

IBM® WebSphere® Commerce



Catalog Manager 用户指南

版本 5.4

IBM® WebSphere® Commerce



Catalog Manager 用户指南

版本 5.4

注意

在使用本资料及其支持的产品之前，请阅读第 113 页的『声明』中的信息。

第一版，第一修订版（2002 年 4 月）

本版本适用于下列产品：

- IBM WebSphere Commerce，版本 5.4（程序 5724-A18）

以及以上所列产品的所有后续发行版和修订版，直到在新版本中另有声明为止。确认您正在使用本产品级别的正确版本。

通过您当地的 IBM 代表或 IBM 分部可订购出版物。以下地址不备有出版物。

IBM 欢迎您提出宝贵意见。您可以将意见通过以下任何一种方式发送给我们：

1. 通过电子的形式将意见发送到以下列出的电子邮件地址。

torrcf@ca.ibm.com

如果希望得到答复，请务必包含完整的网络地址。

2. 邮寄至以下地址：

IBM Canada Ltd. Laboratory B3/KB7/8200/MKM
8200 Warden Avenue
Markham, Ontario, Canada L6G 1C7

当您发送信息给 IBM 后，即授予 IBM 非专有权，IBM 可以它认为合适的任何方式使用或分发此信息，而无须对您承担任何责任。

目录

在开始之前	v
本书中使用的约定	v
本书面向的读者	vi
在何处查找更多信息	vi

第 1 部分 Catalog Manager 概述 . . . 1

第 2 部分 转换、装入和抽取数据 . . . 3

第 1 章 简介 5

实用程序	7
管理工具	8

第 2 章 转换文本 9

启动文本转换工具	10
使用文本模式编辑视图	11
创建模式文件	11
打开模式文件	11
保存模式文件	11
编辑模式文件	11
添加元素	11
除去元素	12
替换元素	12
将元素上移一行	12
将元素下移一行	12
添加属性	12
除去属性	12
替换属性	12
将属性上移一行	12
将属性下移一行	13
更改模式文件结构	14
准备模式文件以将数据从 XML 转换为字符限定变量格式	14
使用转换命令编辑视图	15
创建命令	15
除去命令	15
编辑或替换命令	15
将命令上移一行	15
将命令下移一行	16
清除命令	16
使用转换处理视图	16

第 3 章 转换 XML 数据 17

启动 XSL 编辑器	18
处理映射规则构建区域	19
创建映射规则构建区域	19
修改映射规则构建区域	19
删除映射规则构建区域	19
使用 XSL 编辑器	20
创建映射规则	20

元素至元素映射	20
属性至属性映射	20
创建定制映射表达式	20
删除映射规则	21
处理 XML 转换	21
定制映射规则表	21
显示完整的 XSL 规则 / 值表达式	21

第 4 章 生成 DTD 和模式 23

设置 DTD 生成程序	23
生成 DTD	24
生成模式和详细的 XML 文件	26

第 5 章 解析标识 27

设置标识解析器	27
设置标识解析器如何处理时间戳记	28
设置标识解析器如何处理存储	28
设置标识解析器如何处理数据库驱动程序	29
确定如何处理数据	29
选择 load 方法	29
选择 update 方法	30
选择 mixed 方法	30
使用标识解析技术	30
使用标识解析器指定属性文件	31
使用属性文件生成标识	32
使用带有组合键的属性文件	33
使用带有级联的主键的属性文件	34
使用内部别名解析	35
使用内部别名标识解析的部分示例	36
使用唯一索引解析	36
唯一索引解析的部分示例	37
将数据装入 MEMBER 表	38
使用 REFKEYS 表创建外部关系	39
错误疑难解答	40

第 6 章 装入数据 41

设置装入程序	41
忽略输入文件中的元素	42
将 NULL 插入列	42
装入时间戳记和日期数据	42
装入当前时间戳记	43
装入当前时间戳记的示例	44
添加持续时间至当前时间戳记的示例	45
管理事件队列	46
与不同的数据库软件和操作系统一起运行	46
替换组件	49
使用产品顾问搜索空间同步	50
定制产品顾问搜索空间同步	53
确定在使用装入程序时如何处理数据	54
选择装入方法	54
选择导入方法	54

选择 SQL 导入方法	55
其它注意事项	55
装入大文档	56
疑难解答技巧	57

第 7 章 抽取数据 59

创建抽取过滤器	59
设置抽取程序	62

第 8 章 使用装入程序软件包记录程序 . . 63

在您的环境中为 Windows NT、Windows 2000、AIX、Linux 和 Solaris 系统配置记录	63
设置类路径变量的示例	63
指定 com.ibm.wca.logging.configFile 系统属性的示例	63
为装入程序软件包定制记录	64
处理程序	65
过滤器	66
格式	66
示例: WCALoggerConfig.xml 和 WCALogger.dtd	67
WCALoggerConfig.xml	67
WCALogger.dtd	68

第 9 章 使用装入程序软件包错误报告程序 69

第 10 章 配置装入程序软件包命令和脚本 71

第 3 部分 使用 Web Editor 73

第 11 章 设置 Web Editor 75

配置 Web Editor	76
编辑 webeditor.properties 文件	76
更改临时文件的位置	77
使用 DTD 生成程序创建 XML 表单描述文件	78
定制 XML 表单描述	82
编辑表单名称	82
更改字段描述	83
添加下拉菜单	83
添加字段帮助	84
定制搜索结果和工作会话列表	85
编辑 weProcessList 文件	87

编辑 webeditor.xml 文件	88
-------------------------------	----

第 12 章 处理产品目录 89

使用 Web Editor 将记录添加到表	89
使用 Web Editor 修改表中的记录	90
使用 Web Editor 从表中删除记录	91

第 4 部分 命令参考 93

第 13 章 DTD 生成命令 95

用于 Windows、AIX、Linux 和 Solaris 系统的 DTD 生成命令	95
用于 iSeries 系统的 DTD 生成命令	96

第 14 章 抽取命令 99

用于 Windows、AIX、Linux 和 Solaris 系统的抽取命令	99
用于 iSeries 系统的抽取命令	100

第 15 章 标识解析命令 101

用于 Windows、AIX、Linux 和 Solaris 系统的标识解析命令	101
用于 iSeries 系统的标识解析命令	102

第 16 章 装入命令 105

用于 Windows、AIX、Linux 和 Solaris 系统的装入命令	105
用于 iSeries 系统的装入命令	106

第 17 章 文本转换命令 109

用于 Windows、AIX、Linux 和 Solaris 系统的文本转换命令	109
用于 iSeries 系统的文本转换命令	110

第 18 章 XML 转换命令 111

用于 Windows、AIX、Linux 和 Solaris 系统的 XML 转换命令	111
用于 iSeries 系统的 XML 转换命令	111

声明 113

商标和服务标记	114
-------------------	-----

在开始之前

《IBM WebSphere Commerce Catalog Manager 用户指南》提供关于 WebSphere Commerce Catalog Manager 的信息。特别地，它提供有关以下主题的详细信息：

- 使用 Catalog Manager 的工具和实用程序转换、装入和抽取数据。
- 使用 Catalog Manager Web Editor 处理产品目录数据
- Catalog Manager 命令

本书中使用的约定

本书使用以下约定：

黑体字	表示命令或图形用户界面 (GUI) 控件 (例如字段、按钮或菜单选项名称)。
等宽字	表示完全按显示原样输入的文本示例以及文件名和目录路径。
斜体字	用于强调和表示可用自己的值替换的变量。
	表示特定于 WebSphere Commerce Windows® 版的信息。
	表示特定于 WebSphere Commerce Windows NT® 版的信息。
	表示特定于 WebSphere Commerce Windows 2000® 版的信息。
	表示特定于 WebSphere Commerce AIX® 版的信息。
	表示特定于 WebSphere Commerce Solaris™ Operating Environment 版的信息。
	表示特定于 WebSphere Commerce Linux 版的信息。
	表示特定于 WebSphere Commerce IBM @server™ iSeries™ (原来称为 AS/400®) 版的信息。
	表示特定于 DB2® 通用数据库的信息。
	表示特定于 Oracle® 数据库的信息。
	表示特定于 WebSphere Commerce 商务版的信息。
	表示特定于 WebSphere Commerce 专业版的信息。

本书面向的读者

需要了解如何使用 WebSphere Commerce Catalog Manager 的站点开发者、管理员和研究者，以及其它需要了解其功能的商务用户应当阅读本书。

尤其是需要了解如何使用 WebSphere Commerce Catalog Manager 各种功能的 WebSphere Commerce 商店开发者和站点管理员应当阅读本书。



正在执行有计划扩展的 WebSphere Commerce 商店开发者和站点管理员也应当具有以下领域的知识：

- 数据库技术
- Enterprise JavaBeans™ 组件体系结构
- 超文本标记语言（HTML）
- Java™
- JavaServer Pages™ 技术
- VisualAge® for Java™ 企业版版本 3.5 或更高版本
- 可扩展标记语言（XML）

类别和产品经理应当阅读本指南以了解如何使商店开发者知道产品目录、产品和商品需求。类别和产品经理还应当了解如何使用 Web Editor 更新商店产品目录。

在何处查找更多信息

可以从 WebSphere Commerce Web 站点获得此用户指南的可移植文档格式（PDF）。请查看 WebSphere Commerce 技术库 Web 站点以获取本文档的最新版本以及与 WebSphere Commerce 相关的其它信息。

-  http://www.ibm.com/software/webservers/commerce/wc_be/lit-tech-general.html
-  http://www.ibm.com/software/webservers/commerce/wc_pe/lit-tech-general.html

第 1 部分 Catalog Manager 概述

WebSphere Commerce Catalog Manager 提供了一个一般工具箱，此工具箱提供了可以按照要求的顺序排列在一起以解决特定的产品目录管理问题的各种功能。它非常灵活，足以处理对 WebSphere Commerce 模式的定制。

Catalog Manager 提供一种能力，使其可以使用 XML 作为管理信息的方式从多个源将信息聚集到统一的 WebSphere Commerce 系统并将所有那些不同的数据重新映射为标准的产品目录和产品定义格式。

Catalog Manager 为您提供执行以下操作的方式：

- 以 ASCII 和 XML 格式将数据从多个输入源导入到 WebSphere Commerce
- 将数据从 ASCII 格式转换为 XML 格式并再转换回来
- 将数据从一种 XML 格式重新映射为另一种 XML 格式
- 将信息从多个输入流聚集到已聚集的数据库
- 通过 Web 浏览器界面创建 / 编辑 / 删除数据

Catalog Manager 包含以下部分：

- Catalog Manager 装入程序软件包

此软件包主要由用于准备和装入数据到 WebSphere Commerce 数据库的命令实用程序构成。您可以使用装入程序软件包装入大量数据和更新 WebSphere Commerce 数据库中的数据。

装入程序软件包还允许您执行以下操作：

- 从数据库抽取数据作为 XML 文档
- 将 XML 数据转换为备用 XML 格式
- 将数据在字符限定变量格式与 XML 数据格式之间转换

- Catalog Manager 管理工具

Catalog Manager 还包含以下两个带有用户界面的工具以帮助管理其功能：

- 文本转换工具
- XSL 编辑器

- Catalog Manager Web Editor

Web Editor 使您能通过 Web 浏览器创建、删除和更改产品目录数据。

第 2 部分 转换、装入和抽取数据

第 1 章 简介

Catalog Manager 装入程序软件包主要由用于准备和装入数据到 WebSphere Commerce 数据库的命令实用程序构成。您可以使用装入程序软件包装入大量数据和更新 WebSphere Commerce 数据库中的数据。装入程序软件包允许您执行以下操作:

- 从数据库抽取数据作为 XML 文档
- 将 XML 数据转换为备用 XML 格式
- 将数据在字符限定变量格式与 XML 数据格式之间转换

Catalog Manager 还包含以下两个带有用户界面的工具以帮助管理其功能:

- 文本转换工具
- XSL 编辑器

实用程序

WebSphere Commerce 装入程序软件包包含用于准备和装入数据到 WebSphere Commerce 数据库的实用程序。您可以使用这些实用程序装入大量和少量数据以及更新 WebSphere Commerce 数据库中的数据。

装入过程由将数据移动到 WebSphere Commerce 数据库的必要步骤构成:

1. 使用 DTD 生成程序生成 DTD
2. 使用标识解析器解析输入文件中的标识
3. 使用装入程序装入数据

装入程序软件包还包含用于从数据库抽取数据作为 XML 文档和将 XML 数据转换为备用 XML 格式的实用程序。

装入程序软件包由以下实用程序构成:

- **文本转换程序**

文本转换程序将数据在字符限定变量格式与 XML 数据格式之间转换。

请参阅第 9 页的第 2 章,『转换文本』以获得更多信息。

- **XML 转换程序**

XML 转换程序更改、聚集和重新映射 XML 文档中的数据为备用 XML 格式以供其他用户或系统需要时使用。

请参阅第 17 页的第 3 章,『转换 XML 数据』以获得更多信息。

- **DTD 生成程序**

DTD 指定可在 XML 数据文档中使用的结构元素和标记定义。例如, DTD 可以列出将在文档中使用的元素并指定每个元素可以接受的属性。

DTD 生成程序是一个实用程序, 它可以根据数据库模式为装入程序创建要使用的 DTD。 DTD 描述了装入程序在其中导入数据的表和列。 DTD 生成程序还可以创建可与 WebSphere Catalog Manager Web Editor 一起使用的模式和详细的 XML 文档。

DTD 生成程序根据您数据必须符合的目标数据库生成 DTD。此 DTD 将在整个装入过程中使用。 DTD 生成程序只需运行一次。

请参阅第 23 页的第 4 章,『生成 DTD 和模式』以获得更多信息。

- **标识解析器**

标识解析器是一个实用程序,它为 XML 元素生成它们需要的标识。如果 XML 内容已经提供标识,则不必运行标识解析器。

标识解析器使用相关标识更新一系列 XML 元素。这一步非常重要,因为装入程序软件包 XML 文件直接映射到目标数据库模式。因此,XML 元素就必须具有标识。

标识解析器包含一个发生错误时生成异常文档的错误报告程序。

请参阅第 27 页的第 5 章,『解析标识』以获得更多信息。

- **装入程序**

装入程序将有效和格式良好的 XML 用作输入将数据装入到数据库中。XML 文档元素映射为数据库中的表名,元素属性映射为列。装入程序是将数据装入系统的最普通的方法。

此装入程序允许对表进行列级别的更新。它也允许您从数据库中删除数据。

以下示例显示了从格式良好和有效的 XML 输入到装入程序的摘录:

```
<ADDRBOOK
  ADDRBOOK_ID="11801"
  MEMBER_ID="100"
  DISPLAYNAME="Friends"
  DESCRIPTION="All my friends"
  TYPE="P"
/>
```

在以上示例中,ADDRBOOK 是表名,要更新的列由 ADDRBOOK 元素的属性表示。

装入实用程序包含以下功能部件:

- **错误报告程序**

装入程序包含一个发生错误时生成异常文档的错误报告程序。

- **产品顾问搜索空间同步**

产品顾问搜索空间是通过从 WebSphere Commerce 产品目录抽取信息并以适合搜索的格式表示信息创建的。产品目录更新后,搜索空间和产品目录变为不同步。要避免搜索空间和产品目录同步的延迟,您可以启用装入程序的产品顾问搜索空间同步功能。

如果启用了产品顾问空间搜索同步,通过将命令信息添加到调度程序作业表 SCHCONFIG 和调度程序作业实例表 SCHSTATUS 中(如下表所示),装入程序将调度适当的搜索空间命令。

表	方式	搜索空间命令	条件
CATENTRY	更新	UpdateSearchSpaces	更新发布标志
	删除	RemoveProductFromAllSearchSpaces	删除产品
CATENTDESC	添加	UpdateSearchSpaces	添加描述 / 语言
	更新	UpdateSearchSpaces	更新发布标志 更新描述数据
	删除	UpdateSearchSpaces	除去描述 / 语言
LISTPRICE	添加	UpdateSearchSpaces	添加标价
	更新	UpdateSearchSpaces	更新标价
	删除	UpdateSearchSpaces	删除标价
ATTRVALUE	添加	UpdateSearchSpaces	添加用户定义的价格
	更新	UpdateSearchSpaces	更新用户定义的价格
	删除	UpdateSearchSpaces	删除用户定义的价格
CATENTATTR	添加	UpdateSearchSpaces	添加属性
	更新	UpdateSearchSpaces	更新属性
	删除	UpdateSearchSpaces	删除属性
CATGPENREL	添加	AddProductsToSearchSpace	添加产品到类别
	删除	RemoveProductsFromSearchSpace	从类别除去产品

请参阅第 41 页的第 6 章, 『装入数据』以获得更多信息。

- **抽取程序**

抽取程序对数据库使用查询, 将选定的数据子集从数据库抽取到 XML 文档。将从数据库中抽取的数据通过使用抽取过滤器 XML 文档指定。

抽取程序的功能与装入程序相反。使用抽取程序可以从 WebSphere Commerce 数据库中以 XML 文件的格式抽取选择的数据子集。例如, 可以抽取与即将来临的节日相关的产品数据, 或者从统一数据库中抽取信息以用于其它系统。

请参阅第 59 页的第 7 章, 『抽取数据』以获得更多信息。

装入程序软件包也具有记录程序功能部件。装入程序软件包中的每个实用程序都创建消息, 以提示成功、失败和错误, 并提供程序跟踪信息。

管理工具

Catalog Manager 包含以下工具以帮助管理其功能:

文本转换工具

文本转换工具帮助管理员准备信息，这些信息是在处理将数据在字符限定的变量格式与 XML 数据格式的转换时所需要的。

XSL 编辑器

XML 转换程序使用可扩展样式表语言 (XSL) 定义将 XML 文件转换为其它 XML 文件的规则。XSL 编辑器中的映射功能为管理员提供了一个可视界面，管理员用它可以创建从源文档类型定义 (DTD) 中的元素到目标 DTD 中的元素的关联。

第 2 章 转换文本






Catalog Manager 为您提供了将来自其它工具（例如电子表格程序）的 ASCII 文件输出转换为 XML 数据格式（这种格式可以被输入到 WebSphere Commerce 数据库）的能力。


文本转换工具帮助准备信息，这些信息是在处理将数据在字符限定的变量格式与 XML 数据格式的转换时所需要的。提供了以下视图：

1. “文本模式编辑视图” 允许您创建和修改要在转换时使用的 XML 模式文件。
2. “转换命令编辑视图” 允许您创建和修改用于运行转换过程的实际命令。
3. “转换命令处理视图” 允许您启动转换过程。

启动文本转换工具

要启动文本转换工具，请使用 WebSphere Commerce 目录中提供的适当的脚本或命令：

-  `drive:\WebSphere\CommerceServer\bin\TextTrans.cmd`
-  `drive:\Program Files\WebSphere\CommerceServer\bin\TextTrans.cmd`
-  `/usr/WebSphere/CommerceServer/bin/textttrans.sh`
-   `/opt/WebSphere/CommerceServer/bin/textttrans.sh`

 在 iSeries 环境中，管理员应当首先将运行文本转换工具必需的文件复制到他们的 Windows NT 或 Windows 2000 机器上。以下是关于如何执行此操作的示例：

1. 在 Windows 机器上创建一个新的目录（例如 `drive:\TextTrans`）。
2. 在 `drive:\XMLTrans` 下创建以下新的子目录：

```
\bin
\lib\loader
\wcsadmin
```

3. 根据以下图表将文件从 iSeries 机器复制到这些目录：

从		至
/QIBM/ProdData/WebCommerce		drive:\TextTrans
/bin		\bin
/TextTrans.cmd	----->	\TextTrans.cmd
/lib/loader		\lib\loader
/wcmxmlp.jar	----->	\wcmxmlp.jar
/wcmxslt.jar	----->	\wcmxslt.jar
/wcsadmin		\wcsadmin
/acsxml.ico	----->	\acsxml.ico
/swing.jar	----->	\swing.jar
/TextTransformerUI.zip	----->	\TextTransformerUI.zip
/TextTransform.cnt	----->	\TextTransform.cnt
/TextTransform.hlp	----->	\TextTransform.hlp
/TextTransform.reg	----->	\TextTransform.reg

4. 通过在以下文本：

```
set lib=%WCS_HOME%\lib\loader
```

之前插入以下文本：

```
set WCS_HOME=drive :\TextTrans
```

修改 Windows 机器上的 `TextTrans.cmd` 文件。

5. 通过运行 Windows 机器上的 `TextTrans.cmd` 文件启动文本转换工具。

使用文本模式编辑视图

以下过程与创建和修改 XML 模式文件以用于在字符限定变量格式与 XML 格式之间转换数据有关。

创建模式文件

要创建新的模式文件，请执行以下操作：

1. 启动文本转换工具。
2. 选择**文件 > 新建**，或单击工具栏上的“新建”图标。
3. 选择路径并输入要创建的 XML 模式文件名。

注：缺省文件名是“Default.xml”。

4. 选择 **CSV 格式**。

注：不要选择 **WebSphere Commerce Suite 格式**。此选项只在先前的 WebSphere Commerce 版本中使用。

5. 当完成后单击**确定**。

现在您可以通过以下指示的步骤创建新的元素和属性。

打开模式文件

要打开模式文件，请执行以下操作：

1. 启动文本转换工具。
2. 选择**文件 > 打开**，或单击工具栏上的“打开”图标。
3. 选择要打开的模式文件。
4. 当完成后单击**确定**。

现在您可以通过以下描述的步骤修改元素和属性。

保存模式文件

要保存模式文件，请执行以下操作：

1. 启动文本转换工具。
2. 要保存您对模式所做的任何更改，请选择**文件 > 保存**或单击工具栏上的“保存”图标。
3. 要用新的名称保存模式副本，请执行以下操作：
 - a. 选择**文件 > 另存为**。
 - b. 选择路径并输入要创建的 XML 文件名。
 - c. 当完成后单击**确定**。

编辑模式文件

要编辑模式文件，请如以上所述打开模式文件并遵循此过程。

添加元素

要添加元素，请执行以下操作：

1. 在“元素列表”字段输入新元素名称。

2. 单击“添加行”图标。

除去元素

要除去元素，请执行以下操作：

1. 选择元素名称。
2. 单击“除去行”图标。

替换元素

要替换元素，请执行以下操作：

1. 在“元素列表”字段输入新元素名称。
2. 选择要替换的元素名称。
3. 单击“替换行”图标。

将元素上移一行

要将元素上移一行，请执行以下操作：

1. 选择要上移的元素名称。
2. 单击“上移行”图标。

将元素下移一行

要将元素下移一行，请执行以下操作：

1. 选择要下移的元素名称。
2. 单击“下移行”图标。

添加属性

要添加属性，请执行以下操作：

1. 在“元素列表”字段输入新属性名称。
2. 单击“添加行”图标。

除去属性

要除去属性，请执行以下操作：

1. 选择属性名称。
2. 单击“除去行”图标。

替换属性

要替换属性，请执行以下操作：

1. 在“元素列表”字段输入新属性名称。
2. 选择要替换的属性名称。
3. 单击“替换行”图标。

将属性上移一行

要将属性上移一行，请执行以下操作：

1. 选择要上移的属性名称。
2. 单击“上移行”图标。

将属性下移一行

要将属性下移一行，请执行以下操作：

1. 选择要下移的属性名称。
2. 单击“下移行”图标。

更改模式文件结构

“文本模式编辑视图”的下部窗格中的“文件结构”视图描述字符限定变量文件的布局。以下字段是预期的文件结构中必需的：

字段分隔符:

指定分隔属性值的定界符。缺省值是逗号（“,”）。

记录分隔符:

指定分隔数据记录的定界符。缺省值是“
”（这等同于对 `\r\n` 的实体引用。）

字符串定界符:

指定表示字符串开始和结束边界的定界符。缺省值是单引号（“'”）。

内含页眉:

布尔值，如果在文本数据文件中有页眉行则指定为“true”，如果在文本数据文件中没有页眉行则指定为“false”。如果有页眉行，则页眉行必须符合标记名称的 XML 规则，因为在此情况下页眉将被用作 XML 标记名称。缺省值是“false”。

页眉行数:

指定存在于文本数据文件中的页眉行的行数。缺省值是零（“0”）。

准备模式文件以将数据从 XML 转换为字符限定变量格式

如果正在使用“文本模式编辑视图”创建的 XML 模式文件将数据从 XML 格式转换为字符限定变量格式，则必须在处理转换之前使用文本编辑器将模式文件中的数据类型指定从“CSV 格式”更改为“XML 格式”。

使用转换命令编辑视图

使用“转换命令编辑视图”，可以创建新的命令文件、打开现有的命令文件或保存对命令文件的更改。缺省命令文件名是“Manifest.txt”。

您可以创建新命令、除去命令、用编辑的信息替换命令或更改命令顺序。

注：只要命令表被更新，命令文件就会自动保存。

创建命令

要创建命令，请执行以下操作：

1. 指定源文件，源文件可以是字符分隔变量格式的文件（带 .csv 扩展名）或 XML 格式的文件（带 .xml 扩展名）。
2. 指定要在转换时使用的 XML 模式文件。
3. 指定要在转换过程中创建或修改的输出文件的名称（即存储新数据的地方），文件名可以是 XML 格式的文件（带 .xml 扩展名）或字符分隔变量格式的文件（带 .csv 扩展名）。
4. 指定命令方式。
如果要创建输出文件，请选择**创建**，或者，如果要将输出数据附加到现有的数据文件，请选择**附加**。
5. 单击“添加行”图标。

