的以下方法:
 - `getContentClassId()`
在版本 5.0 PU1 中删除了该方法。
 - `getContentKeyword()`
在版本 5.0 PU1 中删除了该方法。
 - `getContentType()`
在版本 5.0 PU1 中删除了该方法。
 - `getDeveloperName()`
在版本 5.0 PU1 中删除了该方法。
 - `getLongDescription()`

- `getSizeInKB()`
- `getVersion()`
- `isBundle()`
- 在 `IContentList.java` 中, `getContentSummaries` 方法的 `java.util.Collection` 返回类型变更为 `java.util.Collection<IContentSummary>`。
- 添加了 `IContentService.java` 的以下字段:
 - `public static final java.lang.String DETAILS_INCLUDE_CUSTOM_FIELDS`
 - `public static final java.lang.String DETAILS_INCLUDE_RESOURCE_BINARIES`
 - `public static final java.lang.String TARGET_INTERFACE_WAP`
 - `public static final java.lang.String TARGET_INTERFACE_WEB`
- `IContentService.java` 发生了以下变更:
 - 添加了方法 `searchContent(java.lang.String searchQuery, long categoryId, java.util.Map<java.lang.String, java.lang.Boolean> sortKeyMap, int numberToReturn, int startIndex)`。
 - 方法 `IContentList searchContent(long categoryId, java.lang.String searchStr, long[] contentTypeIds, int numberToReturn, int startIndex)` 已过时。
改用 `searchContent(String, long, Map, int, int)`。
- 在 `IContentService.java` 中, 方法 `getContentDetails(java.util.Collection<IContentSummary> contentSummaries, IContentDetailsCriteria criteria)` 的 `java.util.Collection` 参数变更为 `java.util.Collection<IContentSummary>`。
- 在 `IContentService.java` 中, 以下已过时方法的 `java.util.Collection` 参数变更为 `java.util.Collection<IContentSummary>`:
 - `getContentDetails(java.util.Collection<IContentSummary> contentSummaries, java.util.BitSet filter, long campaignId, long bundleId, boolean isSkipTrial)`
 - `getContentDetails(java.util.Collection<IContentSummary> contentSummaries, java.util.BitSet filter, long campaignId, long bundleId, boolean retailPrice, boolean isSkipTrial)`
- 在 `IContentService.java` 中, 所有 `getContentDetails` 方法的 `java.util.Collection` 返回类型都变更为 `java.util.Collection<IContentDetails>`。

- 在 `IContentService.java` 中, `getPurchases` 方法的 `java.util.Collection` 返回类型变更为 `java.util.Collection<IPurchaseDetails>`。
- 在 `IContentService.java` 中, `getSupportedModels` 方法的 `java.util.Collection` 返回类型变更为 `java.util.Collection<IModel>`。
- 在 `IContentService.java` 中, `getBookmarks` 方法的 `java.util.Collection` 返回类型变更为 `java.util.Collection<IContentSummary>`。
- 在 `IContentService.java` 中, `getCampaigns` 方法的 `java.util.Collection` 返回类型变更为 `java.util.Collection<ICampaign>`。
- 添加了 `IContentType.java` 的以下方法:
 - `getContentClassId()`
 - `getContentKeyword()`
 - `getContentType()`
 - `getCustomFields()`
 - `getDeveloperName()`
 - `getLongDescription()`
 - `getSizeInKB()`
 - `getVersion()`
 - `isBundle()`
- 在 `IDownloadService.java` 中, 以下方法的 `java.util.Map` 参数变更为 `java.util.Map<java.lang.String,java.lang.Object>`:
 - `pushSMSContent(java.lang.String mobileId, byte[] contentBinary, java.lang.String mimeType, java.lang.String contentType, java.util.Map<java.lang.String,java.lang.Object> params)`
 - `pushSMSContent(java.lang.String ticket, java.util.Map<java.lang.String,java.lang.Object> params)`
- 向 `IGiftingService.java` 中添加了以下方法:


```
giftContent(long contentId, long campaignId, IApiContext
recipientCtx, java.lang.String message, int giftedDownloads,
int giftedSubscriptions,
java.util.Map<java.lang.String,java.lang.String>
billingParams)
```
- 在 `IGiftingService.java` 中, 以下方法的 `java.util.Collection` 返回类型变更为 `java.util.Collection<IGifting>`:
 - `getGiftingsByGifter()`
 - `getGiftingsByRecipient()`

- 在 `ISystemService.java` 中，`getContentTypes()` 方法的 `java.util.Collection` 返回类型变更为 `java.util.Collection<IContentType>`。
- 在 `ISystemService.java` 中，`getCountries()` 方法的 `java.util.Collection` 返回类型变更为 `java.util.Collection<ICountry>`。
- 在 `ISystemService.java` 中，`getLocales()` 方法的 `java.util.Collection` 返回类型变更为 `java.util.Collection<ILocale>`。
- 在 `ISystemService.java` 中，以下方法的 `java.util.Collection` 返回类型变更为 `java.util.Collection<IModel>`:
 - `getModels()`
 - `getModels(java.lang.String manufacturer)`
- `IUserService.java` 的以下方法已过时:
 - `getCategoryService(IApiContext context)`
改用 `getCategoryService(String)`。
 - `getCategoryService(java.lang.String authKey)`
 - `getContentService(IApiContext context)`
改用 `getContentService(String)`。
 - `getContentService(java.lang.String authKey)`
 - `getDownloadService(IApiContext context)`
改用 `getDownloadService(String)`。
 - `getDownloadService(java.lang.String authKey)`
 - `getGiftingService(IApiContext context)`
改用 `getGiftingService(String)`。
 - `getGiftingService(java.lang.String authKey)`
 - `getMessageService(IApiContext context)`
改用 `getMessageService(String)`。
 - `getMessageService(java.lang.String authKey)`
 - `getStreamingService(IApiContext context)`
改用 `getStreamingService(String)`。
 - `getStreamingService(java.lang.String authKey)`
 - `getSystemService(IApiContext context)`
改用 `getSystemService(String)`。
 - `getSystemService(java.lang.String authKey)`
 - `getUserService(IApiContext context)`

改用 `getUserService(String)`。

- `getUserService(java.lang.String authKey)`
- 在 `IUserService.java` 中, `setUserPreferences()` 方法的 `java.util.Map` 参数变更为 `java.util.Map<java.lang.String, java.lang.Object>`。
- 在 `IUserService.java` 中, `setUserProperties()` 方法的 `java.util.Map` 参数变更为 `java.util.Map<java.lang.String, java.lang.Object>`。
- 在 `IUserService.java` 中, `signup()` 方法的 `java.util.Map` 参数变更为 `java.util.Map<java.lang.String, java.lang.String>`。
- 在 `IUserService.java` 中, `getUserPreferences()` 方法的 `java.util.Map` 返回类型变更为 `java.util.Map<java.lang.String, java.lang.Object>`。
- 在 `IUserService.java` 中, 以下方法的 `java.util.Map` 返回类型变更为 `java.util.Map<java.lang.String, java.lang.String>`:
 - `getUserProperties()`
 - `getUserProperties(long subscriberId)`
 - `getUserProperties(java.lang.String userName)`

用户配置文件 API 的变更

新版本中的用户配置文件 API 所发生的变更和添加的内容如下所示:

- 添加了接口 `UserDeviceModel.java`。
- `User.java` 的以下方法已过时:
 - `getAttributes()`
 - `getAttribute(java.lang.String name)`
 - `getAttribute(java.lang.String name, java.lang.Object defaultValue)`
 - `getCreateDate()`
 - `getLastLogin()`
 - `hasLoggedIn()`
 - `isConfirmed()`
 - `setAttribute(java.lang.String name, java.lang.Object value)`
 - `setAttributes(java.util.Hashtable attributes)`
 - `setCreateDate(java.util.Date date)`
 - `setHasLoggedIn(boolean value)`
 - `updateLastLogin()`
- `UserManager.java` 具有以下新字段:

- `public static final long DEVICE_MODEL_IGNORE`
- `public static final long DEVICE_MODEL_UNKNOWN`
- `userManager.java` 的以下字段已过时：
 - `public static final int ID`
 - `public static final int LAST_NAME`
 - `public static final int FIRST_NAME`
 - `public static final int LOGIN_NAME`
 - `public static final int STATUS`
 - `public static final int KEY`
- `userManager.java` 的以下方法已过时：
 - `accountExists(java.lang.String userId)`
 - `disableUser(java.lang.String userId)`
改用 `disableUser(User)`。
 - `doAccountExists(java.lang.String userId)`
 - `doEnableUser(java.lang.String userId)`
改用 `doEnableUser(User)`。
 - `doGetAllLikeInOrder(java.lang.String[] columns, java.lang.String[] values, java.lang.String[] orders, boolean isDescending, int pageNum, int recPerPage, java.lang.String role)`
 - `doGetAllUsers(java.lang.String role)`
 - `doGetAllUsersContainingFirstName(java.lang.String firstName, java.lang.String role)`
 - `doGetAllUsersContainingId(java.lang.String userId, java.lang.String role)`
 - `doGetAllUsersContainingLastName(java.lang.String lastName, java.lang.String role)`
 - `doGetAllUsersContainingName(java.lang.String name, java.lang.String role)`
 - `doGetAnonymousUser()`
 - `doGetFieldName(int fieldConstant, java.lang.String role)`
 - `doGetUser(java.lang.String userId)`
改用 `doGetUserByLoginId(String)`。
 - `doIsActive(java.lang.String userId)`
 - `doIsAuthenticated(java.lang.String inUserId, java.lang.String inPassword)`

改用 `doAuthenticatePassword(User, String)`。

- `doRemoveUser(java.lang.String userId)`
改用 `doRemoveUser(User)`。
- `enableUser(java.lang.String userId)`
改用 `enableUser(User)`。
- `getAllLikeInOrder(java.lang.String[] columns,
java.lang.String[] values, java.lang.String[] orders,
boolean isDescending, int pageNum, int recPerPage)`
- `getAllLikeInOrder(java.lang.String[] columns,
java.lang.String[] values, java.lang.String[] orders,
boolean isDescending, int pageNum, int recPerPage,
java.lang.String role)`
- `getAllUsers()`
- `getAllUsers(java.lang.String role)`
- `getAllUsersContainingFirstName(java.lang.String firstName)`
- `getAllUsersContainingFirstName(java.lang.String firstName,
java.lang.String role)`
- `getAllUsersContainingId(java.lang.String userId)`
- `getAllUsersContainingId(java.lang.String userId,
java.lang.String role)`
- `getAllUsersContainingLastName(java.lang.String lastName)`
- `getAllUsersContainingLastName(java.lang.String lastName,
java.lang.String role)`
- `getAllUsersContainingName(java.lang.String name)`
- `getAllUsersContainingName(java.lang.String nameLike,
java.lang.String role)`
- `getAnonymousUser()`
- `getFieldName(int fieldConstant)`
- `getFieldName(int fieldConstant, java.lang.String role)`
- `getUser(java.lang.String userId)`
改用 `getUserByLoginId(String)`。
- `isActive(java.lang.String userId)`
- `isAuthenticated(java.lang.String inUserId, java.lang.String
inPassword)`
改用 `authenticatePassword(User, String)`。
- `removeUser(java.lang.String userId)`
改用 `removeUser(User)`。

