Java XML
Programmer's Reference

Java XML程序员参考手册

Mohammad Akif
Steven Brodhead
Andrei Cioroianu
James Hart
Eric Jung
Dave Writz

〔美〕 等著

马树奇 等译

電子工業出版社
Publishing House of Electronics Industry
北京 · BEIJING

内 容 提 要

Java语言已经成为Internet时代的主流编程语言，XML则是数据描述的优秀工具。Java语言与XML的结合具有广泛而深远的意义，这方面的知识是当今程序设计人员必须掌握的。本书正是以Java和XML为主题，提供了标准的API参考、实例以及编程技巧，使Java程序设计人员学会开发XML应用程序。

版权贸易合同登记号：01-2001-3353

图书在版编目（CIP）数据

Java XML程序员参考手册/（美）阿吉夫（Akif, M.）著；马树奇等译. - 北京：电子工业出版社，2002.5

书名原文：Java XML Programmer's Reference

ISBN 7-5053-7600-4

Ⅰ. J… Ⅱ. ①阿… ②马… Ⅲ. ①JAVA语言 - 程序设计 - 手册 ②可扩充语言，XML - 程序设计 - 手册 Ⅳ. TP312.62

中国版本图书馆CIP数据核字（2002）第030651号

责任编辑：杨 芸

印 刷：北京天竺颖华印刷厂

出版发行：电子工业出版社 http://www.phei.com.cn

北京市海淀区万寿路173信箱 邮编：100036

北京市海淀区翠微东里甲2号 邮编：100036

经 销：各地新华书店

开 本：787×1092 1/16 印张：45 字数：1150千字

版 次：2002年5月第1版 2002年5月第1次印刷

定 价：68.00元

凡购买电子工业出版社的图书，如有缺损问题，请向购买书店调换，若书店售缺，请与本社发行部联系。联系电话：（010）68279077

简　介

本书的目标是为使用XML的Java程序员提供一个快速的参考，介绍在编写XML应用程序时需要用到的编程接口、工具和技术。作为标准参考材料，本书也有一些技术章节对Java XML程序员遇到的主要工具、实用主题和任务提供简要说明。

本书的读者对象

本书首先针对已经对Java、XML（以及XSLT）有所了解，同时又想快速跟上最新的Java/XML API、实现和工具的程序员。

本书的目的是作为Java XML程序员的桌面手册，同时也提供足够详细的参考材料，还有一系列简单的例子解释这些材料的使用。

本书的内容

本书主要讲述Java程序员操作XML数据时可以使用的编程接口和抽象，包括：

- W3C Document Object Model（W3C文档对象模型，DOM 1级、2级和3级）
- Simple API for XML（SAX，XML的简单API）2.0
- Java API for XML Processing（JAXP，XML处理的Java API）1.1分析和转换接口（包括TrAX）
- 描述XML的JDOM（当前版本为Beta 6）API

此外，本书还介绍了以上标准在当前的一些商业产品和开放源代码程序中的实现。我们将看到在使用Java和XML时还有哪些其他的功能。

- Apache XML项目
- Oracle XDK
- IBM Alphaworks
- Java XML @ Sun
- 用于信息应用的XML工具

本书的另一个特色是提供了五种高级Java/XML编程技术的简明指导。这几个较短的章节组成了一个逻辑系列（或“故事”），涉及了一个开发人员在使用Java语言创建基于XML的应用程序（拍卖服务事例）时面临的主要问题。

本书的结构

本书由四个不同部分组成：

第一部分就是第1章，讲述Java和XML的基础知识和预备知识，以及这两种技术之间的

交叉部分。它提供了使用Java编程语言进行XML编程的基础概述。

第二部分包括第2章到第6章，讨论主要的Java XML API，如SAX、DOM（内核和扩展）、JAXP和JDOM等。包括这些API的完整文档，以及散布在文中的那些解释它们用法的例子。