除去命令

要除去命令，请执行以下操作：

1. 选择命令。
2. 单击“除去行”图标。

编辑或替换命令

要编辑或替换命令，请执行以下操作：

1. 单击“编辑命令”图标。
行数据被填写入适当的输入字段。
2. 更改适当的输入字段中的文本。
3. 单击“替换行”图标以更新行。

将命令上移一行

要将命令上移一行，请执行以下操作：

1. 选择命令。
2. 单击“上移行”图标。

注：这将更改转换处理顺序。

将命令下移一行

要将命令下移一行，请执行以下操作：

1. 选择命令。
2. 单击“下移行”图标。

注： 这将更改转换处理顺序。

清除命令

要清除命令，请执行以下操作：

1. 选择命令。
2. 单击“清除输入字段”图标。

这将清除源文件、模式文件和输出文件字段中的文本。

使用转换处理视图

要启动文本转换处理，请执行以下操作：

1. 在“文件”字段中输入或浏览至参数文件的名称。
2. 单击**处理**。

“处理”按钮下面的输出区域显示指示转换处理状态的信息。您可以单击文本区域底部的“保存”按钮保存输出信息，或者单击“清除”按钮清除所有状态信息。

第 3 章 转换 XML 数据

可扩展样式表语言 (XSL) 提供以下语言:

1. 用于指定 XML 文档格式化的语言
2. 描述如何将 XML 文件转换成其它结构规则的文件的语言

XSL 转换能力可用于将 XML 文件转换成另一个符合不同 XML 模式或 DTD 的 XML 文件。

要将 XML 文件转换成备用的 XML 格式, 必须使用转换 XSL 规则文件指定转换的规则。

以下示例使用 MemberSubsystem.xsl 作为转换 XSL 规则文件, 而日语作为本地语言, 转换 MemberSubsystemExtracted.xml 中的数据。



```
java com.ibm.wca.XMLTransformer.XMLTransformer -infile MemberSubsystemExtracted.xml  
-transform MemberSubsystem.xsl -outfile TransMbrStr.xml -param 'language="-10''
```









```
QWEECOMM/TRNWCSXML INFILE(MemberSubsystemExtracted.xml)  
TRANSFORM(MemberSubsystem.xsl) INSTRROOT(/QIBM/UserData/WebCommerce/instances/my_inst)  
OUTFILE(TransMbrStr.xml) PARAM('language=-10')
```

XML 转换程序使用 XSL 定义将 XML 文件转换成另一个 XML 文件的规则。XSL 编辑器中的映射功能提供了一个可视界面, 您可以用它建立从源 DTD 中元素到目标 DTD 中元素的关联。如果提供两个 DTD, 您就可以开发 XSL 规则, 这种规则确定如何将符合第一个 (源) DTD 的 XML 文件转换成符合第二个 (目标) DTD 的文件。

启动 XSL 编辑器

要启动 XSL 编辑器, 请使用 WebSphere Commerce 目录中提供的适当的脚本或命令:

-  `drive:\WebSphere\CommerceServer\bin\XSLEditor.cmd`
-  `drive:\Program Files\WebSphere\CommerceServer\bin\ XSLEditor.cmd`
-  `/usr/WebSphere/CommerceServer/bin/xsleditor.sh`
-   `/opt/WebSphere/CommerceServer/bin/xsleditor.sh`

 在 iSeries 环境中, 管理员应当首先将运行 XSL 编辑器必需的文件复制到他们的 Windows NT 或 Windows 2000 机器上。以下是关于如何执行此操作的示例:

1. 在 Windows 机器上创建一个新的目录 (例如 `drive:\XMLTrans`)。
2. 在 `drive:\XMLTrans` 下创建以下新的子目录:

```
\bin
\lib\loader
\wcsadmin
```

3. 根据以下图表将文件从 iSeries 机器复制到这些目录:

从		至
/QIBM/ProdData/WebCommerce		drive:\XMLTrans
/bin		\bin
/XSLEditor.cmd	----->	\XSLEditor.cmd
/lib/loader		\lib\loader
/wcmxmlp.jar	----->	\wcmxmlp.jar
/wcmxslt.jar	----->	\wcmxslt.jar
/wcsadmin		\wcsadmin
/config.dtd	----->	\acsxml.ico
/swing.jar	----->	\swing.jar
/XML_transform.ico	----->	\XML_transform.ico
/XMLTransformerUI.zip	----->	\XMLTransformerUI.zip
/XMLTransformWP.xml	----->	\XMLTransformWP.xml
/XMLTransformWP.dtd	----->	\XMLTransformWP.dtd
/XMLRuleConfig.xml	----->	\XMLRuleConfig.xml
/XMLTransform.cnt	----->	\XMLTransform.cnt
/XMLTransform.hlp	----->	\XMLTransform.hlp
/XMLTransform.reg	----->	\XMLTransform.reg

4. 通过在以下文本:

```
set lib=%WCS_HOME%\lib\loader
```

之前插入以下文本:

```
set WCS_HOME=drive:\XMLTrans
```

修改 Windows 机器上的 XSLEditor.cmd 文件。

5. 通过运行 Windows 机器上的 XSLEditor.cmd 文件启动 XML 转换工具。

处理映射规则构建区域

当启动 XSL 编辑器时，它将显示“映射规则构建区域”窗口。使用此窗口管理映射规则构建区域。

创建映射规则构建区域

要创建映射规则构建区域，请执行以下操作：

1. 启动 XML 编辑器。
2. 从下拉菜单中选择 [新建]。
3. 在“名称”字段中输入映射规则构建区域的名称。
4. 在“描述”字段中输入新的映射规则构建区域的简短描述。
5. 在“源模式”字段中输入现有文件的名称或浏览到将用作源模式的现有文件。
6. 在“目标模式”字段中输入现有文件的名称或浏览到将用作目标模式的现有文件。
7. 在“XSL 规则文件”字段中输入将要创建的新规则文件的名称。

您可以在此指定完整路径。如果您不提供路径，将在当前工作目录中创建文件。

8. 单击打开创建并打开新的映射规则构建区域。

修改映射规则构建区域

要修改映射规则构建区域，请执行以下操作：

1. 启动 XML 编辑器。
2. 从下拉菜单中选择您希望修改的映射规则构建区域。
3. 单击打开打开映射规则构建区域。
4. 更新您希望更改的字段。
5. 单击保存保存更改。

删除映射规则构建区域

要删除映射规则构建区域，请执行以下操作：

1. 启动 XML 编辑器。
2. 从下拉菜单中选择您希望删除的映射规则构建区域。
3. 单击“除去”按钮除去此条目。

注： 除去映射规则构建区域不会从磁盘上删除物理文件。

使用 XSL 编辑器

当使用 XSL 编辑器打开映射规则构建区域时，映射规则构建区域显示在“XSL 编辑器”主窗口中。

在“XSL 编辑器”主窗口中，左窗格显示标号为“源模式”的源 DTD 的层次结构视图。右窗格显示标号为“目标模式”的目标 DTD 的层次结构视图。

创建映射规则

元素至元素映射

从源层次结构中选择并拖动元素，然后在目标层次结构中的元素上放下。XSL 规则生成并显示在位于窗口底部的“映射规则”视图中。

这里是生成的 XSL 规则的示例：

```
<xsl:template match="merchant">
  <xsl:element name="MERCHANT">
    </xsl:element>
  </xsl:template>
```

注：所有必需的但不存在的上级关系都将自动生成。

属性至属性映射

从源层次结构中选择并拖动属性，然后在目标层次结构中的属性上放下。XSL 规则生成并显示在位于窗口底部的“映射规则”视图中。

这里是生成的 XSL 规则的示例：

```
<xsl:attribute name="MEADDR1">
  <xsl:apply-templates select="@mecmail1"/>
</xsl:attribute>
```

注：所有必需的但不存在的上级关系都将自动生成。

创建定制映射表达式

要创建定制映射表达式，请先从目标层次结构中选择元素或属性。然后用鼠标右键单击并选择“创建定制表达式”菜单。将显示“创建定制表达式”窗口，其中在两个下拉菜单中有可用的“模板”和“规则表达式”列表。执行以下操作完成定制表达式：

1. 选择要添加定制表达式的模板。
2. 选择要创建的规则表达式（例如**常量表达式**）。
3. 在“值”列中为表中所列的每个参数输入一个值并按 **Enter** 键提交输入的值。
4. 单击**确定**完成创建步骤；或者单击**取消**以取消创建规则。

生成的 XSL 规则基于规则配置文件 (XSLRuleConfig.xml) 中定义的定制表达式。如果需要, 您可以修改规则配置文件并添加新规则。要使规则在“规则表达式”列表中是可用的, 请将该规则的“可见性”属性设置为“true”。

删除映射规则

要删除映射规则, 请执行以下操作:

1. 从“映射规则”表中选择规则。
2. 用鼠标右键单击并选择**删除**。
将删除规则及其所有派生的规则。

注: 已更新的映射规则和已生成的 XSL 规则会自动延续。

处理 XML 转换

要处理 XML 转换, 请执行以下操作:

1. 选择**工具 > 转换**调出“处理转换”窗口。
2. 完成必需字段:
 - a. 在“输入 XML 文件”字段中输入或浏览到源 XML 数据文件的路径和名称。
 - b. 在“XSL 规则文件”字段中输入或浏览到将用于转换的映射规则文件的路径和名称。
如果映射规则构建区域是打开的, 将用映射规则构建区域中当前打开的映射规则文件路径预填此字段。
 - c. 在“输出 XML 文件”字段中输入或浏览到在转换处理过程中将创建的新的 XML 数据文件的路径和名称。
3. 单击**开始**开始 XML 转换处理; 或者单击**关闭**退出此窗口而不处理转换。

定制映射规则表

要定制映射规则表, 请执行以下操作:

1. 要隐藏表中的列, 请用鼠标右键单击表中的单元并选择**隐藏列**。
2. 要显示表中隐藏的列, 请执行以下操作:
 - a. 用鼠标右键单击表中的单元。
 - b. 选择**显示列**调出隐藏列的列表。
 - c. 从列表中选择列。
注: 要选择多个列, 请按住 **Shift** 键然后单击列名。
 - d. 单击**确定**显示选定的列或单击**取消**取消操作。
3. 要显示表中所有隐藏的列, 请用鼠标右键单击表中的某个单元并选择**显示所有列**。
所有列按缺省顺序显示。

显示完整的 XSL 规则 / 值表达式

从“值表达式”或“XSL 规则”列中单击一个单元将调出一个窗口, 其中有选定行的完整的规则内容。

第 4 章 生成 DTD 和模式

DTD 生成程序可创建与装入程序软件包一起使用的 DTD 和模式。DTD 生成程序使用包含数据库表名的输入文件，并仅生成 DTD 或者同时生成 DTD 和模式及描述了数据库的详细 XML 文件，这取决于如何调用“DTD 生成”命令。

设置 DTD 生成程序

装入程序 DTD 直接映射为 WebSphere Commerce 数据库模式。每个表是一个元素，每个列是一个属性。

示例：将装入程序 DTD 映射为数据库模式

CATENTRY 表的 DDL 语句	DTD
<pre>CREATE TABLE "CATENTRY" ("CATENTRY_ID" BIGINT NOT NULL "MEMBER_ID" BIGINT NOT NULL "CATENTTYPE_ID" CHAR(16) NOT NULL "MARKFORDELETE" INTEGER NOT NULL "PARTNUMBER" VARCHAR(64) NOT NULL "MFAPARTNUMBER" VARCHAR(64) "MFNAME" VARCHAR(64) "URL" VARCHAR(254) "FIELD1" INTEGER "FIELD2" INTEGER "FIELD3" DECIMAL(20,5) "FIELD4" VARCHAR(254) "FIELD5" VARCHAR(254) "LASTUPDATE" TIMESTAMP "OID" VARCHAR(64) "ONSPECIAL" INTEGER "ONAUCTION" INTEGER "BUYABLE" INTEGER "BASEITEM_ID" INTEGER "CLASSIFGRP_ID" INTEGER "ITEMSPC_ID" INTEGER "STATE" INTEGER);</pre>	<pre><!ELEMENT CATENTRY EMPTY> <!ATTLIST CATENTRY CATENTRY_ID CDATA #REQUIRED MEMBER_ID CDATA #REQUIRED CATENTTYPE_ID CDATA #REQUIRED MARKFORDELETE CDATA #REQUIRED PARTNUMBER CDATA #REQUIRED MFAPARTNUMBER CDATA #IMPLIED MFNAME CDATA #IMPLIED URL CDATA #IMPLIED FIELD1 CDATA #IMPLIED FIELD2 CDATA #IMPLIED FIELD3 CDATA #IMPLIED FIELD4 CDATA #IMPLIED FIELD5 CDATA #IMPLIED LASTUPDATE CDATA #IMPLIED OID CDATA #IMPLIED ONSPECIAL CDATA #IMPLIED ONAUCTION CDATA #IMPLIED BUYABLE CDATA #IMPLIED BASEITEM_ID CDATA #IMPLIED CLASSIFGRP_ID CDATA #IMPLIED ITEMSPC_ID CDATA #IMPLIED STATE CDATA "1"</pre>

可以通过执行以下操作，设置 DTD 生成程序的工作方式：

1. 创建新的 DTD 生成程序定制程序属性文件。



DB2ConnectionCustomizer.properties 位于 DTDGenerator.zip 压缩文档中。将此文件解压缩并重命名，但保留 .properties 扩展名，并将它放置在类路径中的目录中。**重要信息：**请不要除去或修改现有的 DB2ConnectionCustomizer.properties 文件。

• 

`ISeries_GENWCSDTD_Customizer.properties` 位于 `/QIBM/ProdData/WebCommerce/properties` 目录中。将此文件复制到 `/instroot/xml` 目录，重命名新文件但保留 `.properties` 扩展名，然后对新文件进行任何必要的更改。**重要信息：** 请不要除去或修改原始的 `ISeries_GENWCSDTD_Customizer.properties` 文件。

2. 修改新文件中的数据库驱动程序值。例如：

```
DBVendorName = DB2
DBDriverName = COM.ibm.db2.jdbc.app.DB2Driver
DBURL = jdbc:db2:
```

其中：

• `DBVendorName` 用于选择数据库的类型。

选项如下：

- DB2 通用数据库 iSeries 版 (DB2/iSeries)
- 用于其它操作系统的 DB2 (DB2)
- Oracle 数据库 (Oracle)

• `DBDriverName` 用于选择 JDBC 驱动程序。

选项如下：

- DB2 通用数据库 iSeries 版 (`com.ibm.db2.jdbc.app.DB2Driver`)
- 用于其它操作系统的 DB2 (`COM.ibm.db2.jdbc.app.DB2Driver`)
- Oracle 数据库 (`oracle.jdbc.driver.OracleDriver`)

• `DBURL` 用于指定访问数据库的 URL。

选项如下：

- DB2 通用数据库 iSeries 版 (`jdbc:db2://`)
- 用于其它操作系统的 DB2 (`jdbc:db2:`)
- Oracle 数据库 (`jdbc:oracle:oci8:@`)

3. 将新文件名指定为“DTD 生成”命令的 `customizer` 参数的值。

生成 DTD

`TableNames.txt` 输入文件包含以下数据库表名，每行一个：

```
MEMBER
ADDRBOOK
ADDRESS
```

这里是如何能够调用 DTD 生成程序的示例：

-     

```
java com.ibm.wca.DTDGenerator.GenerateDTD -dbname MALL -dbuser db2inst1
-dbpwd db2ibm -outfile wc.dtd -infile TableNames.txt
```

- 

```
QWEBCOMM/GENWCSDTD DATABASE(DATABASE_NAME) SCHEMA(MALL)
INSTROOT(/QIBM/UserData/WebCommerce/instances/mser)
PASSWD(mypassword) OUTFILE(wc.dtd) INFILE(TableNames.txt)
```


输出文件 wc.dtd 包含以下内容:

```
<!ELEMENT MALL (( MEMBER | ADDRBOOK | ADDRESS)*)>
<!ELEMENT MEMBER EMPTY>
<!ATTLIST MEMBER
  MEMBER_ID      CDATA      #REQUIRED
  TYPE           CDATA      #REQUIRED
  STATE          CDATA      #IMPLIED
>
<!ELEMENT ADDRBOOK EMPTY>
<!ATTLIST ADDRBOOK
  ADDRBOOK_ID    CDATA      #REQUIRED
  MEMBER_ID      CDATA      #REQUIRED
  TYPE           CDATA      #IMPLIED
  DISPLAYNAME    CDATA      #REQUIRED
  DESCRIPTION    CDATA      #IMPLIED
>
<!ELEMENT ADDRESS EMPTY>
<!ATTLIST ADDRESS
  ADDRESS_ID      CDATA      #REQUIRED
  ADDRESSTYPE    CDATA      #IMPLIED
  MEMBER_ID      CDATA      #REQUIRED
  ADDRBOOK_ID    CDATA      #REQUIRED
  ORGUNITNAME    CDATA      #IMPLIED
  FIELD3         CDATA      #IMPLIED
  BILLINGCODE    CDATA      #IMPLIED
  BILLINGCODETYPE CDATA      #IMPLIED
  STATUS         CDATA      #IMPLIED
  ORGNAME        CDATA      #IMPLIED
  ISPRIMARY      CDATA      #IMPLIED
  LASTNAME       CDATA      #IMPLIED
  PERSONTITLE    CDATA      #IMPLIED
  FIRSTNAME      CDATA      #IMPLIED
  MIDDLENAME     CDATA      #IMPLIED
  BUSINESSTITLE  CDATA      #IMPLIED
  PHONE1         CDATA      #IMPLIED
  FAX1           CDATA      #IMPLIED
  PHONE2         CDATA      #IMPLIED
  ADDRESS1       CDATA      #IMPLIED
  FAX2           CDATA      #IMPLIED
  NICKNAME       CDATA      #REQUIRED
  ADDRESS2       CDATA      #IMPLIED
  ADDRESS3       CDATA      #IMPLIED
  CITY           CDATA      #IMPLIED
  STATE          CDATA      #IMPLIED
  COUNTRY        CDATA      #IMPLIED
  ZIPCODE        CDATA      #IMPLIED
  EMAIL1         CDATA      #IMPLIED
  EMAIL2         CDATA      #IMPLIED
  PHONE1TYPE     CDATA      #IMPLIED
  PHONE2TYPE     CDATA      #IMPLIED
  PUBLISHPHONE1  CDATA      #IMPLIED
  PUBLISHPHONE2  CDATA      #IMPLIED
  BESTCALLINGTIME CDATA      #IMPLIED
  PACKAGESUPPRESSION CDATA      #IMPLIED
  LASTCREATE     CDATA      #IMPLIED
  OFFICEADDRESS  CDATA      #IMPLIED
  SELFADDRESS    CDATA      "0"
  FIELD1         CDATA      #IMPLIED
  FIELD2         CDATA      #IMPLIED
  TAXGEOCODE     CDATA      #IMPLIED
  SHIPPINGGEOCODE CDATA      #IMPLIED
>
```

生成模式和详细的 XML 文件

在此示例中，如下调用了 DTD 生成程序：

• 

```
java com.ibm.wca.DTDGenerator.GenerateDTD -dbname SAMPLE
-dbuser johndoe -dbpwd password -xmlTableDesc c:\sample\sample.xml
-outfile tables.dtd -tablenames "employee,staff"
```

• 

```
java com.ibm.wca.DTDGenerator.GenerateDTD -dbname SAMPLE
-dbuser johndoe -dbpwd password -xmlTableDesc usr/sample/sample.xml
-outfile tables.dtd -tablenames "employee,staff"
```

• 

```
QWEBCOMM/GENWCSDTD DATABASE(MYDB) SCHEMA(SAMPLE)
INSTROOT(/QIBM/UserData/WebCommerce/instances/mser) PASSWD(mypassword)
OUTFILE(tables.dtd) TABNAMES('employee,staff') XMLTABDESC(/sample/sample.xml)
```

模式文件在 sample 目录中创建并具有文件名 WCAWebForm.xsd。sample.xml 输出文件包含以下内容：

```
<?xml version="1.0" encoding="UTF-8"?>
<formList xmlns="WCAWebForm.xsd" dbname="SAMPLE" dtdname="tables.dtd">
  <form name="EMPLOYEE">
    <uniqueIndex name="U2" columns="FIRSTNAME, LASTNAME"/>
    <uniqueIndex name="U3" columns="MIDINIT, LASTNAME"/>
    <field name="EMPNO" type="string" maxLength="6" minOccurs='1'
      uniqueKey="true" showColumnInList="true" />
    <field name="FIRSTNAME" type="string" maxLength="32" minOccurs='1'
      showColumnInList="true" />
    <field name="MIDINIT" type="string" maxLength="1" minOccurs='1'
      showColumnInList="true" />
    <field name="LASTNAME" type="string" maxLength="15" minOccurs='1'
      showColumnInList="true" />
    <field name="WORKDEPT" type="string" maxLength="3" showColumnInList="true" />
    <field name="PHONENO" type="string" maxLength="4"/>
    <field name="HIREDATE" type="date" maxLength="10"/>
    <field name="JOB" type="string" maxLength="8"/>
    <field name="EDLEVEL" type="integer" maxLength="5" minOccurs='1' />
    <field name="SEX" type="string" maxLength="1"/>
    <field name="BIRTHDATE" type="date" maxLength="10"/>
    <field name="SALARY" type="decimal" maxLength="9"/>
    <field name="BONUS" type="decimal" maxLength="9"/>
    <field name="COMM" type="decimal" maxLength="9"/>
  </form>
  <form name="STAFF">
    <field name="ID" type="integer" maxLength="5" minOccurs='1'
      uniqueKey="true" showColumnInList="true" />
    <field name="NAME" type="string" maxLength="9" showColumnInList="true" />
    <field name="DEPT" type="integer" maxLength="5" showColumnInList="true" />
    <field name="JOB" type="string" maxLength="5" showColumnInList="true" />
    <field name="YEARS" type="integer" maxLength="5" showColumnInList="true" />
    <field name="SALARY" type="decimal" maxLength="7"/>
    <field name="COMM" type="decimal" maxLength="7"/>
  </form>
</formList>
```

第 5 章 解析标识

要装入到目标数据库中的 XML 数据必须为需要标识的 XML 元素包含标识。要生成或定位 XML 文档中产品目录实体的标识，请调用“标识解析”命令。

标识解析器只为主表解析标识。主表是在 KEYS 或 SUBKEYS 表中列出的表。如果需要为不在 KEYS 或 SUBKEYS 中的表解析标识，请在运行标识解析器之前，将该表添加到 SUBKEYS 表中。

以下是一些您可能希望要使用标识解析器的情况的示例：

- 当需要数据的标识时以 XML 格式装入新内容
- 当数据库中存在对象的标识时更新内容

标识解析器可以提供实际的标识，或使用以下技术解析标识：

- 内部别名解析
- 属性文件规范
- 唯一索引解析

设置标识解析器

可通过执行以下步骤，设置标识解析器如何处理时间戳记、存储和数据库驱动程序：

1. 创建新的标识解析器定制程序属性文件。



• DB2ConnectionCustomizer.properties 位于 IdResGen.zip 压缩文档中。将此文件解压缩并重命名，但保留 .properties 扩展名，并将它放置在类路径中的目录中。**重要信息：** 请不要除去或修改现有的 DB2ConnectionCustomizer.properties 文件。



• ISeries_RESWCSID_Customizer.properties 位于 /QIBM/ProdData/WebCommerce/properties 目录中。将此文件复制到 /instroot/xml 目录，重命名新文件但保留 .properties 扩展名，然后对新文件进行任何必要的更改。**重要信息：** 请不要除去或修改原始的 ISeries_RESWCSID_Customizer.properties 文件。

2. 修改新文件中指定的属性值。
3. 将新文件名指定为“标识解析器”命令的 customizer 参数的值。

设置标识解析器如何处理时间戳记

以下缺省输入时间戳记掩码在标识解析器定制程序属性文件中提供:

```
InputTimeStampFormat.1 = yyyy-DD hh:mm:ss.SSSSSS
InputTimeStampFormat.2 = yyyy-MM-dd hh:mm:ss.SSSSSS
InputTimeStampFormat.3 = yyyy-DD-hh.mm.ss.SSSSSS
InputTimeStampFormat.4 = yyyy-MM-dd-HH.mm.ss.SSSSSS
InputTimeStampFormat.5 = yyyy-MM-dd-hh.mm.ss.SSSSSS
InputTimeStampFormat.6 = yyyy-MM-dd HH:mm:ss.SSSSSS
InputTimeStampFormat.7 = yyyy-DD HH:mm:ss.SSSSSS
```

您可以修改这些时间戳记掩码, 或根据自己的希望在标识解析器定制程序属性文件中添加任意多个掩码。如果添加输入时间戳记, 则必须使用当前顺序的下一个数字。(例如, 如果添加到上面的列表中, 则下一个时间戳记掩码将是 `InputTimeStampFormat.8`。)

您也可以通过修改标识解析器定制程序属性文件中的以下属性的值定制输出时间戳记格式、微秒掩码和特定于数据库的格式。

```
TargetTimeStampFormat = yyyy-MM-dd HH:mm:ss.SSSSSS
MicroSecondMask = SSSSSS
DatabaseSpecificFormat = YYYY-MM-DD HH24:MI:SS
```

设置标识解析器如何处理存储

这里是标识解析器定制程序属性文件中的一部分, 它指定了与持久散列映射相关的属性的缺省值:

```
////////////////////////////////////
/// 0 = Normal hashmap with no backend storage
/// 1 = JDBM
////////////////////////////////////

PersistentStorageType = 0

////////////////////////////////////
/// If PersistentStorageType != 0, set MemoryStorageSize to the maximum size
/// of the hashmap in memory data and after that the hashmap will stream
/// the data to a persistent storage as specified
/// If -1, then it uses the normal hashmap with no backend storage
////////////////////////////////////

MemoryStorageSize = 1
```