有关已过时安全方法的完整列表，请参见 `$CDS_HOME/javadoc/cdsapi` 目录中的 `deprecated-list.html` 文件。

WAP 网关 API 的变更

新版本中的 WAP 网关 API 发生了变更并添加了内容。向 `WAPGatewayAdapter.java` 中添加了以下字段：

- `public static final java.lang.String HANDLE_MSISDN`
- `public static final java.lang.String HANDLE_MSISDN_MAP`
- `public static final java.lang.String HANDLE_UNIQUE_ID`
- `public static final java.lang.String HANDLE_UNIQUE_ID_MAP`

第4章

迁移数据库

本章介绍了如何将数据库从 Content Delivery Server 版本 5.0 和 5.0 PU1 迁移到 Content Delivery Server 版本 5.1。

本章包括以下主题：

- [迁移步骤和标签](#)
- [迁移先决条件](#)
- [配置迁移](#)
- [测试内容](#)
- [配置示例](#)
- [运行迁移](#)

迁移步骤和标签

数据库迁移过程包含一系列**步骤**，其中的每个步骤均可视为一个单独的迁移过程。这些步骤可划分为目录读取/写入步骤或售卖读取/写入步骤。在开始售卖步骤之前，必须先完成目录步骤。读取步骤必须始终在相应的写入步骤之前执行。以下列表显示了单个服务器部署的迁移步骤集必须采用的顺序：

1. 目录读取
2. 目录写入
3. 售卖读取
4. 售卖写入

在运行目录写入步骤时，迁移过程将运行 Content Delivery Server 服务器代码，以便将所有内容传送到提交和验证 workflow。此过程可能需要使用大量内存。为防止出现 OutOfMemory 错误，请在运行目录写入步骤之前设置 EXTRA_JAVA_ARGS 环境变量，以增加 Java 虚拟机允许的最大堆大小。例如，

```
$ EXTRA_JAVA_ARGS="-Xmx1500M" cdsi db migrate db -s code -c -w
```

不要更改最小堆大小。如果将最小堆大小设置为较大的值，则会导致 Java 虚拟机一经启动便请求更多的内存。当 Java 虚拟机需要对 Oracle 进程进行派生处理时，操作系统将无法完成执行请求。这是因为，每次 JVM 虚拟机需要启动进程时，它必须先创建自身的副本，但可能没有足够的可用操作系统内存来执行此操作。

数据库迁移使用**标签**来存储和装入转换的数据。标签是一个目录，在执行读取和写入步骤之间将在其中存储迁移数据。在迁移目录服务器及其关联的所有售卖服务器期间，必须使用相同的标签，因为在执行售卖步骤期间将引用目录数据。

对于远程售卖，请将售卖数据的标签保存在 NFS 可访问（读取和写入）的位置中。应将目录数据的标签复制到远程售卖站点，因为目录标签共享会影响 Content Delivery Server 性能。

与标签关联的目录采用以下结构：

- 目录数据：
/label/catalog
- 每个 Vending Manager 的售卖数据：
/label/vending/VENDING_NAME

如果迁移需要配置文件所定义的定制实现，则需要扩展 Java 类路径以使其包含实现路径。在运行迁移步骤之前，可通过设置 EXTRA_JAVA_CP 环境变量来指定路径。例如：

```
$ EXTRA_JAVA_CP="/tmp/extra.jar" cdsi db migrate db -s code -c -r
```

迁移先决条件

需要满足以下条件才能运行数据库迁移：

- 必须安装至少与 JDK 版本 1.5.0_11 兼容的 Java 运行时环境。
- 必须安装 Content Delivery Server 发行版 5.1。
- 必须安装 Oracle 客户端。有关详细信息，请参见第 27 页上的“环境变量设置”。
- 需要具有数据库服务器的访问权限，以便在其上执行所有写入步骤。
- 在运行软件的系统上安装软件的本地副本，以免出现重定位或重新配置问题。

迁移术语和变量

在本指南的其余部分，术语“源数据库”是指 Sun Java System Content Delivery Server 版本 5.0（或 5.0 PU1，如果适用）数据库。术语“目标数据库”是指 Sun Java System Content Delivery Server 版本 5.1 数据库。

环境变量设置

您必须定义 CDS_HOME 和 ORACLE_HOME 变量，并且必须在 PATH 中指定以下目录：

- CDS_HOME/bin - 其中 CDS_HOME 是 Content Delivery Server 的安装目录。
- ORACLE_HOME/bin - 其中 ORACLE_HOME 是 Oracle 服务器的安装目录，它必须位于运行迁移的同一台计算机上。有关必须运行的 Oracle 数据库版本，请参见《Sun Java™ System Content Delivery Server 5.1 安装指南》；通常，该版本至少为 Oracle Database 9i 版本 9.2.0.6 或 Oracle Database 10g 发行版 1 (10.1.0.2)。

Oracle 主目录可以指服务器安装或客户端安装创建的 Oracle 主目录。对于每一种情况，必须完全安装 Oracle 数据库实用程序。

请注意，如果售卖源和目标数据库运行不同版本的 Oracle 服务器，则可能会出现导出错误。请确保两个数据库使用相同的服务器版本；否则，请确保遵循以下规则：

- Import 实用程序和将数据导入到的数据库（目标数据库）必须位于同一版本的服务器上。
- Export 实用程序版本与源数据库的服务器版本必须相同。

例如，如果源数据库在 Oracle Database 9i 上运行，目标数据库在 Oracle Database 10g 上运行，则 Export 实用程序必须在 Oracle Database 9i 上运行，Import 实用程序必须在 Oracle Database 10g 上运行。

另外，还要确保正确配置 init_env.sh 文件中的路径。

配置迁移

数据库迁移只需要一个配置文件，即 DBMigration.xml，它位于默认目录 \$CDS_HOME/cfg/migration/ 中。此配置文件可用于所有迁移步骤。本节介绍用于创建配置文件的规则，并定义 DBMigration.xml 文件中使用的标记。

可以将创建的数据库迁移配置文件命名为 DBMigration.xml 以外的名称，但它必须位于 \$CDS_HOME/cfg/migration 目录中，并且在执行 cdsi migrate 命令时必须指定该文件。

迁移文件标记约定

在创建配置文件时，请遵循以下一般性规则：

- 在标记之间指定的任何文本数据不能包含空格、回车符或换行符。

正确：

```
<label>/path</label>
```

错误：

```
<label>
/path
</label>
```

- 所有标记和属性名称必须为小写。
- 不要使用迁移程序无法识别的标记。如果在迁移过程中遇到无法识别的标记，则会忽略此类标记并发出警告消息。请注意，所有标记和属性名称必须为小写。
- 如果要覆盖运行的并行线程数，请指定一个正值。有关 `<threads>` 标记的信息，请参见第 28 页上的“配置文件标记”。如果指定 0（默认值），则线程数等于系统上的 CPU 数。

如果指定的线程数大于 Oracle 会话中支持的数量，则会导致出现错误。请确保指定的数量不超过 Oracle 支持的会话数。

另请注意，如果在高端计算机上运行并且 CPU 数大于 Oracle 中支持的会话数，则会出现错误。

配置文件标记

本节定义了 `<dbmigration>` 标记以及可在该标记中使用的标记。`<dbmigration>` 标记是必需的。它是顶层标记并包含所有其他迁移配置标记，而且必须是 XML 文件中的根标记。以下标记在 `<dbmigration>` 中有效：

表 4-1 配置文件标记

标记名称	标记描述
<dbmigration>	<p>此标记是必需的。</p> <p>它是顶层标记配置文件标记。它包含所有其他迁移配置标记，并且必须是 XML 文件中的根标记。</p>
<catalog>	<p>必需的。</p> <p>包含特定于目录的迁移配置部分。以下标记在 <catalog> 中有效: <catalog_events>、<classpath>、<deployment> 和 <server_config>。</p> <p>此标记在 <server> 中有效。</p>
<catalog_database_link>	<p>必需的。</p> <p>用于使远程 Vending Manager 服务器能够访问目录数据库的链接。</p> <p>如果将源 Vending Manager 服务器和源 Catalog Manager 服务器部署在同一个数据库上，并且具有相同的 SID，则按以下方式创建目录数据库链接：</p> <pre>create database link catalog_database_link connect to SRC_CM_PS_APP identified by SRC_CM_PS_APP_Password using 'SRC_CM_SID'</pre> <p>如果将源 Catalog Manager 服务器和源 Vending Manager 服务器部署在不同的数据库上，则按以下方式创建目录数据库链接：</p> <p>首先，修改 \$ORACLE_HOME/network/admin 中的 tnsnames.ora 文件以创建本地服务名称：</p> <pre>database_link_name = (description = (address = (protocol = TCP) (host = source_catalog_database_host) (port = 1521)) connect_data = (server = dedicated) (service_name = source_catalog_sid)))</pre> <p>然后，使用刚创建的本地服务名称创建数据库链接：</p> <pre>create database link catalog_database_link connect to SRC_CM_PS_APP identified by SRC_CM_PS_APP_Password using 'database_link_name'</pre> <p>此标记在 <vending> 中有效。</p>

表 4-1 配置文件标记 (续)