本书的第三部分（第7章到第11章）主要介绍相关工具，大部分介绍的是商业的和开放源代码的Java XML实现，如Oracle XDK、Apache XML项目、IBM Alphaworks（主要介绍WSDL和UDDI）、Sun Java Tools，以及一组可用的轻量级信息应用工具。重复一下，在这部分内容中，作者有意提供了一些工作范例代码，读者可以在自己的机器上运行。

第四部分包括第12章到第16章，介绍了一系列可用的技术。我们采取了案例学习的方式，该案例的目标是建立一个拍卖服务。其中讲到了用Java操纵XML的许多不同方面的技术，最后你将可以构建出这个应用的工作模型。

本书附录讲述的是Ant安装工具的设置和构建方法。这个工具使读程序能够为本书介绍的大部分工具和JAR文件设置Ant目标，以及测试本书的代码。

附录也描述了我们用到的是哪些技术，从何处可以下载，以及如何把它们准备好以便本书代码能使用。

本书结尾是术语表和索引。

使用本书需要的条件

本书的大部分示例代码可以在Ant（一个基于XML的构建工具）上运行。

对本书用到的所有工具，我们都给出了完整的下载和部署的细节（或对已有文档有清楚的指示），附录中有这些工具的详细介绍。

约定

为了帮助你更好地理解本书的内容，有大量约定贯穿于本书当中。

工作范例——是可以下载和自己试验的——一般以下面的形式列出：

范例：Specimen

源码

本节给出了可以编译和运行范例的Java源文件。例如，如果文件名是example.java，你就可以从Wrox web站点http://www.wrox.com/的归档文件中把这个文件下载下来，通常同一章的所有范例都在同一个子目录下。

```
public class Example {
  ...
}
```

示例

这部分描述了用于输入、验证和操纵XML数据的XML文件、模式、DTD和样式表。同样，你可以从Wrox的网站下载类似sample.xml的文件。

```
<?xml version="1.0">
...
```

输出

本节显示了运行源文件后的输出效果，通常是XML的输出形式。某些例程中输出的可能是实际的XML文件，并被写到你的目录下。运行这些源文件的提示信息以黑体显示在输出结果的开始。

```
>java Example sample.xml
<Output></Output>
```

其他的独立代码通常如下显示：

```
import xml.parser.validator;

public class Example{
...
```

关于文本的样式：

- 正文中出现的文件名和代码显示为：courrier.xml
- 用户界面和URL中出现的文本显示为：File/Save As...
- XML元素名、函数名和属性名写为：<xsl:value-of>、concat()、href

此外：

该黑体字写的是重要的、不可遗忘的信息，它与周围的内容直接相关。

背景的样式用于和当前讨论的内容分开。

客户支持

对于本书，Wrox提供三种支持方式：

- 查看www.wrox.com上的勘误表
- 参加p2p.wrox.com上的一对一论坛
- 给技术支持发E-mail进行查询或发出本书的反馈信息

勘误表

你可以到我们的网站www.wrox.com去查看是否有勘误表，只需要浏览本书所在的页面就可以找到指向勘误表的链接。

P2P列表

你可以参加我们在p2p.wrox.com网站的论坛中的一对一的讨论。你可以把自己的问题提交给作者、评论家和同行。在‘Java’部分有一个java_xml邮件列表，你可以选择加入这个邮件列表或是每周接收一份摘要。如果没有时间或工具来接收邮件列表，你也可以搜索我们的在线文档，可以按特定主题或关键字进行搜索。因为这些列表经过了调整，所以你可以迅速地找到准确的信息。邮件会被协调员编辑或转移到正确的位置，使它成为更有效的资源。垃圾邮件和不相关的邮件将被删除，你的电子邮件地址将受到我们特有的Lyris系统的保护，以免被一些自动收集邮件地址的网络工具利用。