您可以通过设置标识解析器定制程序属性文件中 `PersistentStorageType` 的值, 指定标识解析器如何处理持久存储。

- 如果设置 `PersistentStorageType = 0`, 则标识解析器将以“常规”方式运行(其中符号散列映射存在于内存中)。
- 如果设置 `PersistentStorageType = 1`, `JDBM` 将用于保存符号和键。

您可以通过设置标识解析器定制程序属性文件中 `MemoryStorageSize` 的值, 指定内存中存储的记录数目。

- `MemoryStorageSize` 的值为“1”表示在内存中仅保留一条记录。
- `MemoryStorageSize` 的值为“-1”有特殊的含义, 它表示在内存中保留所有记录。

在此情况下, 标识解析器将回复到它的“常规”行为。

设置标识解析器如何处理数据库驱动程序

标识解析器定制程序属性文件中的以下行指定了数据库驱动程序的缺省值:

```
DBVendorName = DB2
DBDriverName = COM.ibm.db2.jdbc.app.DB2Driver
DBURL = jdbc:db2:
```

其中:

- DBVendorName 用于选择数据库的类型。

选项如下:

- DB2 通用数据库 iSeries 版 (DB2/iSeries)
- 用于其它操作系统的 DB2 (DB2)
- Oracle 数据库 (oracle)

- DBDriverName 用于选择 JDBC 驱动程序。

选项如下:

- DB2 通用数据库 iSeries 版 (com.ibm.db2.jdbc.app.DB2Driver)
- 用于其它操作系统的 DB2 (COM.ibm.db2.jdbc.app.DB2Driver)
- Oracle 数据库 (oracle.jdbc.driver.OracleDriver)

- DBURL 用于指定访问数据库的 URL。

选项如下:

- DB2 通用数据库 iSeries 版 (jdbc:db2://)
- 用于其它操作系统的 DB2 (jdbc:db2:)
- Oracle 数据库 (jdbc:oracle:oci8:@)

确定如何处理数据

“标识解析”命令使您可以选择 load、update 或 mixed 方法来处理输入文件。

- 如果文件中的**所有记录都不存在于**数据库中, 请使用 load 方法处理输入文件。
- 如果文件中的**所有记录都存在于**数据库中, 请使用 update 方法处理输入文件。
- 如果文件中**仅部分记录存在于**数据库中, 请使用 mixed 方法处理输入文件。

选择 load 方法

标识解析器的 load 方法用于为装入到数据库中的记录生成新标识。使用此方法可以为记录创建新标识。以下示例用于为新数据生成标识:

-  NT 2000

```
idresgen -dbname db -dbuser user -dbpwd pwd -infile input.xml
-outfile output.xml -method load -customizer customizer -schemaname mall
```

load 方法是缺省值。

-  AIX Solaris Linux

```
./idresgen.sh -dbname db -dbuser user -dbpwd pwd -infile input.xml
-outfile output.xml -method load -customizer customizer -schemaname mall
```

load 方法是缺省值。

• **400**

```
QWEBCOMM/RESWC SID DATABASE(DATABASE_NAME) SCHEMA(MALL)
INSTROOT(/QIBM/UserData/WebCommerce/instances/mser)
PASSWD(mypassword) INFILE(input.xml) OUTFILE(output.xml)
METHOD(*LOAD)
```

选择 update 方法

如果为标识解析器指定 **update** 方法，则输入文件中的记录应当在数据库中已经存在。标识解析器在数据库中定位标识。如果记录在数据库中不存在，标识解析器将无法为此记录解析标识，并且将指出发生了错误。以下示例用于为数据库中已经存在的数据定位标识：

• **NT** **2000**

```
idresgen -dbname db -dbuser user -dbpwd pwd -infile input.xml
-outfile output.xml -method update -customizer customizer -schemaname mall
```

• **AIX** **Solaris** **Linux**

```
./idresgen.sh -dbname db -dbuser user -dbpwd pwd -infile input.xml
-outfile output.xml -method update -customizer customizer -schemaname mall
```

• **400**

```
QWEBCOMM/RESWC SID DATABASE(DATABASE_NAME) SCHEMA(MALL)
INSTROOT(/QIBM/UserData/WebCommerce/instances/mser) PASSWD(mypassword)
INFILE(input.xml) OUTFILE(output.xml) METHOD(*UPD)
```

选择 mixed 方法

如果输入数据文件包含数据库中已经存在的记录以及一些新的记录，则必须使用 **mixed** 方法运行标识解析器。使用此方法，仅当记录在数据库中不存在时标识解析器才为这些记录创建新标识。而现有的标识是从数据库中获取的。以下示例用于为新数据生成标识，并为在数据库中已经存在的数据定位标识：

• **NT** **2000**

```
idresgen -dbname db -dbuser user -dbpwd pwd -infile input.xml
-outfile output.xml -method mixed -customizer customizer -schemaname mall
```

• **AIX** **Solaris** **Linux**

```
./idresgen.sh -dbname db -dbuser user -dbpwd pwd -infile input.xml
-outfile output.xml -method mixed -customizer customizer -schemaname mall
```

• **400**

```
QWEBCOMM/RESWC SID DATABASE(DATABASE_NAME) SCHEMA(MALL)
INSTROOT(/QIBM/UserData/WebCommerce/instances/mser) PASSWD(mypassword)
INFILE(input.xml) OUTFILE(output.xml) METHOD(*MIX)
```

使用标识解析技术

标识解析器可以使用 Java 属性文件确定应当用主表的哪些列查找需要主表标识的表。如果表在 **KEYS** 或 **SUBKEYS** 表中列出，则它是主表。

要使用标识解析器的内部别名解析，将别名放置在 XML 文件的主键属性（identifier）中。之后别名可以用于整个 XML 文件以引用该元素。这一过程使程序不再需要确定创建 XML 文件所必需的唯一索引。

标识解析器也可以分析数据库模式来确定是否存在满足其需求的唯一索引。仅当在属性文件中没有所分析的表的条目或属性文件不存在时，标识解析器才查找唯一索引。如果这些条件为真，则执行唯一索引检查。如果唯一索引存在但不包含表中的主键，则认为此唯一索引有效。

使用标识解析器指定属性文件

标识解析器让您使用备用的 Java 属性文件来描述应当用主条目中的哪些列来查找需要主行标识的表。

缺省属性文件是 `IdResolveKeys.properties`，但是您可以在调用标识解析命令时修改它或指定自己的文件（如果您希望的话）。

-     

`IdResolveKeys.properties` 位于以下目录中：

-  `drive:\WebSphere\CommerceServer\properties`
-  `drive:\Program Files\WebSphere\CommerceServer\properties`
-  `/usr/WebSphere/CommerceServer/properties`
-   `/opt/WebSphere/CommerceServer/properties`

如果在运行标识解析器时不将此文件保存到当前目录中，则可以将此文件保存到类路径环境变量中定义的目录中。您也可以指定至此文件的全路径。

- 

要更改 `IdResolveKeys.properties`，请从 `/QIBM/ProdData/WebCommerce/properties` 目录复制此文件，将它保存到 `/instroot/xml` 目录，然后对新文件进行任何必要的更改。

注：以上目录在 `RESWC SID` 命令使用的类路径中。

使用属性文件生成标识

在以下示例中，需要为 ADDRBOOK 和 ADDRESS 记录分别解析标识 ADDRBOOK_ID 和 ADDRESS_ID。MEMBER 记录的标识是已知的。对于 WebSphere Commerce 数据库，每个记录都需要一个有效标识。另外，ADDRESS 记录中的 ADDRBOOK_ID 需要来自主表的标识，以满足它的外键约束。

```
<MEMBER
  MEMBER_ID="100"
  TYPE="U"
  STATE="1"
/>
<MEMBER
  MEMBER_ID="101"
  TYPE="U"
  STATE="1"
/>
<ADDRBOOK
  MEMBER_ID="100"
  DISPLAYNAME="Friends"           DISPLAYNAME 列的实际值
  DESCRIPTION="All my friends"
  TYPE="P"
/>
<ADDRESS
  ADDRBOOK_ID="@Friends"        引用使用 DISPLAYNAME 值作为查找的 ADDRBOOK
  MEMBER_ID="101"
  NICKNAME="Bob"
  ADDRESS1="15 Brave Developers St."
  CITY="Toronto"
  ZIPCODE="A0A0A0"
  COUNTRY="Canada"
  STATUS="P"
/>
```

您需要属性文件来标识在解析外键列的标识时关系行将使用主要行中的哪些列。以下过程确保以上文件的分析可以正确进行：

在 IdResolveKeys.properties 中，指定以下内容：

```
NAMEDELIMITER=@
SELECTDELIMITER=:

ADDRBOOK=@DISPLAYNAME:DISPLAYNAME
ADDRESS=@NICKNAME:NICKNAME
```

NAMEDELIMITER 和 SELECTDELIMITER 设置了在整个属性文件中使用的定界符，并且这些定界符必须使用一致。

ADDRBOOK=@DISPLAYNAME:DISPLAYNAME 声明了在接收到通讯录记录时，将为通讯录行创建标识。从输入记录中抽取 DISPLAYNAME 字段，并使用此字段组成到新标识的关联。DISPLAYNAME 字符串用于匹配通讯录行 DISPLAYNAME 并解析外键需要的标识。

使用先前的输入示例，在此示例中 DISPLAYNAME 是“Friends”，假定为此记录创建的标识是 12951。DISPLAYNAME 用来作为指向 12951 的键。继续处理下一个记录 ADDRESS，其中 ADDRBOOK_ID 具有“...”形式（它表示跟随在定界符之后的信息将用于查找通讯录标识）。字符串匹配 DISPLAYNAME，返回 12951 并放置在 ADDRBOOK_ID 属性中。

```

<MEMBER
  MEMBER_ID="100"
  TYPE="U"
  STATE="1"
/>
<MEMBER
  MEMBER_ID="101"
  TYPE="U"
  STATE="1"
/>
<ADDRBOOK
  ADDRBOOK_ID="12951"
  MEMBER_ID="100"
  DISPLAYNAME="Friends"
  DESCRIPTION="All my friends"
  TYPE="P"
/>
<ADDRESS
  ADDRESS_ID="13051"
  ADDRBOOK_ID="12951"
  MEMBER_ID="101"
  NICKNAME="Bob"
  ADDRESS1="15 Brave Developers St."
  CITY="Toronto"
  ZIPCODE="A0A0A0"
  COUNTRY="Canada"
  STATUS="P"
/>

```

生成的主键
ADDRBOOK DISPLAYNAME 的值未更改

生成的主键
ADDRESS 引用正确的 ADDRBOOK

使用带有组合键的属性文件

由多于两列组成的键是组合键。您可以在属性文件中定义组合键查找，这是通过指定在 NAMEDELIMITER 和 SELECTDELIMITER 后面跟字段名来实现的。例如，要将显示名称和成员标识的组合作为 ADDRBOOK 记录的查找条件，可以在属性文件中指定以下信息：

```
ADDRBOOK=@DISPLAYNAME@MEMBER_ID:DISPLAYNAME MEMBER_ID
```

然后是以下 XML 输入文件片段：

```

<ADDRBOOK
  MEMBER_ID="100"
  DISPLAYNAME="Friends"
  DESCRIPTION="All my friends"
  TYPE="P"
/>
<ADDRBOOK
  MEMBER_ID="101"
  DISPLAYNAME="Friends"
  DESCRIPTION="All my friends"
  TYPE="P"
/>

```

MEMBER 100 的 ADDRBOOK "Friends"

MEMBER 101 的 ADDRBOOK "Friends"

```

<ADDRESS
  ADDRBOOK_ID="@Friends@100"
  MEMBER_ID="101"
  NICKNAME="Bob"
  ADDRESS1="15 Brave Developers St."
  CITY="Toronto"
  ZIPCODE="A0A0A0"
  COUNTRY="Canada"
  STATUS="P"
/>

```

查找 MEMBER 100 的 ADDRBOOK "Friends" 的主键

解析后将生成以下内容:

```

<MEMBER
  MEMBER_ID="100"
  TYPE="U"
  STATE="1"
/>
<MEMBER
  MEMBER_ID="101"
  TYPE="U"
  STATE="1"
/>
<ADDRBOOK
  ADDRBOOK_ID="12951"
  MEMBER_ID="100"
  DISPLAYNAME="Friends"
  DESCRIPTION="All my friends"
  TYPE="P"
/>
<ADDRBOOK
  ADDRBOOK_ID="12952"
  MEMBER_ID="101"
  DISPLAYNAME="Friends"
  DESCRIPTION="All my friends"
  TYPE="P"
/>
<ADDRESS
  ADDRESS_ID="13051"
  ADDRBOOK_ID="12951"
  MEMBER_ID="101"
  NICKNAME="Bob"
  ADDRESS1="15 Brave Developers St."
  CITY="Toronto"
  ZIPCODE="A0A0A0"
  COUNTRY="Canada"
  STATUS="P"
/>

```

兴趣 ADDRBOOK

ADDRESS 引用正确的 ADDRBOOK

使用带有级联的主键的属性文件

主表 STOREENT 定义主键 STOREENT_ID。STORE (引用 STOREENT 的外表) 定义主键 STORE_ID, 此主键是到主表 STOREENT 的外键。这意味着 STORE_ID 的值必须是 STOREENT_ID 值的其中之一。STORE_ID (外表 STORE 的主键) 因此具有双重角色 — 主键和外键。

让我们假定另一个表 CONTRACT 是 STORE 的外表, 并且 CONTRACT 的外键 STORE_ID 引用 STORE 中的主键 STORE_ID。STORE 表因此是 CONTRACT 表的主表。

由于 STORE 表的 STORE_ID 引用自 STOREENT_ID，而不是创建的，因此标识解析器不为 STORE 表创建内部别名和标识值关联。当 CONTRACT 表尝试从 STORE 表中解析 STORE_ID 时，它将获取空值。

由于此特殊情况，您必须通过在属性文件中创建条目，来明确指定创建内部别名。在 IdResolveKeys.properties 中，指定以下内容：

```
"STORE=@STORE_ID:STORE_ID"
```

这强制标识解析器执行以下操作：

- 当 STORE_ID 作为外部引用被解析时，创建内部别名和标识值关联
- 当为 CONTRACT 表解析 STORE_ID 时，使用该关联

使用属性文件中的 STORE=@STORE_ID:STORE_ID 条目和以下 XML 输入文件片段：

```
<STOREENT
  IDENTIFIER="Out Fashions"
  MEMBER_ID="-2000"
  STOREENT_ID="@storeent_id_1"
  TYPE="G"
/>
<STORE
  STORE_ID="@storeent_id_1"
  STOREGRP_ID="1"
  STORELEVEL="store_level"
/>
<CONTRACT
  CONTRACT_ID="@contract_id_1"
  STATE="0"
  STORE_ID="@storeent_id_1"
/>
```

解析后将生成以下内容：

```
<STOREENT
  IDENTIFIER="Out Fashions"
  MEMBER_ID="-2000"
  STOREENT_ID="10501"
  TYPE="G"
/>
<STORE
  STORE_ID="10501"
  STOREGRP_ID="1"
  STORELEVEL="store_level"
/>
<CONTRACT
  CONTRACT_ID="@contract_id_1"
  STATE="0"
  STORE_ID="10501"
/>
```

使用内部别名解析

要使用标识解析器的内部别名解析，将别名放置在 XML 文件的主键属性（identifier）中。之后别名可以用于整个 XML 文件以引用该元素。这一过程使程序不再需要确定创建 XML 文件所必需的唯一索引。

内部别名在整个文件中必须使用一致。如果通讯录标识 `addrbook_id` 的别名定为 `@addrbook_1`，则文件中所有引用该标识的外键都必须使用 `@addrbook_1`。请注意别名是瞬时的。它们不会被保存，且在不重新引用别名时不能在独立的 XML 文件中使用。

使用内部别名标识解析的部分示例

解析前:

```
<MEMBER
  MEMBER_ID="100"
  TYPE="U"
  STATE="1"
/>
<ADDRBOOK
  ADDRBOOK_ID="@addrbook_1"           ADDRBOOK 的别名
  MEMBER_ID="100"
  DISPLAYNAME="Friends"
  DESCRIPTION="All my friends"
  TYPE="P"
/>
<ADDRESS
  ADDRESS_ID="@address_1"           ADDRESS 的别名
  ADDRBOOK_ID="@addrbook_1"       对 ADDRBOOK 的别名的引用
  MEMBER_ID="101"
  NICKNAME="Bob"
  ADDRESS1="1 Brave Developer St."
  CITY="Toronto"
  ZIPCODE="A3B0F4"
  COUNTRY="Canada"
  STATUS="P"
/>
```

解析后:

```
<MEMBER
  MEMBER_ID="100"
  TYPE="U"
  STATE="1"
/>
<ADDRBOOK
  ADDRBOOK_ID="11801"             已生成的主键
  MEMBER_ID="100"
  DISPLAYNAME="Friends"
  DESCRIPTION="All my friends"
  TYPE="P"
/>
<ADDRESS
  ADDRESS_ID="11901"             已生成的主键
  ADDRBOOK_ID="11801"         引用 ADDRBOOK 条目
  MEMBER_ID="100"
  NICKNAME="Bob"
  ADDRESS1="1 Brave Developer St."
  CITY="Toronto"
  ZIPCODE="A3B0F4"
  COUNTRY="Canada"
  STATUS="P"
/>
```

使用唯一索引解析

在所分析表的属性文件中没有任何条目或属性文件不存在时，使用唯一索引解析（标识解析器的缺省行为）。唯一索引解析使用表上任何指定的唯一索引作为定位标识的方

法。例如，MEMBER_ID 加上 IDENTIFIER 是 CATALOG 表的唯一索引，因此可以用作到 CATALOGDSC 表的主键 CATALOG_ID 的解析点。

要更新数据库的内容，您需要知道数据库中主表的唯一键。可以查询数据库找到唯一键。例如，检索唯一键的 DB2 命令可以显示如下：

```
db2 describe indexes for table schema.tablename show detail
```

唯一索引解析的部分示例

解析前:

```
<MEMBER
  MEMBER_ID="100"
  TYPE="0"
  STATE="1"
/>

<CATALOG
  DESCRIPTION="Winter Catalog"
  IDENTIFIER="WC2001"
  MEMBER_ID="100"
  TPCLEVEL="2"
/>

<CATALOGDSC
  CATALOG_ID="@WC2001@100"           引用回成员“100”的产品目录“WC2001”（注意：顺序很重要。）
  FULLIMAGE="c:\store\img\wc.gif"
  LANGUAGE_ID="-1"
  LONGDESCRIPTION="2001 Winter Catalog"
  SHORTDESCRIPTION="2001 Winter Catalog"
  NAME="InFashion 2001 Winter Catalog"
  THUMBNAIL="c:\store\img\wc_th.gif"
/>
```

解析后:

```
<MEMBER
  MEMBER_ID="100"
  TYPE="0"
  STATE="1"
/>

<CATALOG
  CATALOG_ID="10351"                 自动生成的主键
  DESCRIPTION="Winter Catalog"
  IDENTIFIER="WC2001"
  MEMBER_ID="100"
  TPCLEVEL="2"
/>

<CATALOGDSC
  CATALOG_ID="10351"                 引用正确的产品目录
  FULLIMAGE="c:\store\img\wc.gif"
  LANGUAGE_ID="-1"
  LONGDESCRIPTION="2001 Winter Catalog"
  SHORTDESCRIPTION="2001 Winter Catalog"
  NAME="InFashion 2001 Winter Catalog"
  THUMBNAIL="c:\store\img\wc_th.gif"
/>
```

将数据装入 MEMBER 表

标识解析器为系统为其生成标识的表处理解析。这包含在 KEYS 或 SUBKEYS 表中已注册的任何表和列。此解析有两个部分构成：

1. 确定在数据库中是否存在主表（即在 KEYS 或 SUBKEYS 中列出的表）。
此解析是基于元素的 XML 数据内容使用唯一索引解析或属性文件指定进行的。
2. 确定存在到主表的外键。
这通过使用相关表的外键属性的解析指定完成。

MEMBER 表用作 ORGENTITY、MBRGRP 和 USER 表的“超类”。这将创建“is-a”模式，如果表具有到 MEMBER 表子类型的外键约束，此模式有助于维护引用完整性。因为所有 MEMBER 子类型共享一个公共的基本类型，但是，在子类型中标识必须是唯一的。这意味着在 MBRGRP_ID 和 USER_ID 集中 USER_ID/ORGENTITY_ID 必须是唯一的。要达到这一目的，KEYS 表仅引用 ORGENTITY、MBRGRP 和 USER 表，并为其标识指定互斥范围。每一子类型都有一个主键；每一此类主键还是到 MEMBER 表主键的外键。

MEMBER 及其子类型之间的约束产生了一种情况，在此情况下，MEMMBER 及其子类型不能具有同步标识。为了将 ORGENTITY、MBRGRP 和 USER 表装入系统，标识解析器识别“i-sa”模式并适当地加以处理。以下是标识解析器的 XML 语法：

```
<ORGENTITY
  ORGENTITY_ID="@orgAlias"
  ORGENTITYNAME="Test Org"
  ORGENTITYTYPE="0">
  <ISA>
  <MEMBER
    TYPE="0"
    STATE="1"
  />
  </ISA>
</ORGENTITY>
```

生成以下语法：

```
<MEMBER
  MEMBER_ID="12345"
  TYPE="0"
  STATE="1"
/>
<ORGENTITY
  ORGENTITY_ID="12345"           与 member 元素同步
  ORGENTITYNAME="Test Org"
  ORGENTITYTYPE="0"
/>
```

通过此方法，标识解析器处理 <isa> 子元素并创建同步标识。

使用 REFKEYS 表创建外部关系

创建 REFKEYS 表以用于展示各表之间尚不存在于数据库中的外部关系。一般而言，数据库模式通过创建外键声明来描述了外部关系，此声明将表的列链接到另一个表。如果数据库模式没有定义外部关系，并且标识必须解析为外键，则请执行以下操作：

1. 如以下示例 DDL 中所示，创建 REFKEYS 表：

```
CREATE TABLE "REFKEYS" (  
    "FKTABLE_NAME"    CHAR(18)    NOT NULL ,  
    "FKCOLUMN_NAME"   CHAR(18)    NOT NULL ,  
    "TABLENAME"       CHAR(18)    NOT NULL  
);
```

其中：

FKTABLE_NAME 是外（或“子”）表名

FKCOLUMN_NAME 是外列名

TABLENAME 是主（或“父”）表名

2. 在 REFKEYS 表中创建描述必需的外部关系的条目。

错误疑难解答

如果在解析标识期间发生错误，请参考下表：

错误	使用的方法	可能的原因	可能的解决方案
未解析主键	所有	标识解析器没有解析表中的主键（标识），此表在 KEYS 和 SUBKEYS 表中都没有指定。	在运行标识解析器前向 SUBKEYS 表中添加将解析其主键的表的名称。 而且，请确保外键关系存在于数据库模式中。
	更新	通过查询数据库解析主键。数据库查询是通过使用属性文件条目信息或给定表的唯一索引生成的。属性文件条目具有优先级。	请确保输入文件中的唯一索引信息是正确的。 您可能还希望创建或修改属性文件中适当的条目以生成适当的数据库查询，来解析主键。
未解析外键	所有	标识解析器没有解析表中的外键，此表中的外部引用指向 KEYS 和 SUBKEYS 表中都没有指定的表。	在运行标识解析器前向 SUBKEYS 表中添加引用的表的名称。 而且，请确保外键关系存在于数据库模式中。
		通过使用内部别名和通过查询数据库解析外键。仅当内部别名解析外键失败时才进行数据库查询。	内部别名是通过使用主键并使用属性文件条目生成的。 数据库查询是通过使用属性文件条目信息或给定表的唯一索引生成的。属性文件条目具有优先级。

第 6 章 装入数据

装入数据之前，您必须执行以下操作：

1. 生成与装入程序一起使用的 DTD 和模式（第一次装入数据）

注：如果您正在为商店归档文件装入数据，并用与商店归档文件一起提供的 DTD 创建了 XML 文件，则不必执行这一步。

2. 解析标识（如果需要）

数据必须是 XML 格式，同时具有关联的 DTD。要装入数据，请调用“装入”命令。

设置装入程序

装入程序软件包使您可以通过执行以下操作来设置装入程序如何运行：

- 忽略输入文件中的元素
- 将 NULL 插入列中
- 装入时间戳记和日期数据
- 装入当前时间戳记
- 管理事件队列
- 与不同的数据库软件和操作系统一起运行
- 替换组件
- 使用“产品顾问搜索空间同步”

通过执行以下操作，您可以定制装入程序的这些功能：

1. 创建新的装入程序定制程序属性文件。



• `MassLoadCustomizer.properties` 位于 `MassLoader.zip` 压缩文档中。将此文件解压缩并重命名，但保留 `.properties` 扩展名，并将它放置在类路径中的目录中。**重要信息：**不要除去或修改现有的 `MassLoadCustomizer.properties` 文件。



• `ISeries_LODWCSDTA_Customizer.properties` 位于 `/QIBM/ProdData/WebCommerce/properties` 目录中。将此文件复制到 `/instroot/xml` 目录，重命名新文件但保留 `.properties` 扩展名，然后对新文件进行任何必要的更改。**重要信息：**请不要除去或修改原始的 `ISeries_LODWCSDTA_Customizer.properties` 文件。