标记名称	标记描述
<catalog_events>	<p>不是必需的。</p> <p>表示将目录事件存储到此 Catalog Manager 的事件表中。如果未将 Catalog Manager 与 Vending Manager 部署在一起, 请使用此标记。该值表示售卖代码, 此代码用于提取现有目录事件。使用可选的 pk 属性可以限制迁移的事件。有关详细信息, 请参阅本节中的 <event_pk> 标记描述。</p> <p>此标记在 <catalog> 中有效。</p>
<classpath>	<p>不是必需的。</p> <p>指定一个附加类路径, 在引导服务器时必须将其添加到默认类路径中。如果服务器提交代码依赖于其他库, 则通常会使用此标记。如果使用 <classpath>, 它必须至少包含一个 <element> 标记。</p> <p><element> 表示单个类路径元素。在该标记的值中可以识别以下宏:</p> <ul style="list-style-type: none">• \$cds.dist - 扩展为 Content Delivery Server 分发目录的完整路径。• \$cds.home - 扩展为 Content Delivery Server 主目录的完整路径。 <p>此标记在 <catalog> 中有效。</p>
<coverage>	<p>为验证过程创建定制范围定义。该范围指定必须为某些表验证的记录百分比。</p> <p>每个 coverage 标记可以为表中的数据量指定不同的范围, 或者为表名以及表中要测试的数据量指定不同的范围。它具有以下属性:</p> <ul style="list-style-type: none">• min - 范围匹配所需的最小行数• max - 范围匹配所需的最大行数• name - 用于匹配范围的表名 <p>通过提供最小或最大值或提供表名可指定表范围。</p> <p>这些属性的默认值适用于以下情况:</p> <ul style="list-style-type: none">• 如果未指定 min 属性, 则其默认值为 0。• 如果未指定 max 属性, 则其默认值为可能的最大 64 位带符号值。 <p>有关详细信息, 请参见第 33 页上的“默认范围值”。</p> <p>在进行验证之前, 应将表名和行数传递到 coverage 标记列表。首先, 检查该名称; 如果名称匹配, 则选择该范围。然后, 检查行数是否匹配。如果行数匹配, 则选择该范围。</p> <p>此标记在 <dbmigration> 中有效。</p>
<deployment>	<p>必需的。</p> <p>指定迁移过程在内容重新提交期间使用的部署名称。</p> <p>此标记在 <catalog> 中有效。</p>

表 4-1 配置文件标记（续）

标记名称	标记描述
<destination>	<p>不是必需的。</p> <p>为执行写入步骤期间使用的标签目录指定路径。如果在不同的系统上执行这些步骤，文件系统路径可能会有所不同。</p> <p><destination> 具有一个可选的 style 属性。如果执行写入步骤的系统使用不同的路径分隔符（例如，使用 DOS 而不是 UNIX® 标准），请指定相应的 dos 或 unix 作为 style 属性的值。</p> <p>此标记在 <label> 中有效。</p>
<event_pk>	<p>不是必需的。</p> <p>在 CDS_EVENT 表中指定主键。如果 CDS_EVENT 表中包含该主键，则会将表中的数据提取限制为该主键的值以及具有更大值的主键。迁移仅复制主键值大于或等于指定主键值的 CDS_EVENT 条目。</p> <p>包括具有指定主键的行。</p> <p>此标记在 <vending> 中有效。</p>
<label>	<p>必需的。</p> <p>指定从源数据库中提取的数据在文件系统存储位置。以下标记在 <label> 中有效：<destination> 和 <location>。</p> <p>此标记在 <server> 中有效。</p>
<location>	<p>必需的。</p> <p>指定标签位置的物理路径。</p> <p>此标记在 <label> 中有效。</p>
<server>	<p>如果在命令行中指定服务器代码，则必须使用此标记。</p> <p>包含服务器配置条目。每个 Catalog Manager 及其关联的所有 Vending Manager 必须具有一个 <server> 标记。</p> <p><server> 使用 code 属性，该属性可区分各个服务器条目。配置文件不能包含多个具有相同 code 值的 <server> 标记。</p> <p>以下标记在 <server> 中有效：<catalog>、<label>、<source_dbconf>、<target_dbconf> 和 <vending>。</p> <p>此标记在 <dbmigration> 中有效。</p>
<server_config>	<p>不是必需的。</p> <p>指定用于在内容重新提交期间启动部署的主 Content Delivery Server 配置文件。默认值为 CDS.properties。</p> <p>此标记在 <catalog> 中有效。</p>
<source_account>	<p>必需的。</p> <p>指定源配置的售卖帐户名称。售卖帐户名称用于从数据库配置文件中提取正确的数据库连接信息，并且还用于从源数据库中提取数据。</p> <p>此标记在 <vending> 中有效。</p>

表 4-1 配置文件标记 (续)

标记名称	标记描述
<source_dbconf>	必需的。 指定数据库配置文件 DBConf.xml, 该文件用于访问 Catalog Manager 以及为该 Catalog Manager 迁移的所有 Vending Manager 数据库。该文件采用的标准必须与 Content Delivery Server 版本 5.0 的数据库配置文件兼容。唯一接受的值是文件名。将在 \$CDS_HOME/cfg 目录中搜索该文件。此标记在 <server> 中有效。
<target_account>	不是必需的。 指定目标配置的售卖帐户名称。<target_account> 用于从数据库配置文件中提取正确的数据库连接信息。如果未使用此标记, 可改用 <source_account> 的值。
<target_dbconf>	必需的。 指定数据库配置文件, 该文件用于访问 Catalog Manager 以及为该 Catalog Manager 迁移的所有 Vending Manager 数据库。该文件采用的标准必须与 Content Delivery Server 版本 5.1 的数据库配置文件兼容。唯一接受的值是文件名。将在 \$CDS_HOME/cfg 目录中搜索该文件。此标记在 <server> 中有效。
<threads>	不是必需的。 指定运行迁移时启动的并行线程数。使用整数 0 或更大的值。如果指定 0, 则线程数与位于系统上的 CPU 数相匹配。默认值为 0。 此标记在 <dbmigration> 中有效。
<vending>	如果使用售卖代码, 则必须使用此标记。 包含与单个 Vending Manager 相关的完整配置。<vending> 使用 CODE 属性, 该属性可区分各个服务器条目。配置文件不能包含多个具有相同 code 值的 <vending> 标记。以下标记在 <vending> 中有效: <catalog_database_link>、<event_pk>、<source_account> 和 <target_account>。 此标记在 <server> 中有效。

默认范围值

如果没有 `coverage` 标记，则使用以下默认范围：

```
<verification>
<!-- do full coverage, if a table has less than 100 entries -->
<coverage max="100">100.0</coverage>
<!-- do 21% coverage for tables with (100,10000] entries -->
<coverage min="101" max="10000">21.0</coverage>
<!-- do 5% coverage for tables with more than 10000 entries -->
<coverage min="10001">5.0</coverage>
<!-- do 0.1% coverage for fulfillment_request table -->
<coverage table="fulfillment_request">0.1</coverage>
</verification>
```

将评估这些默认值是否可作为最佳范围值，以及运行验证脚本所花的时间。将按 `coverage` 标记在配置文件中的出现顺序对其进行处理。如果最小和最大范围重叠，扫描将在第一个匹配标记处停止。指定表名的标记是一个例外，将始终先检查这些标记。

可以按文本形式使用浮点值，该值将被指定为 `<coverage>` 标记的文本值。

以下示例显示如何为 `category_item` 表选择 95% 范围：

```
<coverage name="category_item">95</coverage>
```

以下示例显示如何为所有表选择 3% 范围：

```
<coverage>3</coverage>
```

要迁移单个服务器部署，您必须至少定义两个目录步骤（一个读取步骤和一个写入步骤）和两个售卖步骤（一个读取步骤和一个写入步骤）。请参见第 39 页上的“运行迁移”一节，以了解如何在执行期间调用这些步骤。

有关迁移文件示例，请参见第 33 页上的“配置示例”。

配置示例

本节提供了三个配置示例，它们代表了使用最广泛的部署方案。

单个服务器部署

本示例显示单个服务器部署的一个可能的配置文件。

代码示例 4-1 单个服务器部署的样例配置文件

```
<dbmigration>

  <threads>0</threads>

  <server code="server1">
    <source_dbconf>DBConf.xml.source</source_dbconf>
    <target_dbconf>DBConf.xml.target</target_dbconf>
    <label>
      <location>/export/space0</location>
      <!--
      <destination style="dos">o:\mount°|xport\space0</destination>
      -->
    </label>

    <catalog>
      <!--
      <classpath>
        <element>${cds.dist}/lib/external/bcel-5.1.jar</element>
      </classpath>
      -->

      <catalog_events pk="131123">vending1</catalog_events>
      <deployment>cds</deployment>
      <server_config>CDS.properties</server_config>
    </catalog>

    <vending code="vending1">
      <source_account>suncds.vending.server.1</source_account>
      <target_account>java.vending</target_account>
      <catalog_database_link>local_database_link</catalog_database_link>
    </vending>

    <!--
    <coverage name="fulfillment_request">0.1</coverage>
    <coverage max="100">100</coverage>
    <coverage min="101" max="10000">21</coverage>
    <coverage min="10001">5</coverage>
    -->

  </server>

</dbmigration>
```

远程售卖部署

本示例显示一个可能的配置文件，它用于包含一个 Catalog Manager 服务器和两个 Vending Manager 服务器（一个本地服务器和一个远程服务器）的部署。

代码示例 4-2 远程售卖部署的样例配置文件

```
<dbmigration>

  <threads>0</threads>

  <server code="server1">
    <source_dbconf>DBConf.xml.source</source_dbconf>
    <target_dbconf>DBConf.xml.target</target_dbconf>

    <label>
      <location>/export/space0</location>
      <!--
      <destination style="dos">o:\mount°|xport\sace0</destination>
      -->
    </label>

    <catalog>

      <!--
      <classpath>
        <element>${cde.dist}/lib/external/bcel-5.1.jar</element>
      </classpath>
      -->

      <catalog_events pk="131123">vending1</catalog_events>
      <deployment>cde</deployment>
      <server_config>CDS.properties</server_config>
    </catalog>

    <vending code="vending1">
      <source_account>suncde.vending.server.1</source_account>
      <target_account>java.vending</target_account>
      <catalog_database_link>local_database_link</catalog_database_link>
    </vending>

    <vending code="vending2">
      <source_account>suncde.vending.server.2</source_account>
      <target_account>ringtone.vending</target_account>
      <catalog_database_link>src.remote_database_link</catalog_database_link>
    </vending>

    <!--
    <coverage name="fulfillment_request">0.1</coverage>
```

代码示例 4-2 远程售卖部署的样例配置文件（续）

```
<coverage max="100">100</coverage>
<coverage min="101" max="10000">21</coverage>
<coverage min="10001">5</coverage>
-->

</server>

</dbmigration>
```

记录迁移信息

在执行目录写入步骤期间，服务器代码会将所有内容传送到提交和验证工作流。服务器代码中的所有日志记录信息将存储在 `$CDS_HOME/dist/cds/logs` 目录下的 `migration.server.log.*` 文件中。要配置其他日志记录属性（如日志文件的最大大小、日志文件的最大数量以及日志记录阈值），请编辑 `LABEL_DIRECTORY/catalog` 目录中名为 `logging.properties` 的文件。该文件是在执行目录读取步骤期间创建的，仅在目录写入步骤期间使用它。