E-mail支持

如果你想指出本书的错误以便在网站上公布，或向了解本书细节的专家直接询问问题，请给support@wrox.com发送邮件。一封典型的E-mail包含如下内容：

- 书名、ISBN号的后四位数字以及所要问的问题的页码
- 你的姓名、联系方式以及所要问的问题的内容

把你的想法告诉我们

作者和Wrox的工作人员已经竭尽全力使本书阅读起来有趣、有用、同时又有教育性，我们也非常想知道你的想法。Wrox一直很乐于了解你最喜欢的内容，以及我们可以做的改进。对反馈意见我们表示非常感谢，无论是批评还是表扬，都将在未来的版本中考虑采纳。必要的时候，我们会把这些意见和问题转给作者。如果你有什么意见和建议，请发E-mail给：feedback@wrox.com。

作 者 简 介

Mohammad Akif

Mohammad是Sun微系统公司的Java设计师。他近5年生活在4个国家，在马来西亚日立分公司、新加坡富士通分公司、美国Sprint公司和其他《财富》500强企业从事与Java技术相关的大型项目的开发工作。他的强项是为使用这些技术建立“最好的实践基础”。

我的一生得到了许多人的帮助与影响。首先感谢我的祖父Nasim Naqvi，他始终关爱并祝福着我，教给我什么是对、什么是错。感谢我的父母Rehana和Akif，是他们造就了我。感谢支持我的兄弟姐妹Irum和Sultan。感谢我的爱妻Saima，她是我的最佳伴侣。还有我的爱女Sakina，她是世上最乖的小宝贝。

Mohammad撰写了第8章和第10章。

Steven Brodhead

Steven是Centcom公司的总裁，他是XML数据库集成工具的缔造者，具有15年的软件开发经验。他最近开发的产品有iConduit（一个XML数据集成平台）和Tibco的Java CORBA ORB。除了开发产品之外，他还使用J2EE、中间件和XML进行大型企业软件产品集成工作。

Steven是复合型人才，既是企业家、新产品设计师也是技术主管。但当他完成了总计35万行的编程工作之后，他认为他最喜欢做的还是程序员。在他职业生涯的早期，他得到了这样的赞誉：“发明了这么多东西，让人追都追不上！”。

Steven住在科罗拉多州的Springs，他喜欢户外活动，如钓鱼、滑雪、徒步旅行。Steve的教育背景包括在路易斯安那州大学获得化工学士和在芝加哥大学获得MBA学位。

Steven撰写了第7章和第13章。

Andrei Cioroianu

Andrei是Devsphere.com的创始人，他创作了Java的开发工具并提供咨询服务。他的项目涉及到小应用程序、桌面应用程序、Servlets和服务器端的架构。他还编写了《Professional Java XML》一书，并在Java开发者期刊上发表过多篇文章。

Andrei欢迎读者对他撰写的第2章～第6章提出意见，他的E-mail地址为andreic@ziplip.com。

James Hart

James已有两年的Java和开放源代码技术开发经验，他是Web Services Architect（http://www.webservicesarchitect.com）的编辑。他在业余时间推销苹果的Mac产品，同时他还是一个技艺不高的吉他爱好者。

感谢Jim、John、Emma和Helen，是他们帮助我撰写了本书的有关章节，还要感谢Chris对我的日夜支持与关爱。

James撰写了第9章。

Eric Jung

Eric在纽约的Ithaca大学获得了计算机学士学位。他已有十年的工作经验，从事过计算机软件开发、银行、医药、法律、通信、网络公司、电视、广告、市场、房地产以及教育业。在这段时间内，他主要做财富500强企业的顾问。