2. 修改在新的装入程序定制程序属性文件中指定的属性值。
3. 将新文件名指定为“装入”命令的 `customizer` 参数的值。

忽略输入文件中的元素

如果输入文件包含不映射到目标数据库的元素，则可以设置装入程序以忽略装入程序定制程序属性文件中这些元素。请使用 `IgnoreElements` 指定要忽略的元素，并用分号 (;) 分隔这些元素。要忽略诸如 `import`、`literals` 和 `ProductRepository` 等元素，请在装入程序定制程序属性文件中指定以下内容：

```
IgnoreElements = import;literals;ProductRepository
```

将 NULL 插入列

通过将装入程序定制程序属性文件中的 `EnableNULLCheck` 属性设置为 “true”，装入程序就能够将 NULL 插入列。例如：

```
EnableNULLCheck = true
```

因性能原因，此功能在缺省情况下是禁用的。

使用 `NULLStringLiteral` 属性确定您数据中空值的字符串表示法。例如，要设置装入程序以便字符串 “-” 用以表示空值，在装入程序定制程序属性文件中指定以下属性和值：

```
NULLStringLiteral = -
```

缺省情况下，此属性的值是 “NULL”（没有引号）。

装入时间戳记和日期数据

装入程序可以将数据装入到时间戳记和日期数据类型的列中。文档中时间戳记和日期数据的数据格式由可定制的模式确定。用户可以编辑现有模式，或将更多模式添加到现有的模式列表中。

时间戳记或日期数据的检查是对照可用的模式（掩码）进行的。第一个与数据匹配的模式用于在数据装入到数据库之前将数据转换成目标时间戳记格式。

有两种可定制的输出时间戳记模式：`TimeStampFormat.JDBC` 和 `TimeStampFormat.Load`。

1. `TimeStampFormat.JDBC` 在装入程序使用 JDBC 连接来执行操作时使用。
装入程序的 SQL 导入和删除方法使用 JDBC 连接来更新数据库。
2. `TimeStampFormat.Load` 在装入程序使用本机实用程序时使用。
装入程序的 `import` 和 `load` 方法使用本机实用程序。

可通过在装入程序定制程序属性文件中修改或添加掩码来定制时间戳记格式。

已提供以下输入时间戳记掩码:

```
InputTimeStampFormat.1 = yyyy-DD hh:mm:ss.SSSSSS
InputTimeStampFormat.2 = yyyy-MM-dd hh:mm:ss.SSSSSS
InputTimeStampFormat.3 = yyyy-DD-hh.mm.ss.SSSSSS
InputTimeStampFormat.4 = yyyy-MM-dd-HH.mm.ss.SSSSSS
InputTimeStampFormat.5 = yyyy-MM-dd-hh.mm.ss.SSSSSS
InputTimeStampFormat.6 = yyyy-MM-dd HH:mm:ss.SSSSSS
InputTimeStampFormat.7 = yyyy-DD HH:mm:ss.SSSSSS
```

输入日期格式的缺省模式如下:

```
InputDateFormat.1 = MM-dd-yyyy
InputDateFormat.2 = yyyy-dd-MM
InputDateFormat.3 = yyyy-MM-dd
InputDateFormat.4 = MM/dd/yyyy
InputDateFormat.5 = yyyy/dd/MM
InputDateFormat.6 = yyyy-DD
```

您可以修改这些时间戳记和日期掩码, 或根据自己的需要添加任意多个掩码。在装入程序定制程序属性文件中按数字顺序指定这些掩码, 其数字顺序是您希望它们与输入时间戳记相对应的顺序。如果添加输入时间戳记, 则必须使用当前顺序的下一个数字。(例如, 如果添加到上面列表中, 则下一个时间戳记掩码将是 InputTimeStampFormat.8。)

将输入数据格式化时间戳记和日期输出所采用的模式如下:

```
TimeStampFormat.JDBC = yyyy-MM-dd hh:mm:ss.SSSSSS
TimeStampFormat.Load = yyyy-MM-dd-hh.mm.ss.SSSSSS

DateFormat.JDBC = yyyy-MM-dd
DateFormat.Load = yyyy-MM-dd
```

一般而言, 输出日期和时间戳记格式不是定制的。

装入当前时间戳记

装入程序可以基于日时时钟的读数, 将值插入到时间戳记数据类型的列中。例如, WebSphere Commerce 中报价的 STARTDATE 和 ENDDATE 的值可基于报价插入到表中的时间。要支持此功能, 装入程序软件包使用 MLTIME 表来保存时间戳记实例。此表的模式如下:

```
table MLTIME
(
  INSTANCEID BIGINT not null,
  MLTIMESTAMP TIMESTAMP
)
```

可通过在装入程序定制程序属性文件中更改以下属性来定制该表的名称及其列。

```
TimestampTableName = MLTIME
TimestampIdColumn = INSTANCEID
TimestampValueColumn = MLTIMESTAMP
```

指定当前时间戳记值的输入数据取决于时间戳记字符串模式。以下掩码用于指定时间戳记的持续时间:

```
%D for days
%M for months
```

%Y for years
%H for hours
%m for minutes
%s for seconds

可通过在装入程序定制程序属性文件中修改或添加掩码来定制当前时间戳记格式。已提供以下输入掩码:

```
InputCurrentTimestampFormat.1 = CURRENT TIMESTAMP  
InputCurrentTimestampFormat.2 = CURRENT TIMESTAMP %D DAYS  
InputCurrentTimestampFormat.3 = CURRENT TIMESTAMP %D DAYS %M MONTHS  
InputCurrentTimestampFormat.4 = CURRENT TIMESTAMP %D DAYS %M MONTHS %Y YEARS  
InputCurrentTimestampFormat.5 = CURRENT TIMESTAMP %Y YEARS %M MONTHS %D DAYS  
InputCurrentTimestampFormat.6 = SYSDATE  
InputCurrentTimestampFormat.7 = ADDDAYS(SYSDATE,%D)  
InputCurrentTimestampFormat.8 = ADDDAYS(ADDMONTHS(SYSDATE,%M),%D)  
InputCurrentTimestampFormat.9 = ADDDAYS(ADDMONTHS(ADDEYEARS(SYSDATE,%Y),%M),%D)
```

当前时间戳记的输入数据与指定的模式相匹配。如果数据与指定的输入模式相匹配, 则该模式将用于分析输入数据, 并且在将输入数据插入到数据库之前, 装入程序将把数据转换成适当的输出格式。如果按顺序排列脚注号, 可以将新模式添加到上面列表中。

指定当前时间戳记有两种目标输出格式:

1. `CurrentTimestampFormat.Load` 在装入程序正在以装入或导入方式操作时使用。
2. `CurrentTimestampFormat.JDBC` 在装入程序使用 JDBC 在数据库中插入、更新或删除值时使用。

装入程序中的缺省目标模式如下:

```
CurrentTimestampFormat.Load = CURRENT TIMESTAMP %Y YEARS %M MONTHS %D DAYS  
%h HOURS %m MINUTES %s SECONDS  
CurrentTimestampFormat.JDBC = CURRENT TIMESTAMP %Y YEARS %M MONTHS %D DAYS  
%h HOURS %m MINUTES %s SECONDS
```

这些模式的属性也可在装入程序定制程序属性文件中定制。当定制 `CurrentTimestampFormat.Load` 和 `CurrentTimestampFormat.JDBC` 属性时, 您应当确保结果语句的语法对于给定的数据库管理系统来说是有效的。

`CurrentTimestampLiteral` 属性由装入程序用于早期确定时间戳记列的值是否是当前时间戳记格式, 从而避免进行占用太多资源的计算来确定该值不是字符串表示法的时间戳记。

```
CurrentTimestampLiteral = CURRENT TIMESTAMP
```



对于 DB2, 此属性的缺省值是 `CURRENT TIMESTAMP`。



Oracle 数据库的缺省值是 `SYSDATE`。

装入当前时间戳记的示例

为装入程序提供了以下信息来更新 `OFFER_ID` 为 10123 的报价。开始日期的值为 “`CURRENT TIMESTAMP`”, 结束日期的值为 “`CURRENT TIMESTAMP + 14 DAYS`”。

```
<OFFER
  OFFER_ID="10123"
  STARTDATE="CURRENT_TIMESTAMP">
  ENDDATE="CURRENT_TIMESTAMP + 14 DAYS"
/>
```

装入程序识别列 STARTDATE 和 ENDDATE 采用的是数据库中的时间戳记数据类型。根据 CurrentTimeStampLiteral 属性，确定这些值是否以当前时间戳记格式指定。STARTDATE 的值与 InputCurrentTimeStampFormat.1 模式相匹配，并且转换成由 CurrentTimeStampFormat.JDBC 属性指定的模式。ENDDATE 的值与 InputCurrentTimeStampFormat.2 属性的格式相匹配，并且也转换成由 CurrentTimeStampFormat.JDBC 属性指定的模式。

添加持续时间至当前时间戳记的示例

装入程序向您提供了将持续时间添加到当前时间戳记的能力。例如，您可能希望在不输入特定日期的情况下装入报价。要实现此目的，您必须创建结束日期，此日期是开始日期之后的一段持续时间。以下示例在 DB2 下工作正常：

```
<Offer
  Startdate="Current Timestamp"
  Enddate="Current Timestamp +14 Days +4 Months +1 Year +0 Hours
    +0 Minutes +0 Seconds"
/>
```

但是，如果要采用不依赖于平台的方式处理当前时间戳记持续时间，则必须通过在装入程序定制程序属性文件中修改掩码来定制当前时间戳记格式。以下是已定制的当前时间戳记属性规范的示例：

```
CurrentTimeStampLiteral=Current Timestamp

InputCurrentTimeStampFormat.0=Current Timestamp
InputCurrentTimeStampFormat.1=Current Timestamp %D Days
InputCurrentTimeStampFormat.2=Current Timestamp %M Months
InputCurrentTimeStampFormat.3=Current Timestamp %Y Years
InputCurrentTimeStampFormat.4=Current Timestamp %D Days %M Months
InputCurrentTimeStampFormat.5=Current Timestamp %D Days %M Months %Y Years
InputCurrentTimeStampFormat.5=Current Timestamp %H Hours %m Minutes %s Seconds

CurrentTimeStampFormat.JDBC=Current Timestamp %D Days %M Months %Y Years
  %H Hours %m Minutes %s Seconds
```

使用报价示例和这些属性规范，报价的结束日期与 InputCurrentTimeStampFormat.5 模式相匹配。这使用 CurrentTimeStampFormat.JDBC 生成以下报价信息。

```
<Offer
  Startdate="Current Timestamp"
  Enddate="Current Timestamp +14 Days +4 Months +1 Year +0 Hours +0 Minutes +0 Seconds"
/>
```

上面示例显示了装入程序如何能够输入多个当前时间戳记格式，并将它们适当地格式化成期望的输出格式。以下示例显示了您如何能够处理不依赖于平台的格式，并将它们映射成特定于平台的输出格式。

```
<Offer
  Startdate="Now"
  Enddate="Now +14D +4M +1Y"
/>
```

```

CurrentTimestampLiteral=Now

InputCurrentTimestampFormat.0=Now
InputCurrentTimestampFormat.1=Now %DD
InputCurrentTimestampFormat.2=Now %MM
InputCurrentTimestampFormat.3=Now %YY
InputCurrentTimestampFormat.4=Now %DD %MM
InputCurrentTimestampFormat.5=Now %DD %MM %YY
InputCurrentTimestampFormat.5=Sysdate %HH %mm %ss

CurrentTimestampFormat.JDBC=AddYears (AddMonths (AddDays (AddHours (AddMinutes (AddSeconds
(Sysdate,%s),%m),%H),%D),%M),%Y)

```

注：以上语句只是个示例。只用于说明假想的数据库管理系统的定制功能。它对于 DB2 或 Oracle 数据库无效。

使用报价示例和这些属性规范，报价的结束日期将与 InputCurrentTimestampFormat.5 模式相匹配。这使用 CurrentTimestampFormat.JDBC 生成以下报价信息。

```

<Offer
  Startdate="Current Timestamp"
  Enddate="AddYears (AddMonths (AddDays (AddMinutes (AddSeconds (Sysdate,0),0),0),14),4),1)"
/>

```

管理事件队列

可通过在装入程序定制程序属性文件中修改设置来定制装入程序的事件队列。例如：

```

QueueLowCount = 35
QueueHighCount = 90

```

当队列中的元素数目达到上限时，填充队列的事件源将被阻塞，从而阻止更多事件参与排队。当队列中的元素数目降至低于下限时，队列再次开始接受事件。

与不同的数据库软件和操作系统一起运行

可通过在装入程序定制程序属性文件中修改以下元素的参数以指定不同的数据库软件和操作系统来定制装入程序，使其与不同的数据库和操作系统一起运行。

- 数据库连接命令
- 数据库装入表命令
- 数据库导入命令
- 调用装入的系统命令

要定制其中某一项，请在装入程序定制程序属性文件中除去命令前的双斜杠注释字符 (//)，并修改缺省值。

数据库连接命令： 如果希望修改缺省值（缺省值假定您正在使用 DB2），则可以更改数据库连接命令的参数。

```
DBConnectCommand = connect to {0} user {1} using {2};
```

其中：

- 0 = 数据库名
- 1 = 数据库用户
- 2 = 用户密码

数据库装入表命令: 如果希望修改缺省值, 则可以更改数据库装入表命令的参数。

```
DBLoadTableCommand = load from {0} of del modified by coldel{1}  
chardel{2} insert into {3} ({4});
```

其中:

- 0 = 文件名
- 1 = 列定界符
- 2 = 字符定界符
- 3 = 表名
- 4 = 列名, 以逗号 (,) 分隔

数据库导入命令: 如果希望修改缺省值, 则可以更改数据库导入命令的参数。

```
DBImportCommand = import from {0} of del modified by coldel{1} chardel{2}  
insert_update into {3} ({4});
```

其中:

- 0 = 文件名
- 1 = 列定界符
- 2 = 字符定界符
- 3 = 表名
- 4 = 列名, 以逗号 (,) 分隔

调用装入的系统命令: 如果希望修改缺省值, 则可以更改调用装入的系统命令的参数。此命令运行由装入程序生成的本机装入和导入脚本。

•  DB2

```
DBLoadCommand = db2c1pex DB2 -z {0} -astvf {1}
```

其中:

0 = 日志文件名
1 = 命令文件名

例如, 对于运行在 AIX 上的 DB2, DBLoadCommand 属性的值如下:

```
db2 -tvf {1} -z {0}
```



```
DBLoadCommand = sqlldr log={0} control={1} USERID={2}
```

其中:

0 = 日志文件名
1 = 命令文件名
2 = 数据库用户名

为不同数据库和操作系统组合使用以下设置:



使用 sqlimport、load、import 或 delete 方法, 为运行在 Windows NT 或 Windows 2000 上的 DB2 设置类路径系统环境变量以包含 db2/dbconnect.zip。



对于运行在 AIX、Solaris 或 Linux 环境中的 DB2, 请执行以下操作:

- 使用 sqlimport、load、import 或 delete 方法, 设置类路径系统环境变量以包含 db2/dbconnect.zip。
- 使用 load 和 import 方法, 在装入程序定制程序属性文件中修改以下属性。

```
/**
 * Connection command. (Default is for DB2)
 * parameter 0 = dbName
 * parameter 1 = dbUser
 * parameter 2 = userPasswd
 */
DBConnectCommand = connect to {0} user {1} using {2};

/**
 * Load Data into Table command. (Default is for DB2)
 * parameter 0 = filename
 * parameter 1 = column delimiter
 * parameter 2 = character delimiter
 * parameter 3 = name of the table
 * parameter 4 = name of the columns, separated by comma(,)s
 */
DBLoadTableCommand = load from {0} of del modified by coldel{1}
insert into {3} ({4});

/**
 * Insert Data into Table command. (Default is for DB2)
 * parameter 0 = filename
 * parameter 1 = column delimiter
 * parameter 2 = character delimiter
 * parameter 3 = name of the table
```





```

    * parameter 4 = name of the columns, separated by comma(,)s
*/

DBUpdateTableCommand = import from {0} of del modified by coldel{1}
insert_update into {3} ({4});
/**
 * System command to invoke load (Default is for DB2)
 * parameter 0 = logFileName
 * parameter 1 = commandFileName
*/

DBLoadCommand = db2 -z {0} -tf {1}

```

- 
 使用 sqlimport、load、import 或 delete 方法，在装入程序定制程序属性文件中为运行在 iSeries 上的 DB2 修改以下属性：

```

/**
 * The connect string.
*/

ConnectionStringID = jdbc:db2://

/**
 * The JDBC driver information.
*/

JDBCDriverName = com.ibm.db2.jdbc.app.DB2Driver
DbVendorName=DB2/iSeries

/**
 * Custom writer for load/import methods.
*/

WriterName=com.ibm.wca.MassLoader.Writer.ISeriesWriter

```

- 






使用 sqlimport、load、import 或 delete 方法，为运行在 Windows NT、Windows 2000、AIX、Solaris 或 Linux 环境中的 Oracle 数据库设置类路径系统环境变量以包含 oracle/dbconnect.zip。

替换组件

通过在装入程序定制程序属性文件中提供以下元素，可以用您希望替换缺省实现的类值替换装入程序组件：

ParserName

要使用的分析程序的名称

ValidatorName

要使用的验证程序的名称

FormatterName

要使用的格式化程序的名称

JDBCFormatterName

使用 SQL 导入方法时格式化程序的名称

WriterName

要使用的编写器的名称

JDBCWriterName

使用 SQL 导入方法时编写器的名称

例如，要用编写器 `com.abc.writer.SpecialWriter` 替换装入程序的缺省编写器（`DefaultWriter`），请在装入程序定制程序属性文件中指定以下内容：

```
WriterName = com.abc.writer.SpecialWriter
```

装入程序将使用 “`com.abc.writer.SpecialWriter`” 执行写功能。

使用产品顾问搜索空间同步

要使用 “产品顾问搜索空间同步”，请执行以下操作。

1. 为同步创建名称为 “`PASyncInfo.xml`” 的 XML 配置信息文件。
2. 在 `PASyncInfo.xml` 中，指定 `PASync.xsd` 作为要使用的 XML 模式。例如：

```
<PASync
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation='PASync.xsd'
```

提供了 `PASync.xsd` 文件。以下文本显示了 `PASync.xsd` 的内容：

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:element name="PASync">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="SearchScheme" />
        <xsd:element ref="Command" minOccurs="1" maxOccurs="unbounded" />
      </xsd:sequence>
      <xsd:attribute name="member" type="xsd:string" use="required" />
      <xsd:attribute name="store" type="xsd:string" use="required" />
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="SearchScheme">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="RelatedTable" />
        <xsd:element ref="Search" minOccurs="1" maxOccurs="unbounded" />
      </xsd:sequence>
      <xsd:attribute name="tableName" type="xsd:string" use="required" />
      <xsd:attribute name="primary" type="xsd:string" use="required" />
      <xsd:attribute name="columnName" type="xsd:string" use="required" />
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="RelatedTable">
    <xsd:complexType>
      <xsd:attribute name="tableName" type="xsd:string" use="required" />
      <xsd:attribute name="from" type="xsd:string" use="required" />
      <xsd:attribute name="to" type="xsd:string" use="required" />
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="Search">
    <xsd:complexType>
      <xsd:attribute name="value" type="xsd:string" use="required" />
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="Command">
    <xsd:complexType>
      <xsd:attribute name="tableName" type="xsd:string" use="required" />
      <xsd:attribute name="idColumnName" type="xsd:string" use="required" />
      <xsd:attribute name="addCommand" type="xsd:string" />
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

```

        <xsd:attribute name = "updateCommand" type="xsd:string" />
        <xsd:attribute name = "deleteCommand" type="xsd:string" />
    </xsd:complexType>
</xsd:element>

```

```
</xsd:schema>
```

3. 在 PASyncInfo.xml 中，指定需要执行同步的成员标识和商店实体标识。例如：

```

member = "-2000"
store = "10351"

```

4. 在 PASyncInfo.xml 中的搜索模式元素下，指定构成搜索空间的 CATGROUP 标识。例如：

```

<SearchScheme
  tableName = "catgroup"
  primary = "CATGROUP_ID"
  colName = "identifier" >

  <RelatedTable
    tableName = "catgpenrel"
    from = "CATGROUP_ID"
    to = "CATENTRY_ID" />

  <Search value="Pants" />
  <Search value="Shirts" />

</SearchScheme>

```

在示例中指定了“Pants”和“Shirts”。您可以按照自己的需要指定任意多个 CATGROUP 标识。

5. 在 PASyncInfo.xml 中，指定属性以确定调度什么命令。例如：

```

<Command tableName = "CATENTRY" idColumnName = "CATENTRY_ID"
  updateCommand = "UpdateSearchSpaces"
  deleteCommand = "RemoveProductsFromAllSearchSpaces"
/>

<Command tableName = "CATENTDESC" idColumnName = "CATENTRY_ID"
  addCommand = "UpdateSearchSpaces"
  updateCommand = "UpdateSearchSpaces"
  deleteCommand = "UpdateSearchSpaces"
/>

<Command tableName = "LISTPRICE" idColumnName = "CATENTRY_ID"
  addCommand = "UpdateSearchSpaces"
  updateCommand = "UpdateSearchSpaces"
  deleteCommand = "UpdateSearchSpaces"
/>

<Command tableName = "ATTRVALUE" idColumnName = "CATENTRY_ID"
  addCommand = "UpdateSearchSpaces"
  updateCommand = "UpdateSearchSpaces"
  deleteCommand = "UpdateSearchSpaces"
/>

<Command tableName = "CATENTATTR" idColumnName = "CATENTRY_ID"
  addCommand = "UpdateSearchSpaces"
  updateCommand = "UpdateSearchSpaces"
  deleteCommand = "UpdateSearchSpaces"
/>

<Command tableName = "CATGPENREL" idColumnName = "CATENTRY_ID"
  addCommand = "AddProductsToSearchSpace"

```

```
        deleteCommand = "RemoveProductsFromSearchSpace"  
    />
```

```
</PASync>
```

6. 在新的装入程序定制程序属性文件中，指定 XML 配置信息文件。例如：

```
PASyncDocumentURL = PASyncInfo.xml
```

7. 在新的装入程序定制程序属性文件中，启用同步。例如：

```
PASyncEnabled = true
```

8. 对“装入”命令使用 SQL 导入或删除方法。以下是装入程序的适当的 XML 输入示例：

```
<store-asset>
```

```
    <catentry  
      CATENTRY_ID="10351"  
      MEMBER_ID="-2000"  
      PARTNUMBER="000051"  
      CATENTTYPE_ID="ProductBean"  
      MFPARTNUMBER="m000051"  
      MARKFORDELETE="0"  
      BUYABLE="1"  
    />
```

```
    <catentry  
      CATENTRY_ID="10352"  
      MEMBER_ID="-2000"  
      PARTNUMBER="000052"  
      CATENTTYPE_ID="ProductBean"  
      MFPARTNUMBER="m000052"  
      MARKFORDELETE="0"  
      BUYABLE="1"  
    />
```

```
    <catentry  
      CATENTRY_ID="10353"  
      MEMBER_ID="-2000"  
      PARTNUMBER="000053"  
      CATENTTYPE_ID="ProductBean"  
      MFPARTNUMBER="m000053"  
      MARKFORDELETE="0"  
      BUYABLE="1"  
    />
```

```
    <catentry  
      CATENTRY_ID="10358"  
      MEMBER_ID="-2000"  
      PARTNUMBER="000058"  
      CATENTTYPE_ID="ProductBean"  
      MFPARTNUMBER="m000058"  
      MARKFORDELETE="0"  
      BUYABLE="1"  
    />
```

```
    <catentry  
      CATENTRY_ID="10365"  
      MEMBER_ID="-2000"  
      PARTNUMBER="000065"  
      CATENTTYPE_ID="ProductBean"  
      MFPARTNUMBER="m000065"  
      MARKFORDELETE="0"  
      BUYABLE="1"  
    />
```

```
    <catentry  
      CATENTRY_ID="10372"
```

```

MEMBER_ID="-2000"
PARTNUMBER="000072"
CATENTTYPE_ID="ProductBean"
MFPARTNUMBER="m000072"
MARKFORDELETE="0"
BUYABLE="1"
/>

<catgpenrel
  CATGROUP_ID="10354"
  CATENTRY_ID="10372"
  CATALOG_ID="10351"
  SEQUENCE="3"
/>

<catgpenrel
  CATGROUP_ID="10354"
  CATENTRY_ID="10365"
  CATALOG_ID="10351"
  SEQUENCE="4"
/>

<catgpenrel
  CATGROUP_ID="10354"
  CATENTRY_ID="10358"
  CATALOG_ID="10351"
  SEQUENCE="5"
/>

<catgpenrel
  CATGROUP_ID="10355"
  CATENTRY_ID="10372"
  CATALOG_ID="10351"
  SEQUENCE="3"
/>

</store-asset>

```

注：禁用“产品顾问搜索空间同步”可提供更好的装入程序性能；因此，请仅在需要时才使用此功能。

定制产品顾问搜索空间同步

装入程序软件包使您可以通过修改装入程序定制程序属性文件来定制“产品顾问搜索空间同步”以执行以下操作：

- **启用或禁用同步**

可通过在装入程序定制程序属性文件中指定 `true` 或 `false` 作为以下属性的值来启用或禁用同步：

```
PASyncEnabled = true
```