`migration.log` 文件包含迁移消息（调试消息除外），可以在 `$CDS_HOME/dist/cds/logs` 目录中找到该文件。在解决迁移期间遇到的问题时，请检查迁移日志文件。有关迁移日志文件的详细信息，请参见第 5 章中的第 55 页上的“消息”。

测试内容

现在，测试是通过内容重新提交进行处理的。请根据 Content Delivery Server 的部署需要修改 `SubmissionVerifiedWorkflows.xml` 文件。有关信息，请参见《Sun Java™ System Content Delivery Server 5.1 安装指南》。

迁移 CDS_EVENT 表

使用 `<event_pk>` 标记可以优化 CDS_EVENT 表的迁移过程。事件表主要用于报告目的。它提供了一些信息，可帮助您缩短迁移单个售卖服务器的执行时间。可通过使用 `<event_pk>` 标记来指定迁移的第一个主键。您可以在服务器启用并运行后迁移事件表的其余部分。还必须相应地修改 `userid` 参数。有关 `<event_pk>` 标记的描述，请参见第 28 页上的“配置文件标记”。

例如，如果配置文件中指定的 <event_pk> 为 12345，则可以使用以下命令来迁移 CDS_EVENT 表的其余部分：

```
$ exp 'userid=CDS_VS_OWNER/dbpasswordx@CONNECT' \  
      'file=CDS_EVENT.dmp' \  
      'constraints=n' 'grants=n' 'indexes=n' \  
      'log=export.log' \  
      'query="where cds_event_id < 12345"' \  
      'tables=CDS_EVENT' \  
      'feedback=1000'  
  
$ imp 'userid=CDS_VS_OWNER/dbpasswordx@CONNECT' \  
      'tables=CDS_EVENT' \  
      'log=import.log' \  
      'file=CDS_EVENT.dmp' \  
      'ignore=y' \  
      'feedback=1000'
```

如果目录服务器是单独部署的（即，在与售卖服务器不同的服务器上运行），请修改 query 参数以区分目录和售卖事件。请使用

```
'query="where server_instance_id = 0 and cds_event_id < 12345"'
```

来选择目录事件并将其写入到 PS_OWNER 模式中。请使用

```
'query="where server_instance_id != 0 and cds_event_id < 12345"'
```

来选择售卖事件并将其写入到 VS_OWNER 模式中。

还必须相应地修改 userid 参数。<connect> 参数是 Oracle 客户端工具可解析为服务器连接的名称。如果未在系统上设置命名服务，则可以使用数据库实例的直接寻址：

```
$ exp 'userid="CDS_PS_OWNER/passwordx@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(  
(PROTOCOL=TCP)(HOST=olddb.wireless.com)(PORT=1521)))(CONNECT_DATA=(SID=  
old_sid)))"' \  
      'file=CDS_EVENT.dmp' \  
      'constraints=n' 'grants=n' 'indexes=n' \  
      'log=export.log' \  
      'query="where cds_event_id < 12345"' \  
      'tables=CDS_EVENT' \  
      'feedback=1000'  
  
$ imp 'userid="CDS_PS_OWNER/passwordx@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(  
(PROTOCOL=TCP)(HOST=newdb.wireless.com)(PORT=1521)))(CONNECT_DATA=(SID=  
new_sid)))"' \  
      'tables=CDS_EVENT' \  
      'log=import.log' \  
      'feedback=1000'
```

```
'file=CDS_EVENT.dmp' \  
'ignore=y' \  
'feedback=1000'
```

有关 Oracle 命名模式的详细信息，请参阅 Oracle 文档。

迁移单独目录部署的事件

如果新部署的 Catalog Manager 是单独运行的，则必须指定 <catalog_event> 标记。在指定该标记时，会将所有售卖事件表中特定于目录的事件迁移到目录事件表中。有关 <catalog_event> 标记的描述，请参见第 28 页上的“配置文件标记”。

迁移版本的支持设备信息

本节中的信息适用于以下情况：Catalog Manager 管理员更改了一个或多个内容项的支持设备。

通常，提交到 Content Delivery Server 版本 5.0（或 5.0 PU1）以及版本 5.1 数据库的任何版本都可以具有任意数量的衍生内容版本（由 DRM 创建）。即使未选择 DRM 保护，也仍然会创建衍生版本。Catalog Manager 管理员可以强制为任何衍生版本指定支持设备属性值，从而修改功能匹配例程的输入内容，该例程可确定允许将哪些版本下载到哪些设备上。

由于在迁移期间将内容传送到提交和验证 workflow，因此，无法预测在旧数据库中创建的衍生版本将在新数据库中变为哪个衍生版本，尤其是两个数据库中的版本号可能并不相同。

这种因素会妨碍迁移过程将当前为衍生版本选择的支持设备与新创建的衍生版本相匹配。迁移扩展了支持设备属性列表，以包含为最初提交版本的所有衍生版本选择的设备。

版本的支持设备信息迁移过程是使用以下方法完成的：如果 Catalog Manager 管理员为原始版本的任何衍生版本定制了支持设备属性，则会合并单个原始版本的所有衍生版本的支持设备属性值。在重新提交原始版本期间，将使用该值作为支持设备属性。虽然在重新提交期间使用此属性，但仅衍生版本继承该属性。不会更改最初为原始版本定义的属性值。

运行迁移

在部署 XML 配置文件后，即可开始执行迁移步骤。要执行单个步骤，请执行以下命令：

```
$ cdsi migrate db -s code -c | -v code -r | -w -f file
```

其中

- `-s code` 是服务器代码
- `-c` 表示目录操作
- `-v code` 表示特定售卖服务器上的售卖操作
- `-r` 表示读取操作
- `-w` 表示写入操作
- `-f file` 指定备用配置文件

如果配置文件名称不是 `DBMigration.xml`，请使用此选项。

例如，要运行目录读取步骤，请输入以下命令：

```
$ cdsi migrate db -s code -c -r
```

要使用备用配置文件运行售卖写入步骤，请输入以下命令：

```
$ cdsi migrate db -s code -v code -w -f file
```

在运行数据库迁移命令时，必须指定 `-c` 或 `-v` 以及 `-r` 或 `-w`。

默认数据库迁移文件 `DBMigration.xml` 位于 `$CDS_HOME/cfg/migration/` 目录中。

注 – 在启动迁移过程之后，请不要修改配置文件。如果对该文件进行任何更改，则必须从头开始重新执行迁移过程。

要完成迁移，您必须对 `Catalog Manager` 服务器和正在迁移的所有 `Vending Manager` 服务器执行所有读取和写入步骤。在迁移开始时，`Catalog Manager` 服务器必须处于只读模式，或者必须关闭运行该服务器的应用服务器。只读表示不能对 `Catalog Manager` 数据库进行任何修改，即，开发者不能提交任何内容，管理员不能进行任何修改。如果不满足这些要求，则可能会在迁移期间破坏数据的完整性。

要迁移 `Vending Manager` 服务器，必须关闭其应用服务器。

在迁移期间部署 `Content Delivery Server` 时，请切记以下几点：

- 如果在源数据库服务器上实际执行迁移读取步骤，则可能会加快这些步骤的运行速度。

- 如果在目标数据库服务器上实际执行迁移写入步骤，则可能会加快这些步骤的运行速度。

如果在与源和目标数据库服务器相同的计算机上运行 Content Delivery Server，则可以忽略此项。您不必在数据库服务器上启动 Content Delivery Server。

迁移单个服务器部署

对于单个服务器部署，请执行以下迁移步骤：

1. 限制对 Catalog Manager 和 Developer Portal 的访问。
2. 执行目录读取步骤。
3. 在目标部署上执行目录写入步骤。
4. (可选) 验证 Catalog Manager 迁移是否成功完成。

您可以使用 Content Delivery Server 提供的迁移验证脚本。有关如何运行该脚本的信息，请参见第 51 页上的“验证迁移完成情况”。

5. 授予对源 Vending Manager 模式的以下系统级权限：

- create any synonym 对 *source_vending_schema_VS_APP* 的访问权限

运行以下命令：

```
grant create any synonym to source_vending_schema_VS_APP
```

- drop public synonym 对 *source_vending_schema_VS_APP* 的访问权限

运行以下命令：

```
grant drop public synonym to source_vending_schema_VS_APP
```

- create database link 对 *source_vending_schema_VS_APP* 的访问权限

运行以下命令：

```
grant create database link to source_vending_schema_VS_APP
```

通过使用这些步骤，Vending Manager 应用程序可以在执行售卖读取步骤期间创建同义词和数据库链接。

注 – 如果未授予这些权限，售卖读取步骤将会失败；无法读取 `category_item` 表提取数据，因为在提取期间，只有在创建同义词后才能访问 Catalog 部分的 `resource_instance`。

或者，也可以直接登录到每个 Catalog Manager 和 Vending Manager 数据库并授予公共访问权限。

6. 如果 Catalog Manager 和 Vending Manager 指向同一个数据库（如果不是，请跳到步骤 7），则按以下方式创建指向 `source_vending_schema_VS_APP` 的数据库链接：

a. 以数据库的 `source_vending_schema_VS_APP` 用户身份登录到源 Vending Manager 数据库。

b. 创建指向 `source_vending_schema_VS_APP` 的数据库链接：

```
create database link catalog_database_link
connect to SRC_CM_PS_APP identified by SRC_CM_PS_APP_Password
using 'SRC_CM_CID'
```

通过在此 Vending Manager 数据库服务器的 `DBMigration.xml` 文件中的 `<vending>` 标记部分添加 `<catalog_database_link>` 标记来提供此数据库链接。有关 `<catalog_database_link>` 标记的描述，请参见第 28 页上的“配置文件标记”。

7. 如果 Catalog Manager 和 Vending Manager 指向不同的数据库，则按以下方式创建指向 `source_vending_schema_VS_APP` 的数据库链接：

a. 以系统用户身份登录到源 Vending Manager 数据库。

b. 通过在位于 `$ORACLE_HOME/network/admin` 的网络配置文件 `tnsnames.ora` 中添加以下条目，为远程数据库名称创建本地服务名称：

```
database_link_name =
  (description =
    (address = (protocol = TCP) (host = source_catalog_database_host)
    (port = 1521))
    (connect_data =
      (server = dedicated)
      (service_name = source_catalog_sid)
    )
  )
```

c. 以数据库的 `source_vending_schema_VS_APP` 用户身份登录到源 Vending Manager 数据库。

d. 使用刚创建的本地服务名称创建数据库链接:

```
create database link catalog_database_link
connect to SRC_CM_PS_APP identified by SRC_CM_PS_APP_Password
using 'database_link_name'
```