Eric与他的妻子Leah和爱犬Lulu居住在科罗拉多州的Englewood。业余时间，他喜欢滑雪、徒步旅行和阅读。

感谢JM，是你鼓励我并把我推向事业的顶峰。
我永远想念你。

Eric撰写了第11章。

Dave Writz

Dave是威斯康星州密尔沃基市Cornerstone咨询公司的负责人和合伙创始人，该公司从事分布式应用程序的开发、服务工作。

在他的职业生涯中，Dave致力于使用面向对象的分析、设计和编程技术建立软件解决方案。他领导了Internet和Intranet的规划小组，并评估、选择、介绍了Java、XML、SOAP、EJB和其他先进的新技术。作为一名优秀的顾问、设计师和开发者，Dave帮助财富1000强和其他公司成功地利用信息技术在竞争中获利。

Dave要感谢“爸爸的乖女儿”Katie、Rachel和Julie，是他们的爱和支持鼓舞并激励着他完成各项工作。

Dave撰写了第12章、第14章、第15章和第16章。

目　录

第1章 XML范例

本章通过范例和讨论提供了对Java和XML在一起使用时的看法。有效使用XML要求对它的优势和弱点有清醒的理解。但要做到这一点，同时又兼顾所有的Java使用细节的话，需要至少整整一本书的篇幅。因此，虽然涉及了XML语言和相关技术规范的某些方面，本书并不试图全面和正式地介绍XML。我们假定读者已经具有XML的一些基本概念，如组成、元素、属性以及它们的正式语法定义。

要详细了解XML，可以参考它的实际技术规范（参见下一节关于XML技术规范的解释）：W3C XML 1.0 Recommendation，它可在如下位置找到：http://www.w3c.org/TR/REC-xml。了解它的内容的一个好方法是参阅如下的文档：http://www.xml.com/axml/testaxml.htm，它是XML的原创作者之一Tim Bray所写，有一些额外的注解和评论。

本章的结尾还给出了其他的资源和说明。我们将通过一系列Java范例来抓住这种语言的一些重要功能，其中包括：

- 分析（Parsing）
- 名称空间（Namespaces）
- 样式表转换（Stylesheet Transformation）
- 字符编码（Character Encoding）
- 处理指令（Processing Instructions）
- 文档类型定义（Document Type Definition：DTD） 及实体（Entities）
- XML模式（XML Schema）

本章的主要目的是介绍用Java语言进行XML处理的主要元素，集中讲述以上的课题，同时提供了一组示例供参考。

下面推荐一些好的XML介绍性书籍，包括：Learning XML, Eric T. Ray, 2001, O'Reilly, ISBN 0-596000-46-4，以及Beginning XML, Curt Cagle et al., Wrox Press, 2000, ISBN 1-861003-41-2。更高级的材料有：Java and XML, Brett McLaughlin, O'Reilly, 2000, ISBN 0-596000-16-2和Professional Java XML, Kal Ahmed et al., Wrox Press, 2001, ISBN 1-861004-01-X。

XML组织机构

World Wide Web Consortium（世界网络协会，http://www.w3.org）是提供技术报告的国际组织，这些报告是由一个广受尊敬的理论家和爱好者组成的团体书写和修改的。由于它的声誉和在互联网建设方面的地位，这些文档被很多编程社区团体广泛采纳，把它作为真正普遍的解决方案的最大希望。不仅如此，来自所有主要的IT公司的支持和个人协作更加速了这些标准的传播和采用。W3C发布报告（通常在实际生活中以发布“通知”开始）的方式如下：

W3C报告	说明
工作草案（Working Draft）	这份文档代表W3C对某个领域的进一步的工作表示支持，并将和外部协作伙伴一起工作，但它不是技术规范
最终调用工作草案（Last Call Working Draft）	当对评审和注释进行深入评估时，一份报告可能有三个星期处于这种状态，之后，主管人员将把它转为待选或推荐状态
待选建议（Candidate Recommendation）	这种报告已经在某种级别得到大多数人的赞同，正在更为广泛的社区团体中征集实现方案
推荐建议（Proposed Recommendation）	在这个阶段，相关的专家组已经达成一致意见，主管已经给咨询委员会提交了报告。这个阶段还要求给出可供演示的工作原型
建议（Recommendation）	本阶段表示，一组标准已经通过W3C的评审并得到认可，具有足够的质量，可以在更广泛的范围内传播，进入商业和学术开发的主流。W3C承诺支持这个标准