- **指定用于同步的配置信息文件**

可以通过在装入程序定制程序属性文件中设置以下属性值，来指定同步使用的 XML 配置信息文件。

```
PASyncDocumentURL = PASyncInfo.xml
```

- 指定调度查询长度

可以通过在装入程序定制程序属性文件中设置以下属性值，来指定调度查询的长度：

```
PAScheduleQueryLength = 30
```

此属性的值应在 20 到 900 的范围之内。

- 指定调度的开始时间

可以通过在装入程序定制程序属性文件中指定绝对时间戳记、当前时间戳记或当前时间戳记与持续时间作为 `PAScheduledStartTime` 属性的值，来指定调度的开始时间。

注：时间戳记的格式必须适用于您的数据库。

以下是将在装入后 5 分钟运行调度作业的 DB2 示例：

```
PAScheduledStartTime = CURRENT TIMESTAMP + 5 MINUTES
```

 以下是将立即运行作业的 Oracle 数据库的示例：

```
PAScheduledStartTime = SYSDATE
```

确定在使用装入程序时如何处理数据

装入程序为使用“装入”命令处理数据提供了以下选项：


- 装入
- 导入
- 使用 SQL 导入功能

在装入数据之前，您应当确定哪种处理的方法将产生最好的结果。






选择装入方法

在以下情况下请考虑使用装入方法：

- 如果您知道数据是干净的，并且数据库中不包含任何数据
- 如果您知道数据是干净的，并且知道数据库不包含正在装入的数据
- 如果您知道数据是干净的，目标表不包含任何主键，并且知道数据库不包含正在装入的数据
- 如果您主要关注的是装入时间
- 如果数据库是本地 DB2 数据库

 通过使用装入方法，数据装入数据库。如果数据已经存在，则命令将因为键重复错误而失败，并显示一条重复错误消息。

选择导入方法

     通过对 DB2 使用导入方法，数据也同样装入到数据库中。如果数据已经存在，数据不会被删除，但是将会用新值更新它。在以下任何情况下请考虑使用此方法：

- 如果数据库管理系统是 DB2
- 如果不知道数据是否是干净的

- 如果您不得不在列级别上更新较大的同类数据集
- 如果您主要关注的不是装入时间
- 如果导入数据的表具有主键

400 通过使用装入方法，数据也同样装入到数据库中。如果数据已经存在，数据不会被删除，但是将会用新值更新它。在以下任何情况下请考虑使用此方法：

- 如果不知道数据是否是干净的
- 如果数据在数据库中已经存在
- 如果您主要关注的不是装入时间
- 如果导入数据的表具有主键

选择 SQL 导入方法

使用 SQL 导入方法时，JDBC 或 SQL 语句用于向数据库更新或插入数据。如果数据还没有存在，则将插入数据，并且更新现有的数据。在以下任何情况下请考虑使用此方法：

- 如果您正在更新现有的数据并需要列级别的更新
使用此方法，对约束违反和数据类型错误有更好的错误报告。
- 如果您知道某些数据不是干净的
- 如果您主要关注的是数据库完整性
- 如果数据库不是本地的
- 如果正在使用产品顾问搜索空间同步

其它注意事项

- 关于使用装入方法的限制

装入方法不能插入或更新位数据字段中的数据。

DB2 通过使用此装入方法，只有新记录插入到数据库中；不更新现有记录。

DB2 装入方法只可以用于本地、非远程、DB2 数据库。

- 关于使用导入方法的限制

要使用导入方法，数据库管理系统必须是 DB2。

导入方法不能插入或更新位数据字段中的数据。

通过使用此导入方法，装入程序仅插入或更新定义了主键的表；导入方法不能在不具有主键的表中插入或更新数据。如果输入记录仅有主列的值，则将拒绝该记录。

- **SQL 导入方法与装入方法的比较**

SQL 导入方法检查数据一致性（包含外部引用）并允许您更新现有的数据。装入方法不会如此。

- **导入方法与 SQL 导入方法的比较**

导入方法和 SQL 导入方法执行相似的功能。通常情况下，导入方法更快但是需要磁盘空间来存储临时文件。

导入方法仅可插入或更新定义了主键的表；而 SQL 导入方法不要求表定义主键。

- **根据使用的数据库产品比较方法**

导入和装入方法使用为 DB2 优化过的本机实用程序，而 SQL 导入方法使用 JDBC 调用（此调用对许多数据库产品都是通用的）。

- **进一步考虑**

delete 方法用于从数据库中删除输入 XML 文档中的数据。此元素必须包含表主键或唯一索引的值。如果正在删除的数据与另一个表中的数据具有从属关系，并启用了“删除级联”，则也将删除从属数据。

如果正在使用“产品顾问搜索空间同步”，则必须使用 SQL 导入方法装入数据。

装入大文档

当使用装入程序软件包实用程序将大文档装入到数据库中时，请考虑以下项：

- **Java 虚拟机 (JVM) 堆大小**

缺省情况下，分配给 JVM 堆的最大内存数是 64 MB。如果不增加此内存，JVM 在装入过程期间最终可能内存不足。分配给 JVM 堆的最大内存数可通过使用 Java 命令中的 JVM -mx 选项进行改变。

- **跟踪记录**

跟踪记录程序在装入 XML 大文档时可能耗尽 JVM 堆。跟踪信息主要用于在运行失败时调试运行。如果跟踪装入过程不是必需的，则应当关闭跟踪。当关闭跟踪时，将有明显的性能改进。可通过修改记录配置 XML 文档来关闭跟踪。

缺省的记录配置文件是 WCALoggerConfig.xml。要关闭跟踪记录，请将装入程序的跟踪记录程序配置从：

```
<logger type="trace">
  <handler type="file">
    <filePath>MassLoadTrace.log</filePath>
    <filter type="Any">
      <messageType name="PUBLIC" />
    </filter>
  </handler>
</logger>
```

更改为：

```
<logger type="trace">
  <handler type="file">
    <filePath>MassLoadTrace.log</filePath>
    <filter type="Any">
      </filter>
  </handler>
</logger>
```

关于修改 WCALoggerConfig.xml 文件的更多信息，请参阅第 64 页的『为装入程序软件包定制记录』。

• 提交计数

在 SQL 导入方式下运作时装入程序的缺省提交计数是：1。因此，缺省情况下，每次对数据库进行更新或插入时都要提交事务。要改进装入程序对于大文档的性能，应当增加提交计数。建议值是“100”；但它还可以更高，这取决于服务器上的物理内存数、DBMS 事务日志大小等。

装入程序的提交计数是通过使用“装入”命令的 `-commitcount count` 选项来更改的（其中 `count` 是在提交事务之前执行的语句数）。

疑难解答技巧

如果装入数据时进展异乎寻常的缓慢，则装入程序的记录程序中可能存在没有正确配置的文件处理程序。这可能源于以下情况之一：



- 调用装入程序的用户没有许可权写入目录或更新记录配置文档中指定的文件。
- 在记录配置文档中指定为文件位置的目录不存在。
- 在记录配置文档中指定为文件位置的驱动器没有足够的空间。

当您更正任一此类问题时，您可能需要通过修改记录配置文档（缺省情况下为 WCALoggerConfig.xml）更改指定的文件位置。关于文件处理程序和 WCALoggerConfig.xml 文件的更多信息，请参阅第 64 页的『为装入程序软件包定制记录』。

第 7 章 抽取数据

要使用抽取程序从数据库中抽取数据，必须使用抽取过滤器文件来指定希望从数据库中抽取的数据。使用的抽取过滤器取决于希望抽取的数据类型。

以下示例使用 MemberSubsystemFilter.xml 作为抽取过滤器，从数据库中抽取成员子系统数据。

- 
- ```
java com.ibm.wca.MassExtract.Extract -filter MemberSubsystemFilter.xml
-outfile MemberSubsystemExtracted.xml -dbname mall -dbuser myname
-dbpwd mypassword -customizer MemberSubsystemCustomizer
```
- 
- ```
QWEBCOMM/EXTWCSDTA FILTER(MemberSubsystemFilter.xml)
OUTFILE(MemberSubsystemExtracted.xml) DATABASE(database_name)
SCHEMA(mall) INSTRROOT(/QIBM/UserData/WebCommerce/instances/msr)
PASSWD(mypassword)
```

创建抽取过滤器

以下抽取过滤器示例从 CATGROUP、CATGRPDESC、CATGRPREL、CATENTRY、CATENTSHIP、OFFER、CATENTREL、CATGPENREL、CATENTDESC 和 ATTRVALUE 表中抽取类别和产品信息：

```
<sqlx>
<!-- ***** -->
<!-- extract Category information -->
<!-- ***** -->

<functionDef id="Category" description="Extract Categories" schemaentity="catgroup">
  <paramDef name=":lastRecord" type="string" value="10301" description="Last record
before loading new data" />
  <body>
    select * from catgroup where catgroup_id > :lastRecord
  </body>
</functionDef>

<execute id="Category" description="Extract Categories" schemaentity="catgroup">
  <param name=":lastRecord" type="string" value="10300" description="Last record
before loading new data" />
</execute>

<functionDef id="Category Description" description="Extract Category Descriptions
for a Locale" schemaentity="catgrpdesc">
  <paramDef name=":lastRecord" type="string" value="10300" description="Last record
before loading new data" />
  <body>
    select * from catgrpdesc where catgroup_id > :lastRecord
  </body>
</functionDef>

<execute id="Category Description" description="Extract Category Descriptions
for a Locale" schemaentity="catgrpdesc">
  <param name=":lastRecord" type="string" value="10300" description="Last record
before loading new data" />
```

```

</execute>

<functionDef id="Category Relationship" description="Extract Category-Relations
for a Locale" schemaentity="catgrprel">
  <paramDef name=":lastRecord" type="string" value="10300" description="Last record
before loading new data" />
  <body>
    select * from catgrprel where catgroup_id_child > :lastRecord
  </body>
</functionDef>

<execute id="Category Relationship" description="Extract Category-Relations for
a Locale" schemaentity="catgrprel">
  <param name=":lastRecord" type="string" value="10300" description="Last record
before loading new data" />
</execute>

<!-- ***** -->
<!-- extract Product information -->
<!-- ***** -->

<functionDef id="Product" description="Extract Product" schemaentity="catentry">
  <paramDef name=":lastrecord" type="string" value="10300" description="Last record
before loading new data" />
  <body>
    select * from catentry where catentry_id > :lastrecord
  </body>
</functionDef>

<execute id="Product" description="Extract Product" schemaentity="catentry">
  <param name=":lastrecord" type="string" value="10300" description="Last record
before loading new data" />
</execute>

<functionDef id="Product Relationship" description="Extract Product Ship
information" schemaentity="catentrel">
  <paramDef name=":lastrecord" type="string" value="10300" description="Last record
before loading new data" />
  <body>
    select * from catentrel where catentry_id_child > :lastrecord
  </body>
</functionDef>

<execute id="Product Relationship" description="Extract Product Ship information"
schemaentity="catentrel">
  <param name=":lastrecord" type="string" value="10300" description="Last record
before loading new data" />
</execute>

<functionDef id="Product Description" description="Extract Product Description"
schemaentity="catentdesc">
  <paramDef name=":lastrecord" type="string" value="10300" description="Last record
before loading new data" />
  <body>
    select * from catentdesc where catentry_id > :lastrecord
  </body>
</functionDef>

<execute id="Product Description" description="Extract Product Description"
schemaentity="catentdesc">
  <param name=":lastrecord" type="string" value="10300" description="Last record
before loading new data" />
</execute>

<functionDef id="Product Ship" description="Extract Product Ship information"
schemaentity="catentship">
  <paramDef name=":lastrecord" type="string" value="10300" description="Last record
before loading new data" />
  <body>
    select * from catentship where catentry_id > :lastrecord
  </body>
</functionDef>

```

```

<execute id="Product Ship" description="Extract Product Ship information"
  schemaentity="catentship">
  <param name=":lastrecord" type="string" value="10300" description="Last record
  before loading new data" />
</execute>

<functionDef id="Category Product Relationship" description="Extract Category
Product Relations" schemaentity="catgpenrel">
  <paramDef name=":lastrecord" type="string" value="10300" description="Last record
  before loading new data" />
  <body>
    select * from catgpenrel where catgroup_id > :lastrecord
  </body>
</functionDef>

<execute id="Category Product Relationship" description="Extract Category Product
Relations" schemaentity="catgpenrel">
  <param name=":lastrecord" type="string" value="10300" description="Last record
  before loading new data" />
</execute>

<!-- ***** -->
<!-- Extract Product Attribute Information -->
<!-- ***** -->

<functionDef id="Product Attribute Values" description="Extract Product Attribute
values for a Locale" schemaentity="attrvalue">
  <paramDef name=":lastrecord" type="string" value="10300" description="Last record
  before loading new data" />
  <body>
    select * from attrvalue where catentry_id > :lastrecord
  </body>
</functionDef>

<execute id="Product Attribute Values" description="Extract Product Attribute values
for a Locale" schemaentity="attrvalue">
  <param name=":lastrecord" type="string" value="10300" description="Last record
  before loading new data" />
</execute>

<!-- ***** -->
<!-- Extract Product Price Information -->
<!-- ***** -->

<functionDef id="Offer" description="Extract Offer" schemaentity="offer">
  <paramDef name=":lastrecord" type="string" value="10300" description="Last record
  before loading new data" />
  <body>
    select * from offer where catentry_id > :lastrecord
  </body>
</functionDef>

<execute id="Offer" description="Extract Offer" schemaentity="offer">
  <param name=":lastrecord" type="string" value="10300" description="Last record
  before loading new data" />
</execute>

</sqlx>

```

设置抽取程序

可通过执行以下操作更改抽取程序使用的数据库驱动程序:

1. 创建新的抽取程序定制程序属性文件。



- `DB2ConnectionCustomizer.properties` 位于 `MassExtract.zip` 压缩文档中。将此文件解压缩并重命名, 但保留 `.properties` 扩展名, 并将它放置在类路径中的目录中。**重要信息:** 请不要除去或修改现有的 `DB2ConnectionCustomizer.properties` 文件。



- `ISeries_EXTWCSDTA_Customizer.properties` 位于 `/QIBM/ProdData/WebCommerce/properties` 目录中。将此文件复制到 `/instroot/xml` 目录, 重命名新文件但保留 `.properties` 扩展名, 然后对新文件进行任何必要的更改。**重要信息:** 请不要除去或修改原始的 `ISeries_EXTWCSDTA_Customizer.properties` 文件。

2. 修改新文件中的数据库驱动程序值。
3. 将新文件名指定为“抽取”命令的 `customizer` 参数的值。

以下是抽取程序定制程序属性文件中的摘录:

```
DBVendorName = DB2
DBDriverName = COM.ibm.db2.jdbc.app.DB2Driver
DBURL = jdbc:db2:
```

其中:

- `DBVendorName` 用于选择数据库的类型。
选项如下:
 - DB2 通用数据库 iSeries 版 (`DB2/iSeries`)
 - 用于其它操作系统的 DB2 (`DB2`)
 - Oracle 数据库 (`Oracle`)
- `DBDriverName` 用于选择 JDBC 驱动程序。
选项如下:
 - DB2 通用数据库 iSeries 版 (`com.ibm.db2.jdbc.app.DB2Driver`)
 - 用于其它操作系统的 DB2 (`COM.ibm.db2.jdbc.app.DB2Driver`)
 - Oracle 数据库 (`oracle.jdbc.driver.OracleDriver`)
- `DBURL` 用于指定访问数据库的 URL。
选项如下:
 - DB2 通用数据库 iSeries 版 (`jdbc:db2://`)
 - 用于其它操作系统的 DB2 (`jdbc:db2:`)
 - Oracle 数据库 (`jdbc:oracle:oci8:@`)

第 8 章 使用装入程序软件包记录程序

装入程序软件包中的每个实用程序都创建消息，以提示成功、失败和错误，并提供程序跟踪信息。

装入程序软件包中的实用程序引用 WCALoggerConfig.xml 文件。

-  NT 2000 AIX Solaris Linux

此文件存在于在类路径系统环境变量中指定的路径中。它也可以由 com.ibm.wca.logging.configFile Java 系统属性指定。

-  400

此文件位于 */instroot/xml* 目录。

WCALoggerConfig.xml 确定每个实用程序提供哪些记录信息以及显示或存储信息的位置。可以定制此文件并指定应创建何种类型的日志，以及记录何种类型的消息。

在您的环境中为 Windows NT、Windows 2000、AIX、Linux 和 Solaris 系统配置记录。

要在您的环境中设置记录，必须设置类路径系统环境变量来包含文件 WCALoggerConfig.xml 或指定 com.ibm.wca.logging.configFile 系统属性。

设置类路径变量的示例

例如，如果 WCALoggerConfig.xml 文件位于 Windows NT 机器上的目录 d:\WebSphere\CommerceServer\xml\loader 中，则可以使用以下语句设置类路径变量：

```
SET CLASSPATH=%CLASSPATH%;D:\WebSphere\CommerceServer\xml\loader
```

指定 com.ibm.wca.logging.configFile 系统属性的示例

要指定 com.ibm.wca.logging.configFile 系统属性，请在调用 Java 解释器时使用 -D 选项。以下是一个示例：

```
java -Dcom.ibm.wca.logging.configFile=D:\ice_tea\src\classlib\logger\xml\WC.xml  
com.ibm.wca.DTDGenerator.GeneratedDTD
```

为装入程序软件包定制记录

要为装入程序软件包定制记录，请使用 WCALoggerConfig.xml 文件。

-     







此文件存在于在类路径系统环境变量中指定的路径中。它也可以由 com.ibm.wca.logging.configFile Java 系统属性指定。

- 

此文件位于 */instroot/xml* 目录。

WCALoggerConfig.xml 包含一个或多个 component 标记，例如 <component name="DTDGenerator">。您可以在每个这些标记中，添加 logger 和 handler。您不应当改变系统提供的 utility 和 logger 标记，但是可以将 handler 标记添加到 logger。关于在此文件中可包含的内容的信息，请参阅 WCALogger.dtd 文件。

装入程序软件包日志位于以下目录的 messages.txt 文件中：

-  *drive:\WebSphere\CommerceServer\instances\instance_name\logs*
-  *drive:\Program Files\WebSphere\CommerceServer\instances\instance_name\logs*
-  */usr/WebSphere/CommerceServer/instances/instance_name/logs*
-   */opt/WebSphere/CommerceServer/instances/instance_name/logs*
-  */QIBM/UserData/WebCommerce/instances/instance_name/logs*

处理程序

要将处理程序添加到记录程序，请在 WCALoggerConfig.xml 文件中指定处理程序类型。您可以将一个或多个处理程序添加到记录程序。注意，每个处理程序都有它自己的属性和下级标记，这些都不必应用到其它处理程序。处理程序类型包含以下类型：

处理程序类型	描述和属性
console	将消息发送到标准输出（通常情况下是命令行）
file	在文本文件中存储消息 必须将 "<filePath>log_path</filePath>" 添加到此处理程序以作为下级标记。
multifile	创建文件的循环记录 必须指定 "<filePath>log_path</filePath>"。将创建日志文件 1 到 <i>n</i> 。 您可以添加以下属性： MaxFiles 整数，它表示在擦除第一个日志文件之前使用多少日志文件 MaxKBFileSize 整数，它表示每个日志文件中可以存储的最大千字节数
     database	以循环记录的方式将消息存储在 DB2 表中。您可以添加以下属性： brand 数据库品牌名称。DB2 是当前支持的唯一的数据库。 maxRows 在擦除最旧的条目之前，表中存储的最大记录条数 您可以包含 "<jdbc/>" 作为下级标记并包含以下属性： url 在 JDBC 中用来访问数据库的 URL（例如，"Jdbc:db2:"wcm"，其中 "wcm" 是数据库的名称）。数据库必须在运行实用程序之前就存在。 table 将在其中记录消息的数据库表的名称。它必须用以下 DB2 语句来创建： "CREATE TABLE"tablename (KEY char(13) FOR BIT DATA NOT NULL, COMPONENTNAME VARCHAR(30), ENTRY VARCHAR(2000), PRIMARY KEY(key))" userid 数据库用户名。必须给用户指定许可权来更新表。以下 DB2 语句将实现此操作： "GRANT SELECT, UPDATE, INSERT, DELETE ON TABLE"tablename TO USER userid" password 指定的用户名的数据库密码

以下示例将类型“database”的处理程序添加到记录程序：

```
<handler type="database" brand="DB2" maxRows="50">
  <jdbc url="jdbc:db2:wcm"
    table="wcm.log"
    userid="wasuser"
    password="123456"/>
  <filter type="Any">
    <messageType name="FATAL"/>
  </filter>
</handler>
```

```

        <messageType name="ERROR"/>
        <messageType name="WARNING"/>
    </filter>
</handler>

```

过滤器

可将过滤器添加到处理程序或从处理程序中将其除去，以包含或排除消息类型。如果记录程序没有过滤器，将不记录任何消息。每一过滤器标记都有列出消息类型的下级 `messageType` 标记，通常这些消息类型是以下类型之一：

- INFO
- ERROR
- FATAL
- WARNING

其它消息类型列于 `WCALogger.dtd` 文件中，但是一般而言，其中的大部分类型都不与装入程序软件包一起使用。

过滤器类型包含以下类型：

过滤器类型	描述和属性
Any	在日志文件中包含标志为指定的 <code>messageType</code> 类型之一的任何消息 例如，如果 <code>messageType</code> 列表包含 <code>ERROR</code> 并且应用程序生成 <code>ERROR</code> 类型消息，则记录此消息。
All	要求消息在包含在日志中之前必须具有所有指定的 <code>messageType</code> 类型属性
Exclude	记录 <code>messageType</code> 标记列表中没有指定的所有消息

以下将过滤器添加到处理程序的示例允许记录 `FATAL` 和 `ERROR` 消息类型，而其它消息将被忽略：

```

<handler type="file">
  <filter type="Any">
    <messageType name="FATAL"/>
    <messageType name="ERROR"/>
  </filter>
</handler>

```

格式

您可以为消息格式化指定两种格式化程序类型中的一种：

格式化程序类型	描述和属性
safe (缺省)	如果在属性文件中找不到消息，将阻止设置异常 此格式化程序创建消息来指示资源丢失。
xml	按 XML 格式来格式化消息 如果无法找到消息，则此格式化程序也将写一条消息而不是设置异常。

示例: WCALoggerConfig.xml 和 WCALogger.dtd

WCALoggerConfig.xml

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE WCALoggerConfig SYSTEM "WCALogger.dtd">
<WCALoggerConfig>
  <component name="MassLoader">
    <logger type="message">
      <handler type="file">
        <filePath>c:\temp\out.txt</filePath>
        <filter type="Any">
          <messageType name="FATAL"/>
          <messageType name="ERROR"/>
          <messageType name="WARNING"/>
          <messageType name="INFO"/>
        </filter>
      </handler>
    </logger>
    <logger type="trace">
      <handler type="file">
        <filePath>out2.txt</filePath>
        <filter type="Any">
          <messageType name="PUBLIC"/>
        </filter>
      </handler>
    </logger>
    <logger type="typedMessage">
      <handler type="file">
        <filePath>tout.txt</filePath>
        <filter type="Any">
          <messageType name="FATAL"/>
          <messageType name="ERROR"/>
          <messageType name="WARNING"/>
          <messageType name="INFO"/>
        </filter>
      </handler>
    </logger>
    <logger type="progress">
      <handler type="console" format="safe">
        <filter type="Any">
          <messageType name="FATAL"/>
          <messageType name="ERROR"/>
          <messageType name="WARNING"/>
          <messageType name="INFO"/>
        </filter>
      </handler>
    </logger>
  </component>
  <component name="DTDGenerator">
    <logger type="message">
      <handler type="console">
        <filter type="Any">
          <messageType name="FATAL"/>
          <messageType name="ERROR"/>
          <messageType name="WARNING"/>
          <messageType name="INFO"/>
        </filter>
      </handler>
    </logger>
    <logger type="trace">
      <handler type="console">
        <filter type="Any">
          <messageType name="FATAL"/>
          <messageType name="ERROR"/>
        </filter>
      </handler>
    </logger>
  </component>

```

```

        <messageType name="WARNING"/>
        <messageType name="INFO"/>
    </filter>
</handler>
</logger>
</component>
</WCALoggerConfig>

```

WCALogger.dtd

```

<!-- This DTD describes how a WCALoggerConfig XML can be structured.
A WCALoggerConfig XML document is the input configuration file for
the WCALoggerFactory class.
-->

<!ELEMENT WCALoggerConfig (component)+>

<!ELEMENT component (logger)+>
<!ATTLIST component name CDATA #REQUIRED>
<!ELEMENT logger (handler+,messageFile?)>
<!ATTLIST logger type (message | trace | typedMessage | progress) "typedMessage">

<!-- messageFile is an optional default properties files that can be used to
make messages locale specific
-->
<!ELEMENT messageFile (#PCDATA)>
<!ELEMENT handler (filePath?, filter, jdbc?)>
<!ATTLIST handler
type ( file|multiFile|console|error|textArea|database|ejbQueue|queue ) "console">

<!-- maxFiles & maxKBFileSize only applies to the multiFile type of handler
-->
<!-- filePath & encoding applies only when the handler is of type file or
multiFile
-->
<!ATTLIST handler maxFiles CDATA #IMPLIED>
<!ATTLIST handler maxKBFileSize CDATA #IMPLIED>
<!ATTLIST handler encoding CDATA #IMPLIED>
<!ATTLIST handler format (safe | xml) "safe">
<!-- maxRecords & brand are only applicable to database handler type
-->
<!ATTLIST handler maxRecords CDATA #IMPLIED>
<!ATTLIST handler brand (DB2) #IMPLIED>
<!-- the jdbc tag must be present within a database handler type tag
-->
<!ELEMENT jdbc EMPTY>
<!ATTLIST jdbc url CDATA #IMPLIED>
<!ATTLIST jdbc table CDATA #IMPLIED>
<!ATTLIST jdbc userid CDATA #IMPLIED>
<!ATTLIST jdbc password CDATA #IMPLIED>