通过在此 Vending Manager 数据库服务器的 DBMigration.xml 文件中的 <vending> 标记部分添加 <catalog_database_link> 标记来提供此数据库链接。有关 <catalog_database_link> 标记的描述, 请参见第 28 页上的“配置文件标记”。

8. 如果将 Catalog Manager 和 Vending Manager 部署在不同的数据库上, 则在位于 \$CDS_HOME/cfg 目录下的 DBSysUser.xml 文件中为数据库创建数据库系统用户和密码:

```
<DBUsers>
  <DBServer name = "DBServer1">
    <User>system</User>
    <Password>dbpassword</Password>
  </DBServer>

  <DBServer name = "DBServer2">
    <User>system</User>
    <Password>dbsyspassword</Password>
  </DBServer>
</DBUsers>
```

有关 DBSysUser.xml 文件的详细信息, 请参见《Sun Java™ System Content Delivery Server 5.1 安装指南》。

9. 编辑此远程 Vending Manager 的 DBmigration.xml 文件以指向修改后的 (源) DBConf.xml 文件。
10. 完全关闭 Content Delivery Server 的单个服务器。
11. 执行售卖读取步骤。
12. 执行售卖写入步骤。
13. (可选) 验证 Vending Manager 迁移是否成功完成。
14. 启动新的 (目标) 服务器。

远程售卖部署的迁移准则

要确保售卖读取步骤成功完成，请检查您是否具有从 Vending Manager 访问目录模式的权限，该模式可能部署在单独的计算机上。如果使用代理服务器，请确保为数据库使用的端口启用代理。

迁移远程售卖部署

对于至少具有一个远程 Vending Manager 的部署（即，在与 Catalog Manager 不同的服务器上运行 Vending Manager），请执行以下迁移步骤。该方案假定至少有一个 Vending Manager（表示为 VM）在单个服务器模式下与 Catalog Manager 一起运行。远程 Vending Manager 表示为 RVM。

此过程的第一部分介绍如何迁移源 Catalog Manager 和 Vending Manager。此过程的第二部分介绍如何迁移远程 Vending Manager。

1. 限制对 Catalog Manager 和 Developer Portal 的访问。

2. 执行目录读取步骤。

3. 在目标部署上执行目录写入步骤。

4. （可选）验证 Catalog Manager 迁移是否成功完成。

您可以使用 Content Delivery Server 提供的迁移验证脚本。有关如何运行该脚本的信息，请参见第 51 页上的“验证迁移完成情况”。

5. 授予对源 Vending Manager (VM) 模式的以下系统级权限：

- create any synonym 对 `source_vending_schema_VS_APP` 的访问权限
运行以下命令：

```
grant create any synonym to source_vending_schema_VS_APP
```

- drop public synonym 对 `source_vending_schema_VS_APP` 的访问权限
运行以下命令：

```
grant drop public synonym to source_vending_schema_VS_APP
```

- create database link 对 `source_vending_schema_VS_APP` 的访问权限
运行以下命令：

```
grant create database link to source_vending_schema_VS_APP
```

通过使用这些步骤，Vending Manager 应用程序可以在执行售卖读取步骤期间创建同义词和数据库链接。

注 – 如果未授予这些权限，售卖读取步骤将会失败；无法读取 `category_item` 表提取数据，因为在提取期间，只有在创建同义词后才能访问 Catalog 部分的 `resource_instance`。

或者，也可以直接登录到每个 Catalog Manager 和 Vending Manager 数据库并授予公共访问权限。

6. 以 `source_vending_schema_VS_APP` 用户身份登录到源 Vending Manager (VM) 数据库。
7. 如果 Catalog Manager 和 Vending Manager (VM) 指向同一个数据库，则按以下方式创建指向 `source_vending_schema_VS_APP` 的数据库链接：
 - a. 以数据库的 `source_vending_schema_VS_APP` 用户身份登录到源 Vending Manager (VM) 数据库。
 - b. 创建数据库链接：

```
create database link catalog_database_link
connect to SRC_CM_PS_APP identified by SRC_CM_PS_APP_Password
using 'SRC_CM_CID'
```

通过在此 Vending Manager 数据库服务器的 `DBMigration.xml` 文件中的 `<vending>` 标记部分添加 `<catalog_database_link>` 标记来提供此数据库链接。有关 `<catalog_database_link>` 标记的描述，请参见第 28 页上的“配置文件标记”。

8. 如果 Catalog Manager 和 Vending Manager (VM) 指向不同的数据库，则按以下方式创建指向 `source_vending_schema_VS_APP` 的数据库链接：
 - a. 以系统用户身份登录到源 Vending Manager (VM) 数据库。
 - b. 通过在位于 `$ORACLE_HOME/network/admin` 的网络配置文件 `tnsnames.ora` 中添加以下条目，为远程数据库名称创建本地服务名称：

```
database_link_name =
  (description =
    (address = (protocol = TCP) (host = source_catalog_database_host)
    (port = 1521))
    (connect_data =
      (server = dedicated)
      (service_name = source_catalog_sid)
    )
  )
```

- c. 以数据库的 `source_vending_schema_VS_APP` 用户身份登录到源 Vending Manager 数据库。

d. 使用刚创建的本地服务名称创建数据库链接:

```
create database link catalog_database_link
connect to SRC_CM_PS_APP identified by SRC_CM_PS_APP_Password
using 'database_link_name'
```

通过在此 Vending Manager 数据库服务器的 DBMigration.xml 文件中的 <vending> 标记部分添加 <catalog_database_link> 标记来提供此数据库链接。有关 <catalog_database_link> 标记的描述, 请参见第 28 页上的“配置文件标记”。

9. 如果将 Catalog Manager 和 Vending Manager 部署在不同的数据库上, 则在位于 \$CDS_HOME/cfg 目录下的 DBSysUser.xml 文件中为数据库创建数据库系统用户和密码:

```
<DBUsers>
  <DBServer name = "DBServer1">
    <User>system</User>
    <Password>dbpassword</Password>
  </DBServer>

  <DBServer name = "DBServer2">
    <User>system</User>
    <Password>dbsyspassword</Password>
  </DBServer>
</DBUsers>
```

有关 DBSysUser.xml 文件的详细信息, 请参见《Sun Java™ System Content Delivery Server 5.1 安装指南》。

10. 关闭旧的 Catalog Manager 和 (源) Vending Manager (VM) 服务器。
11. 为 VM 执行售卖读取步骤。
12. 为 VM 执行售卖写入步骤。
13. (可选) 验证 VM 迁移是否成功完成。

迁移远程 Vending Manager

1. 通过添加 Catalog 元素来修改源远程 Vending Manager (表示为 RVM) 的 DBConf.xml 文件, 以使远程 Vending Manager 可以访问目录数据库:

```

<CDSDatabase>

  <DBServer name="DBServer1">
    <Hostname>dbhost1host.domain.com</Hostname>
    <Port>port_number</Port>
    <SID>dbsid</SID>
    <Tablespace>USERS</Tablespace>
    <Tablespace name="catalog">USERS</Tablespace>
    <Tablespace name="catalog indexes">USERS</Tablespace>
    <Tablespace name="content">USERS</Tablespace>
    <Tablespace name="content indexes">USERS</Tablespace>
    <Tablespace name="event">USERS</Tablespace>
    <Tablespace name="event indexes">USERS</Tablespace>
    <Tablespace name="subscriber">USERS</Tablespace>
    <Tablespace name="subscriber indexes">USERS</Tablespace>
    <Tablespace name="vending">USERS</Tablespace>
    <Tablespace name="vending indexes">USERS</Tablespace>
    <TempTablespace>TEMP</TempTablespace>
  </DBServer>

  <Catalog>
    <Prefix>5.0_schema</Prefix>
    <Password>dbpassword</Password>
    <DBServer>DBServer2</DBServer>
  </Catalog>

  <Vending name="cds.vending.server.1">
    <Prefix>5.0_schema</Prefix>
    <Password>dbpassword</Password>
    <DBServer>DBServer2</DBServer>
  </Vending>

  <DBServer name="DBServer1">
    <Hostname>dbhost2.domain.com</Hostname>
    <Port>port_number</Port>
    <SID>dbsid2</SID>
    <Tablespace>USERS</Tablespace>
    <Tablespace name="catalog">USERS</Tablespace>
    <Tablespace name="catalog indexes">USERS</Tablespace>
    <Tablespace name="content">USERS</Tablespace>
    <Tablespace name="content indexes">USERS</Tablespace>
    <Tablespace name="event">USERS</Tablespace>
    <Tablespace name="event indexes">USERS</Tablespace>
    <Tablespace name="subscriber">USERS</Tablespace>
    <Tablespace name="subscriber indexes">USERS</Tablespace>
    <Tablespace name="vending">USERS</Tablespace>
    <Tablespace name="vending indexes">USERS</Tablespace>

```

```
<TempTablespace>TEMP</TempTablespace>
</DBServer>

</CDSDatabase>
```

在本示例中，DBServer1 表示部署了源远程 Vending Manager (RVM) 的计算机。DBServer2 表示部署了源 Catalog Manager 的计算机。

Catalog 和 Vending 元素的前缀指定了 5.0（或 5.0 PU1，如果适用）模式。

在完成迁移过程后，请从此文件中删除 Catalog 元素，因为仅迁移过程需要使用此元素。

源远程 Vending Manager (RVM) 数据库的 DBConf.xml 中不包含 Catalog Manager 模式，因此，必须手动添加该信息。

有关数据库配置文件的详细描述，请参见《Sun Java™ System Content Delivery Server 5.1 安装指南》。

2. 在位于 %CDS_HOME/cfg 目录下的 DBSysUser.xml 文件中，为 Catalog 数据库创建数据库系统用户和密码：

```
<DBUsers>
  <DBServer name = "DBServer1">
    <User>system</User>
    <Password>dbpassword</Password>
  </DBServer>

  <DBServer name = "DBServer2">
    <User>system</User>
    <Password>dbsyspassword</Password>
  </DBServer>
</DBUsers>
```

有关 DBSysUser.xml 文件的详细信息，请参见《Sun Java™ System Content Delivery Server 5.1 安装指南》。

3. 以系统用户身份登录到源远程 Vending Manager (RVM) 数据库计算机上，并通过在位于 \$ORACLE_HOME/network/admin 的网络配置文件 tnsnames.ora 中添加以下条目，为远程数据库名称创建本地服务名称：

```
database_link_name =
  (description =
    (address = (protocol = TCP) (host = source_catalog_database_host)
    (port = 1521))
    (connect_data =
      (server = dedicated)
```

```
(service_name = source_catalog_sid)
)
)
```