本章的结尾提供了指向W3C的在线技术规范的链接。

Organization for the Advancement of Structured Information Standards（结构化信息标准推进组织，OASIS）是另外一个重要的XML组织。它拥有“XML工业入口（XML Industry Portal）”，本章结尾提供了相关的邮件列表xml-dev-链接。

XML分析

所谓分析，指的是把代表XML文档的一个无结构的字符序列转换为满足XML语法的结构化组件的过程，包括：声明、注释、元素、属性、处理指令以及字符。这个过程依赖于实际处理XML源数据时所选择的处理模式。有两种基本的模型：

- 使用分析器产生的语法events（事件）序列，例如“start of a <table> element”，直接输入到用户的处理中。
- 把嵌套的元素（或称“节点”，node）、XML源结构读入一个树中（这个树就代表你自己的程序），然后通过与这些树形数据类型相关的API操作XML数据。

这两种模型导致了两个不同的XML分析标准：

- 用一个实现了Simple API for XML（SAX）的分析器进行基于事件的分析
- 使用W3C Document Object Model（DOM）、JDOM或其他的文档模型进行基于树形结构的分析

SAX是由一组精心设计的“回调函数方法”组成的编程接口，可以被实际的XML分析器或驱动程序调用。由一个未被SAX指定的独立进程来控制原始字符序列的读入，以及调用高级别事件（包括分析错误）到注册的“处理器”类中。

下面介绍的是SAX模型的基本分析机制。完整的SAX API的介绍参见第2章。

DOM（W3C Level 1 Recommendation产生于1998年10月，Level 2 Recommendation产生于2000年11月）指定了一个节点类型的层次结构以及用户可以对它进行的处理操作。DOM的“Java绑定（Java Bindings）”（实现并不仅限于Java）完全是通过接口来定义的，所以，

如果你在实际代码使用了这种模型，就不能使用由某种分析器产生的特定的具体树形对象。例如，如果使用了DOM，就不可能使用new操作符明确地创建节点类型，相反，你必须使用工厂方法，如由Document类型实现的createElement()。

这样做的好处是，你的程序不需要了解实现DOM接口的具体对象，从而可以在不影响代码的情况下，自由转换软件供应商。

DOM将在第3章和第4章介绍。

Simple API for XML（SAX）

当前SAX的版本是2.0。尽管SAX 1.0版在很大程度上仍然是工业标准，但最初发布的1.0版现在有几个广受抨击的接口。下面几个图表给出了在SAX分析过程中的不同结构之间的关系。

SAX的主页是： http://www.megginson.com/SAX/index.html。

因为理解SAX 1.0和SAX 2.0之间的对应关系和区别非常重要，所以我们先来看SAX 1.0。

SAX 1.0

分析程序开始的第一件事是获取一个org.xml.sax.Parser接口的实现（这里用的是Sun Microsystems的分析器）：

```
org.xml.sax.Parser parser = new com.sun.xml.parser.Parser();
```

这个Parser接口提供了直接从文件中读取XML数据的方法，也能接收很多包装过的输入格式（如流、读程序或URL）来用作org.xml.sax.InputSource（可以把想要的用于输入的字符编码赋值给它）：

```
InputSource is = new InputSource();
is.setCharacterStream(new FileReader("xmlfile.xml"));
```