<!ELEMENT filter (messageType+)>
<!ATTLIST filter type (Any | All | Exclude ) "Any">

<!-- the messageType attribute name is one of these JLog IRecordType
constants
-->
<!ELEMENT messageType EMPTY>
<!ATTLIST messageType name ( NONE | ALL | INFO |
INFORMATION | WARN | WARNING | ERR | ERROR |
FATAL | DEFAULT_MESSAGE | API | CALLBACK |
ENTRY_EXIT | ENTRY | EXIT | ERROR_EXC |
MISC_DATA | OBJ_CREATE | OBJ_DELETE |
PRIVATE | PUBLIC | STATIC | SVC | PERF |
LEVEL1 | LEVEL2 | LEVEL3 ) "ALL">
<!ELEMENT filePath (#PCDATA)>

```

第 9 章 使用装入程序软件包错误报告程序

装入程序和标识解析器包含一个错误报告程序，如果发生错误此程序将生成异常文档。

缺省情况下，异常文档将写入以下目录：

-      输入文档所在目录
-  `/instroot/logs`

要指定异常文档写入哪个目录，请使用 Java 属性

`com.ibm.wcm.ErrorReporterDir`。Windows NT 环境中用于装入程序的一个示例按如下方式开始：

```
java -Dcom.ibm.wcm.ErrorReporterDir=d:\massloaderrors  
com.ibm.wca.MassLoader.MassLoad -dbname . . .
```

注意： 用户应当有权写入指定的目录。

以下是错误报告程序的示例 DTD (`store-all-error.dtd`)：

```
<!ENTITY % TABLE "calrule | catentry">  
<!ELEMENT store-asset (error, (%TABLE;)*)>  
<!ELEMENT message (#PCDATA) >  
<!ELEMENT error ( message ) >  
<!ATTLIST error  
  locus          CDATA      #REQUIRED  
  id             CDATA      #REQUIRED  
>  
<!ELEMENT calrule (error)>  
<!ATTLIST calrule  
  identifier      CDATA      #REQUIRED  
  calrule_id     CDATA      #REQUIRED  
  calcode_id     CDATA      #REQUIRED  
  startdate      CDATA      #IMPLIED  
  taxcgr_id      CDATA      #IMPLIED  
  enddate        CDATA      #IMPLIED  
  sequence       CDATA      #REQUIRED  
  combination    CDATA      #REQUIRED  
  calmethod_id   CDATA      #REQUIRED  
  calmethod_id_qfy CDATA      #REQUIRED  
  flags          CDATA      #REQUIRED  
  field1         CDATA      #IMPLIED  
  field2         CDATA      #IMPLIED  
>  
<!ELEMENT catentry (error)>  
<!ATTLIST catentry  
  catentry_id    CDATA      #REQUIRED  
  member_id     CDATA      #REQUIRED  
  catenttype_id  CDATA      #REQUIRED  
  partnumber     CDATA      #IMPLIED  
  mfpartment     CDATA      #IMPLIED  
  mfname         CDATA      #IMPLIED  
  markfordelete  CDATA      #REQUIRED  
  url            CDATA      #IMPLIED  
  field1         CDATA      #IMPLIED  
  field2         CDATA      #IMPLIED  
  lastupdate    CDATA      #IMPLIED
```

```

field3          CDATA    #IMPLIED
onspecial       CDATA    #IMPLIED
onauction       CDATA    #IMPLIED
field4          CDATA    #IMPLIED
field5          CDATA    #IMPLIED
buyable         CDATA    #IMPLIED
>

```

以下是来自装入程序的样本错误报告文档:

```

<?xml version="1.0"?>
<!DOCTYPE store-asset SYSTEM "store-all-error.dtd">
<store-asset>
  <error
    locus="Parser"
    id="SAXParseFatalError" >
    <message>
      Error The string "--" is not permitted within comments. : 155 : 18
    </message>
  </error>
  <calrule
    calcode_id="30"
    enddate="2100-01 10:20:30.000000"
    calmethod_id="-47"
    identifier="7"
    taxcgry_id="9"
    calmethod_id_qfy="-46"
    startdate="1900-01-01-00.00.00.000000"
    flags="1"
    combination="2"
    calrule_id="44"
    sequence="9.0E+1">
    <error
      locus="Writer"
      id="SQLException" >
      <message>
        A SQL Exception was received [IBM][CLI Driver][DB2/NT] SQL0530N
        The insert or update value of the FOREIGN KEY
        "JANTONY.CALRULE.F_CALRULE4" is not equal to any value of the
        parent key of the parent table. SQLSTATE=23503
      </message>
    </error>
  </calrule>
  <catentry
    catentry_id="10118"
    member_id="-2001"
    partnumber="1254"
    mfpartnumber="sku-163"
    mfname="InFashion"
    markfordelete="0"
    buyable="1"
    field1="abc" >
    <error
      locus="Formatter"
      id="FormattingError" >
      <message>
        Error when formatting value for CATENTRY.FIELD1 : abc with error
        [class java.lang.NumberFormatException(abc)].
      </message>
    </error>
  </catentry>
</store-asset>

```

第 10 章 配置装入程序软件包命令和脚本

要启动装入程序软件包并运行其命令，请使用 WebSphere Commerce 目录中提供的脚本或命令：

-  `drive:\WebSphere\CommerceServer\bin`
-  `drive:\Program Files\WebSphere\CommerceServer\bin`
-  `/usr/WebSphere/CommerceServer/bin`
-   `/opt/WebSphere/CommerceServer/bin`
-  QWEBCOMM 本机库

脚本和命令如下所示：

 **400**

GENWCSDTD

DTD 生成命令

RESWC SID

标识解析命令

EXTWCSDTA

抽取命令

LODWCSDTA

装入命令

TRNWCSTXT

文本转换命令

TRNWC XML

XML 转换命令

 **NT**  **2000**

dttdgen.cmd

DTD 生成命令

idresgen.cmd

标识解析命令

massextract.cmd

抽取命令

massload.cmd

装入命令

txttransform.cmd

文本转换命令

xmltransform.cmd

XML 转换命令

**dtdgen.sh**

DTD 生成外壳程序脚本

idresgen.sh

标识解析外壳程序脚本

massextract.sh

抽取外壳程序脚本

massload.sh

装入外壳程序脚本

txttransform.sh

文本转换外壳程序脚本

xmltransform.sh

XML 转换外壳程序脚本

第 3 部分 使用 Web Editor

本部分描述如何管理和使用 Catalog Manager Web Editor。







Web Editor 使您能通过 Web 浏览器创建、删除和更改产品目录数据。用于查看和更新信息的数据输入表单是 Web Editor 的主要功能。最简单的情况就是，表单与 WebSphere Commerce 数据库中的表相对应。管理员可以选择使用所提供的缺省表单，或者定制可用的表单。

注： Web Editor 使用 Internet Explorer 5 或更新的版本。

第 11 章 设置 Web Editor




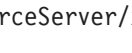
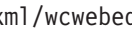
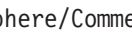
管理员可以配置 Web Editor 以向 WebSphere Commerce 提供数据，包括对 WebSphere Commerce 模式的扩展和定制。Web Editor 并非设计为特定于 WebSphere Commerce 的特定实例的一组 Web 输入表单。Web Editor 的设计目的是使其灵活并且可定制，以支持不同组织的个人需要和角色。

数据库视图是已存储的对数据库中一个或多个表的查询。在 Catalog Manager 安装期间，在以下目录的每个特定于系统的子目录 (db2、oracle 和 os400) 中保存了一个用于创建 WebSphere Commerce 产品逻辑视图的样本文件：

-  *drive:\WebSphere\CommerceServer\schema*
-  *drive:\Program Files\WebSphere\CommerceServer\schema*
-  */usr/WebSphere/CommerceServer/schema*
-   */opt/WebSphere/CommerceServer/schema*
-  */QIBM/ProdData/WebCommerce/schema*

此样本文件 *wcs.view.sql* 包含一个产品视图，创建该产品视图用于组合多个表中有关产品的信息。此文件包含产品视图的 SQL 数据定义。管理员可以研究此文件以计划开发他们自己的数据库视图。

缺省的 XML 表单描述文件 (*forms51_be.xml*) 包含为使在 WebSphere Commerce 数据库中添加、编辑、删除数据更容易而设计的表单。此文件的副本位于以下目录中：

-  *drive:\WebSphere\CommerceServer\xml\wcwebeditor\xml*
-  *drive:\Program Files\WebSphere\CommerceServer\xml\wcwebeditor\xml*
-  */usr/WebSphere/CommerceServer/xml/wcwebeditor/xml*
-   */opt/WebSphere/CommerceServer/xml/wcwebeditor/xml*
-  */instroot/xml/wcwebeditor/xml*

此配置文件可以按现状使用，或者由设置 Web Editor 的管理员进行更改或增强。要定制 XML 表单描述文件，请参阅本部分随后包含的指导。

注：DTD 生成程序可以自动创建 Web Editor 将使用的表单。

配置 Web Editor

本部分提供关于如何配置 Web Editor 的信息。尽管安装过程一开始已进行了配置，但是管理员可以使用此信息进行某些其它操作，例如重新配置 Web Editor 以便使用另一数据库。

编辑 webeditor.properties 文件

Web Editor 具有某些在 webeditor.properties 文件中设置的应用程序参数。该文件位于以下目录:

-  `drive:\WebSphere\AppServer\installedApps\WC_Enterprise_App_demo.ear\wcwebeditor.war\WEB-INF\classes\webeditor.properties`
-  `drive:\Program Files\WebSphere\AppServer\installedApps\WC_Enterprise_App_demo.ear\wcwebeditor.war\WEB-INF\classes\webeditor.properties`
-  `/usr/WebSphere/AppServer/installedApps/WC_Enterprise_App_demo.ear\wcwebeditor.war\WEB-INF\classes\webeditor.properties`
-   `/opt/WebSphere/AppServer/installedApps/WC_Enterprise_App_demo.ear\wcwebeditor.war\WEB-INF\classes\webeditor.properties`
-  `/QIBM/UserData/WEBASADV4/was_instance_name\installedApps\WC_Enterprise_App_wcs_instance_name.ear\wcwebeditor.war\WEB-INF\classes`

通过编辑 webeditor.properties 文件，管理员可以执行某些操作，例如更改用于描述在 Web Editor 中显示的表单的文件。以下是 webeditor.properties 文件内容的示例:

```
# Properties file for WebEditor

(The following specifies where the customized process list and the Catalog Manager
utility configuration envelopes are located.)
# URI Location of Process and WCM Subsystem configuration envelopes
ProcessConfigFile=file:///D:/WebSphere/CommerceServer/xml/wcwebeditor/xml/weProcessList.xml

(The following specifies where the forms are defined.)
# Location of Forms file
FormsURL=file:///D:/WebSphere/CommerceServer/xml/wcwebeditor/xml/forms51_be.xml

(The following setting specifies where the XML-to-HTML style sheet is located)
# Location of XML to HTML StyleSheet
StyleSheetURI=file:///D:/WebSphere/CommerceServer/xml/wcwebeditor/xsl/webeditor.xsl

(The following setting specifies the location for temporary files)
# Location of Temporary Directory
#temp.dir=

# WebSphere datasource. This is used to build form drop-down lists and field
# default values. However, when publishing and searching, the Web Editor utilizes
# the WCM subsystems ID Resolver, Mass Loader, and Mass Extractor. Database access
# for these WCM subsystems must be configured separately.
# Name of WAS database source
dbsource=jdbc/WebSphere Commerce DB2 DataSource demo

# Name of WAS database. If specified, this takes preference over the database name
# value in the forms XML file. This value will be utilized as a parameter when
# invoking the WCM subsystems such as the Mass Loader.
dbname=all

(The following setting specifies the character set to use. By default, it is
set to the single-byte character set. For other countries, choose from the values
that are commented out.)
```

```

# Encoding Character Set
Encoding=ISO-8859-1

# CN
#Encoding=gb2312
# TW
#Encoding=Big5
# KR
#Encoding=EUC-KR
# JP
#Encoding=Shift_JIS

(The following setting is used to allow images to be previewed. The hostname of
the server plus any leading directory information before the image information that
is stored in WebSphere Commerce should be specified here. The WebSphere Commerce
catalog information is appended to this value to construct an image URL.)
# Specifies the base href location for images
# This should be set so that this info plus the info stored in WCS
# (e.g. /image/char.gif) combines to create a URL to an image

#imageRootURL=http://%HOSTDOMAIN
imageRootURL=http://localhost/webeditor

(The following setting is used to set a date format for the application. If these
properties are not available (i.e., they are commented out), then the Java-locale
specific format will be used. Below is a reference table for setting these values.)
# Use these properties to specify a date format if the Java locale-specific
# format is not desired

#dateFormat=yyyy-MM-dd
#dateTimeFormat=yyyy-MM-dd HH:mm:ss

# Symbol Meaning                Presentation                Example
# -----
# G      era designator         (Text)                      AD
# y      year                   (Number)                    1996
# M      month in year          (Text & Number)             July & 07
# d      day in month           (Number)                    10
# h      hour in am/pm          (1~12) (Number)            12
# H      hour in day            (0~23) (Number)            0
# m      minute in hour         (Number)                    30
# s      second in minute       (Number)                    55
# S      millisecond            (Number)                    978
# E      day in week            (Text)                      Tuesday
# D      day in year            (Number)                    189
# F      day of week in month   (Number)                    2 (2nd Wed in July)
# w      week in year           (Number)                    27
# W      week in month          (Number)                    2
# a      am/pm marker           (Text)                      PM
# k      hour in day            (1~24) (Number)            24
# K      hour in am/pm          (0~11) (Number)            0
# z      time zone              (Text)                      Pacific Standard Time
# '      escape for text        (Delimiter)
# ''     single quote           (Literal)                    '

```







更改临时文件的位置

通过除去注释标记并将值添加到 `webeditor.properties` 文件中的 `temp.dir` 属性，可以更改临时文件的位置。

此外，`java.io.tmpdir` Java 属性用于确定创建临时文件的位置。

使用 DTD 生成程序创建 XML 表单描述文件

forms51_be.xml 是 XML 表单描述文件的示例。它提供 Web Editor 使用的一组表单。此文件的副本位于以下目录中：

-  `drive:\WebSphere\CommerceServer\xml\wcwebeditor\xml`
-  `drive:\Program Files\WebSphere\CommerceServer\xml\wcwebeditor\xml`
-  `/usr/WebSphere/CommerceServer/xml/wcwebeditor/xml`
-   `/opt/WebSphere/CommerceServer/xml/wcwebeditor/xml`
-  `/instroot/xml/wcwebeditor/xml`

以下步骤描述系统管理员可以如何使用 DTD 生成程序添加新的 XML 表单。

注：在执行以下描述的过程之前，请重命名 Web Editor 目录中现有的 forms51_be.xml。此外，在以下过程用新名称创建输出文件，然后重新配置 Web Editor 以便使用新创建的文件。关于执行此操作的指导，请参阅前一部分。

要创建 XML 表单，请执行 DTD 生成命令。

以下步骤描述如何创建 XML 表单：

1. 创建名称为“tables.txt”的临时文件，该文件包含您想在表单中使用的表的名称。在每行上输入一个表名，如以下示例所示：

```
catentry
catentdesc
catentship
inventory
```

2. 将 tables.txt 保存到 DTD 生成命令所在的目录。（请参阅第 71 页的第 10 章，『配置装入程序软件包命令和脚本』获取此命令的安装位置。）
3. 在操作系统命令提示符下，切换到 DTD 生成命令所在的目录。

4. 输入如下语句运行 DTD 生成命令:

▶ NT ▶ 2000

```
dtdgen -infile tables.txt -outfile tables51.dtd
-dbname dbname -dbuser userid -dbpwd password
-xmlTableDesc tableFORMS.xml -schemaname schema -propfile filename
```

▶ AIX ▶ Solaris ▶ Linux

```
./dtdgen.sh -infile tables.txt -outfile tables51.dtd
-dbname dbname -dbuser userid -dbpwd password
-xmlTableDesc tableFORMS.xml -schemaname schema -propfile filename
```

▶ 400

```
QWEBCOMM/GENWCSDTD DATABASE(database) SCHEMA(schema)
INSTROOT(instroot) PASSWD(password) OUTFILE(tables51.dtd)
INFILE(tables.txt) XMLTABDESC(tableFORMS.xml)
```

表描述开关 (-xmlTableDesc 或 XMLTABDESC) 使得 DTD 生成程序在创建 DTD 之外, 还创建表的新表单描述。

▶ NT ▶ 2000 ▶ AIX ▶ Solaris ▶ Linux

-propFile 选项指定外部

属性文件的名称, 帮助文本、缺省值和字段描述信息可以存储在此文件中。

5. 如前一部分所述重新配置 Web Editor 以便使用新创建的文件。
6. 在 WebSphere 高级管理控制台中重新启动 Web Editor。要执行此操作, 请遵循以下步骤:
 - a. 展开 **WebSphere 管理域**。
 - b. 展开**企业应用程序**。
 - c. 用鼠标右键单击 **WebSphere Commerce 企业应用程序 — demo**, 选择**停止**。
 - d. 请等待, 直到消息指示应用程序已经停止。
 - e. 用鼠标右键单击 **WebSphere Commerce 企业应用程序 — demo**, 选择**启动**。
 - f. 请等待, 直到消息指示企业应用程序已经启动。
7. 要在 Web 浏览器中查看新表单, 请打开以下 URL:

`https://host_name:8000/wcm/webeditor`

其中 *host_name* 是 WebSphere Application Server 的全限定 HTTP 主机名。

Web Editor 在浏览器窗口显示所有表名的列表。

定制 XML 表单描述

本部分描述管理员如何能够增强 Web Editor 显示的表单。

通过在 XML 表单描述文件本身或在独立的属性文件中设置属性和值，可以定制和增强 XML 表单描述。

注：此属性文件的名称应当指定为 formList 标记中的 resourcePackage 属性值。如果该文件名在类路径中的目录的子目录中出现，则该文件名应当使用数据包（点）规范。

下表列出了管理员可以更改的 Web Editor 的表单字段属性。

字段属性	描述
Currency	使得以特定于语言环境的数字分隔符显示值（例如，对于美国，逗号用作千位分隔符）
DbColumn	用于将字段名映射到属性文件键 如果正在使用特定于语言环境的属性文件，则此条目中的值必须与在属性文件中输入的值相匹配。DTD 生成程序在此条目后附加一个模式。
DefaultValue	指定用户填写新表单时数据输入表单上显示的值 此属性可以由 DTD 生成程序设置为数据库缺省值。该值可以是静态字符串，但也可以包含一个对单行表的 SQL 标量查询。要检索外部数据，可以在此查询中使用用户定义的函数。例如： DefaultValue="SELECT CURRENT TIMETAMP FROM EXEC" 其中 EXEC 如下定义和填充： CREATE TABLE EXEC (A CHAR(1)); INSERT INTO A VALUES('A');
dynamicSqlSelectionList	使为每一表单重新构建下拉菜单
FieldDescription	提供在表单上的输入字段旁显示的描述 DTD 生成程序在创建此属性时使用列上的注释（如果有的话）。如果列上没有注释，则缺省值是列名。 此属性可以在特定于语言环境的属性文件或在 XML 表单描述文件中设置。如果在属性文件中指定了值，则该值优先。

字段属性	描述
FieldHelp	<p>当表单上的字段在焦点中时，将提供字段的简要帮助描述，该描述在浏览器的底部消息栏上显示。</p> <p>缺省情况下，这包含提示输入给定字段的数据以及数据的列类型的消息。</p> <p>此属性可以在特定于语言环境的属性文件或在 XML 表单描述文件中设置。如果在属性文件中指定了值，则该值优先。</p>
formatNumber	<p>用于告诉 Web Editor 无论如何不要处理数字</p> <p>将 formatNumber 属性设置为 “false” 将输入的值视为字符串，在抽取查询期间（此时值不象字符串那样加引号）除外。此属性的缺省值是 “true”。</p>
Hidden	指示不在表单上显示值，但是作为 HTML 隐藏字段仍然是可用的
HideOnCreate	<p>指示在编辑新表单时字段是可用的</p> <p>类似于 showInCreateMode=“false”；但是将字段名作为隐藏类型添加。</p>
Maxlength	<p>指定数据库列的长度</p> <p>用于确保用户不会输入长度大于数据库可存储长度的值。</p>
MinOccurs	<p>指示字段是否是必需的</p> <p>值 “1” 表示字段是必需的，值 “0” 表示字段是可选的。</p>
Name	指定数据库列的名称。
readOnly readOnlyForCreate readOnlyForEdit	<p>控制用户何时可以编辑字段以及何时只能读取字段</p> <p>readOnly=“true” 表示字段总是处于只读方式。 readOnlyForCreate=“true” 表示在编辑新表单时字段处于只读方式。 readOnlyForEdit=“true” 表示在编辑现有表单时字段处于只读方式。</p>
ShowColumnInList	<p>指定字段在设置为 “true” 时是构成数据的多记录视图的列之一</p> <p>DTD 生成程序将前六列设置为 “true”。</p>

字段属性	描述
showInCreateMode	<p>用于向创建的表单隐藏字段</p> <p>将 showInCreateMode 属性设置为 “false” 以在所显示的已创建的表单上隐藏字段。此属性的缺省值是 “true”。</p>
ShowInSearchMode	<p>用于向搜索条件页面隐藏某些列</p> <p>如果此属性设置为 “false”，则给定字段将不在搜索表单上显示。</p>
SqlSelectionList	<p>创建用户在编辑新表单时可以从中选择的下拉菜单</p> <p>查询必须返回一列或二列的结果集。第一列是用户可以从选择的标签的列表；第二列是将存储在数据库表中的实际值的列表。如果只使用一列，则所显示的值将存储在表中。此功能对增强外键关系非常有帮助。</p>
Type	<p>指示数据库列的类型以及应对数据执行什么类型的验证检查</p> <p>例如，如果类型是 “integer”，则应用程序确保只在此字段中输入有效的整数。</p>
UniqueKey	<p>指示字段是表的主键</p> <p>如果此属性设置为 “true”，则应用程序对输入此列的值实施唯一约束。如果不对数据库中的记录扩展，则只对装入到应用程序的记录执行此检查。</p>
ValidateInput	<p>用于关闭验证检查</p> <p>此属性允许用户在数字字段中输入将由标识解析器或 XML 转换程序处理的文本。</p>

编辑表单名称

XML 表单描述中的 form 标记同时具有 name 和 displayName 属性。

- name 属性必须设置为它所代表的数据库表或视图的名称。
- displayName 属性提供在 Web Editor 应用程序中使用方便的名称。此外，表单的 displayName 属性可以在 formList 标记的 resourcePackage 属性中指定的属性文件中设置。（resourcePackage 属性用于多语言支持。）

注： 如果条目在属性文件中出现，则它优先于在 XML 表单描述文件中输入的条目。

更改字段描述

以下是 XML 表单描述文件中的字段描述的示例:

```
<form name = "CATALOG.CAENTRY"
.
.
<field name="MEMBER_ID"
  showInCreateMode="false"
  fieldDescription="MEMBER_ID"
  type="integer"
  maxLength="19"
  defaultValue=""
.
.
<field name="CATENTTYPE_ID"
  fieldDescription="CATENTTYPE_ID"
  type="string"
  maxLength="16"
  defaultValue=""
.
.
```

您可能想将其更改为:

```
<form name = "CATALOG.CAENTRY"
  displayName="Product"
.
.
<field name="MEMBER_ID"
  showInCreateMode="false"
  fieldDescription="Member Identifier"
  type="integer"
  maxLength="19"
  defaultValue=""
.
.
<field name="CATENTTYPE_ID"
  fieldDescription="Product Type"
  type="string"
  maxLength="16"
  defaultValue=""
.
.
```

添加下拉菜单

您可以将选择列表以下拉菜单的形式添加到 Web Editor 表单中, 以使表单更易于完成, 并且限制用户从有效的选择集中进行选择。

要刷新选择列表, 管理员可以重新启动 Web Editor。还有一个表单描述字段属性 `dynamicSqlSelectionList`, 管理员可以将其设置为 “true” 以确保每次都重新装入下拉菜单。

选择列表的查询可以具有返回一个或两个列的结果集。如果结果返回两个列, 则第二列包含存储的实际值, 第一列包含用户标签。

通过在 `field` 标记的 `sqlSelectionList` 属性中输入 SQL 查询或通过在 XML 表单描述文件中创建枚举, 可以创建选择列表。两个方法都在以下示例中显示:

```

<field name="MEMBER_ID"
  showInCreateMode="false"
  fieldDescription="Member Identifier"
  .
  .
  readOnly="false"
  sqlSelectionList="select orgentityname,orgentity.orgentity_id
    from member,orgentity where member.type='0'
    and member.member_id=orgentity.orgentity_id"
  .
  .
  fieldHelp=""
  .
  .
  </field>
  <field name="MARKFORDELETE"
    fieldDescription="Mark for Delete"
    type="NMTOKEN"
    sqlSelectionList=""
  >
  <datatype source="integer">
    <enumeration label="No" value="0"/>
    <enumeration label="Yes" value="1"/>
  </datatype>
  </field>
>

```

WebSphere Commerce 数据库表中有字段使用 1 和 0 表示真或假。要创建对用户更直观的字段，可以建立这些字段的枚举。Catalog Manager 提供了一个 SQL 脚本，用于创建名为“NUMDESC”的枚举表。此脚本名为“createEnum.sql”。管理员可以如以下示例所示使用 ENUMDESC 表编辑 sqlSelectionList 属性来创建选择列表。