4. 在登录到（源）远程 Vending Manager (RVM) 数据库后，授予对 SRC_RVM_VS_APP 的以下权限：

- create any synonym 对 SRC_RVM_VS_APP 的访问权限
运行以下命令：

```
grant create any synonym to SRC_RVM_VS_APP
```

- drop public synonym 对 SRC_RVM_VS_APP 的访问权限
运行以下命令：

```
grant drop public synonym to SRC_RVM_VS_APP
```

- create database link 对 SRC_RVM_VS_APP 的访问权限
运行以下命令：

```
grant create database link to SRC_RVM_VS_APP
```

5. 以 SRC_RVM_VS_APP 用户身份登录到源远程 Vending Manager 数据库。

6. 使用刚创建的本地服务名称创建数据库链接：

```
create database link catalog_database_link
connect to SRC_CM_PS_APP identified by SRC_CM_PS_APP_Password
using 'database_link_name'
```

通过在此 Vending Manager 数据库服务器的 DBMigration.xml 文件中的 <vending> 标记部分添加 <catalog_database_link> 标记来提供此数据库链接。有关 <catalog_database_link> 标记的描述，请参见第 28 页上的“配置文件标记”。

7. 编辑此远程 Vending Manager 的 DBmigration.xml 文件以指向修改后的（源）DBConf.xml 文件。

8. 部署目标远程 Vending Manager (RVMX)，并编辑目标 DBConf.xml 文件以指向 Catalog Manager 模式。

以下样例 DBConf.xml 文件指向所添加的 Catalog Manager 模式：

```
<CDSDatabase>

  <DBServer name="DBServer1">
    <Hostname>dbhost1host.domain.com</Hostname>
    <Port>port_number</Port>
    <SID>dbsid</SID>
    <Tablespace>USERS</Tablespace>
    <Tablespace name="catalog">USERS</Tablespace>
    <Tablespace name="catalog indexes">USERS</Tablespace>
    <Tablespace name="content">USERS</Tablespace>
    <Tablespace name="content indexes">USERS</Tablespace>
    <Tablespace name="event">USERS</Tablespace>
    <Tablespace name="event indexes">USERS</Tablespace>
    <Tablespace name="subscriber">USERS</Tablespace>
    <Tablespace name="subscriber indexes">USERS</Tablespace>
    <Tablespace name="vending">USERS</Tablespace>
    <Tablespace name="vending indexes">USERS</Tablespace>
    <TempTablespace>TEMP</TempTablespace>
  </DBServer>

  <Catalog>
    <Prefix>5.1_schema</Prefix>
    <Password>dbpassword</Password>
    <DBServer>DBServer1</DBServer>
  </Catalog>

  <Vending name="cds.vending.server.1">
    <Prefix>5.1_schema</Prefix>
    <Password>dbpassword</Password>
    <DBServer>DBServer2</DBServer>
  </Vending>

  <DBServer name="DBServer2">
    <Hostname>dbhost2.domain.com</Hostname>
    <Port>port_number</Port>
    <SID>dbsid2</SID>
    <Tablespace>USERS</Tablespace>
    <Tablespace name="catalog">USERS</Tablespace>
    <Tablespace name="catalog indexes">USERS</Tablespace>
    <Tablespace name="content">USERS</Tablespace>
    <Tablespace name="content indexes">USERS</Tablespace>
    <Tablespace name="event">USERS</Tablespace>
    <Tablespace name="event indexes">USERS</Tablespace>
    <Tablespace name="subscriber">USERS</Tablespace>
  </DBServer>
</CDSDatabase>
```

```
<Tablespace name="subscriber indexes">USERS</Tablespace>
<Tablespace name="vending">USERS</Tablespace>
<Tablespace name="vending indexes">USERS</Tablespace>
<TempTablespace>TEMP</TempTablespace>
</DBServer>

</CDSDatabase>
```

在本示例中，DBServer1 表示部署了目标 Catalog Manager 的计算机。DBServer2 表示部署了目标远程 Vending Manager (RVM) 的计算机。

Catalog 和 Vending 元素的前缀指定了 5.1 模式。

9. 更新 DBMigration.xml，以使其指向修改后的 DBConf.xml（目标 DBConf 文件）。

远程 Vending Manager 部署的 DBConf.xml 中不包含 Catalog Manager 模式信息，因此，必须手动添加该信息。

10. 在部署了远程 Vending Manager 的计算机上创建 *label_directory/catalog* 目录。

label_directory 是文件系统位置，从源数据库中提取的数据将按照 DBMigration.xml 文件中 <label> 标记指定的方式存储到该位置。

11. 从部署了 Catalog Manager 的计算机上的 *label_directory/catalog* 目录中复制 .completed、.success 和 *.SSO 文件。

在确定 Catalog Manager 数据库中是否正确填充了数据时，需要使用 .completed 和 .success 文件。.SSO 文件包含有关重新测试的内容和 ID 的数据。

12. 编辑此远程 Vending Manager 的 DBmigration.xml 文件以指向修改后的（源）DBConf.xml 文件。

通过在此 Vending Manager 数据库服务器的 DBMigration.xml 文件中的 <vending> 标记部分添加 <catalog_database_link> 标记来提供此数据库链接。

13. 关闭远程 Vending Manager (RVM)。

14. 为 RVM 执行售卖读取步骤。

15. 为 RVM 执行售卖写入步骤。

16.（可选）验证 RVM 迁移是否成功完成。

17. 在目标部署上启动 RVM。

对每个远程 Vending Manager 重复步骤 1 至步骤 17。

更正路径名称差异

请记住，在执行目录写入步骤时，如果在使用网络路径时应用服务器和数据库服务器的 Content Delivery Server 主目录路径不匹配，则可能会出现错误。要更正此问题，请执行以下步骤：

1. 打开 `$CDS_HOME/deployment/deployment_name/conf/SubmissionVerifierWorkflows.xml` 文件。
2. 保存此文件的备份副本。
3. 将 JAR 文件的所有绝对路径替换为数据库服务器上的合适路径。
4. 执行目录写入步骤。

验证迁移完成情况

可以使用验证迁移脚本来验证数据库迁移是否成功完成。要运行验证脚本，请在提示符下输入以下命令：

```
$ cdsi migrate verifydb -s code -c | -v code -f file
```

其中

- `-s code` 是服务器代码
- `-c` 表示目录操作
- `-v code` 表示特定售卖服务器上的售卖操作
- `-f file` 指定备用配置文件

如果配置文件名称不是 `DBMigration.xml`，请使用此选项。

需要使用 `-c` 或 `-v` 标志来标识目录或售卖步骤。

例如，在验证 Catalog Manager 迁移时，请输入以下命令：

```
$ cdsi migrate verifydb -s code -c
```

在验证 Vending Manager 迁移时，请输入以下命令：

```
$ cdsi migrate verifydb -s code -v code
```

在完成迁移验证后，可能需要将某些内容类型设置为只读，您必须运行 SQL 命令将这些内容类型指定为只读，然后才能部署最新版本的 Content Delivery Server。有关详细信息，请参见第 1 章（第 4 页上的“迁移 Content Delivery Server”）中的步骤 12。

注 – Catalog Search Service 和 Vending Search Service 组件使用 Solr 搜索服务器来编索内容以及处理搜索查询。请注意，在完成迁移并启动目标服务器后，生成索引可能需要一些时间。如果在迁移后立即运行搜索查询，搜索结果可能不会从全部内容集中返回匹配内容。

迁移后修改

在 Catalog Manager 和 Vending Manager 迁移完成后，必须更新对现有 Vending Manager Account Server 信息所做的任何更改，以便在 Catalog Manager 和 Vending Manager 之间启用信息同步。要更新该信息，请执行以下步骤：

1. 登录到 Catalog Manager。
2. 单击主菜单栏中的“帐户”。
控制台将显示 Vending Manager Server 的“帐户管理”页。
3. 单击要修改的 Vending Manager Server 帐户。
4. 单击“编辑”。
根据需要，更新帐户信息。要了解如何更新帐户信息，请单击控制台上的“帮助”。
5. 对每个 Vending Manager Server 帐户重复步骤 3 和步骤 4。

生成搜索索引

在完成迁移并启动目标服务器后，需要生成搜索索引。Catalog Search Service 和 Vending Search Service 组件使用 Solr 搜索服务器来编索内容以及处理搜索查询。对于每个服务，必须为 Catalog Manager 和 Vending Manager 中的内容创建索引。搜索索引是独立于内容数据库进行管理的。

要为 Catalog Manager 生成搜索索引，请登录到 Catalog Manager 管理控制台，然后在浏览器的地址字段中输入以下 URL：

```
http://host:port/admin/main/search_update?action=rebuild
```

host:port 是运行 Catalog Manager 的服务器的主机名和端口号。在生成过程完成后，将显示一条消息。

要为 Vending Manager 生成搜索索引，请登录到 Vending Manager 管理控制台，然后在浏览器的地址字段中输入以下 URL：

```
http://host:port/vsadmin/main/search_update?action=rebuild
```

host:port 是运行 Vending Manager 的服务器的主机名和端口号。在生成过程完成后，将显示一条消息。

请注意，生成索引可能需要一些时间。如果在迁移后立即运行搜索查询，搜索结果可能不会从全部内容集中返回匹配内容。有关搜索索引的详细信息，请参见《Sun Java™ System Content Delivery Server 5.1 系统管理指南》。

数据库迁移故障排除

本章提供了将数据库从 Sun Java System Content Delivery Server 版本 2005Q4 迁移到 Sun Java System Content Delivery Server 版本 5.1 的故障排除信息。

本章包括以下主题：

- [消息](#)
- [错误消息](#)
- [警告消息](#)
- [失败步骤故障排除](#)

消息

数据库迁移和验证脚本提供了几种日志记录和跟踪级别，以加快故障排除过程的速度。将使用以下消息级别：

- **调试** — 用于调试的信息。调试消息可用于跟踪迁移过程，并在故障发生时检查该过程，以确定故障发生的原因。不会在终端上输出消息，而仅将其保存在跟踪文件中。调试消息的数量可能很大，因此，跟踪文件可能会变得非常大。在启动新的迁移会话之前，您需要先删除或归档 `migration.log` 和 `migration.trace.log` 文件。会话表示迁移 Catalog Manager 及其对应的所有 Vending Manager。
- **信息** — 这些消息向用户通知状态信息，或者通知用户已做出与数据有关的决定。此类消息还用于向用户通知高级迁移进度。
- **警告** — 当迁移过程遇到与数据库逻辑要求不一致的数据时，这些消息将向用户发出警告。不过，迁移过程可以修复该问题并继续执行。警告还用于向用户通知在配置文件中发现的不一致问题。每次发出警告消息时，迁移过程始终可以继续执行当前操作。