输入源对各种可能的数据源进行打包，从而使它们对分析器而言看起来基本相同。

这个过程可以用图1.1来说明。黑色圆点代表“挂钩”，通过它你可在默认功能上（如果有的话）注册自定义的处理器和修改工具。

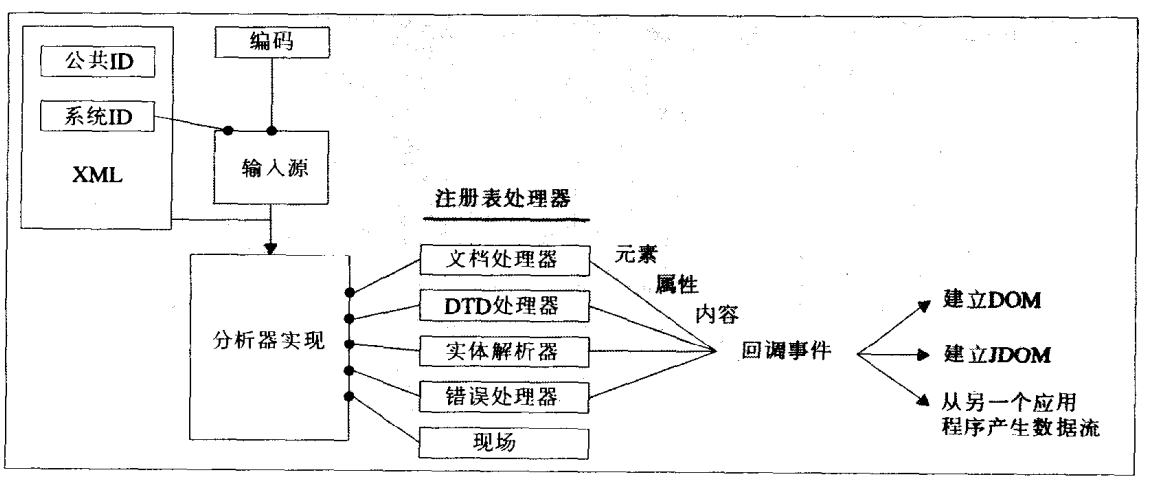

图1.1

现在，分析器已经准备好读取数据，并开始从字符XML输入产生的语法事件中产生事件。先挂接（或称为“注册”）一个实现org.xml.sax.DocumentHandler的对象：

```
parser.setDocumentHandler(handler);
```

类似地，你也可以挂接DTDHandler、ErrorHandler和EntityResolver的实现，用来处理DTD事件、分析错误信息和请求信息，以便自己定制对内部和外部实体的解决方案（这些在介绍SAX的章节中通过很多例子给出了详细解释）。最后，你也可以通过挂接java.util.Locale来指定与特定场所相关的错误信息和其他信息的输出。至此，你可以调用如下函数了：

```
parser.parse(is);
```

它将调用注册过的对象里的方法。例如，在分析过程中遇到元素时，会调用DocumentHandler对象的方法startElement()。

请注意，SAX经常被用做输入器，来创建XML结构的内存表示，如DOM树或JDOM树。许多返回DOM树的分析器都是由SAX回调函数驱动的，通过它完成往DOM树中增添元素和其他节点的处理过程。

还要注意的是，你不需要自己实现所有的处理器，相反，你可以使用一些常用的帮助类，例如SAX 1.0的org.xml.sax.HandlerBase类（尽管它很有争议）。HandlerBase类实现了所有必需的处理器接口，只需要简单地扩展这些类，对你想处理的事件的处理方法进行重载就可以了。

SAX 2.0

SAX 2.0是对SAX 1.0的升级，以提供对XML文档中名称空间的支持，保证独立开发的XML文档不会发生命名冲突。同时，底层模型进行了某些简化，改进了底层驱动器的可插入性，对通过JAXP获取SAX分析器的方法提供了更多的支持。这些目标是通过支持设置和测试标准，以及设立一个固定接口，测试与开发商相关的功能和特性等方法实现的。

由于Parser接口饱受非议，现在你可以用XMLReader来代替它。另一个受批评的接口DocumentHandler已经被程序包org.xml.sax中的ContentHandler取代了。除此以外，SAX 2.0处理过程中的组件示意图与前面几乎相同，所以在图1.2中没有完整地画出来。现在，我们获取org.xml.sax.XMLReader实现的方法（使用Xerces SAX分析器）如下：