```

fieldDescription="On Special"
.
.
readOnly="false"
sqlSelectionList="select description,value from
  numdesc where columnname='ALL' and type='YESNO'"
fieldHelp=""
.
.

```

添加字段帮助

对于在焦点中的当前字段，在 Web 浏览器窗口的底部将显示字段帮助。XML 表单描述文件中的 field 标记上的属性 fieldHelp 或属性文件中的 fieldHelp 键可用于设置此值。

例如，属性文件可以包含以下字段帮助指定：

```

CATEGORY.MARKFORDELETE.defaultValue=No
CATEGORY.MARKFORDELETE.fieldDescription=Delete Entry

```

如果没有指定值，则创建缺省消息“在 *field_type* 中输入 *field_name*”。

注： 如果存在包含表单信息的属性文件，则该文件优先于 XML 表单描述文件中的条目。

定制搜索结果和工作会话列表






可以定制在搜索结果页面上显示的列集。XML 表单描述文件中的 `field` 标记有 `showColumnInList` 属性。要在搜索结果中包含字段，请将此属性设置为“true”；否则，它不会在搜索结果视图中作为列之一出现。表单显示时将出现字段。

编辑 weProcessList 文件

weProcessList 文件允许管理员定制 Catalog Manager 实用程序，这些实用程序在处理 Web Editor 工作会话时运行。






-      

weProcessList.xml 文件位于以下目录:

-  `drive:\WebSphere\CommerceServer\xml\wcwebeditor\xml`
-  `drive:\Program Files\WebSphere\CommerceServer\xml\wcwebeditor\xml`
-  `/usr/WebSphere/CommerceServer/xml/wcwebeditor/xml`
-   `/opt/WebSphere/CommerceServer/xml/wcwebeditor/xml`

-      

weProcessListOracle.xml 文件位于以下目录:

-  `drive:\WebSphere\CommerceServer\xml\wcwebeditor\xml`
-  `drive:\Program Files\WebSphere\CommerceServer\xml\wcwebeditor\xml`
-  `/usr/WebSphere/CommerceServer/xml/wcwebeditor/xml`
-   `/opt/WebSphere/CommerceServer/xml/wcwebeditor/xml`

-  weProcessListAS400.xml 文件位于以下目录:

- `/instroot/xml/wcwebeditor/xml`

此文件包含各种不同实用程序的信封模板。它也可以包含对管理员想运行的定制应用程序的引用。

有一组系统变量可以在此文件中使用。例如，系统变量 `%-dbname%` 使得数据库名插入到信封中，此信封为诸如装入程序的实用程序的给定调用生成。XML 表单描述文件包含到这些过程的引用，这些过程指示应当调用哪个以进行添加、编辑或删除。

以下是 weProcessList.xml 文件的示例:

```
<?xml version="1.0" encoding="UTF-8"?>
<processSet>
  <!-- Do not change name of extract -->
  <process name="extract"
    subsystem="com.ibm.wca.MassExtract.extract.ExtractSubSystem">
    <envelope-input xmlns='saf_params.xsd'>
      <param name="-dbname" type="scalar" value="%-dbname%"/>
      <param name="-dbuser" type="scalar" value="%-dbuser%"/>
      <param name="-dbpwd" type="scalar" value="%-dbpwd%"/>
      <param name="-outfile" type="file" reside="local" value="%tempFilePath%"/>
      <param name="-filter" type="file" reside="local" value="%tempFileURI%"/>
    </envelope-input>
  </process>
</processSet>
```

```

</process>
<process name="transformer"
  subsystem="com.ibm.wca.XMLTransformer.XMLTransformerSubSystem">
  <envelope-input xmlns='saf_params.xsd'>
    <param name="-infile" type="file" reside="local" value="%tempFileURI%"/>
    <param name="-transform" type="file" reside="local"
      value="%webEditorDir%/xsl/ViewsToWCS51.XSL"/>
    <param name="-outfile" type="file" reside="local" value="%tempFilePath1%"/>
    <param name="-param" value="root=%-dbname%"/>
    <param name="-param" value="dtdname=%-dtdname%"/>
  </envelope-input>
</process>
<process name="transformerForDelete"
  subsystem="com.ibm.wca.XMLTransformer.XMLTransformerSubSystem">
  <envelope-input xmlns='saf_params.xsd'>
    <param name="-infile" type="file" reside="local" value="%tempFileURI%"/>
    <param name="-transform" type="file" reside="local"
      value="%webEditorDir%/xsl/ViewsToWCS51.XSL"/>
    <param name="-outfile" type="file" reside="local" value="%tempFilePath1%"/>
    <param name="-param" value="root=%-dbname%"/>
    <param name="-param" value="dtdname=%-dtdname%"/>
    <param name="-param" value="forDelete=true"/>
  </envelope-input>
</process>
<process name="resolver"
  subsystem="com.ibm.wca.IdResGen.IdResGenSubSystem">
  <envelope-input xmlns='saf_params.xsd'>
    <param name="-dbname" type="scalar" value="%-dbname%"/>
    <param name="-dbuser" type="scalar" value="%-dbuser%"/>
    <param name="-dbpwd" type="scalar" value="%-dbpwd%"/>
    <param name="-infile" type="file" reside="local"
      value="%previousOutFileAsURI%"/>
    <param name="-outfile" type="file" reside="local"
      value="%tempFilePath2%"/>
    <param name="-propfile" type="file" reside="local"
      value="propertyFiles.IdKeys"/>
    <param name="-method" type="scalar" value="mixed"/>
  </envelope-input>
</process>
<!-- Resolver as first process -->
<process name="resolverFirstProcess"
  subsystem="com.ibm.wca.IdResGen.IdResGenSubSystem">
  <envelope-input xmlns='saf_params.xsd'>
    <param name="-dbname" type="scalar" value="%-dbname%"/>
    <param name="-dbuser" type="scalar" value="%-dbuser%"/>
    <param name="-dbpwd" type="scalar" value="%-dbpwd%"/>
    <param name="-infile" type="file" reside="local" value="%tempFileURI%"/>
    <param name="-outfile" type="file" reside="local" value="%tempFilePath2%"/>
    <param name="-propfile" type="file" reside="local" value="propertyFiles.IdKeys"/>
    <param name="-method" type="scalar" value="mixed"/>
  </envelope-input>
</process>
<process name="loader"
  subsystem="com.ibm.wca.MassLoader.MassLoadSubSystem">
  <envelope-input xmlns='saf_params.xsd'>
    <param name="-dbname" type="scalar" value="%-dbname%"/>
    <param name="-dbuser" type="scalar" value="%-dbuser%"/>
    <param name="-dbpwd" type="scalar" value="%-dbpwd%"/>
    <param name="-commitcount" type="scalar" value="1000"/>
    <param name="-infile" type="file" reside="local" value="%previousOutFileAsURI%"/>
    <param name="-method" type="scalar" value="sqlimport"/>
    <param name="-noprimary" type="scalar" value="insert"/>
  </envelope-input>
</process>
<process name="loaderFirstProcess"
  subsystem="com.ibm.wca.MassLoader.MassLoadSubSystem">
  <envelope-input xmlns='saf_params.xsd'>
    <param name="-dbname" type="scalar" value="%-dbname%"/>
    <param name="-dbuser" type="scalar" value="%-dbuser%"/>
    <param name="-dbpwd" type="scalar" value="%-dbpwd%"/>
    <param name="-commitcount" type="scalar" value="1000"/>
    <param name="-infile" type="file" reside="local" value="%tempFileURI%"/>
    <param name="-method" type="scalar" value="sqlimport"/>
    <param name="-noprimary" type="scalar" value="insert"/>
  </envelope-input>
</process>

```

```

    </envelope-input>
</process>
<process name="loaderForDelete"
  subsystem="com.ibm.wca.MassLoader.MassLoadSubSystem">
  <envelope-input xmlns='saf_params.xsd'>
    <param name="-dbname" type="scalar" value="%-dbname%"/>
    <param name="-dbuser" type="scalar" value="%-dbuser%"/>
    <param name="-dbpwd" type="scalar" value="%-dbpwd%"/>
    <param name="-commitcount" type="scalar" value="1000"/>
    <param name="-infile" type="file" reside="local" value="%previousOutFileAsURI%"/>
    <param name="-delete" type="scalar" value=""/>
  </envelope-input>
</process>
<process name="loaderForDeleteFirstProcess"
  subsystem="com.ibm.wca.MassLoader.MassLoadSubSystem">
  <envelope-input xmlns='saf_params.xsd'>
    <param name="-dbname" type="scalar" value="%-dbname%"/>
    <param name="-dbuser" type="scalar" value="%-dbuser%"/>
    <param name="-dbpwd" type="scalar" value="%-dbpwd%"/>
    <param name="-commitcount" type="scalar" value="1000"/>
    <param name="-infile" type="file" reside="local" value="%tempFileURI%"/>
    <param name="-delete" type="scalar" value=""/>
  </envelope-input>
</process>
<process name="saveToFile"
  cmd="cmd.exe /c c:\temp\theBatchFile.bat"
  args="-infile %tempFilePath% -dbname %-dbname%"
/>
</processSet>

```







注：文件引用是区分大小写的。

下表包含应用程序能够理解的有效替换变量的列表。

% 替换变量	返回
%-dbname%	当前数据库名称
%-dbuser%	数据库用户名
%-dtdname%	用于 XML 文件的 DTD 的 URI 位置
%-dbpwd%	数据库用户名的密码
%tempFilePath%	到临时文件的全路径
%tempFilePath1%	这些是唯一的临时文件名称。它们可以置于信封定义或命令行的模板语法中。例如，如果 %tempFilePath% 置于信封模板中 -infile 参数的 value 属性中，则 Web Editor 将工作会话的数据写入到临时文件位置。
%tempFilePath2%	
在子系统上	临时 URI
%tempFileURI%	临时 URI 是以 %tempFilePath%...%tempFilePath2% 表示的到相同文件的 URI。这不是另外的文件集，而是一种检索使用不同语法返回的相同生成的临时文件的方法。
%tempFileURI1%	
%tempFileURI2%	
%previousOutFileAsURI%	提供前一任务 -outfile 参数作为 URI 的 URI 表示法
%webEditorDir%	Web Editor 的安装位置

编辑 webeditor.xml 文件

Web Editor 使用 webeditor.xml 文件作为缺省的 XSL 样式表。该文件位于以下目录:

-  *drive:\WebSphere\CommerceServer\xml\wcwebeditor\xsl*
-  *drive:\Program Files\WebSphere\CommerceServer\xml\ wcwebeditor\xsl*
-  */usr/WebSphere/CommerceServer/xml/wcwebeditor/xsl*
-   */opt/WebSphere/CommerceServer/xml/wcwebeditor/xsl*
-  */instroot/xml/wcwebeditor/xsl*

通过编辑 webeditor.xml 文件, 管理员可以更改 Web Editor 的输出格式。以下是 webeditor.xml 文件的内容样本:

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
<xsl:output method="html"/>
  <!-- Largest single line entry field value for fields larger than this a
    TEXTAREA is created -->
  <xsl:variable name="maxEntryFieldSize" select="80"/>
  <xsl:template match="/">
    <xsl:apply-templates/>
  </xsl:template>
  <!-- Read Only field -->
  <xsl:template match="readOnly" name="readOnly">
    <xsl:element name="input">
      <xsl:attribute name="name"><xsl:value-of select="@name"/></xsl:attribute>
      <xsl:attribute name="type">hidden</xsl:attribute>
      <xsl:attribute name="value"><xsl:value-of select="@defaultValue"/></xsl:attribute>
      <!--
      <xsl:attribute name="onFocus">this.blur()</xsl:attribute>
      <xsl:attribute name="style">border-style:groove</xsl:attribute>
      -->
    </xsl:element>
    <table border="1" cellpadding="0" cellspacing="0" width="155" bgcolor="#C0C0C0">
      <tr>
        <td>
          <xsl:value-of select="@defaultValue"/>
        </td>
      </tr>
    </table>
  </xsl:template>
</xsl:stylesheet>
```

第 12 章 处理产品目录

以下过程描述如何通过使用 Web Editor 添加、修改和删除包含产品目录数据的数据库表中的记录，来处理产品目录。

使用 Web Editor 将记录添加到表

要使用 Web Editor 将记录添加到表，请遵循以下步骤：

1. 在 Web 浏览器中打开以下 URL:

```
https://host_name:8000/wcm/webeditor
```

其中 *host_name* 是 WebSphere Application Server 的全限定 HTTP 主机名。

Web Editor 数据库登录窗口出现。

2. 输入数据库用户名和密码；然后单击**登录**。

Web Editor 在浏览器窗口的左侧菜单栏中显示表名的列表。

3. 在左侧菜单栏的“添加”子菜单中单击适当的超级链接。

显示适当的表单。

4. 在表单中输入所有必需的数据。

5. 单击**移至工作会话**。

显示表单的工作会话结果。这些结果包含在 Web Editor 工作会话期间所有已执行但未清除或未处理的编辑、添加和删除。

6. 如果您希望从工作会话中除去任何记录更改，请选中您想除去的每一记录更改前的框并单击**清除已选项**。

7. 单击**处理工作会话**将选定的更改提交到数据库。

状态页面显示一条声明过程成功完成的消息。

8. 导航到 Web 站点，单击适当的超级链接并验证所做的更改。

使用 Web Editor 修改表中的记录

要使用 Web Editor 修改表中的记录，请遵循以下步骤：

1. 在 Web 浏览器中打开以下 URL：

```
https://host_name:8000/wcm/webeditor
```

其中 *host_name* 是 WebSphere Application Server 的全限定 HTTP 主机名。

Web Editor 数据库登录窗口出现。

2. 输入数据库用户名和密码；然后单击**登录**。

Web Editor 在浏览器窗口的左侧菜单栏中显示表名的列表。

3. 在左侧菜单栏的“搜索”子菜单中单击适当的超级链接。

显示适当的搜索页面。

4. 通过执行以下操作指定搜索条件：

- a. 选择您希望在搜索中指定的每个属性旁的复选框。
- b. 对于每一选定的属性使用适当的下拉菜单，以选择您希望在搜索中使用的逻辑。
- c. 在每一选定的属性的下一字段中，输入或选择您希望在搜索中使用的值。

5. 单击**查找**。

这将把搜索条件提交到 Web Editor。

显示状态页面，此页面显示符合搜索条件的记录数。

6. 执行以下操作之一：

- 单击**装入数据**查看找到的记录列表。

Web Editor 显示通过查询检索到的记录列表。

转至步骤 6。

- 单击**新建搜索**返回到搜索页面。

返回步骤 3。

7. 选择您希望编辑的记录。

显示适当的表单。

8. 向下滚动到您希望编辑的字段，然后更改其内容。

9. 单击**移至工作会话**。

显示表单的工作会话结果。这些结果包含在 Web Editor 工作会话期间所有已执行但未清除或未处理的编辑、添加和删除。

10. 如果您希望从工作会话中除去任何记录更改，请选中您想除去的每一记录更改前的框并单击**清除已选项**。

11. 单击**处理工作会话**将选定的更改提交到数据库。

状态页面显示一条声明过程成功完成的消息。

12. 导航到 Web 站点，单击适当的超级链接并验证所做的更改。

使用 Web Editor 从表中删除记录

要使用 Web Editor 从表中删除记录，请遵循以下步骤：

1. 在 Web 浏览器中打开以下 URL：

```
https://host_name:8000/wcm/webeditor
```

其中 *host_name* 是 WebSphere Application Server 的全限定 HTTP 主机名。

Web Editor 数据库登录窗口出现。

2. 输入数据库用户名和密码；然后单击**登录**。

Web Editor 在浏览器窗口的左侧菜单栏中显示表名的列表。

3. 在左侧菜单栏的“搜索”子菜单中单击适当的超级链接。

显示适当的搜索页面。

4. 通过执行以下操作指定搜索条件：

- a. 选择您希望在搜索中指定的每个属性旁的复选框。
- b. 对于每一选定的属性使用适当的下拉菜单，以选择您希望在搜索中使用的逻辑。
- c. 在每一选定的属性的下一字段中，输入或选择您希望在搜索中使用的值。

5. 单击**查找**。

这将把搜索条件提交到 Web Editor。

显示状态页面，此页面显示符合搜索条件的记录数。

6. 执行以下操作之一：

- 单击**装入数据**查看找到的记录列表。

Web Editor 显示通过查询检索到的记录列表。

转至步骤 6。

- 单击**新建搜索**返回到搜索页面。

返回步骤 3。

7. 选择您希望删除的每条记录旁的复选框。

8. 单击**移至删除列表**。

9. 在左侧菜单栏的“工作会话”子菜单中单击适当的超级链接。

显示表单的工作会话结果。这些结果包含在 Web Editor 工作会话期间所有已执行但未清除或未处理的编辑、添加和删除。

10. 如果您希望从工作会话中除去任何记录更改，请选中您想除去的每一记录更改前的框并单击**清除已选项**。

11. 单击**处理工作会话**将更改提交到数据库。

状态页面显示一条声明过程成功完成的消息。

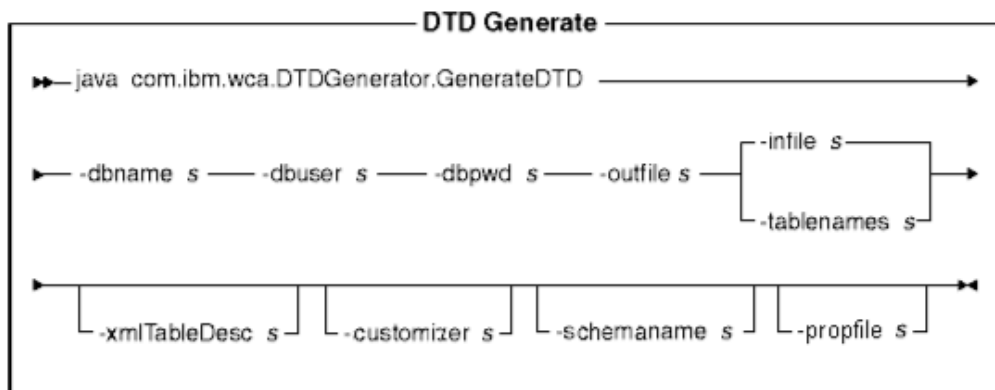
12. 导航到 Web 站点，单击适当的超级链接并验证所做的更改。

第 4 部分 命令参考

第 13 章 DTD 生成命令

此命令创建与装入程序软件包一起使用的 DTD 和模式文件。

用于 Windows、AIX、Linux 和 Solaris 系统的 DTD 生成命令



注:

1. 上图主要用作命令参数的参考。为此命令提供的命令文件或脚本以及在第 71 页的第 10 章,『配置装入程序软件包命令和脚本』下列出的命令文件或脚本充当实际 Java 命令的包装并接受相同的参数;因此,建议您使用命令文件或脚本,而不用直接调用 Java 命令。
2. 指定为此命令的参数文件名可以加在相对或绝对路径的前面。

参数值:

-dbname

目标数据库的名称

-dbuser

连接到数据库的用户的姓名

-dbpwd

连接到数据库的用户的密码

-outfile

输出 DTD 文件的名称

-infile 每行上包含数据库表名的输入文件的名称

-tablename

表的名称,用逗号分隔

-xmlTableDesc

要创建的模式文件的文件路径

-customizer

要使用的定制程序属性文件的名称。DB2ConnectionCustomizer.properties 是缺省文件。定制程序属性文件可以按以下示例所示的方式指定:

-customizer d:\wc\prop\dttdgen.properties

如果此文件存在于在类路径系统环境变量中指定的目录中，则同样的文件可以按以下示例所示的方式指定：

-customizer dtdgen

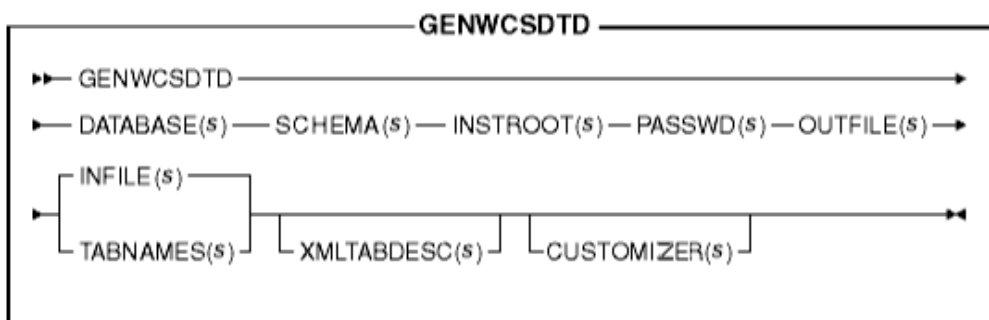
-schemaname

目标数据库模式的名称

-propFile

包含属性的文件，例如，可以存储帮助文本、缺省值和字段描述信息以用于 Web Editor 表单描述的外部属性文件。

用于 iSeries 系统的 DTD 生成命令



注：指定为此命令的参数文件名可以加在相对或绝对路径的前面。

参数值：

DATABASE

目标数据库的名称，如关系数据库目录中所示

SCHEMA

目标数据库模式的名称；这与实例名称相同

INSTROOT

WebSphere Commerce 实例根路径的全名，例如
/QIBM/UserData/WebCommerce/instances/*instance_name*

PASSWD

WebSphere Commerce 实例的密码

OUTFILE

输出 DTD 文件的名称

INFILE

每行上包含数据库表名的输入文件的名称

TABNAMES

表的名称，用逗号分隔

XMLTABDESC

要创建的模式文件的文件路径。此参数是可选的。

CUSTOMIZER

要使用的定制程序属性文件的名称。缺省文件是 `ISeries_GENWCSDTD_Customizer.properties`。定制程序属性文件可以按以下示例所示的方式指定:

```
CUSTOMIZER(/wc/prop/dtdgen.properties)
```

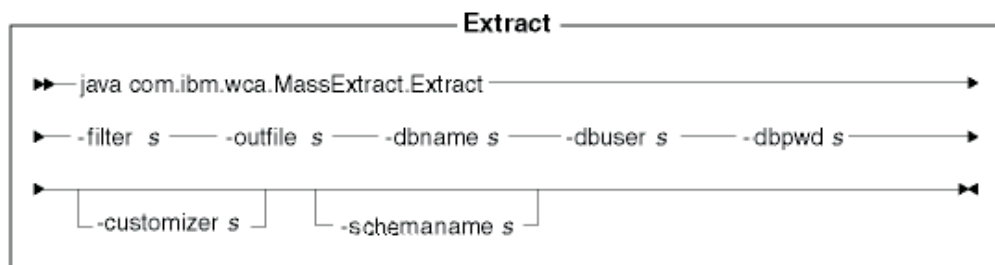
如果此文件存在于在类路径系统环境变量中指定的目录中, 则同样的文件可以按以下示例所示的方式指定:

```
CUSTOMIZER(dtdgen)
```

第 14 章 抽取命令

此命令从数据库中以 XML 文件的格式抽取一个选定的数据子集。

用于 Windows、AIX、Linux 和 Solaris 系统的抽取命令



注:

1. 上图主要用作命令参数的参考。为此命令提供的命令文件或脚本以及在第 71 页的第 10 章,『配置装入程序软件包命令和脚本』下列出的命令文件或脚本充当实际 Java 命令的包装并接受相同的参数;因此,建议您使用命令文件或脚本,而不用直接调用 Java 命令。
2. 指定为此命令的参数的文件名可以加在相对或绝对路径的前面。

参数值:

-filter 抽取过滤器文件的名称

-outfile
输出 XML 文件的名称,已抽取的数据将存储在该文件中

-dbname
从中正在抽取数据的数据库的名称

-dbuser
从中正在抽取数据的数据库的数据库用户名

-dbpwd
与从中正在抽取数据的数据库的用户名相关联的密码

-customizer
要使用的定制程序属性文件的名称。DB2ConnectionCustomizer.properties 是缺省文件。定制程序属性文件可以按以下示例所示的方式指定:

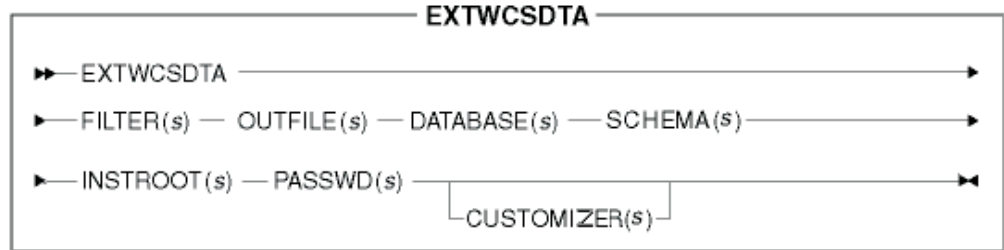
```
-customizer d:\wc\prop\extract.properties
```