- **错误** — 这些消息表示迁移代码中存在严重问题。如果发出错误消息，则会停止当前操作。不过，在该操作实际停止之前，可能会生成其他错误消息。由于迁移运行并行进程，因此，当由于错误而终止某个进程时，将请求停止所有其他进程。在中断这些进程后，它们还会发出错误消息，指出它们在完成之前已中止。

除了“调试”类别中的消息以外，所有消息将输出到控制台并记录到 `$CDS_HOME/dist/cds/logs/migration.log` 文件中。包括调试消息在内的所有消息将记录到 `$CDS_HOME/dist/cds/logs/migration.trace.log` 文件中。

通常，当迁移过程失败时，会在控制台和日志文件中输出一些错误消息和异常跟踪。要了解失败原因并尽可能多地收集问题的相关信息，一定要从头到尾审查进程输出和日志文件以识别原始错误消息。

在由于迁移尝试失败而与支持部门联系时，请提供屏幕输出、日志文件以及日志跟踪文件。根据这些文件中的信息，可能需要使用标签目录内容或数据库副本以全面调查并更正问题。

以下各节更详细地介绍了错误和警告消息。

注 — 在下一节中，消息正文中的 `%s` 符号是在实际消息中显示的数据的占位符。

错误消息

某些错误消息可能表示迁移脚本配置或环境（如 Content Delivery Server 或 Oracle 服务器）出现问题。本节提供了一个错误消息列表，其中包含详细的错误描述和最可能的原因。可能会出现一系列错误消息，因为层叠模块将报告它们所运行的其他模块上出现的异常。请注意执行失败生成的所有错误消息以及与异常关联的消息。

注 — 以 `IE` 开头的错误消息是内部错误消息。如果违反了代码假定条件，则会显示这些消息。此类情况可能表示数据库已损坏或迁移逻辑失败。您必须与支持部门联系以解决此类错误。

CONFIGURATION Exception!

描述： 只要配置无效，就会出现此消息。此错误可能是由不正确的命令行选项或配置文件造成的。

操作： 检查整个迁移过程输出以获取有关错误的详细信息。

```
Exception while running %s
Exception from task %s
Thread %s failed
```



```
Delaying abnormal exit due to running threads
Exception while parsing arguments
Configuration exception
Verification problem
Initialization problem
DB failure, query : %s, params %s
Migration exception
While executing SQL : %s
Exception in bootstrap()
Initialization failed
STEP %s FAILED
cmd process returned code %s
import returned code %s
MR ' %s' interrupts due to pending exceptions !
Exception while running shutdown task %s
Unexpected null %s (%s)
Thread stopped due to pending exception
While processing edition %s
Execution failed on matrix %s, prefix %s, matrix %s
exception while parsing arguments
Configuration error
```

描述: 这些消息是另一个错误的后续错误消息。

操作: 检查整个迁移过程输出以识别原始错误消息。

```
VERIFICATION OF %s FAILED
VERIFICATION FAILED
Target query returned no rows
Source query returned no rows
Exceeded allowed number of failures (%s) for column %s
FAILED verification of %s
Counters don' t match for prefix %s, table %s. Expected: %s,
found: %s
Target query returned no row for %s,PK %s
Verification failed while comparing column %s, at primary key %s
```

描述: 验证失败。

操作: 检查迁移输出以确定验证失败的实际原因并识别发现故障的表。请与支持部门联系，以确定更正该问题需要执行哪些操作。

```
Failed to login to source database as catalog owner user
Failed to find vending user %s from %s
Server code must be specified
Operation can be either read OR write
Server type must be either catalog OR vending
```

operation can be 'read' or 'write' only, and must be specified
server type can be 'catalog' or 'vending' only, and must be specified

vending code must be specified for 'vending' server type
Error parsing configuration file %s

Expected root 'dbmigration' element

code attribute must be set and not empty for <server> tag

Found two or more <server> elements with code %s

Server code ' %s' was not found in config file

Vending code ' %s' not found in configuration for server ' %s'

Catalog event source ' %s' is not defined

Multiple <catalog> tags for server code ' %s'

Multiple <label> tags for server code ' %s'

Multiple <vending> tags with the same code ' %s' found for server code ' %s'

Source database config must be specified

Target database config must be specified

Can't read source database configuration file ' %s'

Can't read target database configuration file ' %s'

Can't find configuration file %s

No deployment name specified for the <catalog> tag

Deployment ' %s', directory ' %s', can not be read or is not a directory

Cannot access deployment configuration file ' %s'

Expected property file (%s) does not exist

<vending> tag must have 'code' attribute present and not empty

Non directory specified for search %s

<event_pk> tag should contain valid long value, got:%s

Valid pricing option name should be specified in

<change_option_price> element

Valid new price should be specified for <change_option_price> element

<source_account> tag expected for <vending code="%s">

No location specified for the label

Failed to find deployment file %s

min attribute value is not a number

max attribute value is not a number

No value specified for coverage element

coverage value is not a double number : %s

coverage value must be in range of [0..100]

Unknown size character %s in <max_file_size> tag

bad file size specified in <max_file_size> : %s

bad integer specified in <max_index> : %s

Cannot find SOURCE vending schema %s

Cannot find SOURCE event schema %s

Failed to create directory %s

%s exists and not a directory

%s already populated. Verify you don't need existing data and remove this path
Label %s does not exist
Label %s is not a directory
Path %s does not exist. Are you sure %s was created ?
%s is not a directory
Preparation for %s was not completed successfully
This migration step was already successfully completed. If you are sure you would like to retry it, remove the following file:
%s
Directory is not writeable: %s
Cannot find target vending schema %s, in %
Cannot find SOURCE catalog schema %s
Cannot find SOURCE fulfillment schema %s
Cannot find target catalog schema %s
No vending server %s found in the database
More than one vending server %s found in the database
Can't establish connection as a system user. Does DBConf have SystemUser and SystemPassword tags ?

描述: 错误是由未正确配置的文件造成的。迁移配置文件或某些部署配置文件无效。

操作: 更正配置文件并重新启动迁移过程。

Missing primary key %s in table %s

描述: 可能是由于数据损坏造成的。

操作: 与支持部门联系，以帮助识别并修复损坏的数据库。

The catalog data has not yet been submitted to the database. You must complete catalog migration before doing this step

描述: 迁移步骤的执行顺序不正确。

操作: 按本指南中指定的顺序执行迁移步骤。有关详细信息，请参见第 4 章。

DATA file doesn't exist for %s:%s
DUMP file doesn't exist for %s:%s(%s)
Expected to find constraints file %s
At least one record expected to be updated for %s

描述: 标签目录不一致。

操作: 确保在执行不同迁移步骤之间没有对标签目录进行任何修改。如果问题仍然存在，请与支持部门联系。

```
migration is installed incorrectly
Unknown CDS DB version %s
Source Database versions mismatch : found %s, expected %s
```

描述：很可能是由于未正确安装 Content Delivery Server 或迁移修补程序造成的。

操作：确保安装完好，并且正确安装了所有迁移修补程序。如果问题仍然存在，请与支持部门联系。

```
export process failed on %s
Failed import for %s_%s
IMPORT process did not leave or left an empty log file ' %s'
%s failed (sqlldr rc is %s)
%s sqlldr did not leave log file. This means id had not done
anything
```

描述：外部 Oracle 进程失败。

操作：在跟踪文件中检查相应外部进程生成的消息。检查位于标签目录中的外部进程日志文件。出现这些错误通常是因为，未设置 ORACLE_HOME 变量或者 PATH 变量不包含 Oracle 二进制文件目录的路径。如果无法确定故障的根本原因，请与支持部门联系。

其他错误消息通常表示迁移程序代码中存在问题。您可能需要与客户支持部门联系以解决该问题。大多数情况下，这些错误是由于意外数据库状态或定制服务器环境造成的。请确保为客户支持部门提供日志文件和跟踪文件。如果提供标签目录内容以及导出的迁移数据库，则也会对问题解决有所帮助。

警告消息

如果出现警告消息，则表明迁移过程中解析的数据库中出现异常情况。

```
Interrupted while waiting for process output
```

描述：监视外部进程生成的输出的线程收到意外中断。外部进程可能已意外终止，或者另一个迁移线程请求中断所有线程。

操作：无需采取任何操作。

```
Process watcher caught exception
```

描述：监视外部进程生成的输出的线程收到意外异常。外部进程可能已意外终止，或者关闭了它的某个输出通道。

操作：无需采取任何操作。

SQL warning: %s

描述: 提交到 Oracle 服务器的 SQL 查询返回警告消息。

操作: 检查警告消息以确定是否需要采取任何操作。

Unknown tag %s found within tag %s
No tags are expected within tag ' %s'

描述: 迁移配置文件的结构不正确。

操作: 检查并修改迁移配置文件。

Caught exception trying to retrieve environment

描述: 在迁移期间尝试检索用于调试的环境变量时发生异常。

操作: 检查异常消息和跟踪文件以确定是否需要采取任何操作。

Terminating task due to pending exceptions
Terminating thread due to pending exceptions
Exiting due to external error conditions, destroying current
process

描述: 正在执行的线程在其他线程上检测到异常。此消息用于通知您未完成当前任务。迁移将由于线程收到原始异常而失败。还会显示错误消息或其他消息。

操作: 请遵循对错误消息采取的操作。

Missing entry with primary key %s in table %s

描述: 将 BLOB 条目装入数据库的进程发现，源数据库中存在一条包含 BLOB 的记录，但目标数据库中却没有该记录。

操作: 与支持部门联系以确定是否需要采取操作。

Null capability id 16 for one of the derived editions of edition %s.

Poorly formatted capability value for id 16, for one of the
derived editions of edition %s : %s.

Edition %s has different capability value for capability %s.
Old=%s, New=%s.

描述: 数据库包含某一版本的支持设备功能的错误记录，或根本不包含任何记录。这表示数据库可能已损坏。

操作: 与支持部门联系以确定需要采取哪些操作（如果有）。

Inconsistency in LCM table, found a mix of temporary/final records

描述: 用于生成新 LARGE_CAPABILITY_MAP 表的逻辑无法生成所需的记录集。

操作: 与支持部门联系以确定需要采取哪些操作（如果有）。

Model to instance mapping count doesn't match expectations.
Expected : %s, found %s

描述: 可能已经给重新提交和版本重新验证过程带来了安全危害。

操作: 与支持部门联系以确定是否需要采取操作。

Specified number of threads is not an integer

描述: 迁移配置文件中的 <threads> 标记值无效。

操作: 检查迁移配置文件并更正错误。

No verification coverage provided for %s, count=%s, defaulting to %s.