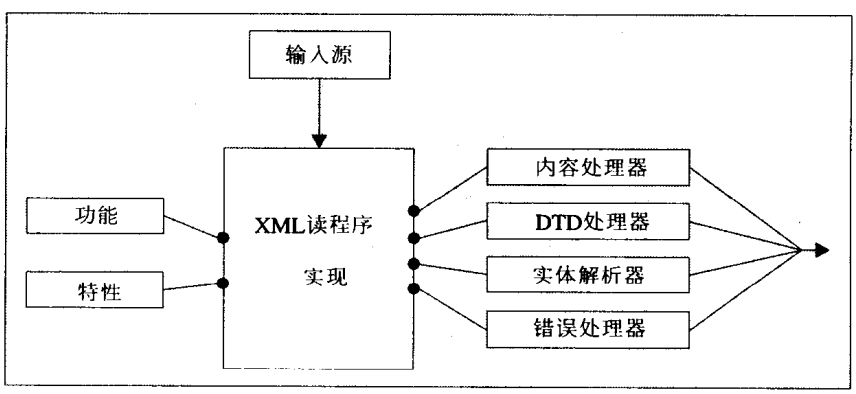

图1.2

```
org.xml.sax.XMLReader reader =
    new org.apache.xerces.parsers.SAXParser();
reader.setContentHandler(handler);
reader.parse(is);
```

但是，创建SAX 2.0分析器的标准机制是使用JAXP（下面的JAXP小节中有介绍）的抽象工厂模式：

- 调用抽象类javax.xml.parsers.SAXParserFactory中的newInstance()方法，获取一个扩展这个类的具体的分析器工厂的引用
- 使用这个工厂得到一个新的javax.xml.parsers.SAXParser
- 从这个SAX分析器中获得一个org.xml.sax.XMLReader

```
SAXParserFactory spf = SAXParserFactory.newInstance();
XMLReader reader = spf.newSAXParser().getXMLReader();
```

现在我们可以在读程序上设置特殊的功能（这里指的是处理名称空间声明的指令）：

```
reader.setFeature(
   "http://xml.org/sax/features/namespace-prefixes",true);
```

注意，上面使用了Uniform Resource Identifier（URI）作为这个功能的名称。后面我们还会详细讲述JAXP。有关SAX的具体内容请参阅第2章。

DOM

DOM是一组操作和查询XML文档结构的接口。由软件开发商自己去实现表示不同XML结构的具体的类。例如：org.w3c.dom.Document接口的实现就是Xerces parser程序包：org.apache.xerces.dom.DocumentImpl的一部分。

但是，你在自己的代码中可能永远不会引用到这些类。你只需获取一个对org.w3c.dom.Document的引用，然后按照W3C规范来操作它。在使用DOM时，你不需要知道这些引用指向的是哪些底层数据类型。

DOM分析器返回的是一个org.w3c.dom.Document的实现，这个实现代表了一个以org.w3c.dom.Node结构树表示的XML源。顶级Node既是一个Node，也是一个Document，因为Document本身是Node的一个特例（或称子接口）（参阅第3章）。

错误处理

XML 1.0定义了三类错误：

- 致命错误（Fatal Errors）：不可修复的错误，如语法不正确
- 错误（Errors）：可修复的错误，如出现了在文档类型定义中没有定义的无效类型
- 警告（Warnings）：任何其他可修复的错误

当你创建了一个分析器例程后，默认的错误处理器通常不做任何事情，也就是说，除非遇到致命错误，否则分析器将保持沉默。为了捕获这些错误，你通常需要向分析器注册一个实现org.xml.sax.ErrorHandler接口的类，以便把自己的错误处理器挂接上去。这既对基于DOM的分析器适用，也对基于SAX的分析器适用。不论是基于DOM还是基于SAX，大部分