如果此文件存在于在类路径系统环境变量中指定的目录中,则同样的文件可以按以下示例所示的方式指定:

```
-customizer extract
```

-schemaname
目标数据库模式的名称

用于 iSeries 系统的抽取命令



注：指定为此命令的参数文件名可以加在相对或绝对路径的前面。

参数值：

FILTER

抽取过滤器文件的名称

OUTFILE

输出 XML 文件的名称，已抽取的数据将存储在该文件中

DATABASE

从中正在抽取数据的数据库的名称，如关系数据库目录中所示

SCHEMA

从中正在抽取数据的数据库模式的名称，这与实例名称相同

INSTROOT

WebSphere Commerce 实例根路径的全名，例如
/QIBM/UserData/WebCommerce/instances/*instance_name*

PASSWD

WebSphere Commerce 实例的密码

CUSTOMIZER

要使用的定制程序属性文件的名称。缺省文件是
ISeries_EXTWCSDTA_Customizer.properties。定制程序属性文件可以按以下
示例所示的方式指定：

```
CUSTOMIZER(/wc/prop/extract.properties)
```

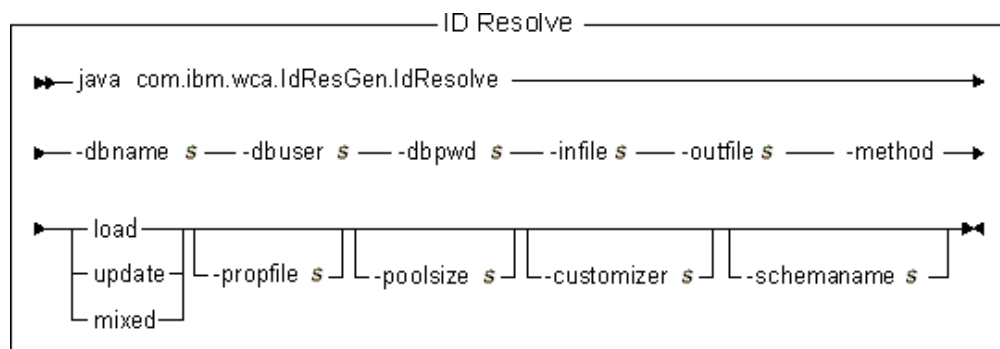
如果此文件存在于在类路径系统环境变量中指定的目录中，则同样的文件可以
按以下示例所示的方式指定：

```
CUSTOMIZER(extract)
```

第 15 章 标识解析命令

此命令在 XML 元素被装入到数据库之前为需要标识的 XML 元素生成标识。

用于 Windows、AIX、Linux 和 Solaris 系统的标识解析命令



注:

1. 上图主要用作命令参数的参考。为此命令提供的命令文件或脚本以及在第 71 页的第 10 章,『配置装入程序软件包命令和脚本』下列出的命令文件或脚本充当实际 Java 命令的包装并接受相同的参数;因此,建议您使用命令文件或脚本,而不用直接调用 Java 命令。
2. 指定为此命令的参数的文件名可以加在相对或绝对路径的前面。

参数值:

-dbname

目标数据库的名称

-dbuser

连接到数据库的用户的姓名

-dbpwd

连接到数据库的用户的密码

-infile 包含表记录描述符的输入 XML 文档的名称

-outfile

要产生的输出 XML 文件的名称;此文件可以用作对装入程序的输入

-method

用于处理输入文件的方法

- 如果文件中的**所有记录都不存在于**数据库中,请使用 load 方法处理输入文件。
- 如果文件中的**所有记录都存在于**数据库中,请使用 update 方法处理输入文件。
- 如果文件中**仅部分记录存在于**数据库中,请使用 mixed 方法处理输入文件。

缺方法是 load。

-profile

以“名称=值”对的格式包含 Java 属性的文本文件。此文件用于为外键标识的查找定义后备列名，并为主表（例如 CATEGORY 和 PRODUCT）查询选择谓词。您可以在此文件中省略定义了唯一索引的表的条目，此唯一索引不包含标识。缺省文件是 IdResolveKeys.properties。尽管此文件具有 .properties 扩展名，但是不要在指定值的时候使用扩展名。

-poolsize

要保留的标识的数目。缺省数目为 50。

-customizer

要使用的定制程序属性文件的名称。DB2ConnectionCustomizer.properties 是缺省文件。定制程序属性文件可以按以下所示的两个实例中任意一个指定：

`-customizer d:\wc\prop\idres.properties`

`-customizer d:\wc\prop\idres`

如果此文件存在于当前目录中，则同样的文件可以按以下示例所示的方式指定：

`-customizer idres.properties`

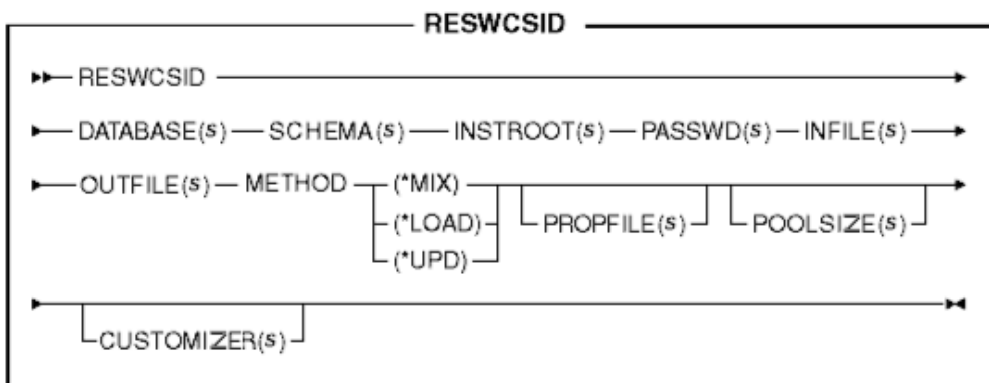
如果此文件存在于在类路径系统环境变量中指定的目录中，则同样的文件可以按以下示例所示的方式指定：

`-customizer idres`

-schemaname

目标数据库模式的名称

用于 iSeries 系统的标识解析命令



注：指定为此命令的参数文件名可以加在相对或绝对路径的前面。

参数值：

DATABASE

目标数据库的名称，如关系数据库目录中所示

SCHEMA

目标数据库模式的名称；这与实例名称相同

INSTROOT

WebSphere Commerce 实例根路径的全名，例如
`/QIBM/UserData/WebCommerce/instances/instance_name`

PASSWD

WebSphere Commerce 实例的密码

INFILE

包含表记录描述符的输入 XML 文档的名称

OUTFILE

要产生的输出 XML 文件的名称；此文件可以用作对装入程序的输入

METHOD

用于处理输入文件的方法

- 如果文件中的**所有记录都不存在于**数据库中，请使用 `load` 方法 (*LOAD) 处理输入文件。
- 如果文件中的**所有记录都存在于**数据库中，请使用 `update` 方法 (*UPD) 处理输入文件。
- 如果文件中**仅部分记录存在于**数据库中，请使用 `mixed` 方法 (*MIX) 处理输入文件。

PROFILE

以“名称=值”对的格式包含 Java 属性的文本文件。此文件用于为外键标识的查找定义后备列名，并为主表（例如 `CATEGORY` 和 `PRODUCT`）查询选择谓词。您可以在此文件中省略定义了唯一索引的表的条目，此唯一索引不包含标识。缺省文件是 `IdResolveKeys.properties`。尽管此文件具有 `.properties` 扩展名，但是不要在指定值的时候使用扩展名。

POOLSIZE

要保留的标识的数目。缺省数目为 50。

CUSTOMIZER

要使用的定制程序属性文件的名称。缺省文件是

`ISeries_RESWCSID_Customizer.properties`。定制程序属性文件可以按以下所示的两个实例中任意一个指定：

```
CUSTOMIZER(/wc/prop/idres.properties)
```

```
CUSTOMIZER(/wc/prop/idres)
```

如果此文件存在于当前目录中，则同样的文件可以按以下示例所示的方式指定：

```
CUSTOMIZER(idres.properties)
```

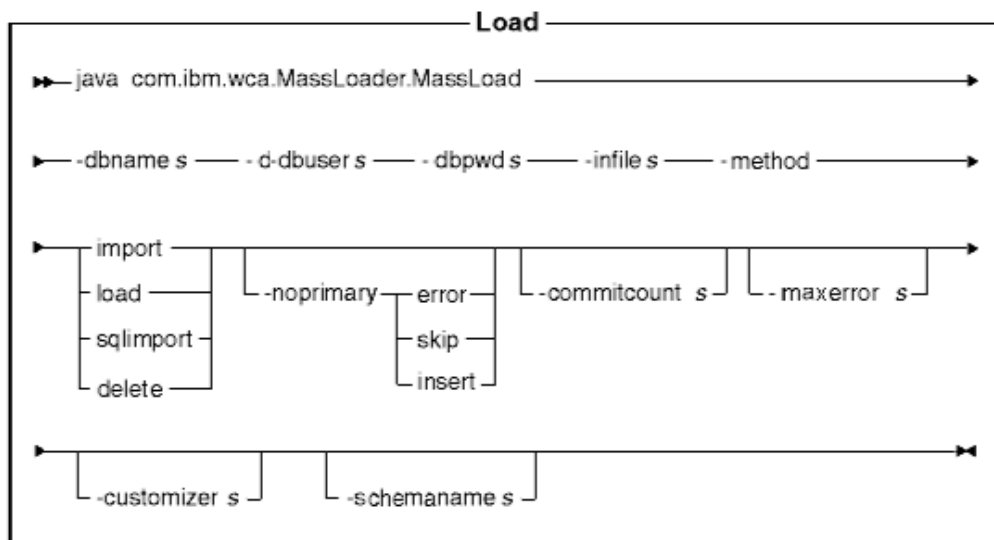
如果此文件存在于在类路径系统环境变量中指定的目录中，则同样的文件可以按以下示例所示的方式指定：

```
CUSTOMIZER(idres)
```


第 16 章 装入命令

此命令装入 XML 输入文件至目标数据库。

用于 Windows、AIX、Linux 和 Solaris 系统的装入命令



注:

1. 上图主要用作命令参数的参考。为此命令提供的命令文件或脚本以及在第 71 页的第 10 章,『配置装入程序软件包命令和脚本』下列出的命令文件或脚本充当实际 Java 命令的包装并接受相同的参数;因此,建议您使用命令文件或脚本,而不用直接调用 Java 命令。
2. 指定为此命令的参数的文件名可以加在相对或绝对路径的前面。

参数值:

-dbname

目标数据库的名称

-dbuser

连接到数据库的用户的姓名

-dbpwd

连接到数据库的用户的密码

-infile 输入 XML 文件的名称

-method

当将数据插入数据库时装入程序使用的操作方式

- load 方法使用来自数据库供应商的本机装入程序。对于本地和远程 Oracle 数据库,您都可以使用 load 方法;但是 load 方法只能用于本地的 DB2 数据库。

- 请使用 import 方法将数据装入到远程 DB2 数据库。如果可从数据库供应商获得 import 或 update 选项，则 import 方法使用这两个选项。如果您为数据库（例如，Oracle）指定此选项，其中 import 或者 update 选项不可用，则将使用 SQL 语句（使用 JDBC）更新数据库。
- SQL 导入（sqlimport）方法可以与本地的和远程的数据库一起使用。
- delete 方法从数据库中删除数据。

如果您在使用产品顾问搜索空间同步，则必须使用 sqlimport 或者 delete 方法。

-noprimary

当丢失输入文件中的记录的主键时，装入程序必须执行的操作。error 选项指示它应当将丢失主键报告为一个错误并终止。skip 选项跳过输入文件中任何不具有主键的记录。insert 选项试图处理（插入或删除）数据。缺省操作为 error。

-commitcount

使用 SQL 更新操作方法时，在发生数据库提交前处理的记录数目。缺省数目为 1。

-maxerror

装入程序终止之前在 SQL 更新操作方法中的错误数。

-customizer

要使用的定制程序属性文件的名称。MassLoadCustomizer.properties 是缺省文件。定制程序属性文件可以按以下示例所示的方式指定：

-customizer d:\wc\prop\ml.properties

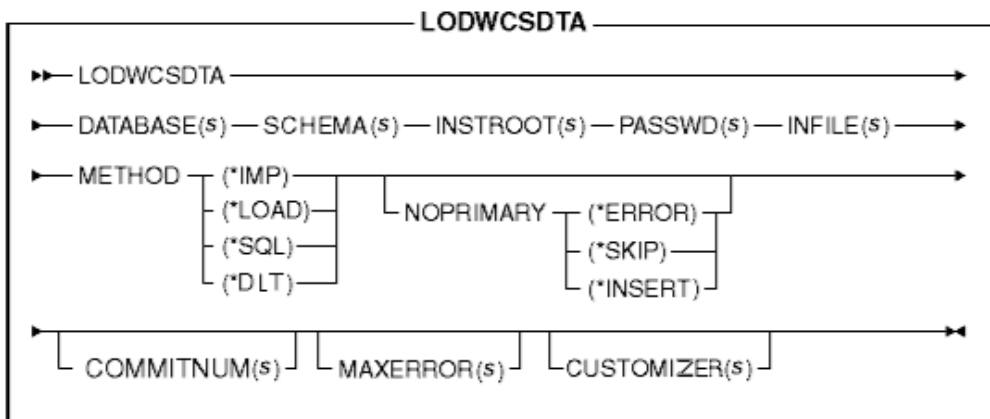
如果此文件存在于在类路径系统环境变量中指定的目录中，则同样的文件可以按以下示例所示的方式指定：

-customizer ml

-schemaname

目标数据库模式的名称

用于 iSeries 系统的装入命令



注：指定为此命令的参数文件名可以加在相对或绝对路径的前面。

参数值：

DATABASE

目标数据库的名称，如关系数据库目录中所示

SCHEMA

目标数据库模式的名称；这与实例名称相同

INSTROOT

WebSphere Commerce 实例根路径的全名，例如
/QIBM/UserData/WebCommerce/instances/*instance_name*

PASSWD

WebSphere Commerce 实例的密码

INFILE

输入 XML 文件的名称

METHOD

当将数据插入数据库时装入程序使用的操作方式

- load 方法 (*LOAD) 使用来自数据库供应商的本机装入程序。对于本地和远程 Oracle 数据库，您都可以使用 load 方法 (*LOAD)；但是 load 方法 (*LOAD) 只能用于本地的 DB2 数据库。
- 请使用 import (*IMP) 选项将数据装入到远程 DB2 数据库。import (*IMP) 方法使用 import 或 update 选项（如果可从数据库供应商获得的话）。如果 import 或者 update 选项不可用，将使用 SQL 语句（使用 JDBC）更新数据库。
- SQL 导入 (*SQL) 方法可以与本地的和远程的数据库一起使用。
- delete (*DLT) 方法从数据库删除数据。

NOPRIMARY

当丢失输入文件中的记录的主键时，装入程序必须执行的操作。error 选项 (*ERROR) 指示它应当将丢失主键报告为错误并终止。skip 选项 (*SKIP) 跳过输入文件中任何不具有主键的记录。insert 选项 (*INSERT) 试图处理（插入或删除）数据。缺省操作为 error。

COMMITNUM

使用 SQL 更新操作方法时，在发生数据库提交前处理的记录数目。缺省数目为 1。

MAXERROR

装入程序终止之前在 SQL 更新操作方法中的错误数。

CUSTOMIZER

要使用的定制程序属性文件的名称。缺省文件是 ISeries_LODWCSDTA_Customizer.properties。定制程序属性文件可以按以下示例所示的方式指定：

```
CUSTOMIZER(/wc/prop/ml.properties)
```

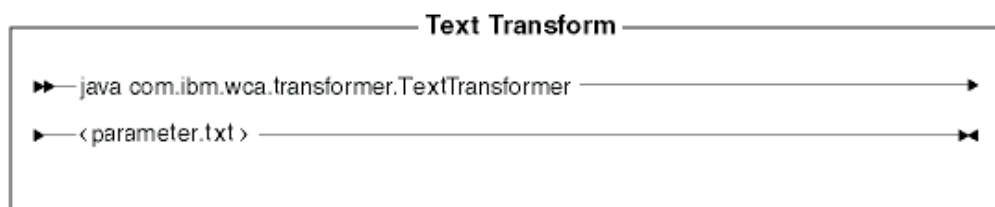
如果此文件存在于在类路径系统环境变量中指定的目录中，则同样的文件可以按以下示例所示的方式指定：

```
CUSTOMIZER(ml)
```

第 17 章 文本转换命令

此命令将数据在字符限定变量格式与 XML 数据格式之间转换。

用于 Windows、AIX、Linux 和 Solaris 系统的文本转换命令



注：上图主要用作命令参数的参考。为此命令提供的命令文件或脚本以及在第 71 页的第 10 章，『配置装入程序软件包命令和脚本』下列出的命令文件或脚本充当实际 Java 命令的包装并接受相同的参数；因此，建议您使用命令文件或脚本，而不用直接调用 Java 命令。

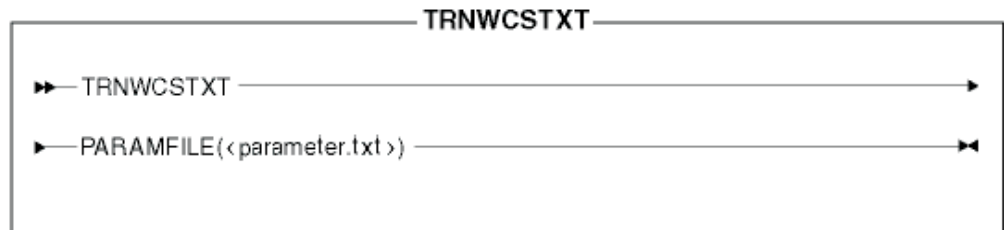
参数值

以下值在参数文件（*parameter.txt*）中指定并以逗号分隔：

- input file
要转换的文件名称
- schema file
将在转换中使用的 XML 模式文件的名称
- output file
将存储已转换数据的输出文件的名称
- transformation method
在将数据添加到输出文件时所使用的的方法。如果要创建新的文件，请指定 **Create**，或者，如果输出数据将附加到现有的数据文件中，请指定 **Append**。

注：此文件也称为“清单”或“命令文件”。

用于 iSeries 系统的文本转换命令



参数值

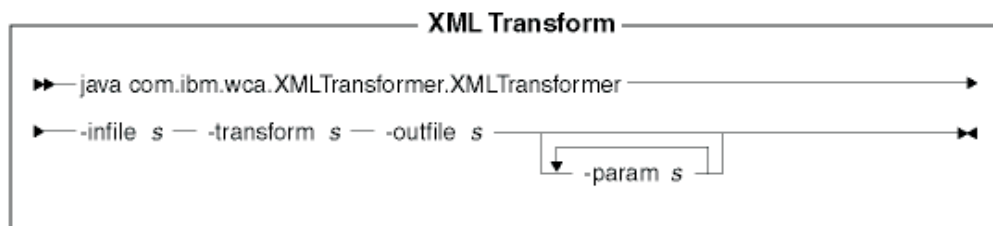
以下值在参数文件 (*parameter.txt*) 中指定并以逗号分隔:

- input file
要转换的文件的名称
- schema file
将在转换中使用的 XML 模式文件的名称
- output file
将存储已转换数据的输出文件的名称
- transformation method
在将数据添加到输出文件时所使用的的方法。如果要创建新的文件, 请指定 **Create**, 或者, 如果输出数据将附加到现有的数据文件中, 请指定 **Append**.

第 18 章 XML 转换命令

此命令将 XML 文件转换为备用 XML 格式。

用于 Windows、AIX、Linux 和 Solaris 系统的 XML 转换命令



注:

1. 上图主要用作命令参数的参考。为此命令提供的命令文件或脚本以及在第 71 页的第 10 章,『配置装入程序软件包命令和脚本』下列出的命令文件或脚本充当实际 Java 命令的包装并接受相同的参数;因此,建议您使用命令文件或脚本,而不用直接调用 Java 命令。
2. 指定为此命令的参数的文件名可以加在相对或绝对路径的前面。

参数值:

-infile 要转换的文件的名称

-transform

转换 XSL 规则文件的名称

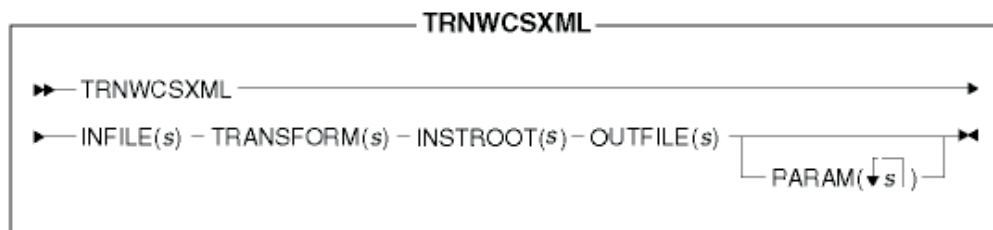
-outfile

输出 XML 文件的名称,转换后的数据将存储在该文件中

-param

要传递给 XSL 规则文件的参数。*此参数是可选的。*此参数可以多次指定以传递多个“名称=值”对。

用于 iSeries 系统的 XML 转换命令



注: 指定为此命令的参数的文件名可以加在相对或绝对路径的前面。

参数值:

INFILE

要转换的文件的名称

TRANSFORM

转换 XSL 规则文件的名称

INSTROOT

WebSphere Commerce 实例根路径的全名，例如
`/QIBM/UserData/WebCommerce/instances/instance_name`

OUTFILE

输出 XML 文件的名称，转换后的数据将存储在该文件中

PARAM

要传递给 XSL 规则文件的参数。**此参数是可选的。**此字符串可包含多个值以传递多个“名称=值”对。

声明

本信息是为在美国提供的产品和服务编写的。IBM 可能在其它国家或地区不提供本文中讨论的产品、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 代理咨询。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证。您可以用书面方式将许可证查询寄往：

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

有关双字节（DBCS）信息的许可证查询，请与您所在国家或地区的 IBM 知识产权部门联系，或用书面方式将查询寄往：

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

本条款不适用联合王国或任何这样的条款与当地法律不一致的国家或地区： 国际商业机器公司以“按现状”的基础提供本出版物，不附有任何形式的（无论是明示的，还是默示的）保证，包括（但不限于）对非侵权性、适销性和适用于某特定用途的默示保证。某些国家或地区在某些交易中不允许免除明示或默示的保证。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；这些更改将编入本书的新版本中。IBM 可以随时对本书中描述的产品和 / 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。该 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。

本程序的被许可方如果要了解有关程序的信息以达到如下目的：（i）允许在独立创建的程序和其它程序（包括本程序）之间进行信息交换，以及（ii）允许对已经交换的信息进行相互使用，请与下列地址联系：

IBM Canada Ltd.
Office of the Lab Director

8200 Warden Avenue, Markham, Ontario L6G 1C7
Canada

只要遵守适当的条件和条款，包括某些情形下的一定数量的付费，都可获得这方面的信息。

本资料中描述的许可程序及其所有可用的许可资料均由 IBM 依据 IBM 客户协议、IBM 国际程序许可证协议或任何同等协议中的条款提供。

此处包含的任何性能数据都是在受控环境中测得的。因此，在其它操作环境中获得的数据可能会有明显的不同。有些测量可能是在开发级的系统上进行的，因此不保证与一般可用系统上进行的测量结果相同。此外，有些测量是通过推算而估计的，实际结果可能会有差异。本文档的用户应当验证其特定环境的适用数据。

涉及非 IBM 产品的信息可从这些产品的供应商、其出版说明或其它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、兼容性或任何其它关于非 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。

所有关于 IBM 未来方向或意向的声明都可随时更改或收回，而不另行通知，它们仅仅表示了目标和意愿而已。

所有 IBM 的价格均是 IBM 当前的建议零售价，可随时更改而不另行通知。经销商的价格可与此不同。

本信息仅用于规划的目的。在所描述的产品可用前，本信息会作更改。

本资料包括日常业务运作中的数据和报告实例。为尽可能表述完整，这些示例包含人名及公司、品牌和产品的名称。所有这些名称都是虚构的，任何与实际商务企业使用的名称和地址类似的名称和地址纯属巧合。

版权许可证:

本资料包括以源语言编写的样本应用程序，这些样本应用程序说明不同操作平台上的编程技术。如果目的是为了开发、使用、经销或分发这样的应用程序，即符合为其编写本样本程序的操作平台的应用程序编程接口的应用程序，则可以任何形式复制、修改、分发这些样本程序，而无须向 IBM 付费。这些示例并未在所有条件下作全面测试。因此，IBM 不能担保或暗示这些程序的可靠性、可维护性或功能。如果目的是为了开发、使用、经销或分发这样的应用程序，即符合 IBM 的应用程序编程接口的应用程序，则可以任何形式复制、修改、分发这些样本程序，而无须向 IBM 付费。

这些样本程序的每份副本或任何部分或其任何派生产品，都必须包含如下形式的版权声明:

©Copyright International Business Machines Corporation 2001.
Portions of this code are derived from IBM Corp. Sample Programs.
©Copyright IBM Corp. 2000, 2001. All rights reserved.

如果您查看本资料的软拷贝，可能看不到照片和彩色图例。

商标和服务标记

以下术语是国际商业机器公司在美国和 / 或其它国家或地区的商标:

AIX
DB2
DB2 Universal Database
IBM
iSeries
OS/400
WebSphere

Microsoft、Windows、Windows NT 和 Windows 2000 是 Microsoft Corporation 在美国和 / 或其它国家或地区的注册商标。

Oracle 是 Oracle Corporation 在美国和 / 或其它国家或地区的注册商标。

Solaris、Java 和所有基于 Java 的商标和徽标是 Sun Microsystems, Inc. 在美国和 / 或其它国家或地区的商标或注册商标。

其它公司、产品和服务名称可能是其它公司的商标或服务标记。



中国印刷