描述: 迁移配置文件指定的验证范围不完整。

操作: 检查迁移配置文件并更正问题。

Source fulfillment_request table reports %s entries per this vending account. %s entries, however, were copied from the source database. Absolute delta : %s
%s entries must be deleted from SUBSCRIBER_GIFT table. Actually deleted %s entries from SUBSCRIBER_GIFT table
%s entries must be deleted from SUBSCRIBER_PURCHASE table. Actually deleted %s entries from SUBSCRIBER_PURCHASE table.

描述: 迁移期间在 FULFILLMENT_REQUEST 表中检测到不一致问题。当该表中的记录与 SUBSCRIBER_PURCHASE 和 SUBSCRIBER_GIFT 表中的所有记录不匹配时，通常会出现这一问题。

操作: 与支持部门联系以确定是否需要采取操作。

Category item %s has no instrumented editions

描述: 在重新提交此版本时，提交工作流不生成任何测试版本。出现此错误很可能是因为，此内容不再适用于任何现有设备。

操作: 检查相应的内容项及其所有版本，以确定其重新提交失败的原因。

Category item %s will be deleted as there are other category items for class %s

Category item %s is marked ' Unstocked' as there are no other category items for class %s

描述: 在 Vending Manager 数据库的类别项处理期间, 包含这些类别项的类具有不再可储存的版本。迁移将删除所有此类版本, 直至只剩下一个版本。然后, 将该版本标记为“取消储存”, 因为在售卖数据库中必须至少有一项表示内容类。

操作: 无。

Target query returned no rows for ' %s' , PK %s

描述: 验证要求存在于源数据库中的所有行在目标数据库中也存在, 不过, 验证表指示如果这些行不存在, 这种情况并不是致命的。此错误通常表示, 在迁移期间由于数据损坏而忽略此表中的某些行。

操作: 与支持部门联系以确定是否需要采取操作

Deployment implements different migration version that does core migration code. Developer portal version: %s, core version : %s

描述: 可能未正确配置迁移文件, 这可能是由于应用的迁移修补程序不正确造成的。请检查所有应用的修补程序的说明, 以验证是否正确按照这些说明进行了操作。

操作: 如果无法通过重新应用迁移修补程序来解决该问题, 请与支持部门联系。

Edition %s doesn' t have content data

描述: 数据库损坏。在迁移期间, 检测到某个版本没有关联的二进制文件。

操作: 与支持部门联系以确定是否需要采取操作

Edition %s failed to verify

描述: 在将指定的版本传送到验证工作流时, 验证将拒绝该版本。数据库仍存储原始版本, 但不会创建任何衍生版本。

操作: 检查相应的内容项及其所有版本, 以确定其重新提交失败的原因。

Edition %s doesn' t have any devices matched to it.

描述: 在进行验证后, 无法确定该版本是否适用于任何当前已知设备。如果在重新测试过程中更改了设备配置文件或版本, 该版本可能不再适用于任何设备。

操作: 检查相关的版本, 并找出不再与任何设备相匹配的原因。

失败步骤故障排除

本节提供的信息可帮助您解决在迁移 Content Delivery Server 数据库时可能遇到的问题。

如果任何读取步骤在迁移期间失败，您需要执行以下操作：

1. 消除导致错误的条件。
2. 删除与读取步骤关联的标签内容。
3. 重新启动该步骤。

目录服务器的标签内容位置为 *label_dir/catalog*。售卖服务器的位置为 *label_dir/vending/target_vending_name*。

如果任何写入步骤在迁移期间失败，数据库可能会处于部分填充状态。在尝试重新运行写入步骤之前，您需要重新对其进行初始化。要重新初始化数据库，请结合此处给出的修改使用第 1 章的第 2 页上的“初始迁移步骤”中所述的 `cdsi db` 命令。

如果目录写入步骤失败，则必须重新对整个数据库执行初始化。要重新初始化整个数据库，请运行以下命令：

```
# cdsi db users
# cdsi db schemas_noc
```

在运行 `cdsi db users` 命令之前，请确保所有软件均未打开到任何数据库前缀的连接。您可以将定制数据库配置文件指定为 `cdsi` 命令的第三个参数。

如果目录写入步骤失败，您只需重新初始化受影响的目录数据库，而将其他售卖数据库和目录数据库保持不变。要重新初始化目录数据库，请运行以下命令：

```
# cdsi db users -cs
```

如果售卖写入步骤失败，您只需重新初始化受影响的售卖数据库，而将其他售卖数据库和目录数据库保持不变。要重新初始化售卖数据库，请运行以下命令：

```
# cdsi db schemas_noc -vs name
```

name 是需要重新初始化的 Vending Manager 服务器的名称（用于 Content Delivery Server 版本 5.1 的服务器）。

如果在失败的写入步骤后无法清除相应的数据库，则可能会导致在重复执行写入步骤时出现不一致的行为。数据库迁移可能无法检测到此类行为。如果迁移的处理时间特别长，则显然是出现这种行为。最终结果是迁移永远也不会完成。

要检测该问题，请将跟踪程序连接到导入进程以查看其是否正确执行。正确的执行过程包括一系列读取和写入系统调用。如果在一分钟内未发出系统调用，则说明导入进程存在问题，并且必须中止迁移。

迁移过程可能会以下列某种方式结束：

- 成功完成
- 失败
- 用户中断进程

请注意，在中断进程时，并非终止迁移过程所产生的所有进程。系统管理员可以识别此类进程并手动将其终止。这些进程可以是 Oracle 导入或导出程序，也可以是 Oracle SQL 加载器程序。即使被手动终止后，进程仍可以将其 Oracle 数据库服务器连接保持为打开状态。如果打开了这些连接，则会禁止使用 `cdsi db` 命令重置相应的数据库。有关如何关闭打开的会话的信息，请参阅 Oracle 文档。

索引

B

备份数据库, 1-2
本地服务名称, 创建, 4-41, 4-44, 4-47
标签, 4-26
部署, 配置, 1-3
部署命令, 2-9

C

catalog 标记, 4-29
Catalog Manager
 迁移要求, 1-4
 限制访问, 1-5
 修改数据库, 1-4
 只读, 1-5
 只读模式, 1-2, 1-4
Catalog Manager 模式, 4-39
catalog_events 标记, 4-30
CDS_EVENT 表, 迁移, 4-36
CDS_HOME 安装目录, 4-27
cdsi db 命令, 5-64
cdsi deploy 命令, 2-9
cdsi migrate 命令, 4-39, 4-51
cdsi 命令, 4-39
cdspm migrate props 命令, 1-3
cdspm migrateprops 命令, 2-10
classpath 标记, 4-30
coverage, 4-30
 指定数据量, 4-30

coverage 标记, 4-30
coverage, 指定数据量, 4-33
初始迁移步骤, 1-2
错误消息, 5-56

D

DBConf.xml 文件, 目标, 4-49
DBConf.xml 文件, 源, 4-45
DBMigration.xml 迁移文件, 4-27
deployment 标记, 4-30
destination 标记, 4-31
单个服务器部署示例, 无测试, 4-34
订户 API 迁移, 1-1
定制集成适配器
 部署, 1-4
读取步骤, 4-25
堆大小设置, 4-26
多售卖部署, 示例, 4-35

E

element 标记, 4-30
event_pk 标记, 4-31

J

集成迁移, 1-3
警告消息, 5-60

L

label 标记, 4-31
location 标记, 4-31

M

migration.server.log 文件, 4-36
目标数据库, 4-27
目录部署, 单独, 4-38
目录和售卖事件, 区分, 4-37

N

内容测试, 4-36

P

配置示例, 4-33
配置属性, 设置, 2-9
配置文件
 数据库
 样例, 4-46, 4-49
配置文件标记
 catalog, 4-29
 catalog_events, 4-30
 classpath, 4-30
 coverage, 4-30
 DBMIGRATION, 4-28
 deployment, 4-30
 destination, 4-31
 element, 4-30
 event_pk, 4-31
 label, 4-31
 location, 4-31
 列表, 4-28
 server, 4-31
 server_config, 4-31
 source_account, 4-31
 source_dbconf, 4-32
 target_account, 4-32
 target_dbconf, 4-32
 threads, 4-32
 vending, 4-32

Q

迁移测试, 1-6
迁移概述
 配置属性迁移文件, 1-3

 迁移署名, 1-3

迁移过程
 测试迁移, 1-6
 阶段, 1-1
迁移后步骤, 4-52
迁移文件标记规则, 4-28
迁移消息级别
 错误, 5-56
 调试, 5-55
 警告, 5-55
 日志, 5-55

R

日志记录属性, 配置, 4-36

S

server 标记, 4-31
server_config 标记, 4-31
source_account 标记, 4-31
source_dbconf 标记, 4-32
数据库
 备份, 1-2
 配置, 4-25, 5-55
 迁移, 4-25, 5-55
数据库配置文件
 样例, 4-46, 4-49
数据库迁移
 标签, 4-26
 步骤, 4-25
 Catalog Manager 状态, 4-39
 单个服务器部署步骤, 4-40
 故障排除, 5-64
 为迁移验证指定定制范围, 4-30
 先决条件, 4-26
 消息级别, 5-55
 运行, 4-39
数据库迁移故障排除, 5-64
数据库实例, 直接寻址, 4-37
署名迁移, 1-3
属性
 编辑迁移文件, 2-9
属性, 配置, 2-9
属性迁移命令, 1-3

属性迁移文件, 1-3
 编辑, 2-9
搜索索引, 生成, 4-52

T

target_account 标记, 4-32
target_dbconf 标记, 4-32
template-home 文件, 2-10
threads 标记, 4-32
threads, 在配置文件中指定, 4-32
tnsnames.ora 文件, 4-41, 4-44, 4-47

V

vending 标记, 4-32

W

网络配置文件, 4-41, 4-44, 4-47
文件
 DBMigration.xml, 4-27
 template-home, 2-10

X

系统权限, 授予, 4-48
写入步骤, 4-25

Y

验证范围, 4-30
验证迁移脚本, 4-51
样例
 数据库配置文件, 4-46, 4-49
样例属性配置文件, 2-11
用户接口迁移, 1-1
源的数据库链接, 创建, 4-42, 4-45, 4-48
源数据库, 4-27

Z

在迁移期间部署 Content Delivery Server, 4-39
支持设备, 迁移版本信息, 4-38

